International Handbooks on Information Systems

Christoph Schwindt
Jürgen Zimmermann *Editors*

Handbook on Project Management and Scheduling Vol. 2

Springer

International Handbooks on Information Systems

Series Editors

Peter Bernus, Jacek Błażewicz, Günter J. Schmidt, Michael J. Shaw

Titles in the Series

M. Shaw, R. Blanning, T. Strader and A. Whinston (Eds.)
Handbook on Electronic Commerce
ISBN 978-3-540-65882-1

J. Błażewicz, K. Ecker, B. Plateau and D. Trystram (Eds.)
Handbook on Parallel and Distributed Processing
ISBN 978-3-540-66441-3

H.H. Adelsberger, Kinshuk, J.M. Pawlowski and D. Sampson (Eds.)
Handbook on Information Technologies for Education and Training
ISBN 978-3-540-74154-1, 2nd Edition

C.W. Holsapple (Ed.)
Handbook on Knowledge Management 1 Knowledge Matters
ISBN 978-3-540-43527-3
Handbook on Knowledge Management 2 Knowledge Directions
ISBN 978-3-540-43848-9

J. Błażewicz, W. Kubiak, I. Morzy and M. Rusinkiewicz (Eds.)
Handbook on Data Management in Information Systems
ISBN 978-3-540-43893-9

P. Bernus, P. Nemes and G. Schmidt (Eds.)
Handbook on Enterprise Architecture
ISBN 978-3-540-00343-4

S. Staab and R. Studer (Eds.)
Handbook on Ontologies
ISBN 978-3-540-70999-2, 2nd Edition

S.O. Kimbrough and D.J. Wu (Eds.)
Formal Modelling in Electronic Commerce
ISBN 978-3-540-21431-1

P. Bernus, K. Merlins and G. Schmidt (Eds.)
Handbook on Architectures of Information Systems
ISBN 978-3-540-25472-0, 2nd Edition

S. Kirn, O. Herzog, P. Lockemann and O. Spaniol (Eds.)
Multiagent Engineering
ISBN 978-3-540-31406-6

J. Błażewicz, K. Ecker, E. Pesch, G. Schmidt and J. Węglarz (Eds.)
Handbook on Scheduling
ISBN 978-3-540-28046-0

F. Burstein and C.W. Holsapple (Eds.)
Handbook on Decision Support Systems 1
ISBN 978-3-540-48712-8

F. Burstein and C.W. Holsapple (Eds.)
Handbook on Decision Support Systems 2
ISBN 978-3-540-48715-9

D. Seese, Ch. Weinhardt and F. Schlottmann (Eds.)
Handbook on Information Technology in Finance
ISBN 978-3-540-49486-7

T.C. Edwin Cheng and Tsan-Ming Choi (Eds.)
Innovative Quick Response Programs in Logistics and Supply Chain Management
ISBN 978-3-642-04312-3

J. vom Brocke and M. Rosemann (Eds.)
Handbook on Business Process Management 1
ISBN 978-3-642-00415-5
Handbook on Business Process Management 2
ISBN 978-3-642-01981-4

T.-M. Choi and T.C. Edwin Cheng
Supply Chain Coordination under Uncertainty
ISBN 978-3-642-19256-2

C. Schwindt and J. Zimmermann (Eds.)
Handbook on Project Management and Scheduling Vol. 1
ISBN 978-3-319-05442-1
Handbook on Project Management and Scheduling Vol. 2
978-3-319-05914-3

More information about this series at
http://www.springer.com/series/3795

Christoph Schwindt • Jürgen Zimmermann
Editors

Handbook on Project Management and Scheduling Vol. 2

Springer

Editors
Christoph Schwindt
Institute of Management and Economics
Clausthal University of Technology
Clausthal-Zellerfeld
Germany

Jürgen Zimmermann
Institute of Management and Economics
Clausthal University of Technology
Clausthal-Zellerfeld
Germany

ISBN 978-3-319-05914-3 ISBN 978-3-319-05915-0 (eBook)
DOI 10.1007/978-3-319-05915-0
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014957172

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This handbook is devoted to scientific approaches to the management and scheduling of projects. Due to their practical relevance, project management and scheduling have been important subjects of inquiry since the early days of Management Science and Operations Research and remain an active and vibrant field of study. The handbook is meant to provide an overview of some of the most active current areas of research. Each chapter has been written by well-recognized scholars, who have made original contributions to their topic. The handbook covers both theoretical concepts and a wide range of applications. For our general readers, we give a brief introduction to elements of project management and scheduling in the first chapter, where we also survey the contents of this book. We believe that the handbook will be a valuable and comprehensive reference to researchers and practitioners in project management and scheduling and hope that it might stimulate further research in this exciting and practically important field.

Short-listing and selecting the contributions to this handbook and working with more than one hundred authors have been a challenging and rewarding experience for us. We are grateful to Günter Schmidt, who invited us to edit these volumes. Our deep thanks go to all authors involved in this project, who have invested their time and expertise in presenting their perspectives on project management and scheduling topics. Moreover, we express our gratitude to our collaborators Tobias Paetz, Carsten Ehrenberg, Alexander Franz, Anja Heßler, Isabel Holzberger, Michael Krause, Stefan Kreter, Marco Schulze, Matthias Walter, and Illa Weiss, who helped us to review the chapters and to unify the notations. Finally, we are pleased to offer special thanks to our publisher Springer and the Senior Editor Business, Operations Research & Information Systems Christian Rauscher for their patience and continuing support.

Clausthal-Zellerfeld, Germany

Christoph Schwindt
Jürgen Zimmermann

Contents

Part XIX Project Management Information Systems

Contents of Volume 1

Part IX Discrete Time-Cost Tradeoff Problems

List of Symbols

Miscellaneous

$:=$	Equal by definition, assignment
$\square$	End of proof
$\lceil z \rceil$	Smallest integer greater than or equal to z
$\lfloor z \rfloor$	Greatest integer smaller than or equal to z
$(z)^+$	Maximum of 0 and z

Sets

$\emptyset$	Empty set
$]a, b[$	Open interval $\{x \in \mathbb{R} \mid a < x < b\}$
$[a, b[$	Half open interval $\{x \in \mathbb{R} \mid a \leq x < b\}$
$]a, b]$	Half open interval $\{x \in \mathbb{R} \mid a < x \leq b\}$
$[a, b]$	Closed interval $\{x \in \mathbb{R} \mid a \leq x \leq b\}$
$\lvert A \rvert$	Number of elements of finite set A
$A \subset B$	A is proper subset of B
$A \subseteq B$	A is subset of B
$A \setminus B$	Difference of sets A and B
$A \cap B$	Intersection of sets A and B
$A \cup B$	Union of sets A and B
$conv(A)$	Convex hull of set A
$f : A \to B$	Mapping (function) of A into B
$\mathbb{N}$	Set of positive integers
$\mathscr{NP}$	Set of decision problems that can be solved in polynomial time by a non-deterministic Turing machine

$\mathscr{O}$	Landau's symbol: for $f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ it holds that $g \in \mathscr{O}(f)$ if there are a constant $c > 0$ and a positive integer n_0 such that $g(n) \leq c\ f(n)$ for all $n \geq n_0$
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Set of n-tuples of real numbers
$\mathbb{R}_{\geq 0}$	Set of nonnegative real numbers
$\mathbb{Z}$	Set of integers
$\mathbb{Z}_{\geq 0}$	Set of nonnegative integers

Projects, Activities, and Project Networks

δ_{ij}	Weight of arc (i, j), start-to-start minimum time lag between activities i and j
$\mathscr{A}$	Set of all maximal feasible antichains of the precedence order (non-dominated feasible subsets)
$\overline{\mathscr{A}}$	Set of all feasible antichains of the precedence order (feasible subsets)
$A \in \overline{\mathscr{A}}$	Feasible antichain (feasible subset)
$\mathscr{A}(S, t)$	Set of activities in execution at time t given schedule S
d_{ij}	Longest path length from node i to node j in project network N
d_{ij}^{max}	Maximum time lag between the starts of activities i and j
d_{ij}^{min}	Minimum time lag between the starts of activities i and j
$\overline{d}$	Prescribed maximum project duration
E	Arc set of directed graph G or project network N
E_i^-	Set of arcs leading to node i
E_i^+	Set of arcs emanating from node i
$\mathscr{F}$	Set of all minimal forbidden sets
$F \in \mathscr{F}$	Minimal forbidden set
$G = (V, E)$	Directed graph with node set V and arc set E (precedence graph)
i, j	Activities or events of the project
(i, j)	Arc with initial node i and terminal node j
n	Number of activities of the project, without project beginning 0 and project completion $n + 1$
$N = (V, E, \delta)$	Project network with node set V, arc set E, and arc weights δ
p_i	Duration (processing time) of activity i
$Pred(i)$	Set of immediate predecessors of activity i in project network N
$\overline{Pred}(i)$	Set of all immediate and transitive predecessors of activity i in project network N
$Succ(i)$	Set of all immediate successors of activity i in project network N
$\overline{Succ}(i)$	Set of all immediate and transitive successors of activity i in project network N
TE	Transitive closure of the arc set

V	Node set of direct graph G or project network N; Set of activities in an activity-on-node network
V^a	Set of real activities in an activity-on-node network

Resources and Skills

Π_k	Set of periods associated with partially renewable resource k
k	Single (renewable, nonrenewable, partially renewable, or storage) resource
$K = \lvert\mathscr{R}\rvert$	Number of renewable resources
$l \in \mathscr{L}$	Single skill
$L = \lvert\mathscr{L}\rvert$	Number of skills
$L_i = \lvert\mathscr{L}_i\rvert$	Number of skills required by activity i
$\mathscr{L}$	Set of skills
$\mathscr{L}_i$	Set of skills required by activity i
$\mathscr{L}_k$	Set of skills that can be performed by resource k
r_{ik}	Amount of resource k used by activity i
$r_{ik}(t)$	Amount of resource k used by activity i in the t-th period of its execution
r_{il}	Number of resource units with skill l required by activity i
$r_k(S,t)$	Amount of resource k used at time t given schedule S
R_k	Capacity or availability of resource k
$R_k(t)$	Capacity of renewable resource k in period t
$\mathscr{R}$	Set of (discrete) renewable resources (e.g., workers)
$\mathscr{R}_l$	Set of workers possessing skill l
$\mathscr{R}^n$	Set of nonrenewable resources
$\mathscr{R}^p$	Set of partially renewable resources
$\mathscr{R}^s$	Set of storage resources
wc_i	Work content of activity i
$wl_{ik} = p_i \cdot r_{ik}$	Workload of renewable resource k incurred by activity i
$WL_k = R_k \cdot \overline{d}$	Workload capacity of renewable resource k

Multi-Modal Project Scheduling

m	Execution mode
$\mathscr{M}_i$	Set of alternative execution modes for activity i
$M_i = \lvert\mathscr{M}_i\rvert$	Number of modes of activity i
p_{im}	Duration of activity i in execution mode m
r_{ikm}	Amount of resource k used by activity i in execution mode m
x	Mode assignment with $x_{im} = 1$, if activity i is processed in execution mode $m \in \mathscr{M}_i$

	Staff assignment with $x_{ikl} = 1$, if a worker of resource k performs activity i with skill l

Discrete Time-Cost Tradeoff

b	Budget for activity processing
$c_i(p_i)$	Cost for processing activity i with duration p_i ($= c_{im}$ with $p_i = p_{im}$)
c_{im}	Cost of executing activity i in mode m
p_{im}	Duration of activity i in mode m

Multi-Project Problems

α_q	Dummy start activity of project q
ω_q	Dummy end activity of project q
d_q	Due date for completion of project q
$\overline{d}_q$	Deadline for completion of project q
n_q	Number of real activities of project q
$q \in Q$	Single project
Q	Set of projects
V_q	Set of activities of project q

Project Scheduling Under Uncertainty and Vagueness

λ	Arrival rate of projects
$\mu_{\hat{z}}(z)$	Membership function of fuzzy set $\hat{z}$
π_σ	Probability of scenario σ $(\sum_{\sigma \in \Sigma} \pi_\sigma = 1)$
$\sigma \in \Sigma$	Single scenario
Σ	Set of scenarios
Σ_i	Set of scenarios for activity i
$E(\tilde{x})$	Expected value of $\tilde{x}$
$f_{\tilde{x}}(x)$	Probability density function (pdf) of random variable $\tilde{x}$ $(= \frac{dF_{\tilde{x}}}{dx}(x))$
$F_{\tilde{x}}(x)$	Cumulative probability distribution function (cdf) of random variable $\tilde{x}$ $(= P(\tilde{x} \leq x))$
$\tilde{p}_i$	Random duration of activity i
$P(A)$	Probability of event A
p_i^{min}, p_i^{max}	Minimum and maximum duration of activity i
$\hat{p}_i$	Fuzzy duration of activity i
$var(\tilde{x})$	Variance of $\tilde{x}$

$\tilde{x}, \tilde{\xi}$	General random variables
x_α	α-quantile $(F_{\tilde{x}}(x_\alpha) = \alpha)$
z	(Crisp) Element from set Z
$\hat{z}$	General fuzzy set

Objective Functions

α	Continuous interest rate
$\beta = e^{-\alpha}$	Discount rate per unit time
c_i^F	Cash flow associated with the start or completion of activity i
$c_i^{F-} > 0$	Disbursement $-c_i^F > 0$ associated with activity or event i
$c_i^{F+} > 0$	Payment $c_i^F > 0$ associated with activity or event i
c_k	Cost for resource k per unit
$C_{max} = S_{n+1}$	Project duration (project makespan)
$f(S)$	Objective function value of schedule S (single-criterion problem); Vector $(f_1(S), \ldots, f_\nu(S))$ of objective function values (multi-criteria problem)
$f(S, x)$	Objective function value of schedule S and mode assignment x
f_μ	Single objective function in multi-criteria project scheduling
LB	Lower bound on minimum objective function value
npv	Net present value of the project
$\mathscr{PF}$	Pareto front of multi-criteria project scheduling problem
UB	Upper bound on minimum objective function value
w_i	Arbitrary weight of activity i

Temporal Scheduling

C_i	Completion time of activity i
EC_i	Earliest completion time of activity i
ES	Earliest schedule
ES_i	Earliest start time of activity i
LC_i	Latest completion time of activity i
LS	Latest schedule
LS_i	Latest start time of activity i
S	Schedule
S_i	Start time of activity i or occurrence time of event i
TF_i	Total float of activity i

Models and Solution Methods

ϕ_{ij}^{k}	Amount of resource k transferred from activity i to activity j
ρ_{mut}	Mutation rate
σ_{pop}	Population size
ℓ	Activity list $(i_1, i_2, \ldots, i_n)$
$\mathscr{C}$	Set of activities already scheduled (completed set)
$\mathscr{D}$	Decision set containing all activities eligible for being scheduled
$S^{\mathscr{C}}$	Partial schedule of activities $i \in \mathscr{C}$
t	Time period, start of period $t + 1$
T	Last period, end of planning horizon

Computational Results

$\Delta_{LB}^{\emptyset}$	Average relative deviation from lower bound
Δ_{LB}^{max}	Maximum relative deviation from lower bound
$\Delta_{opt}^{\emptyset}$	Average relative deviation from optimum value
Δ_{opt}^{max}	Maximum relative deviation from optimum value
$\Delta_{UB}^{\emptyset}$	Average relative deviation from upper bound
Δ_{UB}^{max}	Maximum relative deviation from upper bound
LB_0	Critical-path based lower bound on project duration
LB^*	Maximum lower bound
n_{best}	Number of best solutions found
$n_{iter}^{\emptyset}$	Average number of iterations
n_{iter}^{max}	Maximum number of iterations
n_{opt}	Number of optimal solutions found
OS	Order strength of project network
p_{feas}	Percentage of instances for which a feasible solution was found
p_{inf}	Percentage of instances for which the infeasibility was proven
p_{opt}	Percentage of instances for which an optimal solution was found
p_{unk}	Percentage of instances for which it is unknown whether there exists a feasible solution
RF	Resource factor of project
RS	Resource strength of project
t_{cpu}^{lim}	CPU time limit
$t_{cpu}^{\emptyset}$	Average CPU time
t_{cpu}^{max}	Maximum CPU time

Three-Field Classification $\alpha \mid \beta \mid \gamma$ for Project Scheduling Problems[1]

Field α: Resource Environment

PS	Project scheduling problem with limited (discrete) renewable resources
PS∞	Project scheduling problem without resource constraints (time-constrained project scheduling problem)
PSc	Project scheduling problem with limited continuous and discrete renewable resources
PSf	Project scheduling problem with limited renewable resources and flexible resource requirements (problem with work-content constraints)
PSS	Project staffing and scheduling problem with multi-skilled resources of limited workload capacity
PSS∞	Project staffing and scheduling problem with limited multi-skilled resources of unlimited workload capacity
PSp	Project scheduling problem with limited partially renewable resources
PSs	Project scheduling problem with limited storage resources
PSt	Project scheduling problem with limited (discrete) time-varying renewable resources
$MPSm, \sigma, \mu$	Multi-mode project scheduling problem with m limited (discrete) renewable resources of capacity σ and μ nonrenewable resources
MPS	Multi-mode project scheduling problem with limited renewable and nonrenewable resources
MPS∞	Multi-mode project scheduling without resource constraints (time-constrained project scheduling problem)

Field β: Project and Activity Characteristics

The second field $\beta \subseteq \{\beta_1, \beta_2, \ldots, \beta_{13}\}$ specifies a number of project and activity characteristics; $\circ$ denotes the empty symbol.

$\beta_1 : mult$	Multi-project problem
$\beta_1 : \circ$	Single-project problem
$\beta_2 : prec$	Ordinary precedence relations between activities

[1]The classification is a modified version of the classification scheme introduced in Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999) Resource-constrained project scheduling: notation, classification, models, and methods. Eur J Oper Res 112:3–41.

$\beta_2 : temp$	Generalized precedence relations between activities (minimum and maximum time lags between start or completion times of activities)
$\beta_2 : feed$	Feeding precedence relations between activities
$\beta_3 : \overline{d}$	Prescribed deadline $\overline{d}$ for project duration
$\beta_3 : \circ$	No prescribed maximum project duration
$\beta_4 : bud$	Limited budget for activity processing
$\beta_4 : \circ$	No limited budget for activity processing
$\beta_5 : p_i = sto$	Stochastic activity durations
$\beta_5 : p_i = unc$	Uncertain activity durations from given intervals
$\beta_5 : p_i = fuz$	Fuzzy activity durations
$\beta_5 : \circ$	Deterministic/crisp activity durations
$\beta_6 : c_i = sto$	Stochastic activity cost
$\beta_6 : c_i = unc$	Uncertain activity cost from given intervals
$\beta_6 : c_i = fuz$	Fuzzy activity cost
$\beta_6 : \circ$	Deterministic/crisp activity cost
$\beta_7 : Poi$	Stochastic arrival of projects with identical project network according to Poisson process
$\beta_7 : \circ$	Immediate availability of project(s)
$\beta_8 : act = sto$	Set of activities to be executed is stochastic
$\beta_8 : \circ$	Set of activities to be executed is prescribed
$\beta_9 : pmtn$	Preemptive problem, activities can be interrupted at any point in time
$\beta_9 : pmtn/int$	Preemptive problem, activities can be interrupted at integral points in time only
$\beta_9 : l\text{-}pmtn/int$	Preemptive problem, activities can be interrupted at integral points in time, the numbers of interruptions per activity are limited by given upper bounds
$\beta_9 : \circ$	Non-preemptive problem (activities cannot be interrupted)
$\beta_{10} : r_{il} = 1$	Each activity requires at most one resource unit with skill l for execution
$\beta_{10} : \circ$	Each activity i requires an arbitrary number of resource units with skill l for execution
$\beta_{11} : cal$	Activities can only be processed during certain time periods specified by activity calendars
$\beta_{11} : \circ$	No activity calendars have to be taken into account
$\beta_{12} : s_{ij}$	Sequence-dependent setup/changeover times of resources between activities i and j
$\beta_{12} : \circ$	No sequence-dependent changeover times
$\beta_{13} : nestedAlt$	The project network is given by a nested temporal network with alternatives, where only a subset of the activities must be executed
$\beta_{13} : \circ$	No alternative activities have to be taken into account

Field γ: Objective Function

f	General (regular or nonregular) objective function
reg	Regular objective function
mac	General mode assignment cost
$staff$	General project staffing cost (project staffing and scheduling)
rob	Robustness measure
$mult$	General multi-criteria problem
$f_1/f_2/\ldots$	Multi-criteria problem with objective functions $f_1, f_2, \ldots$
C_{max}	Project duration
$\Sigma c_i^F \beta^{C_i}$	Net present value of project
$\Sigma c_k \max r_{kt}$	Total availability cost (resource investment problem)
$\Sigma c_k \Sigma r_{kt}^2$	Total squared utilization cost (resource leveling)
$\Sigma c_k \Sigma o_{kt}$	Total overload cost (resource leveling)
$\Sigma c_k \Sigma \Delta r_{kt}$	Total adjustment cost (resource leveling)
$\Sigma c_i(p_i)$	Total cost of activity processing (time-cost tradeoff problem)
wT	Weighted project tardiness

Examples

$PS \mid prec \mid C_{max}$	Basic resource-constrained project scheduling problem (RCPSP)
$PS \mid temp, pmtn \mid C_{max}$	Preemptive resource-constrained project scheduling problem with generalized precedence relations
$MPS\infty \mid prec, \overline{d} \mid \Sigma c_i(p_i)$	Discrete time-cost tradeoff problem (deadline version)
$MPS \mid temp \mid \Sigma c_i^F \beta^{C_i}$	Multi-mode resource-constrained net present value problem with generalized precedence relations
$PS \mid prec \mid C_{max}/\Sigma r_{kt}^2$	Bi-criteria resource-constrained project scheduling problem (project duration, total squared utilization cost)
$PS \mid prec, p_i = sto \mid C_{max}$	Stochastic resource-constrained project scheduling problem

Project Management and Scheduling

Christoph Schwindt and Jürgen Zimmermann

1 Projects, Project Management, and Project Scheduling

Nowadays, *projects* are omnipresent. These unique and temporary undertakings have permeated almost all spheres of life, be it work or leisure, be it business or social activities. Most frequently, projects are encountered in private and public enterprizes. Due to product differentiation and collapsing product life cycles, a growing part of value adding activities in industry and services is organized as projects. In some branches, virtually all revenues are generated through projects. The temporary nature of projects stands in contrast with more traditional forms of business, which consist of repetitive, permanent, or semi-permanent activities to produce physical goods or services (Dinsmore and Cooke-Davies 2005, p. 35).

Projects share common characteristics, although they appear in many forms. Some projects take considerable time and consume a large amount of resources, while other projects can be completed in short time without great effort. To get a clear understanding of the general characteristics of a project, we consider the following two definitions of a project, which are taken from Kerzner (2013, p. 2) and PMI (2013, p. 4).

1. "A project can be considered to be any series of activities and tasks that:

   - have a specific objective to be completed within certain specifications,
   - have defined start and end dates,
   - have funding limits (if applicable),
   - consume human and nonhuman resources (i.e., people, money, equipment),
   - are multifunctional (i.e., cut across several functional lines)."

C. Schwindt (✉) • J. Zimmermann
Institute of Management and Economics, Clausthal University of Technology,
Clausthal-Zellerfeld, Germany
e-mail: christoph.schwindt@tu-clausthal.de; juergen.zimmermann@tu-clausthal.de

2. "A project is a temporary endeavor undertaken to create a unique product, service, or result."

According to these definitions, we understand a project as a one-time endeavor that consists of a set of activities, whose executions take time, require resources, and incur costs or induce cash flows. Precedence relations may exist between activities; these relations express technical or organizational requirements with respect to the order in which activities must be processed or with respect to their timing relative to each other. Moreover, the scarcity of the resources allocated to the project generally gives rise to implicit dependencies among the activities sharing the same resources, which may necessitate the definition of additional precedence relations between certain activities when the project is scheduled. A project is carried out by a project team, has a deadline, i.e., is limited in time, and is associated with one or several goals whose attainment can be monitored.

Typical examples for projects are:

- construction of a building, road, or bridge,
- development of a new product,
- reorganization in a firm,
- implementation of a new business process or software system,
- procurement and roll-out of an information system,
- design of a new pharmaceutical active ingredient, or
- conducting an election campaign.

Project management deals with the coordination of all initiating, planning, decision, execution, monitoring, control, and closing processes in the course of a project. In other words, it is the application of knowledge, skills, tools, and techniques to project tasks to meet all project interests. According to the Project Management Institute standard definition (PMI 2013, p. 8), managing a project includes

- identifying requirements,
- establishing clearly understandable and viable objectives,
- balancing the competing demands for time, quality, scope, and cost, and
- customizing the specifications, plans, and approach to the concerns and expectations of the different stakeholders.

Consequently, successful project management means to perform the project within time and cost estimates at the desired performance level in accordance with the client, while utilizing the required resources effectively and efficiently.

From a project management point of view, the life cycle of a project consists of five consecutive phases, each of which involves specific managerial tasks (cf., e.g., Lewis 1997; Klein 2000). At the beginning of the first phase, called *project conception*, there is only a vague idea of the project at hand. By means of some feasibility studies as well as economic and risk analyses it is decided whether or not a project should be performed. In the *project definition phase* the project objectives and the organization form of the project are specified. In addition, the

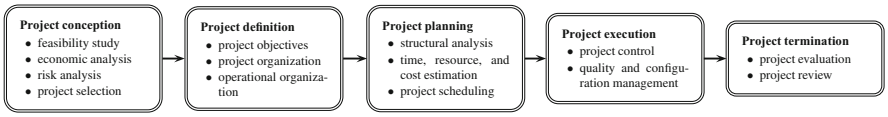


Fig. 1 Project life cycle

operational organization in the form of a roadmap (milestone plan) is conceived. In the *project planning phase* the project is decomposed into precedence-related activities. Then, for each activity the duration, the required resources, and the cost associated with the execution of that activity are estimated. Furthermore, the precedence relations among the activities are specified. Finally, a project schedule is determined by some appropriate planning approach (project scheduling). After these three phases the project is ready for implementation and the *project execution phase* starts. By monitoring the project progress, project management continuously evaluates whether or not the project is performed according to the established baseline schedule. If significant deviations are detected the plan has to be revised or an execution strategy defined in the planning phase is used to bring the project back to course. Moreover, quality and configuration management are performed in this phase (Turner 2009; PMI 2013). The final *project termination phase* evaluates and documents the project execution after its completion. Figure 1 summarizes the five phases of the project life cycle. Next, we will consider the project scheduling part of the planning phase in more detail.

Project scheduling is mainly concerned with selecting execution modes and fixing execution time intervals for the activities of a project. One may distinguish between time-constrained and resource-constrained project scheduling problems, depending on the type of constraints that are taken into account when scheduling the project. In time-constrained problems it is supposed that the activities are to be scheduled subject to precedence relations and that the required resources can be provided in any desired amounts, possibly at the price of higher execution cost or unbalanced resource usage. In the setting of a resource-constrained project scheduling problem, the availability of resources is necessarily assumed to be limited; consequently, in addition to the precedence relations, resource constraints have to be taken into account. Time-cost tradeoff and resource leveling problems are examples of time-constrained project scheduling problems. These examples show that time-constrained problems also may include a resource allocation problem, which consists in assigning resource units to the execution of the activities over time.

Different types of precedence relations are investigated in this handbook. An ordinary precedence relation establishes a predefined sequence between two activities, the second activity not being allowed to start before the first has been completed. Generalized precedence relations express general minimum and maximum time lags between the start times of two activities. Feeding precedence relations require that an activity can only start when a given minimum percentage

of its predecessor activity has been completed. The difference between generalized and feeding precedence relations becomes apparent when the activity durations are not fixed in advance or when activities can be interrupted during their execution.

Throughout this handbook, the term "resource" designates a pool of identical resource units, and the number of resource units available is referred to as the capacity or availability of the resource. In project scheduling, several kinds of resources have been introduced to model input factors of different types. Renewable resources represent inputs like manpower or machinery that are used, but not consumed when performing the project. In contrast, nonrenewable resources comprise factors like a budget or raw materials, which are consumed in the course of the project. Renewable and nonrenewable resources can be generalized to storage resources, which are depleted and replenished over time by the activities of the project. Storage resources can be used to model intermediate products or the cash balance of a project with disbursements and progress payments. Resources like electric power or a paged virtual memory of a computer system, which can be allotted to activities in continuously divisible amounts, are called continuous resources. Partially renewable resources refer to unions of time intervals and can be used to model labor requirements arising, e.g., in staff scheduling problems.

A common assumption in project scheduling is that activities must not be interrupted when being processed. There exist, however, applications for which activity splitting may be advantageous or even necessary. Examples of such applications are the aggregate mid-term planning of project portfolios composed of subprojects or working packages and the scheduling of projects in which certain resources cannot be operated during scheduled downtimes. The preemptive scheduling problems can be further differentiated according to the time points when an activity can be interrupted or resumed. Integer preemption problems assume that an activity can only be split into parts of integral duration, whereas continuous preemption problems consider the general case in which activities may be interrupted and resumed at any point in time.

An important attribute of a project scheduling problem concerns the number of execution modes that can be selected for individual activities. The setting of a single-modal problem premises that there is only one manner to execute an activity or that an appropriate execution mode has been selected for each activity before the scheduling process is started. A multi-modal problem always comprises a mode selection problem, the number of alternative modes for an activity being finite or infinite. Multiple execution modes allow to express resource-resource, resource-time, and resource-cost tradeoffs, which frequently arise in practical project scheduling applications.

With respect to the scheduling objectives, one may first distinguish between single-criterion and multi-criteria problems. A problem of the latter type includes several conflicting goals and its solution requires concepts of multi-criteria decision making like goal programming or goal attainment models. Second, objective functions can be classified as being regular or non-regular. Regular objective functions are defined to be componentwise nondecreasing in the start or completion times of the activities. Obviously, a feasible instance of a problem with a regular objective

function always admits a solution for which no activity can be scheduled earlier without delaying the processing of some other activity. Since in this case, the search for an optimal schedule can be limited to such "active" schedules, problems with regular objective functions are generally more tractable than problems involving a non-regular objective function.

A further attribute of project scheduling problems refers to the level of available information. The overwhelming part of the project scheduling literature addresses deterministic problem settings, in which it is implicitly assumed that all input data of the problem are precisely known in advance and no disruptions will occur when the schedule is implemented. In practice, however, projects are carried out in stochastic and dynamic environments. Hence, it seems reasonable to account for uncertainty when deciding on the project schedule. This observation leads to stochastic project scheduling problems or project scheduling problems under interval uncertainty, depending on whether or not estimates of probability distributions for the uncertain parameters are supposed to be available. Fuzzy project scheduling problems arise in a context in which certain input data are vague and cannot be specified on a cardinal scale, like assessments by means of linguistic variables.

Finally, project scheduling problems may be categorized according to the distribution of information or the number of decision makers involved. Most work on project scheduling tacitly presumes that the projects under consideration can be scheduled centrally under a symmetric information setting, in which there is a single decision maker or all decision makers pursue the same goals and are provided access to the same information. However, in a multi-project environment, decentralized decision making may be the organization form of choice, generally leading to an asymmetric information distribution and decision makers having their own objectives. In this case, a central coordination mechanism is needed to resolve conflicts and to achieve a satisfying overall project performance.

Table 1 summarizes the classification of project scheduling problems considered in this handbook. For further reading on basic elements and more advanced concepts of project scheduling we refer to the surveys and handbooks by Artigues et al. (2008), Demeulemeester and Herroelen (2002), Hartmann and Briskorn (2010), and Józefowska and Węglarz (2006).

2 Scope and Organization of the Handbook

Given the long history and practical relevance of project management and scheduling, one might be tempted to suppose that all important issues have been addressed and all significant problems have been solved. The large body of research papers, however, that have appeared in the last decade and the success of international project management and scheduling conferences prove that the field remains a very active and attractive research area, in which major and exciting developments are still to come.

Table 1 Classification of project scheduling problems

Attributes	Characteristics
Type of constraints	Time-constrained problem
	Resource-constrained problem
Type of precedence relations	Ordinary precedence relations
	Generalized precedence relations
	Feeding precedence relations
Type of resources	Renewable resources
	Nonrenewable resources
	Storage resources
	Continuous resources
	Partially renewable resources
Type of activity splitting	Non-preemptive problem
	Integer preemption problem
	Continuous preemption problem
Number of execution modes	Single-modal problem
	Multi-modal problem
Number of objectives	Single-criterion problem
	Multi-criteria problem
Type of objective function	Regular function
	Non-regular function
Level of information	Deterministic problem
	Stochastic problem
	Problem under interval uncertainty
	Problem under vagueness
Distribution of information	Centralized problem (symmetric distribution)
	Decentralized problem (asymmetric distribution)

This handbook is a collection of 62 chapters presenting a broad survey on key issues and recent developments in project management and scheduling. Each chapter has been contributed by recognized experts in the respective domain. The two volumes comprise contributions from seven project management and scheduling areas, which are organized in 19 parts. The first three areas are covered by Vol. 1 of the handbook, the remaining four areas being treated in Vol. 2. The covered topics range from basic project scheduling problems and their generalizations through multi-project planning, project scheduling under uncertainty and vagueness, recent developments in general project management and project risk management to applications, case studies, and project management information systems. The following list provides an overview of the handbook's contents.

- Area A: Project duration problems in single-modal project scheduling
  - Part I: The Resource-Constrained Project Scheduling Problem
  - Part II: The Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations

  - Part III: Alternative Resource Constraints in Project Scheduling
  - Part IV: Preemptive Project Scheduling
- Area B: Alternative objectives in single-modal project scheduling
  - Part V: Non-Regular Objectives in Project Scheduling
  - Part VI: Multi-Criteria Objectives in Project Scheduling
- Area C: Multi-modal project scheduling
  - Part VII: Multi-Mode Project Scheduling Problems
  - Part VIII: Project Staffing and Scheduling Problems
  - Part IX: Discrete Time-Cost Tradeoff Problems
- Area D: Multi-project problems
  - Part X: Multi-project scheduling
  - Part XI: Project Portfolio Selection Problems
- Area E: Project scheduling under uncertainty and vagueness
  - Part XII: Stochastic Project Scheduling
  - Part XIII: Robust Project Scheduling
  - Part XIV: Project Scheduling Under Interval Uncertainty and Fuzzy Project Scheduling
- Area F: Managerial approaches
  - Part XV: General Project Management
  - Part XVI: Project Risk Management
- Area G: Applications, case studies, and information systems
  - Part XVII: Project Scheduling Applications
  - Part XVIII: Case Studies in Project Scheduling
  - Part XIX: Project Management Information Systems

The parts of Areas A to E, devoted to models and methods for project scheduling, follow a development from standard models and basic concepts to more advanced issues such as multi-criteria problems, project staffing and scheduling, decentralized decision making, or robust optimization approaches. Area F covers research opportunities and emerging issues in project management. The chapters of the last Area G report on project management and scheduling applications and case studies in various domains like production scheduling, R&D planning, make-or-buy decisions and supplier selection, scheduling in computer grids, and the management of construction projects. Moreover, three chapters address the benefits and capabilities of project management information systems.

Most chapters are meant to be accessible at an introductory level by readers with a basic background in operations research and probability calculus. The intended audience of this book includes project management professionals, graduate students

in management, industrial engineering, computer science, or operations research, as well as scientists working in the fields of project management and scheduling.

3 Outline of the Handbook

Area A of this handbook is dedicated to single-modal project scheduling problems in which the activities have to be scheduled under precedence relations and resource constraints and the objective consists in minimizing the duration (or makespan) of the project. In practice, these project scheduling problems have a large range of applications, also beyond the field of proper project management. For example, production scheduling and staff scheduling problems can be modeled as single-modal project scheduling problems. In order to model specific practical requirements like prescribed minimum and maximum time lags between activities, availability of materials and storage capacities, or divisible tasks, project scheduling models including generalized precedence relations, new types of resource constraints, or preemptive activities have been proposed. These extensions to the basic model are also addressed in this portion of the handbook.

Part I is concerned with the classical resource-constrained project scheduling problem RCPSP. Solution methods for the RCPSP have been developed since the early 1960s and this problem is still considered the standard model in project scheduling. In Chap. 1 Rainer Kolisch reviews shifts, schedule types, and schedule-generation schemes for the RCPSP. A shift transforms a schedule into another schedule by displaying sets of activities. Based on the introduced shifts, different types of schedules, e.g., semi-active and active schedules, are defined. Furthermore, two different schedule-generation schemes are presented. The serial schedule-generation scheme schedules the activities one by one at their respective earliest feasible start times. The parallel schedule-generation scheme is time-oriented and generates the schedule by iteratively adding concurrent activities in the order of increasing activity start times. Variants of the two schemes for the resource-constrained project scheduling problem with generalized precedence relations and for the stochastic resource-constrained project scheduling problem are discussed as well. Chapter 2, written by Christian Artigues, Oumasr Koné, Pierre Lopez, and Marcel Mongeau, surveys (mixed-)integer linear programming formulations for the RCPSP. The different formulations are divided into three categories: First, time-indexed formulations are presented, in which time-indexed binary variables encode the status of an activity at the respective point in time. The second category gathers sequencing formulations including two types of variables. Continuous natural-date variables represent the start time of the activities and binary sequencing variables are used to model decisions with respect to the ordering of activities that compete for the same resources. Finally, different types of event-based formulations are considered, containing binary assignment and continuous positional-date variables. In Chap. 3 Sigrid Knust overviews models and methods for calculating lower bounds on the minimum project duration for the RCPSP. Constructive and destructive bounds are

presented. The constructive lower bounds are based on the relaxation or Lagrangian dualization of the resource constraints or a disjunctive relaxation allowing for activity preemption and translating precedence relations into disjunctions of activities. Destructive lower bounds arise from disproving hypotheses on upper bounds on the minimum objective function value. Knust reviews destructive lower bounds for the RCPSP that are calculated using constraint propagation and a linear programming formulation. Chapter 4 by Anurag Agarwal, Selcuk Colak, and Selcuk Erenguc considers meta-heuristic methods for the RCPSP. Important concepts of heuristic methods as well as 12 different meta-heuristics are presented. Amongst others, genetic algorithms, simulated annealing methods, and ant-colony optimization are discussed. A neuro-genetic approach is presented in more detail. This approach is a hybrid of a neural-network based method and a genetic algorithm.

Part II deals with the resource-constrained project scheduling problem with generalized precedence relations RCPSP/max. Generalized precedence relations express minimum and maximum time lags between the activities and can be used to model, e.g., release dates and deadline of activities or specified maximum makespans for the execution of subprojects. In Chap. 5 Lucio Bianco and Massimiliano Caramia devise lower bounds and exact solution approaches for the RCPSP/max. First, a new mathematical formulation for the resource-unconstrained project scheduling problem is presented. Then, they propose a lower bound for the RCPSP/max relying on the unconstrained formulation. The branch-and-bound method is based on a mixed-integer linear programming formulation and a Lagrangian relaxation based lower bound. The mixed-integer linear program includes three types of time-indexed decision variables. The first two types are binary indicator variables for the start and the completion of activities, whereas the third type corresponds to continuous variables providing the relative progress of individual activities at the respective points in time. Chapter 6 presents a constraint satisfaction solving framework for the RCPSP/max. Amedeo Cesta, Angelo Oddi, Nicola Policella, and Stephen Smith survey the state of the art in constraint-based scheduling, before the RCPSP/max is formulated as a constraint satisfaction problem. The main idea of their approach consists in establishing precedence relations between activities that share the same resources in order to eliminate all possible resource conflicts. Extended optimizing search procedures aiming at minimizing the makespan and improving the robustness of a solution are presented. Chapter 7, written by Andreas Schutt, Thibaut Feydy, Peter Stuckey, and Mark Wallace, elaborates on a satisfiability solving approach for the RCPSP/max. First, basic concepts such as finite domain propagation, boolean satisfiability solving, and lazy clause generation are discussed. Then, a basic model for the RCPSP/max and several expansions are described. The refinements refer to the reduction of the initial domains of the start time variables and the identification of incompatible activities that cannot be in progress simultaneously. The authors propose a branch-and-bound algorithm that is based on start-time and/or conflict-driven branching strategies and report on the results of an experimental performance analysis.

Part III focuses on resource-constrained project scheduling problems with alternative types of resource constraints. The different generalizations of the

renewable-resources concept allow for modeling various kinds of limited input factors arising in practical applications of project scheduling models. Chapter 8, written by Sönke Hartmann, considers the resource-constrained project scheduling problem with time-varying resource requirements and capacities RCPSP/t. After a formal description of the problem, relationships to other project scheduling problems are discussed and practical applications in the field of medical research and production scheduling are treated. The applicability of heuristics for the RCPSP to the more general RCPSP/t is analyzed and a genetic algorithm for solving the RCPSP/t is presented. In Chap. 9 Jacques Carlier and Aziz Moukrim consider project scheduling problems with storage resources. In particular, the general project scheduling problem with inventory constraints, the financing problem, and the project scheduling problem with material-availability constraints are discussed. For the general problem setting, in which for each storage resource the inventory level must be maintained between a given safety stock and the storage capacity, two exact methods from literature are reviewed. The financing problem corresponds to the single-resource case in which the occurrence times of the project events replenishing the storages are fixed and no upper limitation on the inventory levels are given. This problem can be solved by a polynomial-time shifting algorithm. Eventually, the authors explain how the general problem can be solved efficiently when the storage capacities are relaxed and a linear order on all depleting events is given. Chapter 10, written by Grzegorz Waligóra and Jan Węglarz, is concerned with the resource-constrained project scheduling problem with discrete and continuous resources DCRCPSP. First, the authors survey the main theoretical results that have been achieved for the continuous resource allocation setting. Then, the DCRCPSP with an arbitrary number of discrete resources and a single continuous resource with convex or concave processing rate, respectively, is analyzed. For the case of concave processing rates, a solution method based on feasible sequences of activity sets is presented. In Chap. 11 Ramon Alvarez-Valdes, Jose Manuel Tamarit, and Fulgencia Villa discuss the resource-constrained project scheduling problem with partially renewable resources RCPSP/π. After the definition of the problem, the authors review different types of requirements of real-world scheduling problems that can be modeled using partially renewable resources and survey the existing solution procedures for RCPSP/π. Preprocessing procedures and two heuristic approaches, a GRASP algorithm and a scatter search method, are treated in detail.

Part IV is devoted to preemptive project scheduling problems, in which activities can be temporarily interrupted and restarted at a later point in time. In some applications, especially if vacation or scheduled downtimes of resources are taken into account, the splitting of activities may be unavoidable. Chapter 12 by Sacramento Quintanilla, Pilar Lino, Ángeles Pérez, Francisco Ballestín, and Vicente Valls considers the resource-constrained project scheduling problem Maxnint_PRCPSP under integer activity preemption and upper bounds on the number of interruptions per activity. Existing procedures for the RCPSP are adapted to solve the Maxnint_PRCPSP, and procedures tailored to the Maxnint_PRCPSP are presented. In addition, the chapter reviews a framework for modeling different kinds of precedence relations when activity preemption is allowed. In Chap. 13 Christoph

Schwindt and Tobias Paetz first present a survey on preemptive project scheduling problems and solution methods. Next, they propose a continuous preemption resource-constrained project scheduling problem with generalized feeding precedence relations, which includes most of the preemptive project scheduling problems studied in the literature as special cases. Based on a reduction of the problem to a canonical form with nonpositive completion-to-start time lags between the activities, structural issues like feasibility conditions as well as upper bounds on the number of activity interruptions and the number of positive schedule slices are investigated. Moreover, a novel MILP problem formulation is devised, and preprocessing and lower bounding techniques are presented.

Area B of the handbook is dedicated to single-modal project scheduling problems with general objective functions, including multi-criteria problems. Non-regular objective functions motivated by real-world applications are, e.g., the net present value of the project, the resource availability cost, or different resource leveling criteria. In practice, project managers often have to pursue several conflicting goals. Traditionally, the respective scheduling problems have been tackled as single-objective optimization problems, combining the multiple criteria into a single scalar value. Recently, however, more advanced concepts of multi-criteria decision making received increasing attention in the project scheduling literature. Based on these concepts, project managers may generate a set of alternative and Pareto-optimal project schedules in a single run.

Part V treats project scheduling problems with single-criteria non-regular objective functions. These problems are generally less tractable than problems involving a regular objective function like the project duration because the set of potentially optimal solutions must be extended by non-minimal points of the feasible region. The resource-constrained project scheduling problem with discounted cash flows RCPSPDC is examined in Chap. 14. The sum of the discounted cash flows associated with expenditures and progress payments defines the net present value of the project, and the problem consists in scheduling the project in such as way that the net present value is maximized. Hanyu Gu, Andreas Schutt, Peter Stuckey, Mark Wallace, and Geoffrey Chu present an exact solution procedure relying on the lazy clause generation principle. Moreover, they propose a Lagrangian relaxation based forward-backward improvement heuristic as well as a Lagrangian method for large problem instances. Computational results on test instances from the literature and test cases obtained from a consulting firm provide evidence for the performance of the algorithms. In Chap. 15 Savio Rodrigues and Denise Yamashita present exact methods for the resource availability cost problem RACP. The RACP addresses situations in which the allocation of a resource incurs a cost that is proportional to the maximum number of resource units that are requested simultaneously at some point in time during the project execution. The resource availability cost is to be minimized subject to ordinary precedence relations between the activities and a deadline for the project termination. An exact algorithm based on minimum bounding procedures and heuristics for reducing the search space are described in detail. Particular attention is given to the search strategies and the selection of cut candidates. The authors report on computational results on

a set of randomly generated test instances. Chapter 16, written by Vincent Van Peteghem and Mario Vanhoucke, considers heuristic methods for the RACP and the RACPT, i.e., the RACP with tardiness cost. In the RACPT setting, a due date for the project completion is given and payments arise when the project termination is delayed beyond this due date. Van Peteghem and Vanhoucke provide an overview of existing meta-heuristic methods and elaborate on a new search algorithm inspired by weed ecology. In Chap. 17 Julia Rieck and Jürgen Zimmermann address different resource leveling problems RLP. Resource leveling is concerned with the problem of balancing the resource requirements of a project over time. Three different resource leveling objective functions are discussed, for which structural properties and respective schedule classes are revisited. A tree-based branch-and-bound procedure that takes advantage of the structural properties is presented. In addition, several mixed-integer linear programming formulations for resource leveling problems are given and computational experience on test sets from the literature is reported. In Chap. 18 Symeon Christodoulou, Anastasia Michaelidou-Kamenou, and Georgios Ellinas present a literature review on heuristic solution procedures for different resource leveling problems. For the total squared utilization cost problem they devise a meta-heuristic method that relies on a reformulation of the problem as an entropy maximization problem. First, the minimum moment method for entropy maximization is presented. This method is then adapted to the resource leveling problem and illustrated on an example project.

Part VI covers multi-criteria project scheduling problems, placing special emphasis on structural issues and the computation of the Pareto front. Chapter 19, written by Francisco Ballestín and Rosa Blanco, addresses fundamental issues arising in the context of multi-objective project scheduling problems. General aspects of multi-objective optimization and peculiarities of multi-objective resource-constrained project scheduling are revisited, before a classification of the most important contributions from the literature is presented. Next, theoretical results for time- and resource-constrained multi-objective project scheduling are discussed. In addition, the authors provide a list of recommendations that may guide the design of heuristics for multi-objective resource-constrained project scheduling problems. Chapter 20, contributed by Belaïd Aouni, Gilles d'Avignon, and Michel Gagnon, examines goal programming approaches to multi-objective project scheduling problems. After presenting a generic goal programming model, the authors develop a goal programming formulation for the resource-constrained project scheduling problem, including the project duration, the resource allocation cost, and the quantity of the allocated resources as objective functions. In difference to the classical resource allocation cost problem, the model assumes that the availability cost refers to individual resource units and is only incurred in periods during which the respective unit is actually used.

Area C of this handbook is devoted to multi-modal project scheduling problems, in which for each activity several alternative execution modes may be available for selection. Each execution mode defines one way to process the activity, and alternative modes may differ in activity durations, cost, resource requirements, or resource usages over time. The project scheduling problem is then complemented

by a mode selection problem, which consists in choosing one execution mode for each activity. Multi-modal problems typically arise from tradeoffs between certain input factors like renewable or nonrenewable resources, durations, or cost. Other types of multi-modal problems are encountered when multi-skilled personnel has to be assigned to activities with given skill requirements or when the resource requirements are specified as workloads rather than by fixed durations and fixed resource demands.

Part VII deals with multi-modal project scheduling problems in which the activity modes represent relations between activity durations and demands for renewable, nonrenewable, or financial resources. This problem setting allows for modeling resource-resource and resource-time tradeoffs, which frequently arise in practical project management. In Chap. 21 Marek Mika, Grzegorz Waligóra, and Jan Węglarz provide a comprehensive overview of the state of the art in multi-modal project scheduling. One emphasis of the survey is on the basic multi-mode resource-constrained project duration problem MRCPSP, for which they review mixed-integer linear programming formulations, exact and heuristic solution methods, as well as procedures for calculating lower bounds on the minimum project duration. Moreover, they also revisit special cases and extensions of the basic problem as well as multi-mode problems with financial and resource-based objectives. Chapter 22, written by José Coelho and Mario Vanhoucke, presents a novel solution approach to the multi-mode resource-constrained project scheduling problem MRCPSP, which solves the mode assignment problem using a satisfiability problem solver. This approach is of particular interest since it takes advantage of the specific capabilities of these solvers to implement learning mechanisms and to combine a simple mode feasibility check and a scheduling step based on a single activity list. A capital-constrained multi-mode scheduling problem is investigated in Chap. 23 by Zhengwen He, Nengmin Wang, and Renjing Liu. The problem consists in selecting activity modes and assigning payments to project events in such a way that the project's net present value is maximized and the cash balance does not go negative at any point in time. The execution modes of the activities represent combinations of activity durations and associated disbursements. In Chap. 24 Philipp Baumann, Cord-Ulrich Fündeling, and Norbert Trautmann consider a variant of the resource-constrained project scheduling problem in which the resource usage of individual activities can be varied over time. For each activity the total work content with respect to a distinguished resource is specified, and the resource usages of the remaining resources are determined by the usage of this distinguished resource. A feasible distribution of the work content over the execution time of an activity can be interpreted as an execution mode. The authors present a priority-rule based heuristic and a mixed-integer linear programming formulation, which are compared on a set of benchmark instances.

Part VIII addresses different variants of project staffing and scheduling problems. In those problem settings, the execution of a project activity may require several skills. It then becomes necessary to assign appropriate personnel to the activities and to decide on the skills with which they contribute to each activity. Isabel Correia and Francisco Saldanha-da-Gama develop a generic mixed-integer

programming formulation for project staffing and scheduling problems, which is presented in Chap. 25. The formulation captures various features like unary multi-skilled resources, which contribute with at most one skill to each activity, workload capacities of the resources, multi-unit skill requirements of the activities, and generalized precedence relations. This framework is illustrated by providing MILP models for two project staffing and scheduling problems discussed in the literature, the multi-skill project scheduling problem MSPSP and the project scheduling problem with multi-purpose resources PSMPR. In Chap. 26 Carlos Montoya, Odile Bellenguez-Morineau, Eric Pinson, and David Rivreau present a heuristic method for the MSPSP, which is based on integrating column generation and Lagrangian relaxation techniques. The MSPSP consists in assigning the multi-skilled resources to the activities so as to minimize the project duration under ordinary precedence relations between the activities. The authors develop two master problem formulations, which are heuristically solved by iteratively considering restricted versions of the master problem defined on a pool of variables. In each iteration, new variables with negative reduced cost are entered into the pool, which are identified via respective pricing problems. The required dual multipliers are obtained from solving the LP relaxation of the current restricted master problem by alternating iterations of a subgradient procedure for the Lagrangian dual and simplex iterations. Project staffing and scheduling problems of type PSMPR are discussed in Chap. 27. In difference to the MSPSP, the availability of each resource is limited by a maximum workload that can be processed in the planning horizon, and a general staffing cost function is considered. The staffing cost depends on the assignment of resources to skill requirements of the activities. Haitao Li devises an exact algorithm for the general problem with convex staffing cost. The hybrid Benders decomposition method starts from hierarchically dividing the problem into a relaxed master problem covering the assignment decisions and a feasibility subproblem modeling the scheduling decisions. Both levels are linked by top-down instructions and a bottom-up feedback mechanism adding Benders cuts to the relaxed master problem when the scheduling problem is infeasible. The feasibility of the scheduling problem is checked using a constraint programming algorithm. In Chap. 28 Cheikh Dhib, Ameur Soukhal, and Emmanuel Néron address a generalization of the MSPSP in which an activity can be interpreted as a collection of concurrent subactivities requiring a single skill each and possibly differing in durations. Moreover, it is assumed that the subactivities must be started simultaneously, but may be interrupted and resumed individually at integral points in time. The authors propose a mixed-integer linear programming formulation of the problem and describe priority-rule based solution methods, which are based on the parallel schedule-generation scheme.

Discrete time-cost tradeoff problems, which are the subject of **Part IX**, represent a type of multi-modal project scheduling problems that are frequently encountered in practice. This type of problems occur when the processing of certain activities can be sped up by assigning additional resources, leading to higher execution cost. In Chap. 29 Joseph Szmerekovsky and Prahalad Venkateshan provide a literature review on the classical discrete time-cost tradeoff problem DTCTP. Furthermore,

they discuss a new integer programming formulation for a version of the DTCTP with irregular start time costs of the activities. For the special case where the start time costs represent the net present value of an activity, the formulation is compared to three alternative MILP models in an extensive computational experiment. In Chap. 30 Mario Vanhoucke studies three extensions of the DTCTP and an electromagnetic meta-heuristic algorithm to solve these problems. The setting of the DTCTP with time-switch constraints presupposes that activities can only be processed in certain time periods defined by given work/rest patterns. In addition to the direct activity costs, the objective function of the DTCTP with work continuity constraints also includes costs for the supply of resources required by groups of activities; this variant of the problem can be reduced to the basic DTCTP. Finally, the DTCTP with net present value optimization is considered.

Area D of the handbook is dedicated to project planning problems involving several individual projects. We distinguish between multi-project scheduling problems, for which the set of projects to be scheduled is assumed given, and project portfolio selection problems, dealing with the choice of the projects to be actually performed. In both scenarios, there may exist dependencies between the individual projects, for example due to precedence relations between activities of different projects or due to the joint requirements for resources.

Part X deals with the first type of multi-project problems. When scheduling concurrent projects, an important question concerns the distribution of information. In the basic multi-project scheduling problem, it is assumed that all planning data are available to a single decision maker, who may centrally schedule the entire project portfolio. On the other hand, decentralized multi-project scheduling covers the situation in which information is distributed over different decision makers, who may pursue individual targets. In this case, a central coordination mechanism is needed to resolve conflicts between the individual projects. In Chap. 31 Jos Fernando Gonçalves, Jorge Jos de Magalhes Mendes, and Mauricio Resende provide a literature overview on basic multi-project scheduling problems BMPSPS. Furthermore, they develop a biased random-key genetic algorithm for the variant of the problem in which a separable polynomial function in the tardiness, the earliness, and the flow time overrun of all projects is to be minimized subject to precedence relations and the limited availability of shared resources. The decentralized multi-project scheduling problem DRCMPSP is addressed in Chap. 32. In their contribution, Andreas Fink and Jörg Homberger discuss implications of the distributed character of the problem. In addition, they provide a classification scheme of different types of DRCMPSP, categorizing problems according to the basic problem structure, the number of decision makers, the distribution of information, and the local and global objectives. The chapter also contains an extensive discussion and classification of solution approaches presented in literature, including auction and negotiation based coordination schemes.

Part XI focuses on project portfolio selection problems. Often there are more projects on offer than resources available to carry them out. In this case project management has to choose the right project portfolio for execution. In Chap. 33 Ana Fernández Carazo considers multi-criteria problems in which the performance

of a portfolio is measured according to a set of conflicting goals. First she identifies a number of key factors characterizing multi-criteria project portfolio selection problems and discusses the different ways in which those factors have been modeled in the literature. Based on this analysis, a proposal for a general project portfolio selection model is developed, which synthesizes various features of previous models. Finally, a binary nonlinear multi-criteria programming formulation of the new model is provided. Walter Gutjahr in Chap. 34 surveys models for project portfolio selection problems which include learning and knowledge depreciation effects. Different types of learning curves are reviewed and it is explained how these models have been used in the context of project staffing and scheduling problems. For the integration of skill development into project portfolio selection models, a mixed-integer nonlinear programming formulation is proposed. Moreover, analytical results for continuous project portfolio investment problems under skill development are reviewed, for which it is assumed that projects can also be partially funded.

Area E of the handbook covers the realm of project scheduling under uncertainty and vagueness, an issue that is widely recognized as being highly relevant to practical project management. Stochastic scheduling problems refer to decision situations under risk, in which quantities like activity durations or activity costs are defined as random variables with known distributions and the objective consists in optimizing the expected value of some performance measure. A solution to such a stochastic problem is commonly given by a policy that is applied when the project is executed. Robust project scheduling is concerned with the problem of finding a predictive baseline schedule that still performs well in case of disruptions or adverse scenarios. Interval uncertainty designates a situation in which only lower and upper bounds can be estimated with sufficient accuracy, but no probability distributions are known. Finally, the concept of fuzzy sets allows to model situations in which vague information, which is only available on an ordinal scale, should be taken into account.

Part XII addresses different types of stochastic project scheduling problems. Chapter 35, contributed by Wolfram Wiesemann and Daniel Kuhn, deals with the stochastic time-constrained net present value problem. Both the activity durations and the cash flows associated with the activities are supposed to be independent random variables. Having discussed the relevance and challenges of stochastic net present value problems, the authors review the state of the art for two variants of the problem. If the activity durations are assumed to be exponentially distributed, the problem can be modeled as a discrete-time Markov decision process with a constant discount rate, for which different exact solution procedures are available. Alternatively, activity durations and cash flows can be represented using discrete scenarios with given probabilities. The resulting stochastic net present value problem SNPV can be formulated as a mixed-integer linear program. Several heuristic solution approaches from literature are outlined. In Chap. 36 Evelina Klerides and Eleni Hadjiconstantinou examine the stochastic discrete time-cost tradeoff problem SDTCTP. They survey the literature on static and dynamic versions of the deadline and the budget variant of this problem. For the dynamic budget

variant of SDTCTP it is shown that the problem can be formulated as a multi-stage stochastic binary program with decision-dependent uncertainty. Furthermore, the authors present effective methods for computing lower bounds and good feasible solutions, which are respectively based on a two-stage relaxation and a static mode selection policy. The resource-constrained project scheduling problem with random activity durations SRCPSP is the subject of Chap. 37. Maria Elena Bruni, Patrizia Beraldi, and Francesca Guerriero give an overview of models and methods that have been proposed for different variants of this problem. They develop a heuristic based on the parallel schedule-generation scheme, which in each iteration determines the predictive completion times of the scheduled activities by solving a chance-constrained program. The presented approach is innovative in two respects. First, the use of joint probabilistic constraints allows to relax the traditional assumption that the start time of an activity can be disturbed by at most one predecessor activity at a time. Second, similar to robust project scheduling approaches, a solution to the problem is a predictive baseline schedule that is able to absorb a large part of possible disruptions. The objective, however, still consists, for given confidence level, in finding a schedule with minimum makespan. Hence, the problem to be solved can be viewed as a dual of a robust scheduling problem. The heuristic is illustrated on a real-life construction project. Chapter 38, by Saeed Yaghoubi, Siamak Noori, and Amir Azaron, tackles a multi-criteria multi-project scheduling problem in which projects arrive dynamically according to a Poisson process. Activity durations and direct costs for carrying out activities are assumed to be independent random variables. The execution of the projects is represented as a stochastic process in a queueing network with a maximum number of concurrent projects, each activity being performed at a dedicated service station. The expected values of the activity durations and the direct costs are respectively nonincreasing and nondecreasing functions of the amount of a single resource that is assigned to the service station. The problem consists in allocating the limited capacity of the resource in such a way that the mean project completion time is minimized, the utilization of the service stations is maximized, and the probability that the total direct cost exceeds the available budget is minimum. The authors apply continuous-time Markov processes and particle swarm optimization to solve this multi-objective problem using a goal attainment technique.

Part XIII comprises two chapters on robust optimization approaches to project scheduling problems under uncertainty. The basic idea of robust project scheduling consists in establishing a predictive baseline schedule with a diminished vulnerability to disturbances or adverse scenarios and good performance with respect to some genuine scheduling objective. There are many ways in defining the robustness of a schedule. For example, a schedule may be considered robust if it maximizes the probability of being implementable without modifications. Alternatively, the robustness may refer to the genuine objective instead of the feasibility; a robust schedule then typically optimizes the worst-case performance. In difference to stochastic project scheduling, robust project scheduling approaches do not necessarily presuppose information about the probability distributions of the uncertain input parameters of the problem. In Chap. 39 Öncü Hazır, Mohamed Haouari, and

Erdal Erel discuss a robust discrete time-cost tradeoff problem in which for the activity cost associated with a given mode an interval of possible realizations is specified, but no probability distribution is assumed to be known. The authors devise a mixed-integer programming formulation for this problem. The objective function is defined to be the sum of all most likely activity mode costs plus the maximum surplus cost that may be incurred if for a given number of activities, the direct cost does not assume the most likely but the highest value. The latter number of activities may be used to express the risk attitude of the decision maker. In addition, six categories of time-based robustness measures are presented and a two-phase scheduling algorithm for placing a project buffer at minimum additional cost is outlined. Based on this algorithm, the relationship between the required budget augmentation and the average delay in the project completion time can be analyzed. The robust resource-constrained project scheduling problem with uncertain activity durations is investigated in Chap. 40 by Christian Artigues, Roel Leus, and Fabrice Talla Nobibon. Like in the preceding chapter, it is assumed that no probability distributions are available; the sets of possible realizations of activity durations may form intervals or finite sets. The problem is formulated as a minimax absolute-regret model for which the objective is to find an earliest start policy that minimizes the worst-case difference between the makespan obtained when implementing the policy and the respective optimum ex-post makespan. An exact scenario-relaxation algorithm and a scenario-relaxation based heuristic are presented for this problem.

Part XIV is devoted to project scheduling problems under interval uncertainty and to fuzzy project scheduling. In Chap. 41 Christian Artigues, Cyril Briand, and Thierry Garaix survey results and algorithms for the temporal analysis of projects for which the uncertain activity durations are represented as intervals. The temporal analysis computations provide minimum and maximum values for the earliest and latest start times of the activities and the total floats. Whereas the earliest start times can be calculated as longest path lengths like in the case of fixed activity durations, the computation of the latest start times is less simple. Two algorithms with polynomial time complexity are presented. Interestingly, the maximum total float of the activities can also be computed efficiently, whereas the computation of the minimum total floats constitutes an $\mathcal{NP}$-hard problem. The chapter elaborates on a recent branch-and-bound algorithm for the latter problem. Hua Ke and Weimin Ma in Chap. 42 study a fuzzy version of the linear time-cost tradeoff problem in which the normal activity durations are represented as fuzzy variables. The authors survey literature on time-cost tradeoff problems under uncertainty and vagueness. Using elements of credibility theory, the concepts of expected values, quantiles, and probabilistic constraints can be translated from random to fuzzy variables. Based on these concepts, three fuzzy time-cost tradeoff models are proposed, respectively, providing schedules with minimum α-quantile of the total cost, with minimum expected cost, and with maximum credibility of meeting the budget constraint. In addition, a hybrid method combining fuzzy simulations and a genetic algorithm for solving the three models is presented.

Area F addresses managerial approaches to support decision makers faced with increasingly complex project environments. Complex challenges arise, for example,

when dealing with project portfolios, or when a project is performed on a client-contractor basis and the goals of both parties must be streamlined, or when risks arise from several sources and these risks are not independent from each other. These and further challenges are discussed in the two parts of Area F.

Part XV is concerned with general project management issues, covering project portfolio management, relational partnerships and incentive mechanisms, and specific challenges encountered in product development and engineering projects. In Chap. 43 Nicholas Hall contrasts the rapid growth of project activities in firms with the lack of trained project management professionals and research-based project management concepts. He proposes 11 areas for future research to reduce the gap between the great practical importance and the limited theoretical foundations of project management in these areas. Chapter 44 by Peerasit Patanakul addresses issues that arise in multi-project environments. These issues comprise the assignment of project managers to projects, organizational factors that enhance multi-project management, and alternative roles of a project management office. New product development constitutes a classical application area of project management procedures and tools. Nevertheless, managing product innovation is still a challenging task, due to the uncertainty associated with the development process and the strategic importance of its success. In Chap. 45 Dirk Pons provides guidelines from a systems engineering perspective, emphasizing on the management of human resources in the development process. Another traditional application area of project management is the construction industry. Construction projects involve two main parties: the contractor and the client receiving the project deliverables provided by the contractor. The concept of partnering tries to overcome the adversarial relation between contractor and client, which still tends to prevail in many construction projects. In Chap. 46 Hemanta Doloi examines key factors that are crucial for successful partnering and draws conclusions from a survey conducted in the Australian construction industry. Chapter 47, written by Xianhai Meng, deals with incentive mechanisms, which are frequently used to enhance project performance, especially in the construction industry. The author discusses different kinds of incentives and disincentives that are related to project goals such as time, cost, quality, and safety. A case study of a road construction project gives insight into the practical application of incentive mechanisms. Project complexity is a prominent cause for project failure. Hence, it is vitally important for managers to know about sources of complexity. In Chap. 48 Marian Bosch-Rekveldta, Hans Bakker, Marcel Hertogh, and Herman Mooi identify drivers of complexity. Based on a literature research and six case studies analyzing the complexity of engineering projects, they provide a framework for evaluating project complexity. The framework comprises technical, organizational, and external sources of project complexity.

Part XVI deals with project risk management. Since the importance of projects has grown and revenues from project work may constitute a considerable share of a firm's total income, managing project risk is vitally important as it helps to identify threats and to mitigate potential damage. In Chap. 49, Chao Fang and Franck Marle outline a framework for project risk management, which considers not only single risks separately but also interactions between risks. The authors

show how interactions can be captured in a matrix-based risk network and provide a quantitative method to analyze such a network. Chapter 50 is concerned with risk management for software projects. Paul Bannerman reviews empirical research on the application of risk management in practice, the effectiveness of risk management, and factors that hinder or facilitate the implementation of risk management. He describes different perspectives on risk management in order to show the wide range of approaches and to identify avenues for further research. An important goal of risk management is to identify risks and to decide on the risks that should be mitigated. This decision is frequently based on a ranking of the identified risks. In Chap. 51 Stefan Creemers, Stijn Van de Vonder, and Erik Demeulemeester survey the different ranking methods that were proposed in the literature. In particular, they consider so-called ranking indices that provide a ranking of activities or risks based on their impact on the project objectives. They show that the ranking methods may differ in their outcome and evaluate their performance with a focus on the risk of project delay.

The last **Area G** proves evidence for the relevance of concepts developed in the preceding parts of this handbook to the practice of project management and scheduling. The area covers different domains beyond proper project scheduling and puts the concepts treated in the previous parts into the perspective of real-life project management. It includes chapters on project scheduling applications, case studies, and project management information systems.

Part XVII collects six industrial applications of resource-constrained project scheduling, where different models and methods presented in previous chapters are put into practice. In particular, test, production, and workflow scheduling problems are considered. Chapter 52, written by Jan-Hendrik Bartels and Jürgen Zimmermann, reports on the problem of scheduling destructive tests in automotive R&D projects. The planning objective consists in minimizing the number of required experimental vehicles. The problem is modeled as a multi-mode resource-constrained project scheduling problem with renewable and storage resources, in which the required stock must be built up before it can be consumed. In addition to different variants of a priority-rule based heuristic, an activity-list based genetic algorithm is proposed. Both heuristic approaches prove suitable for solving large-scale practical problem instances. In Chap. 53 Roman Čapek, Přemysl Šůcha, and Zdeněk Hanzálek describe a scheduling problem with alternative process plans, which arises in the production of wire harnesses. In such a production process, alternative process plans include production operations that can be performed in different ways, using fully or semi-automated machines. A mixed-integer linear programming model for a resource-constrained project scheduling problem with generalized precedence relations, sequence-dependent setup times, and alternative activities is presented. Furthermore, a heuristic schedule-construction procedure with an unscheduling step is proposed, which can be applied to large problem instances. Chapter 54 is concerned with the scheduling of jobs with large computational requirements in grid computing. An example of such jobs are workflow applications, which comprise several precedence-related computation tasks. A computer grid is a large-scale, geographically distributed, dynamically

reconfigurable, and scalable hardware and software infrastructure. Marek Mika and Grzegorz Waligóra present three models for scheduling the computation and transmission tasks in grids, differing in their assumptions with respect to the workflow applications and computer networks. For the models with distributed resources and sequence-dependent setup times, resource allocation and scheduling algorithms are presented. For the model in which transmission tasks compete for scarce network resources it is shown how a feasible resource allocation can be determined. Chapter 55 by Haitao Li considers make-or-buy and supplier selection problems arising in conjunction with the scheduling of operations in make-to-order supply chains. A multi-mode resource-constrained project scheduling problem is formulated to minimize the total supply chain cost, in which synergies and interactions between sourcing and scheduling decisions are captured. The total supply chain cost involves the total fixed cost, cost of goods sold, and total pipeline stock cost and depends on the selected activity modes. The proposed solution algorithm draws on the hybrid Benders decomposition framework exposed in Chap. 27. The relaxed master problem (RMP) covers the assignment decisions, whereas the subproblem (SP) is concerned with the scheduling of the operations. The feasibility of an optimal RMP solution is checked by solving the respective SP. If the SP is feasible, an optimal solution has been found; otherwise, the algorithm identifies some cause of infeasibility and adds respective cuts to the RMP, which is then solved again. A numerical example is discussed to demonstrate the scope and depth of decision-support offered by the solutions of the model for purchasing and program managers. In Chap. 56 Arianna Alfieri and Marcello Urgo apply a project scheduling approach to make-to-order systems for special-purpose machinery like instrumental goods or power generation devices, in which products are assembled in the one-of-a-kind production mode. They present a resource-constrained project scheduling problem with feeding precedence relations and work content constraints and explain its application to a real-world case of machining center production. In Chap. 57 Matthew Colvin and Christos Maravelias apply multi-stage stochastic programming to the development process of new drugs. The problem consists in scheduling a set of drugs, each of which has to undergo three trials. If one trial fails, the development of the related drug is canceled. The required resources are limited and the objective is to maximize the expected net present value of the project. After an introduction to stochastic programming and endogenous observations of uncertainty, a mixed-integer multi-stage stochastic programming model is presented. Some structural properties of the problem are discussed and three solution methods including a branch-and-cut algorithm are developed.

Part XVIII presents two case studies in project scheduling. In Chap. 58 Maurizio Bevilacqua, Filippo Ciarapica, Giovanni Mazzuto, and Claudia Paciarotti combine concepts of robust project scheduling and multi-criteria project scheduling to tackle a construction project for an accommodation module of an oil rig in the Danish North Sea. To guarantee an efficient use of the resources, the project management identified the minimization of the project duration and the leveling of the manpower resources as primary goals. Using historical data from 15 past projects, the means and the standard deviations of the activity durations could be

estimated with sufficient accuracy. To obtain a robust baseline schedule for the project, project buffers and feeding buffers were inserted in the schedule according to the lines of Goldratt's Critical Chain methodology. Compared to the traditional CPM method, the presented robust goal programming approach was able to reduce the project duration by 14 % and to improve the resource utilization by more than 40 %. In Chap. 59 Jiuping Xu and Ziqiang Zeng consider a multi-criteria version of the discrete time-cost tradeoff problem, which is called the discrete time-cost-environment-tradeoff problem DTCETP. They assume that normal activity durations are represented as triangular fuzzy numbers and that for each period there exists a limit on the total cost incurred by the processing and crashing of activities. This cash flow constraint can be modeled as a renewable resource whose capacity coincides with the cost limit. The capacity is taken up according to the requirements of alternative execution modes. In sum, the problem can be formulated as a fuzzy multi-criteria multi-mode resource-constrained project scheduling problem. Four objective functions are taken into account: the total project cost, the project duration, the total crashing costs of activities, and the quantified environmental impact of the project. Xu and Zeng develop an adaptive hybrid genetic algorithm for this problem and describe its application to the Jinping-II hydroelectric construction project on the Yalong River in the Sichuan-Chongqing region. Both the input data of the case study and the computed schedule are provided. The performance of the algorithm is evaluated based on a sensitivity analysis with respect to the objective weights and the results obtained with two benchmark heuristics.

Project management information systems PMIS play a crucial role in the transfer of advanced project management and scheduling techniques to professional project management. **Part XIX** addresses the question of the actual contribution of PMIS on the project performance, studies the effects of PMIS on decision making in multi-project environments, and investigates the project scheduling capabilities of commercial PMIS.

Based on a PMIS success model and a survey conducted among project managers, Louis Raymond and François Bergeron in Chap. 60 empirically assess the impact of PMIS on decision makers and project success. Their model comprises five constructs: the quality of the PMIS, the quality of the PMIS information output, the use of the PMIS, the individual impacts of the PMIS, and the impacts of the PMIS on project success. Each construct is measured using several criteria. Structural equation modeling with the partial least squares method is used to analyze the relationships between the different dimensions and to test the validity of six research hypotheses. The results obtained show that the use of PMIS in professional project management significantly contributes to the efficiency and effectiveness of individual project managers and to the overall project performance. Chapter 61 presents a related study in which Marjolein Caniëls and Ralph Bakens focus on the role of PMIS in multi-project environments, where project managers handle multiple concurrent but generally less complex projects. After a survey of the literature on multi-project management and PMIS the research model is introduced, which contains six constructs: the project overload, the information overload, the PMIS information quality, the satisfaction with PMIS, the use of PMIS information,

Table 2 Overview of project scheduling problems treated in the handbook, respective acronyms used in the literature, and three-field notations of Brucker et al. (1999)

Chaps.	Project scheduling problem	Acronym	Three-field notation
1 – 4	Resource-constrained project scheduling problem	RCPSP	$PS \mid prec \mid C_{max}$
5 – 7	Resource-constrained project scheduling problem with generalized precedence relations	RCPSP/max	$PS \mid temp \mid C_{max}$
8	Resource-constrained project scheduling problem with time-varying resource requirements and capacities	RCPSP/t	$PSt \mid prec \mid C_{max}$
9	Project scheduling problems with storage resources		$PSs \mid temp \mid C_{max}$
10	Discrete-continuous resource-constrained project scheduling problem	DCRCPSP	$PSc \mid prec \mid C_{max}$
11	Resource-constrained project scheduling problem with partially renewable resources	RCPSP/π	$PSp \mid prec \mid C_{max}$
12	Integer preemptive resource-constrained project scheduling problem with limited number of interruptions per activity	Maxnint_ PRCPSP	$PS \mid prec, l\text{-}pmtn/int \mid C_{max}$
13	Continuous preemptive resource-constrained project scheduling problem with generalized precedence relations	PRCPSP/max	$PS \mid temp, pmtn \mid C_{max}$
14	Resource-constrained project scheduling problem with discounted cash flows	RCPSPDC	$PS \mid prec, \overline{d} \mid \Sigma c_i^F \beta^{C_i}$
15	Resource availability cost problem	RACP	$PS\infty \mid prec, \overline{d} \mid \Sigma c_k \max r_{kt}$
16	Resource availability cost problems	RACP, RACPT	$PS\infty \mid prec, \overline{d} \mid \Sigma c_k \max r_{kt}$, $PS\infty \mid prec \mid \Sigma c_k \max r_{kt} + wT$
17	Resource leveling problems	RLP	$PS\infty \mid temp, \overline{d} \mid \Sigma c_k \Sigma r_{kt}^2$, $PS\infty \mid temp, \overline{d} \mid \Sigma c_k \Sigma o_{kt}$, and $PS\infty \mid temp, \overline{d} \mid \Sigma c_k \Sigma \Delta r_{kt}$
18	Resource leveling problem	RLP	$PS\infty \mid prec, \overline{d} \mid \Sigma c_k \Sigma r_{kt}^2$
19	Multi-objective time- and resource-constrained project scheduling problems	MOPSPs, MORCPSPs	$PS\infty \mid prec \mid mult$, $PS \mid prec \mid mult$
20	Multi-objective resource-constrained project scheduling	MORCPSPs	$PS \mid prec \mid mult$

(continued)

Table 2 (continued)

Chaps.	Project scheduling problem	Acronym	Three-field notation
21	Multi-modal resource-constrained project scheduling problems		$MPS \mid prec \mid f$
22	Multi-mode resource-constrained project scheduling problem	MRCPSP	$MPS \mid prec \mid C_{max}$
23	Multi-mode capital-constrained net present value problem	MNPV	$MPSs \mid prec \mid \Sigma c_i^F \beta^{C_i}$
24	Project scheduling problem with work content constraints		$PSf \mid prec \mid C_{max}$
25	Project staffing and scheduling problems		$PSS \mid temp \mid f$
26	Multi-skill project scheduling problem	MSPSP	$PSS\infty \mid prec \mid C_{max}$
27	Project scheduling with multi-purpose resources	PSMPR	$PSS \mid temp \mid staff$
28	Preemptive multi-skill project scheduling problem		$PSS \mid prec, pmtn \mid C_{max}$
29	Discrete time-cost tradeoff problem (deadline version)	d-DTCTP	$MPS\infty \mid prec, \overline{d} \mid \Sigma c_i(p_i)$
	Discrete time-cost tradeoff problem with irregular starting time costs		$MPS\infty \mid prec, \overline{d} \mid f$
30	Discrete time-cost tradeoff problem with time-switch constraints	d-DTCTP-tsc	$MPS\infty \mid prec, \overline{d}, cal \mid \Sigma c_i(p_i)$
	Discrete time-cost tradeoff problem with net present value optimization	d-DTCTP-npv	$MPS\infty \mid prec, \overline{d} \mid \Sigma c_i^F \beta^{C_i}$
31	Basic multi-project scheduling problem	BMPSP	$PS \mid mult, prec \mid f$
32	Decentralized multi-project scheduling problem	DRCMPSP	
33	Multi-criteria project portfolio selection problem		
34	Project selection, scheduling, and staffing with learning problem	PSSSLP	
35	Stochastic net present value problem	SNPV	$PS \mid prec, p_i = sto \mid \Sigma c_i^F \beta^{C_i}$
36	Stochastic discrete time-cost tradeoff problem (budget version)	b-SDTCTP	$MPS\infty \mid prec, bud, p_i = sto \mid C_{max}$
37	Stochastic resource-constrained project scheduling problem	SRCPSP	$PS \mid prec, p_i = sto \mid C_{max}$
38	Markovian multi-criteria multi-project resource-constrained project scheduling problem		$MPSm, 1, 1 \mid mult, prec, bud, p_i = sto, c_i = sto, Poi \mid mult$

(continued)

Table 2 (continued)

Chaps.	Project scheduling problem	Acronym	Three-field notation
39	Robust discrete time-cost tradeoff problem		$MPS\infty \mid prec, \overline{d}, c_i = unc \mid \Sigma c_i(p_i)$
40	(Absolute regret) Robust resource-constrained project scheduling problem	AR-RCPSP	$PS \mid prec, p_i = unc \mid rob$
41	Temporal analysis under interval uncertainty		$PS\infty \mid prec, p_i = unc \mid f$ with $f \in \{ES_i, LS_i, TF_i\}$
42	Fuzzy time-cost tradeoff problem (deadline version)		$MPS\infty \mid prec, \overline{d}, p_i = fuz \mid \Sigma c_i(p_i)$
52	Multi-mode resource-constrained project scheduling problem with storage resources		$MPSs \mid temp, \overline{d} \mid \Sigma c_k \max r_{kt}$
53	Resource-constrained project scheduling problem with generalized precedence relations, sequence dependent setup times, and alternative activities	RCPSP-APP	$PS \mid temp, s_{ij}, nestedAlt \mid C_{max}$
54	Multi-mode resource-constrained project scheduling problems	MRCPSP	$MPS \mid prec \mid C_{max}$
55	Multi-mode resource-constrained project scheduling problem		$MPS \mid prec, \overline{d} \mid mac$
56	Resource constrained project scheduling problem with feeding precedence relations and work content constraints		$PSft \mid feed \mid C_{max}$
57	Stochastic net present value problem in which the set of activities to be executed is stochastic		$PS \mid prec, act = sto \mid \Sigma c_i^F \beta^{C_i}$
58	Robust multi-criteria project scheduling problem		$PS \mid prec, p_i = sto \mid C_{max}/\Sigma r_{kt}^2$
59	Fuzzy multi-criteria multi-mode project scheduling problem	DTCETP	$MPS \mid prec, \overline{d}, bud, p_i = fuz \mid mult$

and the quality of decision making. Based on the results of a survey among project managers, several hypotheses on the relationships between the constructs are tested using the partial least square method. It turns out that project and information overload are not negatively correlated with PMIS information quality and that the quality and use of PMIS information are strongly related to the quality of decision making. In the final Chap. 62, Philipp Baumann and Norbert Trautmann experimentally assess the performance of eight popular PMIS with respect to their project scheduling capabilities. Using the more than 1.500 KSD-30, KSD-60, and KSD-120 instances of the resource-constrained project scheduling problem RCPSP from the PSPLIB library, the impact of different complexity parameters and priority rules on the resulting project durations is analyzed. The results indicate that for the project duration criterion, the scheduling performances of the software packages

differ significantly and that the option of selecting specific priority rules generally leads to schedules of inferior quality as compared to PMIS that do not offer this feature.

Table 2 gives an overview of the different types of project scheduling problems treated in this book. In the literature many of those problems are commonly designated by acronyms, which are provided in the third column of the table. The last column lists the respective designators of the (extended) three-field classification scheme for project scheduling problems proposed by Brucker et al. (1999). The notation introduced there and the classification scheme, which are used in different parts of this handbook, are defined in the list of symbols, which is included in the front matter of this book.

References

Artigues C, Demassey S, Néron E (eds) (2008) Resource-constrained project scheduling: models, algorithms, extensions and applications. Wiley, Hoboken

Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999) Resource-constrained project scheduling: notation, classification, models, and methods. Eur J Oper Res 112:3–41

Dinsmore PC, Cooke-Davies TJ (2005) Right projects done right: from business strategy to successful project implementation. Wiley, San Francisco

Demeulemeester EL, Herroelen WS (2002) Project scheduling: a research handbook. Kluwer, Dordrecht

Hartmann S, Briskorn D (2008) A survey of deterministic modeling approaches for project scheduling under resource constraints. Eur J Oper Res 207:1–14

Józefowska J, Węglarz J (eds) (2006) Perspectives in modern project scheduling. Springer, New York

Kerzner H (2013) Project management: a systems approach to planning, scheduling, and controlling. Wiley, Hoboken

Klein R (2000) Scheduling of resource-constrained projects. Kluwer, Boston

Lewis JP (1997) Fundamentals of project management. Amacom, New York

Project Management Institute, Inc. (2013) A guide to the project management body of knowledge (PMBOK®Guide). PMI, Newtown Square

Turner JR (2009) The handbook of project-based management: leading strategic change in organizations. McGraw-Hill, New York

Part X
Multi-Project Scheduling

Chapter 31
The Basic Multi-Project Scheduling Problem

José Fernando Gonçalves, Jorge José de Magalhães Mendes, and Mauricio G.C. Resende

Abstract In this chapter the Basic Multi-Project Scheduling Problem (BMPSP) is described, an overview of the literature on multi-project scheduling is provided, and a solution approach based on a biased random-key genetic algorithm (BRKGA) is presented. The BMPSP consists in finding a schedule for all the activities belonging to all the projects taking into account the precedence constraints and the availability of resources, while minimizing some measure of performance. The representation of the problem is based on random keys. The BRKGA generates priorities, delay times, and release dates, which are used by a heuristic decoder procedure to construct parameterized active schedules. The performance of the proposed approach is validated on a set of randomly generated problems.

Keywords Genetic algorithm • Meta-heuristics • Multi-project scheduling • Random keys

31.1 Introduction

Managing multiple projects is a complex decision-making process, where a number of projects must share concurrently a set of limited resources. Examples of multi-project environments are new product development, multi-product manufacturing, infrastructure constructions, and maintenance of systems.

J.F. Gonçalves (✉)
LIAAD, INESC TEC, Faculdade de Economia da Universidade do Porto, Porto, Portugal
e-mail: jfgoncal@fep.up.pt

J.J.M. de Mendes
Instituto Superior de Engenharia do Porto, Depto. de Engenharia Civil, Porto, Portugal
e-mail: jjm@isep.ipp.pt

M.G.C. Resende
Algorithms and Optimization Research Department, AT&T Labs Research, Florham Park, NJ, USA
e-mail: mgcr@research.att.com

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_1

The basic multi-project scheduling problem (BMPSP) can be considered an extension of the well-known resource constrained project scheduling problem (RCPSP) where two or more projects which require the same scarce resources are scheduled simultaneously.

There are two main distinguished research fields in multi-project scheduling—the static and the dynamic project environment (Dumond and Mabert 1988). In this chapter we assume a closed project portfolio, which does not change over time. The BMPSP in a static environment has been studied, amongst others, by Fendley (1968), Pritsker et al. (1969), Kurtulus and Davis (1982), Kurtulus and Narula (1985), Lawrence and Morton (1993), Lova et al. (2000), Lova and Tormos (2001, 2002), Gonçalves et al. (2008), Krüger and Scholl (2010), Browning and Yassine (2010), Kumanam and Raja (2011), and Cai and Li (2012).

The existing solution methods apply either a single- or a multi-project approach. The single-project approach is equivalent to the RCPSP, since it merges all projects of the multi-project into an artificial super-project with dummy start and end activities. The multi-project approach keeps the projects separate. The approach considered in this chapter uses a single-project approach.

Scheduling involves the allocation of the given resources to projects to determine the start and completion times of a set of detailed activities. There may be multiple projects contending for limited resources, which makes the solution process more complex. The allocation of scarce resources then becomes a major objective of the problem and several compromises have to be made to solve the problem to the desired level of near-optimality.

In this chapter, we present a biased random-key genetic algorithm (BRKGA) approach to solve the BMPSP. The remainder of the chapter is organized as follows. Section 31.2 describes the problem and presents a conceptual model and Sect. 31.3 reviews the literature. Section 31.4 describes the approach used to solve the BMPSP. Section 31.5 describes the parameterized schedule-generation procedure and Sect. 31.6 reports on some computational experiments. Concluding remarks are made in Sect. 31.7.

31.2 Problem Description

The BMPSP consists of a set Q of projects, where each project $q \in Q$ is composed of activities $j \in V_q$, where activities α_q and ω_q are dummies and represent, respectively, the initial and final activities of project q. Let V be the set of all activities and let $\mathscr{R} = \{1, \ldots, K\}$ represent the set of renewable resources. While being processed, activity $j \in V$ requires r_{jk} units of resource $k \in \mathscr{R}$ during each time instant of its non-preemptable duration p_j. Resource $k \in \mathscr{R}$ has a limited availability of R_k at any point in time. Parameters p_j, r_{jk}, and R_k are assumed to be non-negative and deterministic. The activities are interrelated by the following two kinds of constraints:

- *Precedence constraints*, which force each activity $i \in V$ to be scheduled after all predecessor activities $j \in Pred(i)$ are completed;
- *Resource constraints,* which assure that the processing of the activities is subject to the availability of resources with limited capacities.

For the start and end activities of each project q, we have, for all $q \in Q$, that

$$p_{\alpha_q} = p_{w_q} = 0 \quad \text{and} \quad r_{\alpha_q k} = r_{w_q k} = 0 \quad (k \in \mathscr{R})$$

Activities 0 and $n+1$ are dummy activities, have no duration, and correspond to the start and end of all projects.

The BMPSP consists in finding a schedule for all the activities taking into account precedence constraints and the availability of resources, while minimizing some measure of performance. Let C_j represent the finish time of activity $j \in V$. A schedule can be represented by a vector of finish times $(C_1, \ldots, C_{n+1})$. Let $\mathscr{A}(t)$ be the set of activities being processed at time t. The conceptual model of the BMPSP can be described as

$$\text{Min.} \quad \textit{Measure of Performance}\ (C_1, \ldots, C_n) \tag{31.1}$$

s.t.

$$C_i \leq C_j - p_j \quad (j \in V;\ i \in Pred(j)) \tag{31.2}$$

$$\sum_{j \in \mathscr{A}(t)} r_{jk} \leq R_k \quad (k \in \mathscr{R};\ t \geq 0) \tag{31.3}$$

$$C_j \geq 0 \quad (j \in V) \tag{31.4}$$

According to objective (31.1) we seek to minimize some performance measure. Constraints (31.2) impose the precedence relations between activities, and constraints (31.3) limit the resource usage imposed by the activities being processed at time t to the available capacity. Finally, constraints (31.4) force the finish times to be non-negative.

A variety of measures of performance have been used for the BMPSP. Minimization of project duration has been used widely (Baker 1974). Other measures of performance include: minimization of total project delay, lateness, or tardiness (Kurtulus and Davis 1982), minimization of average project delay (Lova and Tormos 2001), minimization of total lateness or lateness penalty (Kurtulus 1985), minimization of the overall project cost (Talbot 1982), minimization the total cost of delay (Kurtulus and Narula 1985), and maximization of the resource leveling (Woodworth and Willie 1975). In this chapter, we seek to minimize a measure of performance which involves the due date (tardiness), starting time (earliness), and work in process (flow time) of each project (Gonçalves et al. 2008). This performance measure simultaneously incorporates three criteria: tardiness, earliness, and flow time and is described below. The following notation will be used:

$\hat{p}_q$: Target duration for project q.
d_q: Due date for project q.
C_q: Conclusion date for project q in the generated schedule.
S_q: Start date for project q in the generated schedule.
T_q: Tardiness of project $q = \max\{C_q - d_q\ ,\ 0\}$.
E_q: Earliness of project $q = \max\{d_q - C_q\ ,\ 0\}$.
FD_q: Flow time deviation for project $q = \max\{C_q - S_q - \hat{p}_q,\ 0\}$.
LB_0^q: Critical path length of project q.

the performance measure is defined as

$$w^T \sum_q T_q^3 \quad + \quad w^E \sum_q E_q^2 \quad + \quad w^{FD} \sum_q FD_q^2 \tag{31.5}$$

where w^T, w^F, and w^{FD} are parameters defined by the decision maker. Note that the tardiness has an exponent equal to 3 because in the real-world it is considered more important than the earliness or the flow-time (which have an exponent equal to 2). To overcome the problem of not knowing the target duration of a project in a real-world situation, we replace

$$w^T \sum_q FD_q^2 \quad \text{by} \quad w^T \sum_q \frac{\left(C_q - S_q\right)^2}{LB_0^q}$$

In the above model, the constraints for the resources are expressed by condition (31.3). However, there are others types of constraints related with the start of a project which cannot be modeled by that condition. To be able to model this kind of constraint, we add

$$C_{\alpha_q} \geq ES_q \qquad (q \in Q)$$

to the model, where ES_q represents earliest release date for project q. These constraints are enforced in the model implicitly by assigning to the initial activity of each project a duration $ES_q \geq \overline{ES}_q$, i.e.,

$$p_{\alpha_q} = ES_q \geq \overline{ES}_q \qquad (q \in Q)$$

31.3 Literature Review

The BMPSP is a generalization of the RCPSP. Blazewicz et al. (1983) show that the RCPSP, as a generalization of the classical job shop scheduling problem, belongs to the class of $\mathcal{NP}$-hard optimization problems. Therefore the BMPSP, as a generalization of the RCPSP, is also $\mathcal{NP}$-hard.

Exact methods to solve the BMPSP are proposed in the literature. The pioneering work of multi-project scheduling by Pritsker et al. (1969) proposed a zero-one programming approach. Mohanthy and Siddiq (1989) studied the problem of assigning due dates to the projects in a multi-project environment. Drexl (1991) considered a non-preemptive variant of the resource-constrained assignment problem using a hybrid branch-and-bound/dynamic programming algorithm with a Monte Carlo-type upper bounding heuristic. Deckro et al. (1991) formulated the BMPSP as a block angular general integer programming model and employed a decomposition approach to solve large problems. Vercellis (1994) describes a Lagrangian decomposition technique for solving multi-project planning problems with resource constraints and alternative modes of performing each activity in the projects.

Several approaches to the BMPSP using heuristic methods have been proposed in the literature. For example, Fendley (1968) used multi-projects with three and five projects and considered three efficiency measurements in the computational analysis. Kurtulus and Davis (1982) designed multi-project instances whose projects have between 34 and 63 activities and resource requirements for each activity between two and six units.

Kurtulus and Narula (1985) studied penalties due to project delay. Dumond and Mabert (1988) studied the problem of assigning due dates to the projects in a multi-project environment. Tsubakitani and Deckro (1990) proposed a heuristic for multi-project scheduling with resource constraints using the Kurtulus and Davis (1982) approach to select appropriate heuristic decision rules. Bock and Patterson (1990) designed a computational experiment based on the work of Dumond and Mabert (1988) with three factors. Lawrence and Morton (1993) studied the due date setting problem of scheduling multiple resource-constrained projects with the objective of minimizing weighted tardiness costs. Shankar and Nagi (1996) proposed a two-level hierarchical approach consisting of the planning and scheduling stages.

Özdamar et al. (1998) examined different dispatching rules for the tardiness and the net present value objective embedded in a multi-pass heuristic. Ash (1999) proposed a deterministic simulation scheme using available project data to choose an activity scheduling heuristic which not only allows for the establishment of good project schedules, but determines a priori which resources will be assigned to specific project activities.

Lova et al. (2000) developed a multi-criteria heuristic that, lexicographically, improves two criteria: a temporal criterion (mean project delay in relation to the unconstrained critical path duration or multi-project duration increase) and a non-temporal criterion (project splitting, in-process inventory, resource leveling, or idle resources) that can be chosen by the user.

Mendes (2003) presents a genetic algorithm that uses a random-key representation and a modified parallel schedule-generation scheme (SGS).

31.4 Biased Random-Key Genetic Algorithm

We begin this section with an overview of the proposed solution process. This is followed by a discussion of the biased random-key genetic algorithm, including detailed descriptions of the solution encoding and decoding, evolutionary process, and fitness function.

31.4.1 Overview

Considering the difficulty to solve real-world problems with exact methods, a new solution approach is developed that combines a genetic algorithm with a schedule-generation scheme (SGS) that creates parameterized active schedules. The SGS constructs a schedule based on the priorities and delay times of the activities, and the release dates of the projects.

The role of the genetic algorithm (GA) is to evolve the encoded solutions, or *chromosomes*, which encode the vectors of priorities (Π) and delays (Δ) of the activities and the vector of project release dates (ES). For each chromosome, the following phases are applied to decode the chromosome:

1. *Decoding of the priorities*. This phase transforms a part of the chromosome supplied by the genetic algorithm into the vector of activity priorities (Π).
2. *Decoding of the delay times*. This phase transforms a part of the chromosome supplied by the genetic algorithm into the vector of activity delays (Δ).
3. *Decoding of the release dates*. This phase transforms a part of the chromosome supplied by the genetic algorithm into the vector of project release dates (ES).
4. *Schedule generation*. This phase makes use of Π, Δ, and ES, generated in the previous phases, and constructs parameterized active schedules.
5. *Fitness evaluation*: This phase computes the fitness of the solution (or measure of quality of the schedule) according to Eq. (31.5).

Figure 31.1 illustrates the sequence of steps applied to each chromosome generated by the BRKGA.

The remainder of this section details the genetic algorithm, the decoding procedure, and the SGS

31.4.2 Biased Random-Key Genetic Algorithm

Genetic algorithms with random keys, or *random-key genetic algorithms* (RKGA), for solving problems like sequencing, whose solutions can be encoded as permutation vectors, were introduced in Bean (1994). In an RKGA, chromosomes are represented as vectors of randomly generated real numbers in the interval $[0, 1]$.

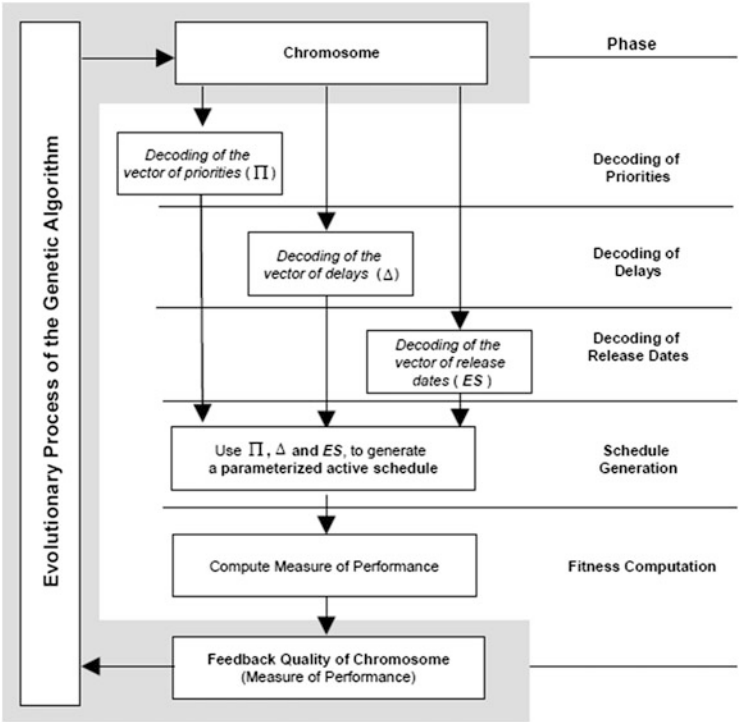


Fig. 31.1 Architecture of the algorithm

The *decoder*, a deterministic algorithm, takes as input a chromosome and associates with it a solution of the combinatorial optimization problem for which an objective value or fitness can be computed.

Random key GAs are particularly attractive for sequencing problems and/or when the chromosomes have several parts (see, for example, Gonçalves and Almeida 2002; Gonçalves and Resende 2004; Gonçalves and Sousa 2011). Unlike traditional GAs, which need to use special repair procedures to handle permutations or sequences, RKGAs move all the feasibility issues into the objective evaluation procedure and guarantee that all offspring formed by crossover correspond to feasible solutions. When the chromosomes have several parts, traditional GAs need to use different genetic operators for each part. However, since RKGAs use parametrized uniform crossovers (instead of the traditional one-point or two-point crossover), they do not need to have different genetic operators for each part.

A *RKGA* evolves a *population* of random-key vectors over a number of *generations* (iterations). The initial population is made up of σ_{pop}^{init} vectors of n_{key} random keys. Each component of the solution vector, or random key, is generated independently at random in the real interval $[0, 1]$. Next, the fitness of each individual is computed by the decoder in generation g, the population is partitioned

into two groups of individuals: a small group of $n_{elit} < \sigma_{pop}^{init}/2$ *elite* individuals, i.e., those with the best fitness values, and the remaining set of $\sigma_{pop}^{init} - n_{elit}$ *non-elite* individuals. To evolve the population of generation g, a new generation of individuals is produced. All elite individuals of the population of generation g are copied without modification to the population of generation $g + 1$. *RKGAs* implement mutation by introducing *mutants* into the population. A mutant is a vector of random keys generated in the same way in which an element of the initial population is generated. At each generation, a small number $n_{mut} < \sigma_{pop}^{init}/2$ of mutants is introduced into the population. With $n_{elit} + n_{mut}$ individuals accounted for in the population of generation $g + 1$, $\sigma_{pop}^{init} - n_{elit} - n_{mut}$ additional individuals need to be generated to complete the σ_{pop}^{init} individuals that make up population $g+1$. This is done by producing $\sigma_{pop}^{init} - n_{elit} - n_{mut}$ offspring solutions through the process of *mating* or *crossover*.

A *biased random-key genetic algorithm,* or *BRKGA* (Gonçalves and Resende 2011a), differs from a *RKGA* in the way parents are selected for mating. While in the *RKGA* of Bean (1994) both parents are selected at random from the entire current population, in a *BRKGA* each offspring is generated combining a parent selected at random from the elite partition in the current population and one selected at random from the rest of the population. Repetition in the selection of a mate is allowed and therefore an individual can produce more than one offspring in the same generation. As in *RKGAs*, *parameterized uniform crossover* (Spears and Dejong 1991) is used to implement mating in *BRKGAs*. Let π_{elit} be the probability that an offspring inherits the vector component of its elite parent. Recall that n_{key} denotes the number of components in the solution vector of an individual. For $l = 1, \ldots, n_{key}$, the l-th component $c(l)$ of the offspring vector c takes on the value of the l-th component $e(l)$ of the elite parent e with probability π_{elit} and the value of the l-th component $\bar{e}(l)$ of the non-elite parent $\bar{e}$ with probability $1 - \pi_{elit}$.

When the next population is complete, i.e., when it has σ_{pop}^{init} individuals, fitness values are computed for all of the newly created random-key vectors and the population is partitioned into elite and non-elite individuals to start a new generation.

A *BRKGA* searches the solution space of the combinatorial optimization problem indirectly by searching the continuous n_{key}-dimensional hypercube, using the decoder to map solutions in the hypercube to solutions in the solution space of the combinatorial optimization problem where the fitness is evaluated.

To specify a biased random-key genetic algorithm, we simply need to specify how solutions are encoded and decoded and how their corresponding fitness values are computed. We specify our algorithm next by first showing how the resource-constrained multi-project scheduling solutions are encoded and then decoded and how their fitness evaluation is performed.

31.4.3 Chromosome Representation

A chromosome represents a solution to the problem and is encoded as a vector of random keys. In a direct representation, a chromosome represents a solution of the original problem, and is usually called *genotype*, while in an indirect representation it does not and special procedures are needed to derive a solution from it usually called *phenotype*.

In the present context, the direct use of schedules as chromosomes is too complicated to represent and manipulate. In particular, it is difficult to develop corresponding crossover and mutation operations. Instead, solutions are represented indirectly by parameters that are later used by a schedule generator to obtain a solution. To obtain the solution (phenotype) we use the parameterized active schedule generator described in Sect. 31.5. Each solution chromosome is made of $2n + m$ genes, where n is the number of activities and m is the number of projects:

$$Chromosome = (\underbrace{gene_1, \ldots, gene_n}_{\text{Priorities}}, \underbrace{gene_{n+1}, \ldots, gene_{2n}}_{\text{Delay Times}}, \underbrace{gene_{2n+1}, \ldots, gene_{2n+m}}_{\text{Release Dates}})$$

The first n genes are used to determine the priorities of each activity. The genes $n + 1$ to $2n$ are used to determine the delay time used at each of the n iterations of the scheduling procedure, which schedules one activity per iteration. The last m genes are used to determine the release dates of each of the m projects.

31.4.4 Decoding of the Activity Priorities

As mentioned in Sect. 31.4.3, the first n genes are used to obtain activity priorities. Activity priorities are values between 0 and 1. The higher the value, the higher the priority will be. Below, we present the decoding procedure for the activity priorities.

Let $TF_j = d_{q(j)} - l_j$, represent the slack of activity j where $d_{q(j)}$ is the due date of the project q to which activity j belongs and l_j is the length of the longest-length path from the beginning of activity j to the end of the project $q(j)$ to which activity j belongs. Furthermore, let TF^{max} be the maximum slack for all activities amongst all projects.

The priority of each activity j is given by an expression which produces priority values between 70 and 100 % of the normalized slack. The priority of each activity j is given by the following expression

$$\Pi_j = \frac{TF_j}{TF^{max}} \times \left(0.7 + 0.3 \times gene_j\right)$$

31.4.5 Decoding of the Delays

Genes $n+1$ to $2n$ are used to determine the delay times Δ_i, used by each scheduling iteration i. The delay time for each activity i is calculated by

$$\Delta_i = gene_i \times 1.5 \times p^{max}$$

where p^{max} is the maximum duration amongst all activity durations. The factor 1.5 was obtained after experimenting with values between 1.0 and 2.0 in increments of 0.1.

31.4.6 Decoding of the Release Dates

The last m genes of each the chromosome (genes $2n + 1$ to $2n + m$) are used to determine the release dates of each project $q \in Q$. The following decoding expression is used to obtain the release date of each project $q \in Q$:

$$ES_q = \overline{ES}_q + gene_{2n+q} \times \left(d_q - \overline{ES}_q\right)$$

Consequently, the duration of the initial activity of each project q is equal to

$$p_{\alpha_q} = ES_q \qquad (q \in Q)$$

31.5 Schedule-Generation Procedure

The set of active schedules is usually very large and contains many schedules with relatively large delay times, having therefore poor quality in terms of the performance measure. To reduce the solution space, parameterized active schedules, introduced by Gonçalves and Beirão (1999) and Gonçalves et al. (2005) are used. The basic idea of parameterized active schedules consists in controlling the delay time allowed for each activity to encounter. By controlling the maximum delay time allowed, one can reduce or increase the solution space. A maximum delay time equal to zero is equivalent to restricting the solution space to non-delay schedules and a maximum delay time equal to infinity is equivalent to allowing general active schedules.

The procedure used to construct parameterized active schedules is based on a schedule-generation scheme that proceeds by time-increments. For each iteration μ, there is a scheduling time t_μ. All activities which are active at t_μ form the active set, i.e.,

$$\mathscr{A}_\mu = \left\{j \in V \mid C_j - d_j \leq t_\mu < C_j\right\}$$

```
procedure GENERATE-PARAMETRIZED-ACTIVE-SCHEDULES (Π, Δ)
1   Initialization: μ := 1; t_1 := 0; A_0 := {0}; Γ_0 := {0};
                C_0 := {0}; R'_k(0) := R_k, (k ∈ R);
2   while |C_μ| < n+2 repeat
3       Update: D_μ;
4       while D_μ ≠ {} repeat
5           Select activity with highest priority:
            j* := argmax_{j∈D_μ} { Π_j };
6           Calculate earliest finish time in terms of precedence only:
            EC_j* := max_{i∈Pred(j)} {C_i} + p_j*;
7           Calculate earliest finish time in terms of precedence and capacity:
            C_j* := p_j* + min{ t ∈ [EC_j* − p_j*, ∞] ∩ Γ_μ | r_j*k ≤ R'_k(τ),
                k ∈ R | r_j*k > 0, τ ∈ [t, t + p_j*] };
8           Update: C_μ := C_μ−1 ∪ { j* };   Γ_μ := Γ_μ−1 ∪ { C_j* };
9           Iteration increment: μ := μ + 1;
10          Update A_μ, D_μ, R'_k(t) | t ∈ [C_j* − p_j*, C_j*], k ∈ R | r_j*k > 0;
11      end while;
12      Determine the time associated with the activity selected at iteration μ;
        t_μ := min {t ∈ Γ_μ−1 | t > t_μ−1};
13  end while;
end GENERATE-PARAMETRIZED-ACTIVE-SCHEDULES;
```

Fig. 31.2 Pseudocode to generate parameterized active schedules

The remaining resource capacity of resource k at instant time t_μ is given by

$$R'_k(t_\mu) = R_k(t_\mu) - \sum_{j \in \mathscr{A}_\mu} r_{jk}$$

All activities that have been scheduled up to iteration μ are contained in the set $\mathscr{C}_\mu$ and Γ_μ denotes the set of finish times of the activities in $\mathscr{C}_\mu$. Let Δ_μ be the delay time associated with the activity being scheduled at iteration μ, and let the set $\mathscr{D}_\mu$ comprise all activities which are precedence-feasible in the interval $[t_\mu, t_\mu + \Delta_\mu]$, i.e.,

$$\mathscr{D}_\mu = \left\{ j \in V \setminus \mathscr{C}_{\mu-1} \mid C_i \le t_\mu + \Delta_\mu \quad \forall i \in Pred(j) \right\}$$

The algorithmic description of the schedule-generation scheme used to generate parameterized active schedules is given by the pseudocode shown in Fig. 31.2.

The basic idea of parameterized active schedules is incorporated in the selection step of the procedure, i.e., in the step

$$j^* := \underset{j \in \mathscr{D}_\mu}{\operatorname{argmax}} \left\{ \Pi_j \right\}$$

The set $\mathscr{D}_\mu$ forces the selection to be made only amongst activities which will have a delay smaller or equal to the maximum allowed delay.

Parameters Π_j (priority of activity j) and $\mathscr{D}_\mu$ (delay for the activity being scheduled at iteration μ) are supplied by the genetic algorithm.

31.6 Computational Results

In the next subsections we present the details of the computational experiments used to illustrate the effectiveness of the algorithm described in this chapter.

31.6.1 Test Problems

The test problems used in the computational experiments are the ones proposed by Gonçalves et al. (2008). These test problems have known optimal solutions equal to zero for the measure of performance described in Sect. 31.2 (i.e., *tardiness* $= 0$, *earliness* $= 0$, and *flow time deviation*$= 0$).

Five types of multi-project instances were used, with 10, 20, 30, 40, and 50 single-project instances, each with 120 activities. For each problem type, 20 instances were used. Since each single-project instance has 120 activities, we have that each multi-project instance has 1,200, 2,400, 3,600, 4,800, and 6,000 activities, respectively. Each activity can use up to four resources. The average number of overlapping projects in execution can be 3, 6, 9, 12, and 15. Table 31.3 shows the combinations of the number of overlapping projects used for the problems with 10, 20, 30, 40, and 50 single-projects.

31.6.2 BRKGA Configuration

Although there is no straightforward way to configure the parameters of a genetic algorithm, our past experience with genetic algorithms based on the same evolutionary strategy (see Gonçalves and Almeida 2002; Gonçalves and Resende 2004, 2011b, 2012, 2013, 2014; Gonçalves et al. 2005, 2008) has shown that good results can be obtained with the values of n_{elit}, n_{mut}, and Crossover Probability (π_{elit}) shown in Table 31.1.

Table 31.1 Range of parameters for the evolutionary strategy

Parameter	Interval
n_{elit}	$(0.10–0.25) \times \sigma_{pop}^{init}$
n_{mut}	$(0.15–0.30) \times \sigma_{pop}^{init}$
Crossover probability (π_{elit})	(0.70–0.85)

Table 31.2 Configuration of the BRKGA for the computational experiments

Population size:	min{0.2 × Number of activities in the multi-project, 250}
Crossover probability:	0.7
Selection:	The top 10 % from the previous population chromosomes are copied to the next generation
Mutation:	20 % of the population chromosomes are replaced with new randomly generated chromosomes
Fitness:	See Eq. (31.5)
Stopping criterion:	50 generations

For the population size we obtained good results by indexing it to the size of the problem, i.e., use small size populations for small problems and larger populations for larger problems. Having in mind this past experience and in order to obtain a reasonable configuration, we conducted a factorial analysis on a small pilot set of problem instances not included in the experimental tests. The configuration shown in Table 31.2 was the best in terms of the sum of fitness values and the number of best results and was held constant for all problem instances in the experiments. The experimental results demonstrate that this configuration not only provides high-quality solutions but it is very robust.

31.6.3 Results

Table 31.3 summarizes the experimental results. It lists the fitness, earliness, tardiness, and flow time deviation for each problem type. Let m be the number of projects in each problem instance. Averages and standard deviations were computed for the 20 problem instances included in each problem type. Columns Avg^1 and SD^1 list averages and standard deviations for the expression

$$\frac{1}{m}\left(w^T \sum_{i=1}^{m} T_i^3 \;+\; w^E \sum_{i=1}^{m} E_i^2 \;+\; w^{FD} \sum_{i=1}^{m} FD_i^2\right)$$

Columns Avg^2 and SD^2 list, respectively, averages and standard deviations for the expression

$$\frac{1}{m}\sum_{i=1}^{m} E_i$$

Columns Avg^3 and SD^3 list, respectively, averages and standard deviations for the expression

$$\frac{1}{m}\sum_{i=1}^{m} T_i$$

and columns Avg[4] and SD[4] list, respectively, averages and standard deviations for the expression

$$\frac{1}{m}\sum_{i=1}^{m} FD_i$$

The last column with heading *% Improv* represents the percentage improvement of the average last generation fitness values on those of the first generation, i.e.,

$$100\,\% \times \frac{(\text{Fitness at first generation} - \text{Fitness at last generation})}{\text{Fitness at first generation}}$$

Table 31.3 shows that all averages of the tardiness are close to zero and that the averages values of the earliness are also close to zero for all instances with more than three overlapping projects. As expected, the fitness obtained gets smaller (i.e., improves) as the number of overlappings of projects increases. This is due to the fact that as the number of overlapping projects increases, so does the flexibility in terms of capacity, therefore allowing for more possibilities of finding good schedules. Finally, the *% Improv* values show that the BRKGA achieves a substantial

Table 31.3 Experimental results

No Proj's	No Overl.	Fitness		Tardiness		Earliness		Flow dev.		No Best	% Improv
		Avg[1]	SD[1]	Avg[2]	SD[2]	Avg[3]	SD[3]	Avg[4]	SD[4]		
10	3	10.35	18.56	0.00	0.00	1.20	1.41	0.38	0.54	17	99.99
20	3	73.14	117.52	0.00	0.00	2.57	2.91	1.07	1.97	17	100.00
	6	0.95	2.10	0.00	0.00	0.42	0.27	0.03	0.07	20	100.00
30	3	210.13	202.81	0.01	0.02	3.92	2.88	1.74	1.45	18	100.00
	6	3.89	7.11	0.00	0.00	0.60	0.40	0.09	0.20	20	100.00
	9	0.48	0.37	0.00	0.00	0.38	0.12	0.02	0.05	19	100.00
40	3	1,324.14	1,282.69	0.06	0.06	9.45	7.29	6.15	4.77	15	100.00
	6	6.18	15.00	0.00	0.00	0.59	0.35	0.11	0.22	18	100.00
	9	4.48	16.52	0.00	0.00	0.50	0.25	0.06	0.21	18	100.00
	12	2.00	4.28	0.00	0.00	0.52	0.26	0.04	0.08	17	100.00
50	3	2,584.49	2,887.14	0.07	0.04	14.68	5.68	7.40	6.42	11	99.91
	6	25.87	57.23	0.00	0.00	0.87	0.60	0.23	0.39	17	100.00
	9	0.73	0.79	0.00	0.00	0.43	0.11	0.02	0.05	20	100.00
	12	1.35	2.16	0.00	0.00	0.50	0.17	0.02	0.05	18	100.00
	15	1.07	1.98	0.00	0.00	0.50	0.15	0.01	0.04	13	100.00

Table 31.4 Average elapsed time for 50 generations

Problem instance type (number of projects):	10	20	30	50
Average elapsed time (in seconds) for 50 generations:	178	449	840	1,860

improvement in the quality of the solutions. Sometimes the average percentage improvement is as large as 100 %.

The computational experiments were run on a PC with a 1.33 GHz AMD Thunderbird CPU on the MS Windows Me operating system and the algorithm was implemented in Visual Basic 6.0. Table 31.4 presents the average computational times, in seconds, for each problem instance and for 50 generations.

31.7 Conclusions

This chapter presents the Basic Multi-Project Scheduling Problem and a solution approach using a biased random-key genetic algorithm. The chromosome representation of the problem is based on random keys. The schedules are constructed using a schedule-generation scheme that generates parameterized active schedules based on priorities, delay times, and release dates generated by the biased random-key genetic algorithm.

The approach is tested on a set of test problems with 10, 20, 30, 40, and 50 projects (having 1,200, 2,400, 3,600, 4,800, and 6,000 activities, respectively). In the computational experiments, the algorithm obtained values near the optimum (zero), therefore validating the effectiveness of the proposed approach.

Acknowledgements This work has been partially supported by funds granted by the ERDF through the Programme COMPETE and by the Portuguese Government through FCT, the Foundation for Science and Technology, project PTDC/EGE-GES/117692/2010.

References

Ash R (1999) Activity scheduling in the dynamic, multi-project setting: choosing heuristics through deterministic simulation. In: Proceedings of the 1999 winter simulation conference, Phoenix, pp 937–941

Baker KR (1974) Introduction to sequencing and scheduling. Wiley, New York

Bean JC (1994) Genetics and random keys for sequencing and optimization. ORSA J Comput 6:154–160

Błażewicz J, Lenstra JK, Rinnooy Kan AHG (1983) Scheduling subject to resource constraints: classification and complexity. Discrete Appl Math 5:11–24

Bock D, Patterson J (1990) A comparison of due date setting, resource assignment, and job preemption heuristics for the multi-project scheduling problem. Decis Sci 21:387–402

Browning TR, Yassine AA (2010) Resource-constrained multi-project scheduling: priority rule performance revisited. Int J Prod Econ 126(2):212–228

Cai Z, Li X (2012) A hybrid genetic algorithm for resource-constrained multi-project scheduling problem with resource transfer time. In: Proceedings of the 2012 IEEE international conference on automation science and engineering (CASE 2012), Seoul, pp 569–574

Deckro R, Winkofsky E, Hebert J, Gagnon R (1991) A decomposition approach to multi-project scheduling. Eur J Oper Res 51:110–118

Drexl A (1991) Scheduling of project networks by job assignment. Manag Sci 37(12):1590–1602

Dumond J, Mabert V (1988) Evaluating project scheduling and due date assignment procedures: an experimental analysis. Manag Sci 34(1):101–118

Fendley L (1968) Towards the development of a complete multiproject scheduling system. J Ind Eng 19(10):505–515

Gonçalves JF, Almeida JR (2002) A hybrid genetic algorithm for assembly line balancing. J Heuristics 8:629–642

Gonçalves JF, Beirão NC (1999) Um algoritmo genético baseado em chaves aleatórias para sequenciamento de operações. Revista da Associação Portuguesa de Investigação Operacional 19:123–137 (in Portuguese)

Gonçalves JF, Resende MGC (2004) An evolutionary algorithm for manufacturing cell formation. Comput Ind Eng 47:247–273

Gonçalves JF, Resende MGC (2011a) Biased random-key genetic algorithms for combinatorial optimization. J Heuristics 17:487–525

Gonçalves JF, Resende MGC (2011b) A parallel multi-population genetic algorithm for a constrained two-dimensional orthogonal packing problem. J Comb Optim 22:180–201

Gonçalves JF, Resende MGC (2012) A parallel multi-population biased random-key genetic algorithm for a container loading problem. Comput Oper Res 39:179–190

Gonçalves JF, Resende MG (2013) A biased random key genetic algorithm for 2D and 3D bin packing problems. Int J Prod Econ 145(2):500–510

Gonçalves JF, Resende MG (2014) An extended akers graphical method with a biased random-key genetic algorithm for job-shop scheduling. Int Trans Oper Res 21(2):215–246

Gonçalves JF, Sousa PSA (2011) A genetic algorithm for lot sizing and scheduling under capacity constraints and allowing backorders. Int J Prod Res 49:2683–2703

Gonçalves JF, Mendes JJM, Resende MGC (2005) A hybrid genetic algorithm for the job shop scheduling problem. Eur J Oper Res 167:77–95

Gonçalves JF, Mendes JJM, Resende MGC (2008) A genetic algorithm for the resource constrained multi-project scheduling problem. Eur J Oper Res 189:1171–1190

Krüger D, Scholl A (2010) Managing and modelling general resource transfers in (multi-) project scheduling. OR Spectr 32(2):369–394

Kumanam S, Raja K (2011) Multi-project scheduling using a heuristic and memetic alogrithm. J Manuf Sci Prod 10(3–4):249–256

Kurtulus I (1985) Multiproject scheduling: analysis of scheduling strategies under unequal delay penalties. J Oper Manag 5(3):291–307

Kurtulus I, Davis E (1982) Multi-project scheduling: categorization of heuristic rules performance. Manag Sci 28:161–172

Kurtulus I, Narula S (1985) Multi-project scheduling: analysis of project performance. IIE Trans 17:58–66

Lawrence S, Morton T (1993) Resource-constrained multi-project scheduling with tardy costs: comparing myopic bottleneck and resource pricing heuristics. Eur J Oper Res 64:168–187

Lova A, Tormos P (2001) Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling. Ann Oper Res 102(1):263–286

Lova A, Tormos P (2002) Combining random sampling and backward-forward heuristics for resource-constrained multi-project scheduling. In: Proceedings of the eight international workshop on project management and scheduling, Valencia, pp 244–248

Lova A, Maroto C, Tormos P (2000) A multicriteria heuristic method to improve resource allocation in multiproject scheduling. Eur J Oper Res 127:408–424

Mendes J (2003) Sistema de apoio à decisão para planeamento de sistemas de produção do tipo projecto. Ph.D. dissertation, Universidade do Porto, Porto (in Portuguese)

Mohanthy R, Siddiq M (1989) Multiple projects multiple resources-constrained scheduling: some studies. Int J Prod Res 27(2):261–280

Özdamar L, Ulusoy G, Bayyigit M (1998) A heuristic treatment of tardiness and net present value criteria in resource constrained project scheduling. Int J Phys Distrib Logist Manag 28:805–824

Pritsker A, Allan B, Watters L, Wolfe P (1969) Multiproject scheduling with limited resources: a zero-one programming approach. Manag Sci 16:93–108

Shankar V, Nagi R (1996) A flexible optimization approach to multi-resource, multi-project planning and scheduling. In: Proceedings of 5th industrial engineering research conference, Minneapolis, pp 263–267

Spears WM, Dejong KA (1991) On the virtues of parameterized uniform crossover. In: Proceedings of the fourth international conference on genetic algorithms, San Diego, pp 230–236

Talbot FB (1982) Resource-constrained project scheduling with time-resource tradeoffs: the nonpreemptive case. Manag Sci 28(10):1197–1210

Tsubakitani S, Deckro R (1990) A heuristic for multi-project scheduling with limited resources in the housing industry. Eur J Oper Res 49:80–91

Vercellis C (1994) Constrained multi-project planning problems: a Lagrangean decomposition approach. Eur J Oper Res 78:267–275

Woodworth BM, Willie CJ (1975) A heuristic algorithm for resource leveling in multi-project, multi-resource scheduling. Decis Sci 6(3):525–540

Chapter 32
Decentralized Multi-Project Scheduling

Andreas Fink and Jörg Homberger

Abstract This chapter is concerned with the decentralized resource-constrained multi-project scheduling problem (DRCMPSP), which is characterized in that individual involved decision makers pursue individual goals, whereas some overall coordination mechanism is needed to resolve conflicts due to the interdependencies between multiple projects. The connection between activities from these projects may result from temporal and resource-orientated constraints. In general, there may be two kinds of autonomous decision makers, on the one hand those that control individual projects, and on the other hand those that control globally available resources. After providing a more detailed description of such kinds of problems and the resulting peculiarities of decentralized decision making, a classification of respective problem types is provided, which leads to related requirements for solution procedures. Overall, there are two basic solution approaches, namely auctions and negotiations. These methods are described in connection with a review of the related literature.

Keywords Auctions • Decentralized decision making • Multi-project scheduling • Negotiations

32.1 Introduction

The kind of management of multiple interconnected projects depends on whether a single decision making entity is basically authorized to centrally schedule all activities of all projects, or not. While the preceding Chap. 31 of this handbook mainly assumes the former situation, this chapter is concerned with the latter case

A. Fink (✉)
Faculty of Economics and Social Sciences, Institute of Computer Science,
Helmut-Schmidt-Universität, Hamburg, Germany
e-mail: andreas.fink@hsu-hamburg.de

J. Homberger
Department of Computer Science, Hochschule für Technik Stuttgart, Stuttgart, Germany
e-mail: joerg.homberger@hft-stuttgart.de

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_2

(decentralized multi-project scheduling). Each individual project is represented by an autonomous self-interested decision maker, which generally aims at an effective scheduling for one's own project. This applies in particular if individual projects stem from different profit-maximizing firms or autonomous business units with conflicting goals. Then it is not appropriate to presuppose full central availability of reliable information about the individual projects and a governance structure where a single project manager is authorized and qualified to define and pursue an overall goal on behalf of all the parties.

The connection between individual projects and respective activities may result from temporal (precedence) and resource-orientated constraints. The involved resources may be distinguished in, firstly, local resources, which are owned by a project decision maker and which are usually used exclusively by activities from the respective project, and, secondly, global (shared) resources, which could be used by any project (i.e., projects generally compete for these resources). As each global resource is owned by someone, only this owner is generally authorized to control the use of the related resource.

Consequently, decentralized multi-project management means that there is no single decision maker that is entitled to hierarchically allocate all resources and instruct which activities from multiple projects have to be executed when. On the contrary, one may have to take into account two kinds of decision makers, on the one hand those that control individual projects, and on the other hand those that control globally available resources. Accordingly, the decentralized resource-constrained multi-project scheduling problem (DRCMPSP) is characterized in that individual decision makers pursue individual, generally conflicting goals, whereas some overall coordination mechanism is needed to resolve conflicts due to the interdependencies between the projects.

Decentralized project management is thus principally a matter of governance, where the organizational structure which constitutes the framework for coordinating multiple projects has to be established at first. For example, the involved decision makers that control the projects and resources may agree on a protocol which formally defines rules on who is entitled and/or obliged to contribute to the overall scheduling procedure in a specific manner. The design of such a protocol is generally aimed at the quality of resulting solutions. This involves some measurement of individual utility for the respective decision makers, yet fairness criteria may also be relevant (e.g., because decision makers might refuse to collaborate at all if the used coordination procedure is deemed as unfair in some way). However, there is no single, obvious concept for measuring quality in practice under consideration of information asymmetry and conflicting preferences of multiple autonomous decision makers.

This intrinsic decentralized organizational character of the DRCMPSP, with multiple autonomous decision makers, must be distinguished from technical distribution, i.e., some distributed implementation of a solution procedure that nonetheless is centrally devised. The latter case would allow to design a distributed solution procedure that supposes that all information about multiple projects is honestly disclosed and thus in principal globally available (see also Chap. 31), while in a

genuine decentralized situation rational decision makers may communicate biased information to influence the project scheduling in their interest (for example, proclaiming overstated lateness penalties to work towards an earlier scheduling of particular project activities). In this chapter we generally consider the genuine decentralized resource-constrained multi-project scheduling problem.

In Sect. 32.2 we formally define the problem, discuss special characteristics and requirements according to the decentralized character of the problem, and mention several application examples. In Sect. 32.3 we describe a general classification of the DRCMPSP with related requirements on solution approaches. In Sect. 32.4 we describe main ideas of decentralized coordination approaches for the DRCMPSP and review the related literature. In Sect. 32.5 we give some conclusion and discuss avenues for future research on the DRCMPSP.

32.2 Problem Description

We first provide a formal definition of the DRCMPSP. Then we discuss the peculiarities that result from the decentralized character of the problem. Eventually we mention applications, which further illustrate the problem.

32.2.1 Formal Problem Definition

Given is a set Q of multiple projects. Each project $q \in Q$ is controlled by a corresponding autonomous project decision maker (agent) PA_q. As usual we assume that the basic project configuration has been established beforehand, which results in the following characteristics. Each project $q \in Q$ consists of a set of activities V_q. The overall set of activities V results from the union $\cup_{q \in Q} V_q$, where $V_q \cap V_p = \emptyset$ for $q \in Q, p \in Q, q \neq p$. Each activity $i \in V$ has a duration (processing time) $p_i \in \mathbb{R}_{\geq 0}$. Once started an activity must not be interrupted.

The processing of activities must observe ordinary finish-start precedence constraints. For each activity $i \in V$ the sets $Pred(i)$ and $Succ(i)$ consist of the (immediate) predecessors and successors, respectively, of activity i. An activity i must not be started before all its predecessors from $Pred(i)$ have been finished. Each project $q \in Q$ includes in V_q dummy start and end activities α_q and ω_q, respectively. For these activities it holds that $p_{\alpha_q} = 0$, $p_{\omega_q} = 0$, $Pred(\alpha_q) = \emptyset$, $Succ(\omega_q) = \emptyset$, $Succ(\alpha_q) = V_q \backslash \{\alpha_q\}$, $Pred(\omega_q) = V_q \backslash \{\omega_q\}$. If there are restrictions on the earliest starting time of a project q this may be modeled by an earliest starting time parameter for the dummy start activity α_q. Furthermore for an activity $i \in V$ there may be a due date d_i for the completion of this activity.

The structure of the overall multi-project can be represented by an activity-on-node project network $G = (V, E)$, where E is the set of precedence relationships which results from all given predecessor and successor relationships. In the usual

static version of the DRCMPSP all information on all projects is generally known from the beginning. In dynamic versions of the problem projects or activities may become known (released) later within the planning horizon.

There is a set of renewable resources $\mathscr{R}$. Each resource $k \in \mathscr{R}$ is available with capacity $R_k \in \mathbb{R}_{>0}$ in each period (time unit). An activity i requires an amount of $r_{ik} \in \mathbb{R}_{\geq 0}$ of resource k in each processing period. The resources are partitioned in a set of global (shared) resources $\mathscr{R}^{global}$ and local resource sets $\mathscr{R}_q^{local}, q \in Q$. An activity $i \in V_q$ can only use resources from $\mathscr{R}^{global}$ and $\mathscr{R}_q^{local}$. Each local resource $k \in \mathscr{R}_q^{local}$ is owned and controlled by the corresponding project decision maker (agent) PA_q. Each global resource $k \in \mathscr{R}^{global}$ is owned and controlled by a corresponding autonomous resource decision maker (agent) RA_k. It may be the case that a resource decision maker owns and controls a group of global resources (i.e., there may be different global resources k_1 and k_2 with $RA_{k_1} = RA_{k_2}$).

Multi-project scheduling means that there are interdependencies between the projects in the set Q. The usual assumption is that the dependency results from the fact that activities from different projects compete for global resources, i.e., $\mathscr{R}^{global} \neq \emptyset$, while precedence constraints only apply within a project q, i.e., for all activities $i \in V_q$ $Pred(i) \subset V_q$ and $Succ(i) \subset V_q$. On the other hand, one may also take into account precedence relationships between activities from different projects, i.e., for some activity $i \in V_q$ it may hold that $Pred(i) \setminus V_q \neq \emptyset$ or $Succ(i) \setminus V_q \neq \emptyset$.

A feasible solution (schedule) S for some scheduling problem is given by starting times S_i and corresponding completion times $C_i = S_i + p_i$ for each activity i such that precedence (temporal) constraints as well as resource constraints are observed. That is, $S_i \geq \max_{j \in Pred(i)} C_j$ for all activities i and $r_k(S,t) \leq R_k$ at time t where $r_k(S,t) = \sum_{i \in V: S_i \leq t < C_i} r_{ik}$ represents the amount of resource k used at time t given schedule S. The completion time of a project $q \in Q$ is given by C_{ω_q}. The overall multi-project schedule S may be decomposed in accordance with the set of partial schedules of the different projects $\{S^q | q \in Q\}$.

The quality of a schedule may be assessed by taking into account different local and global criteria (objectives). First of all, the quality of each project $q \in Q$ may be evaluated individually (locally) by the project decision maker (agent) PA_q, e.g., by either a time-oriented objective function or by a cost-oriented (or profit-oriented) objective function. Typical time-oriented local objective functions are minimization of completion time C_{ω_q} (makespan) or minimization of project delay (tardiness) T_q, i.e., $\max\{0, C_{\omega_q} - d_{\omega_q}\}$ for a given due date d_{ω_q} for this project. With regard to the latter criterion a project's due date is usually considered as a soft constraint with penalization in case of violations. In the literature on the DRCMPSP the due date d_{ω_q} of a project q is often artificially calculated as, firstly, the makespan of project q when neglecting the global resources (i.e., for this criterion, one may need to locally solve a traditional resource-constrained project scheduling problem RCPSP) or, secondly, the critical path length $d_{\alpha_q \omega_q}$ when neglecting all resource constraints (i.e., the longest path from α_q to ω_q when only observing the precedence constraints of the project network).

It may be possible to convert a delay into a cost value by some transformation function which incorporates penalties that arise when the respective project is finished late. Furthermore, special versions of the problem may involve time-dependent costs for resource utilization or time-dependent revenues (cash flows) for completing certain activities. Only if a cost-oriented (or profit-oriented) objective function is available it is possible to directly compare the effect of changes to the project schedule between different projects (assumption of transferable utility), which may be relevant for the design of a coordination procedure.

In addition to project-specific (local) objectives global criteria (coordination goals) evaluate schedules for the multi-project problem as a whole. Because of the decentralized character of the decision problem and related information asymmetries the latter kind of assessment may only be possible in simulation experiments for artificial problem instances but not for a real application. If not implicitly guaranteed by the solution procedure, achieving a feasible solution (in particular a feasible allocation of global resources) may be regarded as the basic goal of the coordination of multiple projects. Assuming feasibility, Pareto-efficiency is a common criterion; that is, it should not be possible to modify the overall schedule in a way such that some local objective function value is improved without worsening any other project. Further global objective functions may involve (1) a weighted sum of the local objective functions (e.g., average project delay), which may be regarded as a proxy for social welfare, (2) something such as the minimum completion time or the minimum delay over all projects, or (3) some fairness measure when dealing with the different projects in comparison.

Based on the three field classification for project scheduling problems (Brucker et al. 1999) we basically consider problems of the kind *PS | mult, prec, inter | private-mult* with different kinds of objective functions taking into account conflicting goals of autonomous self-interested decision makers with interfering job sets in a decentralized decision situation In Sect. 32.3 we classify different versions of the DRCMPSP under consideration of the basic problem structure and the specific kind of decentralization, which leads to respective requirements on solution approaches.

32.2.2 Decentralized Character of the Problem

In a genuine decentralized decision situation there are multiple autonomous decision makers (for the DRCMPSP regarding local project management and maybe also resource management). These decision makers selfishly pursue individual goals, which are generally conflicting. For example, there are tradeoffs between favorably scheduling different projects or utilizing different resources and there is no superordinate decision maker which is entitled to centrally balance individual preferences.

Only if the decentralized character of the multi-project scheduling problem is disregarded, the objective functions of multiple projects might be aggregated in some straightforward way (for example by a weighted sum or some other

transformation of the multiple criteria). However, for a truly decentralized problem it is generally not reasonable to simply aggregate different goals into a single scalar objective function. In an experimental setting it may nevertheless be useful to evaluate overall schedules by an aggregate objective function (for example social welfare under the assumption of transferable utility).

Considering multiple self-interested decision makers, achieving Pareto-efficiency is a basic goal. Pareto-efficient (non-dominated) solutions are characterized by the fact that it is not possible to improve some decision maker's situation without worsening at least one other decision maker's goal attainment. Since for a DRCMPSP instance in general many different Pareto-efficient solutions exist, additional criteria must be devised to possibly select among these. In particular, one may aim at a fair allocation which treats different projects equally in some way (yet this is easier said than done regarding a unique and applicable measure of fairness).

As already mentioned above, one may distinguish the DRCMPSP from distributed scheduling according to an implementation point-of-view. In the latter case multiple projects may take place at different locations and distributed local project manager entities may exist. For certain such application scenarios it might be conceivable to centrally manage all projects by taking care of multiple objectives via some procedure from multi-criteria optimization (by, e.g., aggregating multiple objectives into a single scalar objective function or computing/approximating the set of Pareto-efficient solutions). Only in such a situation it is appropriate to design an agent-based solution approach where each agent is kind of a white box software component which is purposefully designed by a central authority. This general concept of hierarchically designing cooperating agents which eventually pursue a collaborative goal is related to the distributed problem solving paradigm in computer science. According to this paradigm one designs a "system so that the agents solve the problem in a good way, in a distributed way, in an efficient way. [...] there is assumed to be a single body that is able, at design time, to directly influence the preferences of all agents in the system" (Rosenschein and Zlotkin 1994, p. 32). Such an approach in particular serves to cope with complexity (e.g., related to a dynamic character of the problem) but it must be distinguished from a multi-agent coordination mechanism that addresses genuine decentralization with self-interested agents that represent decision makers which are intrinsic to the problem. In the latter case agents have to be regarded as autonomous black box components with no option to centrally control or observe the internal reasoning of the agents.

In order to devise a coordination mechanism for the genuine DRCMPSP one may draw on project manager agents (one agent for each project) and possibly also resource manager agents (which each own one or more global resources). These agents represent corresponding autonomous decision makers with individual goals. Within an overall interaction protocol (which must be agreed on beforehand), the project manager agents aim at acquiring global resources and scheduling activities from their project as well as possible; resource manager agents may aim to sell or utilize their resources in a best possible way. From a global point of view each of these agents is a black box; i.e., there is no centralized control and no complete

global (symmetric) information. Verification of agent behavior is only possible according to checkable rules, which are part of an agreed upon interaction protocol. To clarify this point, we have to distinguish between the possible ways to use symmetric information, which is known to all agents, vs. asymmetric information, which is only known to corresponding project or resource manager agents. It is reasonable to consider all aspects that are observable beforehand or at the latest while the project is running as symmetric information (e.g., activity durations), while other information is not observable (e.g., cost functions, deadlines). Verifying whether an agent completely behaves truthfully is generally impossible; it is only possible to ensure compliance with rules that do not depend on asymmetric (private) information. In consideration of the impossibility of a general verification that stated information conforms to the true preferences of an agent, it is not appropriate to simply presuppose that rational decision makers honestly disclose private information (instead of, e.g., announcing overstated lateness penalties to obtain an earlier scheduling of the affected project activities). Instead it is reasonable to aim at incentive-compatible interaction protocols where lying does not pay-off, i.e., an agent's best strategy would be to behave truthfully.

32.2.3 Application Examples

To illustrate the DRCMPSP we mention some applications which differ with respect to the number of global and local resources, the existence of an actual owner and thus autonomous manager of global resources, the existence of temporal interdependencies between projects, and various kinds of objectives.

- In a manufacturing firm different research projects, product development and engineering projects, as well as regular production projects may compete for shared (global) machine resources, while each project may also have exclusive local resources (e.g., engineers). Usually there are no precedence constraints between activities from different projects. The global resources (machines) are managed by a business unit which aims at maximizing machine utilization. The different kinds of projects are related to different business units, which pursue individual goals such as maximizing throughput or observing deadlines.
- Different contractor firms may collaborate in a make-to-order project (e.g., in a large construction project for a building complex there may be concrete construction, staging, roof work, different kinds of interior work, electric installations, finishing of the house fronts, etc.). Precedence constraints between activities from different projects may generally prevail, but there is no or only one global resource (overall construction site management), since the firms mainly use their own, task-specific resources. The contractor firms are partly paid depending on the completion times of individual activities (milestones) but these payments, the budget restrictions as well as the resource availabilities of the firms may be regarded as private information.

- Some contractor firm may be involved within different customer projects (for example regarding engineering or auditing kind of work). Considering that this firm's resources are limited it needs some resource balancing procedure which is connected to the management of the multiple projects (considering, e.g., respective precedence constraints and activity due dates).

32.3 Classification of Problem Types

We describe a classification of different versions of the DRCMPSP, which also leads to related requirements for applicable solution approaches. In particular, we consider the basic problem structure, the actual character of decentralization and related information-oriented restrictions, and local and global objectives. The resulting classification is summarized in Table 32.1. Concrete problem versions impose respective requirements on solutions methods (the latter will be further discussed and classified in Sect. 32.4).

32.3.1 Basic Problem Structure

The basic problem structure mainly conforms to elements of the centralized resource-constrained multi-project scheduling problem (see Chap. 31). The following description builds on the notation introduced in Sect. 32.2.1.

- Set Q of individual projects
  - Basic type(s) of the individual projects (default: assumptions analogous to RCPSP; otherwise special characteristics such as, e.g., multiple modes or time-dependent resource costs)
  - Number ($\left|\mathscr{R}_q^{local}\right|$) and type (default: renewable) of local resources of project $q \in Q$

Table 32.1 Overview on classification for different versions of the DRCMPSP

Basic problem structure	Number and type of individual projects
	Kind of project interdependencies
	Consideration of randomness
Decision makers and distribution of information	Project decision maker agents (PAs)
	Resource decision maker agents (RAs)
	Global (symmetric) information
	Private (asymmetric) information
Objectives	Individual (local) objectives
	Coordination (global) objectives
	Transferability of utility

- Kind of project interdependencies
  - Default: Activities from different projects compete for global (shared) resources ($\mathscr{R}^{global} \neq \emptyset$)
  - Possible precedence constraints between activities from different projects ($\exists q \in Q \; \exists i \in V_q : Pred(i) \setminus V_q \neq \emptyset \vee Succ(i) \setminus V_q \neq \emptyset$)
- Randomness in the development of events and the future state of the system
  - Deterministic (default) vs. nondeterministic information
  - Static (default) vs. dynamic character of the project portfolio (e.g., consideration of project release times and rolling planning horizon)

32.3.2 Decision Makers and Distribution of Information

The DRCMPSP is characterized by the fundamental assumption that there are autonomous decision maker agents that constitute intrinsic features of the underlying problem. Firstly, individual projects are represented by project manager agents that autonomously pursue individual preferences (local objective function) subject to the coordination needs due to the interdependencies between multiple projects. Secondly, global resources may be represented by resource manager agents. For both kinds of agents, individual preferences (and possibly further aspects of the individual projects and global resources) have to be regarded as private information. Accordingly, while these agents' external behavior is observable, the respective internal reasoning is hidden in a black box.

- Default: Each project $q \in Q$ is controlled by a corresponding autonomous project decision maker (agent) PA_q.
- Each global resource $k \in \mathscr{R}^{global}$ may be owned and controlled by a corresponding autonomous resource decision maker (agent) RA_k
- Global (symmetric) information
  - Example: information related to the activities (such as durations)
- Private (asymmetric) information
  - In particular: information related to individual (local) preferences (e.g., due dates or delay costs)
  - Interest in secrecy: Private information may be distinguished with respect to the agents' willingness to disclose secret information, which restricts information exchange in solution mechanisms.

32.3.3 Objectives

The objectives are closely connected to the particular kind of the decentralized decision situation with the involved decision makers. We mainly distinguish

between individual objectives of the different decision makers and coordination (global) objectives that are related to the assessment of the overall multi-project system.

- Individual (local) objectives
  - Related to projects (e.g., minimize project delay, minimize project completion time, minimize costs due to the resource utilization, maximize discounted cash flow depending on the starting times of particular activities)
  - Related to global resources (e.g., minimize incurred costs, maximize obtained revenues, maximize utilization)
  - Homogeneous or heterogeneous agent types/objectives
- Coordination (global) objectives
  - Always: Obtain an overall feasible schedule
  - Achieve Pareto-efficiency (or minimize distance to nearest Pareto-optimal solution)
  - Maximize some measure of social welfare (e.g., minimize average project delay) or minimize the deviation from the result of a hypothetical centralized solution procedure
  - Consider some measure of fairness
- Is utility transferable between individual projects/project manager agents and/or (global) resources/resource manager agents?
  - If this applies, agents can evaluate changes to a project schedule in monetary terms (as the usual common denominator), which allows for solution approaches that utilize the exchange of money (e.g., most auctions types and certain kinds of negotiations with side payments).

In case of nondeterministic information, robustness may constitute an additional criterion for evaluating schedules (both locally and globally).

32.4 Solution Approaches and Literature Review

Centralized resource-constrained multi-project scheduling has been introduced by Pritsker et al. (1969) and Kurtulus and Davis (1982); for further literature references on this type of problem and related solution approaches see Chap. 31.

Agnetis et al. (2007) provide a survey on decentralized/multi-agent machine scheduling problems and different solution approaches. In the following we focus on the literature on decentralized resource-constrained multi-project scheduling problems (DRCMPSP). We note that the title of many manuscripts (like the title of this chapter) does not include "resource-constrained" yet resource restrictions are nonetheless considered. Furthermore, in the literature on the DRCMPSP there are some papers where the "D" is an abbreviation for "distributed", while otherwise

"D" usually represents "decentralized". Mostly, but not in all cases, this conforms to the foregoing discussion regarding genuine decentralization vs. distributed problem solving. Moreover, some papers use in the title the term "multi-agent" to indicate the decentralized character of the problem and/or planning approach. When considering the related literature it is important to notice possible hidden simplifications that are connected with devising certain procedures in comparison to the underlying decentralized problem. For example, sometimes it is implicitly supposed that autonomous agents act cooperatively and honestly disclose private information or incorporate a global objective function in their decision making. However, it is preferable to work towards designing incentive-compatible mechanisms where a rational agent's individually best strategy would be to simply behave truthfully.

32.4.1 Classification

In the following we mainly distinguish between auction-based solution methods and negotiation-based solution methods. Both kinds of methods usually build on an iterative procedure with two scheduling levels, global scheduling (in particular allocation of shared resources) and local scheduling (individual projects). Wang et al. (2013) distinguish three basic solution concepts: In the top-down decision mode at first an allocation of shared resources is determined; schedules for the individual projects are locally generated subject to the provided global resources. In the bottom-up decision mode local scheduling is done at first and resulting global resource conflicts are detected afterwards, which usually leads to the need to reschedule local projects. When no such idealized hierarchy of global coordination and local scheduling prevails, the coordination between the individual projects may be carried out according to some peer-to-peer architecture.

We note that some solution approaches make use of payments between original decision maker agents (e.g., prices or side payments). This requires transferable utility in connection with cost-oriented or profit-oriented objective functions.

In addition to the autonomous project manager or resource manager agents (see Sect. 32.3.2) a solution approach may be based on a protocol which involves further active software components, which are often also termed agents. However, in contrast to the original project manager or resource manager agents (which genuinely correspond to the original structure of the decentralized scheduling problem) these artificial agents are purposefully devised and designed to play a particular supportive role in the coordination procedure (e.g., some kind of mediator agent MA). These facilitator agents generally act openly. That is, their behavior is traceable similar to a white box software component. Therefore, these artificial agents may be reproduced and simulated as duplicates in parallel by each genuine decision maker agent (instead of using, e.g., a dedicated mediator agent). On the other hand some solution approaches might presuppose trusted third party agents. Such agents are assumed to honestly follow given rules and to not disclose any received information. However, if it is realistic to have such trusted third party agents

Table 32.2 Overview on classification for solution approaches for the DRCMPSP

Basic mechanism	Auction
	Negotiation
Protocols with or without artificial agents	Facilitator agents with specific roles (e.g., auctioneer or mediator agent (MA), activity agents)
	Trusted third party agents (problematic assumption)
Information exchange	Favored schedules or partial schedules
	Transfers (e.g., willingness to pay for some resource at some time)
	Meta-information (e.g., resource requirements, preferences, activity time windows, objective values)

one might even use a traditional centralized solution procedure, which is carried out by such an agent.

Solution methods generally differ with regard to the kind of information exchange between the agents. This issue is closely connected to the basic solution mechanism. For example, an auction approach usually employs a bidding process where agents partly disclose their willingness to pay for some resource at some time. A negotiation procedure might involve that agents submit particular resource requirements or favored partial schedules or compare different schedules. In general it is not possible to ensure that private information is truthfully unveiled (i.e., disclosed information could be misleading), while only for incentive-compatible mechanisms rational agents might provide unbiased information. It is nonetheless possible that agents are not willing to disclose private information to a large extent to prevent that other agents learn about their costs or revenues and thus might be able to strategically exploit this informational edge.

The main elements of the preceding discussion on classification aspects for solution approaches are summarized in Table 32.2.

32.4.2 Auction Approaches

One stream of scheduling research uses market-oriented auctions to allocate resources among different projects to resolve respective conflicts. An auction at first requires some notion of goods, for which bidders submit bids, which signal the desire to acquire goods at an announced price. An allocation scheme allocates goods to bidders in connection with determined prices. Auctions thus require that the involved parties evaluate schedules or changes to schedules in monetary terms (as the common denominator for transferable utility). The involved payments may actually have to be carried out, since otherwise (virtual/artificial payments) decision makers may simply overbid without bearing the consequences.

Since renewable resources are inherently coupled to the time of use, goods are commonly modeled as a combination of a resource and time. A bid involves the

desire of a project decision maker (agent) to acquire a specific resource at a specific time. Accordingly, such kind of preference information will be disclosed by the project manager agents.

Typical auction types suppose a discrete modeling of goods. Regarding the DRCMPSP one usually assumes a discretization of time as an ordered set of time slots. Then, a basic good represents a combination of a resource and one particular discrete time slot. Since non-interruptible activities generally require a resource for a successive set of time slots, these time slots usually constitute complementary goods (i.e., the benefit from a set of time slots is larger than the sum of individual benefits). Then, results from economic theory show that it is not reasonable to devise a market-oriented mechanism that is based on isolated prices for resource consumption for individual time slots and related allocations (Gul and Stacchetti 1999; Wellman et al. 2001). In general there are no respective equilibrium prices that support optimum solutions. Thus an effective outcome is usually not achievable by such auctions which individually determine prices and allocate resources for single time slots (either sequentially or in parallel). Therefore, it is advisable to use combinatorial auctions. There, a single bid comprises a bundle of goods (Cramton et al. 2006), in our case consumption of a resource for a set of consecutive time slots.

In general it proves computationally infeasible to use a single one-shot combinatorial auction to simultaneously make all required allocations for a complete schedule for some DRCMPSP instance (e.g., because an exponential number of bids might be needed). Therefore, in the literature it is usually proposed to use an iterative combinatorial auction approach. Within an iteration bids may or may not be restricted to a time window or to those activities where all predecessors have already been finished. The auction process evolves according to a successive determination or adjustment of resource prices and allocations until all eligible activities or projects have acquired needed resources.

If there is only one global resource, the owner of this resource [i.e., the corresponding autonomous resource decision maker (agent)] may take on the role of the auctioneer. Otherwise some artificial agent may be devised to facilitate the auction process as a dedicated auctioneer agent.

As an early example of the use of auction approaches for scheduling problems we refer to Kutanoglu and Wu (1999), who consider a decentralized type of the job-shop scheduling problem (a special case of the DRCMPSP where a project is constituted by a single job). They describe iterative combinatorial auction procedures, with different ways to determine prices and payments, which progressively lead to an allocation of machine use (resource consumption) for respective time slots to the jobs (bidders), and discuss the relation to Lagrangean-based decomposition approaches. A related market-oriented auction approach is applied for the DRCMPSP by Lee (2002) and Lee et al. (2003) by constituting for each global resource an artificial market where resource time slots are allocated to the different projects by a combinatorial auction procedure. As long as there is no equilibrium considering the supply and demand for the resource time slots, prices are updated within an iterative tâtonnement process. A related market-based

allocation of resource time slots to different projects is described by Kumara et al. (2002).

The iterative combinatorial auction approach is used by Confessore et al. (2007) for the special version of the DRCMPSP with only one disjunctive global resource ($\mathscr{R}^{global} = \{k_1\}$, $R_{k_1} = 1$, $r_{ik_1} \in \{0, 1\}$ for all $i \in V$). Within an iterative process, each local project manager submits a bid (i.e., a price offer for a bundle of time slots for the shared resource) to an auctioneer (in the role of the owner of the shared resource). The auctioneer heuristically solves the $\mathscr{NP}$-hard winner determination problem and the resulting provisional allocation is the basis for the next round of the auction, where losing agents update their bids to acquire wanted time slots of the shared resource. Further iterative auction-based methods for the DRCMPSP are described by Adhau et al. (2012) and Araúzo et al. (2009, 2010). These approaches also take into account more than one global resource with multi-unit resource availability and a dynamic project portfolio with positive project release times.

Some descriptions of auction-based approaches in the literature do not elaborate on the problem how project manager agents that, e.g., pursue some time-oriented objective function reasonably determine bid prices and whether payments according to the resulting allocations are actually put into effect. Without analyzing such kinds of questions it is problematic to address the issue of incentive compatibility and thus it is uncertain how self-interested project manager agents would behave if such a method would be put into practice. Actually, related methods should be regarded as distributed problem solving approaches with agents that are purposefully designed by a central authority (see the discussion in Sect. 32.2.2). In this respect the genuine decentralized character of the DRCMPSP is not addressed, but those approaches primarily aim at a flexible dynamic scheduling procedure for a complex and uncertain problem setting.

Auctions can be regarded as a special kind of a negotiation, where the interaction protocol is built on a bidding and price-based allocation mechanism as described above. For negotiation approaches in general there are more degrees of freedom; related methods are described in the following section.

32.4.3 Negotiation Approaches

Negotiations generally subsume approaches where autonomous agents communicate with each other according to some interaction protocol with the aim to somehow collaboratively search for an agreement on some negotiation issues (Kraus 2001). In each negotiation round one or more alternative offers (possible agreements) are generated by some agents and accepted or rejected by other agents. Depending on the involved exchange of information one may refer to argumentation-based negotiations, where agents unveil additional information (meta-information) on offers or requests for offers to purposefully influence other agents and the negotiation process towards specific regions of the solution space (Rahwan et al. 2003). For the generation of offers as well as for the generation of meta-information the agents

apply individual negotiation strategies. In order to describe and classify the different negotiation approaches from the literature for the DRCMPSP, we use the following characteristic elements (see also Tables 32.3 and 32.4).

- *Problem type*

  Approaches can be distinguished with regard to the considered decision maker agents [project decision maker agents (PAs), resource decision maker agents (RAs)]. Moreover, negotiation approaches differ in regard to supposing and exploiting transferable utility or not. General issues related to the transfer of utility (e.g., by using money as the common denominator of utility) have already been mentioned in Sects. 32.3.3 and 32.4.1. Approaches that do not presuppose transferable utility are partly motivated by the fact that it may be hard to quantify the monetary gain/loss due to different scheduling decisions (taking into account time-oriented objective functions, which are commonly used in the literature on resource-constrained project scheduling). On the other hand there are approaches where prices and money are involved (e.g., prices for using global resource capacities). These approaches can be useful in cases where the agents have monetary objective functions.
- *Protocol*

  In the context of the DRCMPSP we frequently find mediator-based protocols. Within these approaches an artificial mediator agent (MA) can have different tasks: (1) generate offers in a centralized manner, (2) make acceptance decisions in a centralized manner, (3) support the decentralized generation of offers by project agents, and (4) host a voting procedure in order to determine the negotiation outcome (e.g., on the basis of the acceptance decisions of the decision agents). In addition to mediator-based protocols, we also find one alternating offer protocol, one contract net protocol, and one market-based protocol.
- *Information exchange*

  Negotiation approaches for the DRCMPSP search for an agreement on schedules and the allocation of global resources to projects. In the case that the DRCMPSP model at hand considers transferable utility (e.g., side payments between agents), transfer-related agreements are also sought after. Hence, offers in the context of negotiation approaches for the DRCMPSP include schedule-related information and—in the case of transferable utility—also transfer-related information. With respect to the representation of schedule-related information two main approaches can be distinguished. In the first approach schedule-related information simply consists of starting times of some activities (partial schedules) or of all activities (schedule). In the second approach project ordering lists are used as schedule-related information (Wauters et al. 2010, 2012; Homberger 2012). A project ordering list defines the sequence by which the projects have to be scheduled. The latter approach enforces the generation of feasible solutions because the access to global resources is synchronized.

  Negotiation approaches usually require that some private information are partly disclosed and used (e.g., by a mediator agent) to ensure the generation of feasible schedule offers, to generate acceptance decisions, and to determine the

negotiation outcome. Examples for meta-information are resource requirements and activity time windows, which can be used to generate feasible offers with regard to starting times of activities. In some approaches also local objective values or changes on local objective values are exchanged in order to determine the final outcome of the negotiation. If a method's requirements for information disclosure are very extensive (e.g., in the case that local objectives are exchanged) these approaches actually do not fully address the genuine DRCMSP but the case where in principal the assumptions of centralized multi-project management (see Chap. 31) prevail. For some reasons, however, the solution approach might nonetheless be designed in a distributed way. That is, eventually there may be no information asymmetry and it is assumed that all local project manager agents are cooperative to the effect that they do not opportunistically pursue their own goals but they act as a white box software component (which is purposefully designed by a central authority in order to pursue a collaborative goal).

- *Negotiation strategy*

  Negotiations may be regarded as (decentralized) search procedures; thus it is not surprising that related approaches often incorporate heuristics and concepts from metaheuristics. Therefore, agents are provided with local heuristics, and metaheuristic concepts are used in order to schedule activities, to generate schedule offers, and to make acceptance decisions. The heuristic scheduling procedures partly differ from their counterparts which are used in the context of central project scheduling. For example, a local scheduling procedure (e.g., a serial schedule-generation scheme) which is applied by a project agent usually considers rules for synchronization with the local scheduling procedures of other agents (e.g., Wauters et al. 2010, 2012). In order to incorporate metaheuristic concepts into negotiations, two kinds of approaches can be distinguished. In the first approach, an agent applies a complete metaheuristic in each negotiation round (Lau et al. 2005a,b; Mao et al. 2009; Mao 2011). In the second approach, an agent applies only one element from a metaheuristic in each negotiation round (e.g., Homberger 2012; Fink and Homberger 2013). For example, a crossover operator from evolutionary algorithms to generate a new offer from previous ones may be applied, or a probabilistic acceptance rule according to simulated annealing may be utilized to decide about a submitted offer. Related negotiation approaches thus constitute kind of decentralized metaheuristics.

In the following we describe the main concepts of negotiation approaches for specific versions of the DRCMPSP from the literature (see also Tables 32.3 and 32.4). We first consider approaches which have been designed for problems that do not suppose transferable utility (Table 32.3).

The approach of Homberger (2007) consists of a simple mediator-based protocol, which is based on the idea that the project agents iteratively propose a reallocation (offer) of global resource capacities in order to improve their local schedules. In addition to each offer, the corresponding agent has to provide honest information about the potential improvement on its local objective value in the case that the offer is accepted and the proposed reallocation is arranged. Based on this information, the

Table 32.3 Classification of negotiation approaches for the DRCMPSP—part 1

	Homberger (2007)	Wauters et al. (2010, 2012)	Fink and Homberger (2013)	Homberger (2012)
Problem type				
Decision agents	PAs	PAs	PAs	PAs
Local objectives	Time-oriented	Time-oriented	Monetary	Time-oriented
Global objective(s)	Social welfare	Social welfare	Social welfare, Pareto	Social welfare
Utility transfer	No	No	No	No
Protocol				
Artificial agents	MA	MA	MA	MA
Interaction	Mediator-based	Mediator-based	Mediator-based	Mediator-based
Propose offer	PAs (evolution strategy)	MA, PAs (dispersion game)	MA (ant colony opt.)	MA, PAs (evolution strategy)
Accept/reject offer	MA (global objective)	MA (global objective)	PAs (local objective)	PAs (local objective)
Information exchange				
Offer (schedule-related)	Reallocation of global resource capacities	Project ordering list, activity lists	Schedule	Project ordering list
Offer (transfer-related)	–	–	–	–
Meta-information	Improvement on local objective value	Local objective values	–	–
Strategies				
Heuristics	s-SGS, shift	s-SGS, shift	s-SGS, shift	s-SGS
Metaheuristic	Evolutionary algorithm	Learning automata	–	Evolutionary algorithm

mediator iteratively accepts one offer and reallocates the global resource capacities accordingly. This means that local objective values are used by the mediator similarly to a centralized scheduling approach. It is shown that the approach is suitable for solving large DRCMPSP instances.

The approach of Wauters et al. (2010, 2012) is based on a learning concept. The negotiation is mediated in the sense that a mediator supports the decentralized generation of project ordering lists by managing a dispersion game of project agents. Moreover, the agents use learning automata to generate activity lists which are offered to the mediator. Based on these lists (project ordering list and activity lists) the mediator applies a serial schedule-generation scheme (s-SGS) and a forward/backward shifting heuristic in order to generate and evaluate a complete solution (schedule for each project). The approach is based on the assumption

Table 32.4 Classification of negotiation approaches for the DRCMPSP—part 2

	Lau et al. (2005a,b)	Mao et al. (2009) and Mao (2011)	Lau et al. (2006)
Problem type			
Decision agents	PAs, RAs	PAs, RAs	PAs, RAs
Local objectives	Monetary	Monetary	Monetary
Global objective(s)	Social welfare	Social welfare	Social welfare
Utility transfer	Yes	Yes	Yes
Protocol			
Artificial agents	MA	–	–
Interaction	Alternating offer, mediator-based	Market-based	Contract net
Propose offer	PAs, RAs	RAs	RAs
Accept/reject offer	PAs	PAs	PAs
Information exchange			
Offer (schedule-related)	Start time of an activity	Resource time slots	Start time of an activity
Offer (transfer-related)	Price for performing an activity	Slot prices	Price for performing an activity
Meta-information	Processing time, schedule flexibility information (e.g., activity time window, flexibility costs), relative costs	Resource req., activity time window	Processing time, resource req., activity time window
Strategies			
Heuristics	Shift	s-SGS, priorities, shift	Shift
Metaheuristic	Tabu search	Co-evolutionary algorithm	–

that all problem data and solution information can be used by the mediator in a centralized way. The decentralized idea of this approach is that the project agents—participating in the decentralized generation of encoded offers by choosing positions in the project ordering list—use learning procedures to learn positions and activity lists, which increase local utility. The approach obtains high quality solutions for instances of the MPSPLib (Homberger et al. 2008).

Homberger (2012) suggests a meditated negotiation method on the basis of an evolutionary algorithm. The protocol is based on the idea to search for a project ordering list which allows the project agents to generate local schedules with a high solution quality. Therefore, in each negotiation round the mediator generates and offers a population of project ordering lists. These offers are used by the project agents to generate alternative local schedules in a synchronized way. The agents evaluate their schedules and make acceptance decisions regarding the schedules and

the corresponding project ordering lists. Based on these acceptance decisions, the mediator selects some project ordering lists from the current population as parents in order to generate a new (promising) population of project ordering lists by mutation. In order to avoid early stagnation of the negotiation, mutual acceptance rates are defined. The decentralized generation and evaluation of schedules takes into account that project-oriented precedence relations of activities and local objectives are private information. The meditated negotiation method finds better solutions than the approach of Homberger (2007) for MPSPLib instances.

Fink and Homberger (2013) describe an ant-based coordination mechanism for resource-constrained project scheduling with multiple agents and cash flow objectives. It differs from most of the preceding papers in that it considers precedence constraints between activities from different projects and objective functions which involve cash flows. The approach is mediated in the sense that a mediator agent iteratively generates schedules (offers) on the basis of symmetric information (i.e., restrictions according to private preference information are observed) by using a modified ant colony approach. The centrally generated offers are evaluated internally by the local project manager agents; i.e., the approach allows the agents to keep their private objective values (cash flow values) as a secret. Based on the acceptance decisions of the project agents the mediator updates the pheromones according to the ant colony optimization procedure in order to facilitate an effective generation of schedules. It is shown that the adaptation of pheromones depending on acceptance decisions leads to good solutions in comparison to a centralized approach, which can directly use the objective function values of the agents.

The following approaches consider extended versions of the DRCMPSP which involve resource agents and transferable utility (see Table 32.4).

Lau et al. (2005a,b, 2006) use an extended variant of the DRCMPSP in order to model a supply chain network. Each agent has a monetary objective function (minimizing its operation costs). In addition to the basic problem description in Sect. 32.2.1, earliness costs, lead times, and transportation costs are considered. Moreover, activities can be carried out by a set of alternative resources and corresponding resource agents. Therefore, project agents have to select resource agents when scheduling their activities. In order to solve the extended DRCMPSP the authors develop two negotiation approaches, one on the basis of a contract net protocol (Lau et al. 2006), and one on the basis of an alternating-offers protocol (Lau et al. 2005a,b). The alternating-offers protocol is extended by a mediator-based protocol in order to determine the final outcome by a voting rule. In the context of voting the project agents are enforced to disclose relative cost values with regard to their local objective functions. Lau et al. examine the effect of partial information sharing on the overall system performance within the supply chain structure. In particular, both negotiation processes may be improved when project manager agents honestly share time window information. Overall the alternating-offers protocol outperforms the contract net protocol in terms of the achieved social welfare (total operation costs).

Mao et al. (2009) and Mao (2011) model the scheduling of airport ground handling services as a DRCMPSP. The model takes uncertainty of project release times and of activity durations into account and considers prices for reserving timeslots of resources. Each agent has a monetary objective function, which considers resource utilization costs in case of resource agents, and project delay costs in case of project agents. In order to handle uncertainties a cooperative online scheduling approach is developed. It includes a market-based protocol, which determines the interaction of a project agent and necessary resource agents to negotiate resource time slots. The agents behave in a cooperative manner in the sense that they exchange meta-information (secure activity time windows) during the negotiation. This information allows to increase the flexibility of resource agents to shift time slots and thus to reduce the resource utilization costs. Moreover, in order to handle uncertainties of activity durations the agents learn slack times for handling services with a co-evolutionary algorithm. The approach was successfully tested for instances of a deterministic version of the DRCMPSP (instances from the MPSPLib) and for instances of a dynamic version.

Finally we mention the approaches of Yan et al. (2000), Hao et al. (2006) and Chen and Wang (2007), which focus on implementation issues of negotiations according to multi-agent system architectures.

32.5 Conclusions

The preceding sections provide an overview on the DRCMPSP. This includes a comprehensive discussion on the kind of problem and a review on main solution approaches from the literature. It has been shown that it is necessary to clearly state assumptions on the considered problem with regard to the decentralized character of the decision situation. According to the genuine character of the DRCMPSP it is in general not reasonable to readily presuppose that the involved decision maker agents honestly disclose private preference information and cooperatively work towards some overall objective function. Therefore, it is of particular importance to elaborate on the behavior of the involved self-interested agents and on information asymmetry taking into account conflicting individual objectives and goals of the overall coordination procedure.

In our opinion, avenues for future research on the DRCMPSP include more evaluation effort with regard to an experimental and empirical comparison of different solution approaches. For this purpose, existing benchmark instances (e.g., instances of the MPSPLib) should be extended with regard to further aspects of different DRCMPSP models (e.g., considering both project and resource decision agents as well as actual utility transfers between agents). Moreover, the evaluation should address precisely the underlying assumptions concerning private information and supposed agent behavior. It is of particular interest to analyze the connection between solution quality and a varying extent of information exchange or different assumptions on the behavior of the involved agents.

Negotiation approaches from the literature have been mainly evaluated regarding social welfare measures. Only one coordination mechanism has also been evaluated with regard to Pareto-efficiency (Fink and Homberger 2013). So far no approach has been further assessed on the basis of fairness criteria, which may be regarded as an important aspect of coordination mechanisms (Stadtler 2009).

References

Adhau S, Mittal ML, Mittal A (2012) A multi-agent system for distributed multi-project scheduling: an auction-based negotiation approach. Eng Appl Artif Intel 25:1738–1751

Agnetis A, Pacciarelli D, Pacifici A (2007) Combinatorial models for multi-agent scheduling problems. In: Levner E (ed) Multiprocessor cheduling: theory and applications. I-Tech Education and Publishing, Vienna, pp 21–46

Araúzo JAA, Galán JM, Pajares J, López-Paredes A (2009) Multi-agent technology for scheduling and control projects in multi-project environments. An auction based approach. Intel Artif 13:12–20

Araúzo JA, Pajares J, Lopez-Paredes A (2010) Simulating the dynamic scheduling of project portfolios. Simul Model Pract Theory 18:1428–1441

Brucker P, Drexl A, Möhring R, et al (1999) Resource-constrained project scheduling: notation, classification, models, and methods. Eur J Oper Res 112:3–41

Chen Y, Wang S (2007) An agent-based evolutionary strategic negotiation for project dynamic scheduling. Int J Adv Manuf Technol 35:333–348

Confessore G, Giordani S, Rismondo S (2007) A market-based multi-agent system model for decentralized multi-project scheduling. Ann Oper Res 150:115–135

Cramton P, Shoham Y, Steinberg R (2006) Combinatorial auctions. MIT Press, Cambridge

Fink A, Homberger J (2013) An ant-based coordination mechanism for resource-constrained project scheduling with multiple agents and cash flow objectives. Flex Serv Manuf J 25:94–121

Gul F, Stacchetti E (1999) Walrasian equilibrium with gross substitutes. J Econ Theory 87:95–124

Hao Q, Shen W, Zhang Z, et al (2006) Agent-based collaborative product design engineering: an industrial case study. Comput Ind 57:26–38

Homberger J (2007) A multi-agent system for the decentralized resource-constrained multi-project scheduling problem. Int Trans Oper Res 14:565–589

Homberger J (2012) A (μ, λ)-coordination mechanism for agent-based multi-project scheduling. OR Spectr 34:107–132

Homberger J, Tausch C, Wiedemann C (2008) MPSPLib: multi project scheduling problem library. http://mpsplib.com

Kraus S (2001) Automated negotiation and decision making in multiagent environments. In: Luck M, Marik V, Stepankova O (eds) Multi-agent systems and applications. Lecture notes in computer science, vol 2086. Springer, Berlin, pp 150–172

Kumara SRT, Lee Y-H, Chatterjee K (2002) Distributed multiproject resource control: a market-based approach. CIRP Ann Manuf Technol 51:367–370

Kurtulus I, Davis EW (1982) Multi-project scheduling: categorization of heuristic rules performance. Manag Sci 28:161–172

Kutanoglu E, Wu SD (1999) On combinatorial auction and Lagrangean relaxation for distributed resource scheduling. IIE Trans 31:813–826

Lau JSK, Huang GQ, Mak KL, Liang L (2005a) Distributed project scheduling with information sharing in supply chains: part I—an agent-based negotiation model. Int J Prod Res 43:4813–4838

Lau JSK, Huang GQ, Mak KL, Liang L (2005b) Distributed project scheduling with information sharing in supply chains: part II—theoretical analysis and computational study. Int J Prod Res 43:4899–4927

Lau JSK, Huang GQ, Mak KL, Liang L (2006) Agent-based modeling of supply chains for distributed scheduling. IEEE T Syst Man Cybern A 36:847–861

Lee Y-H (2002) Market-based dynamic resource control of distributed multiple projects. Ph.D. dissertation, Pennsylvania State University

Lee Y-H, Kumara SRT, Chatterjee K (2003) Multiagent based dynamic resource scheduling for distributed multiple projects using a market mechanism. J Intell Manuf 14:471–484

Mao X (2011) Airport under control—multiagent scheduling for airport ground handling. Ph.D. dissertation, Tilburg University, Tilburg

Mao X, Roos N, Salden A (2009) Stable multi-project scheduling of airport ground handling services by heterogeneous agents. In: Sierra C, Castelfranchi C, Decker KS, Sichman JS (eds) Proceedings of the 8th international conference on autonomous agents and multiagent systems (AAMAS '09), Budapest, Hungary, vol 1, pp 537–544

Pritsker AAB, Waiters LJ, Wolfe PM (1969) Multiproject scheduling with limited resources: a zero-one programming approach. Manag Sci 16:93–108

Rahwan I, Ramchurn SD, Jennings NR, McBurney P, Parsons S, Sonenberg L (2003) Argumentation-based negotiation. Knowl Eng Rev 18:343–375

Rosenschein JS, Zlotkin G (1994) Designing conventions for automated negotiation. AI Mag 15(3):29–46

Stadtler H (2009) A framework for collaborative planning and state-of-the-art. OR Spectr 31:5–30

Wang L, Zhan D, Nie L, Xu X (2013) Schema and solutions for decentralised multi-project scheduling problem. Int J Comput Appl Technol 46:142–154

Wauters T, Verbeeck K, Vanden Berghe G, De Causmaecker P (2010) A game theoretic approach to decentralized multi-project scheduling. In: van der Hoek W, Kaminka GA, Lespérance Y, Luck M, Sen S (eds) Proceedings of the 9th international conference on autonomous agents and multiagent systems (AAMAS '10), Toronto, pp 1415–1416

Wauters T, Verbeeck K, De Causmaecker P, Vanden Berghe G (2012) Fast permutation learning. In: Hamadi Y, Schoenauer M (eds) Learning and intelligent optimization. Lecture notes in computer science, vol 7219. Springer, Berlin, pp 292–306

Wellman MP, Walsh WE, Wurman PR, MacKie-Mason JK (2001) Auction protocols for decentralized scheduling. Game Econ Behav 35:271–303

Yan Y, Kuphal T, Bode J (2000) Application of multiagent systems in project management. Int J Prod Econ 68:185–197

Part XI
Project Portfolio Selection Problems

Chapter 33
Multi-Criteria Project Portfolio Selection

Ana F. Carazo

Abstract A very common problem in businesses consists in the planning and allocation of a limited set of resources among a set of candidate projects in order to fund them and carry them out within a given time horizon. Several issues must be taken into account during this decision process: multiple and conflicting objectives, different types of constraints, the planning horizon, and the interdependences between some projects (synergies, precedence, complementarity, incompatibility, etc.).

This chapter provides an in-depth analysis of the main contributions that different authors have made in this field under a multi-criteria approach. The study describes the evolution in the treatment of the key aspects that define the problem of project portfolio selection and shows the advantages and disadvantages of the different approaches. Finally, taking into account all the previously mentioned aspects, a global and very flexible mathematical model is presented that will help decision makers to decide how to invest their scarce resources among a set of candidate projects, that is, how to choose a project portfolio.

Keywords Multi-criteria optimization • Nonlinear binary model • Project planning • Project portfolio selection • Synergies

33.1 Introduction

The problem of selecting a project portfolio arises from the everyday dilemma faced by organizations in finding the best possible way to distribute a limited budget among candidate projects to fulfil the needs of the organization. In this process, the projects compete for funds and other resources (such as manpower, equipment, etc.)

A.F. Carazo (✉)
Department of Economics, Quantitative Methods and Economic History,
Pablo de Olavide University, Sevilla, Spain
e-mail: afercar@upo.es

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_3

to fulfil a set of objectives, priorities, and restrictions of the organization. Usually, there are more projects than resources to carry them out. This decision-making process is a difficult task and requires far more than just the insights and intuition of the managers, given that there is a vast amount of information to manage. Making the wrong decisions during this process may have two negative consequences: resources are wasted, and the benefits that would have been derived from allocating such resources to better projects are also lost.

Therefore, these decisions are crucial. As a consequence of their importance, this process has received significant attention from managers and researchers, as reported in different studies (Archer and Ghasemzadeh 1999; Say et al. 2003; Martinsuo and Lehtonen 2007). Traditionally, decision makers in organizations have performed decision making processes based on either their experience alone or by using a mixture of their professional judgment and ranking tools (Moore and Baker 1969; Cooper et al. 2001), such as financial methods (Silvola 2006), scoring models (Lawson et al. 2006), analytical hierarchy process (Feng et al. 2011), or multi-attribute utility (Duarte and Reis 2006). Thus, projects were selected from the highest to the lowest score until the budget available for the period was spent.

However, these approaches are not always feasible for four main reasons:

- They only take into account one objective when choosing the projects, while in most cases decision makers have to deal with multiple conflicting objectives to answer the needs of the organization (Martino 1995).
- They only take into account one constraint, the budget constraint. However, organizations have to deal with other constraints regarding staff, equipment, political factors, etc. (Mavrotas et al. 2008).
- There may be complementarity and incompatibility relationships as well as synergies between the projects, such that the best group of individual projects may not necessarily be the best set (project portfolio) when all these interactions are taken into account (Chien 2002).
- The dynamic nature of the process is not taken into account. The budget constraint usually refers to one period, and all the projects selected start at the same time. This could be restrictive since some projects may be more flexible than others regarding execution time, and a more dynamic approach would encourage a better distribution of resources (Archer and Ghasemzadeh 1999).

These considerations have led to growing interest in other techniques derived from mathematical programming, as they are able to address the aspects mentioned above. This interest is also driven by advances in the technical procedures used to solve the optimization problems generated (Weber et al. 1990).

Also, nowadays, the increasing size and complexity of many organizations makes this process more complicated, which generates the need to seek a global model that helps any organization decide on the project portfolio selection.

The objective of the present chapter is not the design of the model itself, but the *process followed to arrive at this mathematical model*. We think that the

detailed description of this process allows the appreciation of the key stages in the elaboration of the model and contributes to visualizing the way in which the socioeconomic aspects correspond with the mathematical decisions taken. This process can be structured into the following steps, which will be widely developed in the later sections:

- Identification of the fundamental aspects for a suitable project portfolio selection and an analysis of the way these aspects have been treated over the years in the literature on multi-criteria project portfolio selection.
- Contribution of an alternative for each of the considered aspects, showing how they have been incorporated into the mathematical model produced.

This chapter is structured as follows. Section 33.2 identifies and gives an analysis of the treatment and mathematical formulas that different authors have given, over the years, for each of the factors considered fundamental in making a suitable project portfolio selection. Also, a description is given of the deficiencies that have been found in these models. After analysing each of these aspects, in Sect. 33.3 is shown how we have incorporated them into our own global project portfolio selection proposal, and we briefly formalize the mathematical model proposed to solve the problem of project portfolio selection. Finally, this chapter ends with the main conclusions in Sect. 33.4.

33.2 Identification and Analysis of Key Aspects in Project Portfolio Selection

The aspects or factors that we have identified as fundamental for the presentation of a global model helping decision makers to select an efficient project portfolio are: the simultaneous consideration of diverse objectives (multi-objective optimization), in other words dealing with a *multi-criteria decision making* process; the *interaction* or *dependency* between projects allowing the evaluation of all types of relationships between projects (positive or negative synergies, precedence relationships, complementarity between projects, etc.); and *time* as a fundamental aspect in analysing the problem from a more dynamic and complete aspect that allows giving flexibility to the model. The fundamental nature of these aspects is confirmed by the reviewed literature in the field (see for example: Baker 1974; Ghasemzadeh et al. 1999; Graves and Ringuest 2003; Carazo et al. 2010).

In the following sections we will justify the importance of these aspects and motivate why we felt it was key to include them in a proposed mathematical model for the selection and planning of a project portfolio. We will also focus, among the approaches available, on the optimization models as they offer quantitative tools that will allow us to deal with all the indicated aspects in a combined manner.

33.2.1 Multi-Criteria Problem

The first important issue is the observation that there are often multiple conflicting criteria (such as returns, cost, risk, etc.) that have to be taken into account in the decision process. Therefore, the problem must be analyzed using a multi-criteria approach.

Until the decade of the 1970s most project selection decisions were based on selection models which only considered one criterion. Good examples are the economic assessment measurements [net present value (NPV), the internal rate of return (IRR), etc.] that can be found in several studies (Lorie and Savage 1955; Weingartner 1963; Myers 1972). Later, in the decades of the 1980–1990s, the development of the mono-objective optimization models applied to the field of project portfolio selection allowed decision makers to solve the problem in a simplified manner, seeking that set of projects that optimized the main objective of the organization. Thus, they considered only one criterion subject to the restrictions and conditions established by the organization, which led to the presentation of a simplified and unreal solution to the problem. In these models, the hypothesis usually adopted was that projects are not fractionable, that is to say, the decision variables are usually binary, representing the selection, or not, of each of the investment proposals. However, "versions" of the same project can be considered, that is, alternative forms such as monetary funds or other necessary resources, which can be treated as different projects, although taking into account that it only makes sense to carry out one version.

As the use of mono-objective programming developed as a project portfolio selection technique, the first deficiencies began to arise, as the sphere in which decisions are taken in any organization is usually characterized by a set of competing objectives.

This aspect, however, was not considered in most works until the beginning of the 1990s. From that time, different studies can be found (Czajkowski and Jones 1986; Ringuest and Graves 1989; Schniederjans and Wilson 1991; Santhanam and Kyparisis 1995; Lee and Kim 2000, 2001; Badri et al. 2001; Klapka and Piños 2002; Stummer and Heidenberger 2003; Medaglia et al. 2007, 2008) in which the selection process was made with a multiple objective optimization approach (maximizing: profit, revenue, utility, etc. or minimizing: resource use, cost, risk, runtime, etc.).

The multi-objective nature of project selection and planning is evident. When organizations need to assign their scarce resources to a set of projects, they wish to simultaneously optimize several measures or criteria, such as benefits, risks, the value of the chosen portfolio, and so on. The simultaneous optimization of all the objectives provides a set of solutions called efficient or Pareto optimal. This is the set of feasible non-dominated portfolios, i.e., there is no other portfolio able to yield higher values in at least one objective without detriment to any other.

In this field of study, optimization models have received a lot of attention as they can address different types of interdependencies, multiple objectives and constraints and they allow planning the process in time. Specifically, it is customary to use

multi-objective programming models with binary variables to represent the different candidate projects.

In formal terms, the multi-objective problem of project portfolio selection can be expressed by the following general structure:

$$Opt. \quad F(x) = \{f_1(x), f_2(x), \ldots, f_v(x)\} \tag{33.1}$$

$$s.t. \qquad x \in X, \qquad (x_q \in \{0, 1\}\ ; \quad q = 1, 2, \ldots, m) \tag{33.2}$$

where x is the vector of the decision variables $x = (x_1, x_2, \ldots, x_m)$, whose dimension equals the number of initial candidate projects $m = |Q|$ being Q the set of all candidate projects, where $x_q = 1$ indicates that the project q is selected, and $x_q = 0$ otherwise. On the other hand, $f_\mu(x)$ is the objective function that evaluates μ-*th* criterion ($\mu = 1, \ldots, v$), and X is the feasible region.

In this context, different studies present different approaches to the problem depending on the type of information available and how they incorporate the decision maker's preferences into the process. Authors such as Ghasemzadeh et al. (1999) and Medaglia et al. (2008) combine the different objectives into a single function, assigning a weight score that reflects the relative importance that each objective has for the decision maker. A different approach is offered by goal programming techniques, in which the decision maker can set a priori aspiration levels for each objective function (Santhanam and Kyparisis 1995; Lee and Kim 2001; Badri et al. 2001). Other authors do not use a priori information regarding the preferences of decision makers; rather, once they have obtained a first set of efficient portfolios they include such preferences using interactive techniques (Klapka and Piños 2002; Stummer and Heidenberger 2003), or a multi-criteria tool (Electre, Promethee, and Analytic Hierarchy Methods, among others) that allows them to rank the efficient portfolios (Gaytán and García 2009).

It should be noted that among the different approaches to address this multi-objective problem there is no one that is a priori superior to the others. The selection of one of them mainly depends on the type of information available, and how the preferences of the decision maker are incorporated when choosing one of the possible solutions, so that it better fits those preferences. In the model shown in Sect. 33.3, we have chosen the approach that demands the least information from the decision maker, i.e., to generate the set of efficient portfolios and later select one of the solutions by means of an interactive procedure (Carazo et al. 2012). The fact of presenting the set of efficient solutions (efficient frontier) still remains a major challenge (Doerner et al. 2006; Medaglia et al. 2007) in the context of a nonlinear problem with binary variables. Obtaining this set allows us to identify the tradeoffs between the objectives, which helps the decision maker to better understand the situation.

There are many applications related to multi-criteria project portfolio selection that can be found in the literature. Gaytán and García (2009) presented a multi-objective model for selecting transportation infrastructure projects. Badri et al. (2001) considered project investment decisions in health service institutions.

Santhanam and Kyparisis (1995) used their model to select projects in a large service organization. Mavrotas et al. (2008) developed an approach to evaluate and select projects in the context of a university department. Medaglia et al. (2008) employed portfolio selection in a public enterprise providing water and sewerage services, and Ghasemzadeh et al. (1999) did so in a telecommunications manufacturing company, etc. These models assume the managers of the organizations to be able to estimate costs, human resources, benefits, and any other data relating to the involved projects.

Also, it is not only the techniques and their use that have been changing or evolving over the years, but also the problem to be solved. Currently, it is not so much a question of selecting the best projects, with the resources available, but a matter of considering diverse criteria and choosing a set of interdependent projects in time, which respond to the requirements of the organization. This leads to the incorporation of the other factors that will be analysed in the following Sects. 33.2.2 and 33.2.3.

33.2.2 Interdependence Between Projects

The analysis of the literature on project selection shows that the relationships and interdependences between projects appear as a fundamental aspect, which helps to differentiate between selection of a *group of independent projects* and *selection of a project portfolio*. A project portfolio is a set of projects that share resources during a given period, and among which there may be complementarity, incompatibility, or synergies produced by sharing costs and benefits, derived from conducting more than one project at the same time (Fox et al. 1984). This means that it is not sufficient to simply compare two projects, but rather we need to compare groups of projects (Chien 2002) in order to identify the one best adapted to the needs of the organization. Therefore, the project portfolio concept necessitates a global assessment, which is different from the sum of the individual assessments of each of the projects that make up the portfolio, such as would be the case where a group of independent projects were selected.

Although the first studies in the field of project selection already indicated that it was necessary to incorporate the interactions between projects to achieve a better use of resources (Reiter 1963; Reisman 1965; Reiter and Rice 1966; Weingartner 1966; Baker 1974), their treatment was not considered in a broad sense until the end of the 1980s.

The main reason for this was the difficulty supposed in introducing or formalizing the interactions between projects in the traditionally used simple mathematical models (Baker 1974). These interactions were quantitatively formalized in the works of Czajkowski and Jones (1986) and Schmidt (1993) and shortly after in Dickinson et al. (2001). These works presented different models the main problem being that they only quantified relationships between pairs of projects, that is, they did not allow, for example, that the joint implementation of three or more projects

could reduce or increase the need for a certain resource and/or the value of an objective by a certain amount.

The first of the models that allowed a global study of the interactions was that proposed by Santhanam and Kyparisis (1995). They presented a generic and complete model, which was applicable to all *types* of interdependences and to any *degree* of interaction or interdependence.

There is a classification of these types of interdependences accepted by most researchers, as reflected in the studies by Hillier (1967), Baker (1974), Gear and Cowie (1980), Czajkowski and Jones (1986), Schmidt (1993), Santhanam and Kyparisis (1995, 1996), Chien (2002), Verma and Sinha (2002), among others which distinguish between technical, resource and benefit interactions.

- *Technical or result interactions* take place when the accomplishment of a determined project necessarily involves the joint, total or partial accomplishment of another project or projects, or the non-accomplishment of a determined project or group of projects, as the case may be. These types of interactions are accommodated through additional restrictions. Thus, for example, if project q' depends on the execution of the projects contained in the group $Q(q)$, the restriction $x_{q'} \leq x_q,\ q \in Q(q)$ would be added. Serial relationships or precedence relationships could also be considered within these interactions, as Chien (2002) established. These relationships will be analysed in great detail in Sect. 33.2.3 (in relation to time planning).
- *Resource interactions* originate when the simultaneous implementation of two or more projects requires less (or more) resources than if they were carried out separately. It implies that the cost of a project portfolio is inferior (or superior) to the sum of the costs of all its projects (Spradlin and Kutoloski 1999).
- *Benefit interactions* derive from two or more projects producing greater or less benefits when carried out simultaneously than if they were accomplished at different times.

To model the last two types of interactions, additional terms are included in the objective functions, and/or in the restrictions. The total contribution, whether in each of the objective functions or restrictions involved, is given by the sum of the individual contributions of each selected project, plus (or minus) the additional contributions given by the interdependences if these exist. Adding such terms (for example, in one of the cost/benefit functions) leads to the following expression:

$$f_\mu(x_1, x_2, \ldots, x_m) = c_1x_1 + c_2x_2 + \ldots + c_mx_m + y(x_1, x_2, \ldots, x_m) \qquad (33.3)$$

where c_q indicates the individual contribution of the q-th project and the term $y(x)$ represents the algebraic sum of all the additional terms due to the interdependences between projects.

The other aspect of interest when considering these interdependences is the *degree of interdependences* existing between projects, that is, the interrelationships between two, three, or more projects. Over the years, very different ways have been

attempted to quantify the interdependences included in term $y(x)$ of Eq. (33.3). *Three groups of basic formulas* have been evolving, until we currently have the most global of all, presented by Stummer and Heidenberger (2003). A more general variant will be incorporated in the model shown in Sect. 33.3. Next, each one of these groups will be described.

- The *first group of formulas* only considers binary relationships. That is, they only measure the interactions that may exist through the joint accomplishment of two projects. Examples of these can be found in the studies of Czajkowski and Jones (1986) and Schmidt (1993). Thus, if $c_{qq'}$ is the additional contribution if projects q and q' are carried out simultaneously, then, modelled as a quadratic term, the term $y(x)$ of Eq. (33.3) would be given by:

$$y(x_1, x_2, \ldots, x_m) = \sum_{q=1}^{m-1} \sum_{q'=q+1}^{m} c_{qq'} x_q x_{q'} \tag{33.4}$$

  Later, Santhanam and Kyparisis (1995) incorporated an additional term that included the interactions between three projects, so that:

$$y(x_1, x_2, \ldots, x_m) = \sum_{q=1}^{m-1} \sum_{q'=q+1}^{m} c_{qq'} x_q x_{q'} + \sum_{q=1}^{m-2} \sum_{q'=q+1}^{m-1} \sum_{q''=q'+1}^{m} c_{qq'q''} x_q x_{q'} x_{q''} \tag{33.5}$$

  where $c_{qq'q''}$ is the additional contribution of the simultaneous execution of projects q, q' and, q''. This model is used by authors such as Lee and Kim (2000, 2001) and Klapka and Piños (2002).
- The *second group of formulas* allows the interactions between any number of projects (it measures any degree of interactions) and was introduced by Santhanam and Kyparisis (1995) using a polynomial model. This case assumes the existence of different sets of projects σ where $Q_\rho \subseteq Q$ with $(\rho = 1, \ldots, \sigma)$, which contain at least four projects and produce an additional contribution c_ρ, so that expressions (33.4) and (33.5) would be modified in the following manner:

$$y(x) = \sum_{q=1}^{m-1} \sum_{q'=q+1}^{m} c_{qq'} x_q x_{q'} + \sum_{q=1}^{m-2} \sum_{q'=q+1}^{m-1} \sum_{q''=q'+1}^{m} c_{qq'q''} x_q x_{q'} x_{q''} \tag{33.6}$$
$$+ c_1 \prod_{p \in Q_1} x_p + \ldots + c_\sigma \prod_{p \in Q_\sigma} x_p$$

- *The third of the models* is that proposed by Stummer and Heidenberger (2003), who presented a much more generic formula, which generalizes the previous model. These authors assume that additional effects $\underline{c}_\rho$ take place if the portfolio contains at least a number m_ρ of projects that are elements of some subset $Q_\rho \subseteq Q$ with $(\rho = 1, \ldots, \sigma)$ and that additional effects $\bar{c}_\rho$ are produced if

the portfolio contains, at most, a number M_ρ of some subset $Q_\rho \subseteq Q$ with $(\rho = \sigma + 1, \ldots, \sigma')$ so that:

$$y(x) = \sum_{\rho=1}^{\sigma} \underline{c}_\rho y_\rho(x) + \sum_{\rho=1}^{\sigma'} \bar{c}_\rho z_\rho(x) \tag{33.7}$$

where, on the one hand, $y_\rho(x)$ is a function that will be equal to 1 if the portfolio contains at least m_ρ projects of $Q_\rho \subseteq Q$ with $(\rho = 1, \ldots, \sigma)$, that is, if the interaction ρ is activated, and 0 otherwise. On the other hand, $z_\rho(x)$ is another function that will be equal to 1 if, at most, M_ρ projects of $Q_\rho \subseteq Q$ with $(\rho = \sigma + 1, \ldots, \sigma')$ are selected.

The structure of this formula allows the interaction presented by Santhanam and Kyparisis (1995, 1996) to be incorporated as a particular case in expressions (33.4)–(33.7) when it is established that all the projects of the subset are carried out at the same time.

This proposal has been used in later works such as Doerner et al. (2004, 2006), and will be the one used in the model in Sect. 33.3, both for the objective functions and for some of the restrictions that define the feasible set of portfolios.

33.2.3 Project Planning

The third aspect refers to integrating the selection process within a planning horizon. Organizations seek solutions that enable them to plan their resources over several periods of time. In other words, they seek to develop stable, ongoing policies that allow them to reach their overall economic, social, and environmental objectives in the medium to long term. For this reason, managers face the task of having to simultaneously select project portfolios and plan them within a given planning horizon.

In this context, most studies deal with multi-criteria project portfolio selection first and then with the scheduling of the selected projects, or it is assumed that all the projects selected start at period one (Stummer and Heidenberger 2003). However, this approach may result in some projects not being implemented due to lack of resources in a given period. This drawback may be overcome by using models that are more flexible regarding when the projects are launched.

The importance of suitable project planning within the selection process has already been shown by works like those of Ireland (2002), which establish that not all the projects have to start at the same time, because of both the existence of limited resources of each period and differences in the duration and priority of each project.

In order to analyze a correct treatment of project portfolio planning, two aspects will be commented upon in detail. We will justify why it is fundamental to perform the selection and planning of the portfolio projects simultaneously, and we will briefly describe how this scheduling has been approached in the literature over the years.

33.2.3.1 Justification of the Simultaneous Project Selection and Planning Processes

Most works in the field of project portfolio selection carry out an initial selection of the projects that form the portfolio, and schedule them later, establishing when each of the proposals that compose the portfolio must start. This section will justify why it is more appropriate to carry out a joint selection and planning process.

Consider the following example. Suppose that a certain company tries to select and to plan the projects of a portfolio for a specific planning horizon, say the next 4 years, and it has the following information:

- Type A resource (workforce). It has ten workers per year available.
- Type B resources (budget). It has 1,000€/year for financing the portfolio.

The distribution of the resource needs for each of the six candidate projects is shown in Table 33.1.

As additional information, it is known that the experts have established that there is no preference for the accomplishment of any particular project, and that in addition the portfolio should allow the greatest possible number of projects to be financed with the resources available.

We consider a company selecting a project portfolio without accounting for the scheduling of the projects that compose it. First, it would choose the portfolio on the basis of the resources available, supposing that all the projects would begin at the same time and, later, these projects would be distributed in the planning horizon deciding when each must start. In the specific example, the possible portfolios of candidate projects to finance would be:

- $\{q_1, q_3, q_4\}$ with the resulting consumption of resources: type A ($3+4+3=10$ workers per year); type B ($300+500+200=1{,}000$€).
- Other possible options: $\{q_1, q_2\}$, $\{q_1, q_3\}$, $\{q_1, q_4\}$, $\{q_2, q_4\}$, $\{q_3, q_4\}$, $\{q_4, q_6\}$ or to finance some of the projects $\{q_1\}$, $\{q_2\}$, $\{q_3\}$, $\{q_4\}$, or $\{q_6\}$ individually.

If, in the proposed example, it were established that the project portfolio would be selected based on the inclusion of the greatest number of projects, then $\{q_1, q_3, q_4\}$

Table 33.1 Resource needs and duration of each of the candidate projects

Projects/resources	Type A (workers/year)	Type B (budget/year)	Duration
q_1	3	300€	2 years
q_2	4	600€	4 years
q_3	4	500€	2 years
q_4	3	200€	4 years
q_5	11	50€	1 year
q_6	7	800€	2 years
Total resources/Time available	10	1,000€	4 years

would be selected. Once this selection is made, the execution of the selected projects would have to be scheduled. Three possible alternatives are found:

The first considers that all the projects begin at time zero (see Fig. 33.1):

The second and third options are illustrated in Fig. 33.2. In both cases it is attempted to make a more balanced utilization of the resources, not beginning all the projects at the same time but in a way that allows for a more balanced scheduling.

Following a portfolio selection process of selecting first and scheduling later, could lead, as has been commented in Table 33.2, to an underutilization of available resources year on year. An alternative would be to undertake the selection and planning of the projects at the same time. This alternative would provide the portfolio $\{q_1, q_3, q_4, q_6\}$ making it unlikely that resources remain idle. The scheduling is shown in Fig. 33.3.

As can be seen in Table 33.2, this last option, joint selection and planning, allows the best use of the available resources.

Joint selection and planning attempts to determine which projects to carry out, as well as their start times, so that the pursued objectives are reached, with the condition that the available finances and personnel, etc. are not exceeded at any time. Selection and planning must also accommodate the time restrictions, complementarity and interdependences between the projects, as well as other requirements (strategic, segmentation, policies, etc.) that the organization may have.

This time flexibility regarding the start of projects within the planning horizon means that precedence relationships can appear between some projects, that is, certain projects can only start if others (its precursors) have been completed or if at least a certain amount of time has passed since the latter were started. In addition, it may be of interest to the organization to consider the possibility that not all the selected projects finish within the time horizon, deciding in the following period whether or not to continue with such projects.

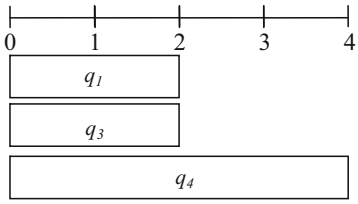


Fig. 33.1 Temporal representation of a portfolio in which all projects begin at start time

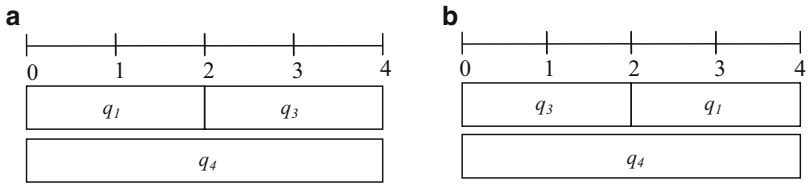


Fig. 33.2 Temporal representation of portfolios in which all the projects do not begin at start time

Table 33.2 Consumption of type A and B resources per year for Figs. 33.1, 33.2, and 33.3

	Fig. 33.1				Fig. 33.2a			
	Resource consumption		Surplus resource		Resource consumption		Surplus Resource	
Year	A	B	A	B	A	B	A	B
1st	3 + 4 + 3 = 10	300 + 500 + 200 = 1,000	0	0	3 + 3 = 6	300 + 200 = 500	4	500
2nd	3 + 4 + 3 = 10	300 + 500 + 200 = 1,000	0	0	3 + 3 = 6	300 + 200 = 500	4	500
3rd	3	200	7	800	4 + 3 = 7	500 + 200 = 700	3	300
4th	3	200	7	800	4 + 3 = 7	500 + 200 = 700	3	300

	Fig. 33.2b				Fig. 33.3			
	Resource consumption		Surplus resource		Resource consumption		Surplus resource	
Year	A	B	A	B	A	B	A	B
1st	4 + 3 = 7	500 + 200 = 700	3	300	3 + 4 + 3 = 10	300 + 500 + 200 = 1,000	0	0
2nd	4 + 3 = 7	500 + 200 = 700	3	300	3 + 4 + 3 = 10	300 + 500 + 200 = 1,000	0	0
3rd	3 + 3 = 6	300 + 200 = 500	4	500	7 + 3 = 10	800 + 200 = 1,000	0	0
4th	3 + 3 = 6	300 + 200 = 500	4	500	7 + 3 = 10	800 + 200 = 1,000	0	0

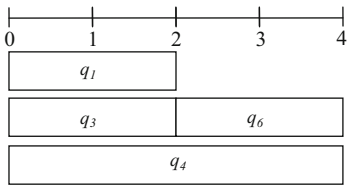


Fig. 33.3 Temporal representation of the portfolio obtained under a simultaneous process of selection and planning

This problem, which in the case of few alternatives seems to be an easy puzzle into which each of its pieces (candidate projects) have simply to be fitted, becomes complex when there are large numbers of combinations, and therefore difficult to solve without the help of a suitable quantitative tool that considers all the aspects.

33.2.3.2 Contributions to Project Selection and Planning in the Literature

The increased complexity of this situation may explain the fact that there are only a few models in the literature that simultaneously address both project selection

and planning within a multi-objective decision-making framework. The main works that have considered it have the main disadvantage of not including the interactions of resources and/or benefits between projects, see for example the paper of Chun (1994), Coffin and Taylor (1996a,b), Ghasemzadeh et al. (1999), Sun and Ma (2005), and Medaglia et al. (2008).

Among the publications we have found that jointly consider time and the interrelationships for project portfolio selection, the following can be highlighted: Dickinson et al. (2001), Stummer and Heidenberger (2003), Doerner et al. (2004, 2006), and Rabbani et al. (2010). The papers of Dickinson et al. (2001) and Rabbani et al. (2010) present the main disadvantages (1) of considering that all the projects must be completed within the time horizon, (2) that the resources remaining from one period cannot be transferred to successive periods, and (3) that they are limited to very elementary interdependences (only admitting binary interdependences). On the other hand, in Stummer and Heidenberger (2003) and Doerner et al. (2004, 2006) we found that, although interrelationships and time are considered as important factors, all the selected projects must begin at the same time.

Consequently, we did not find studies that take all the following aspects into account: flexibility regarding the start of projects within the time horizon; variability of the resources to be consumed in each period, allowing, when possible, to transfer surplus resources to the following period; consideration of the fact that the value of the interrelationships can be different depending on the particular moment in time; and existence of multiple objectives and restrictions for the evaluation of the projects.

To address these aspects, Sect. 33.3 presents a multi-objective binary model that allows the incorporation of all these key aspects under general conditions and which can be applied in both public and private settings.

33.3 Evolution from the Previous Proposals: A More General Mathematical Model

Next, we briefly describe a mathematical model[1] that formulate the multi-criteria project portfolio selection problem in a general way, considering all the key aspects mentioned in the previous section.

Assume an organization with m candidate projects, where $m = |Q|$ and Q is the set of all projects from which efficient portfolios have to be selected according to a set of objectives and some constraints. We are also interested in determining when each selected project will start (t) within a given planning horizon divided into T periods.

[1] A broader description of the model is given in Carazo et al. (2010).

Thus, the decision variables are denoted by x_{qt} and are defined by

$$x_{qt} = \begin{cases} 1, & \text{if project } q \text{ starts at } t \quad (q = 1, \ldots, m; \quad t = 1, \ldots, T) \\ 0, & \text{otherwise} \end{cases} \tag{33.8}$$

and thus $x = (x_{11}, \ldots, x_{1T}, x_{21}, \ldots, x_{2T}, \ldots, x_{m1}, \ldots, x_{mT})$ is a vector with $m \cdot T$ binary variables, which represent one portfolio.

Next, the objective functions used to select the efficient portfolios are shown, followed by the set of constraints that form the feasible set of portfolios:

$$\underset{x}{Opt.} \left\{ \begin{array}{ll} C_{\mu t'}(x) = \sum_{q=1}^{m} \sum_{t=1}^{t'} c_{q\mu(t'+1-t)} \cdot x_{qt} + \sum_{\rho=1}^{\sigma} y_{\rho t'}(x) \cdot \Delta f_{\mu\rho t'} & (\mu = 1, \ldots, v; \quad t' = 1, \ldots, T) \\ C_{\mu}(x) = \sum_{t'=1}^{T} w_{\mu t'} C_{\mu t'}(x) & (\mu = v+1, \ldots, v') \end{array} \right\} \tag{33.9}$$

s.t.

$$\sum_{q=1}^{m} \sum_{t=1}^{t'} r_{qk(t'+1-t)} \cdot x_{qt} + \sum_{\rho=\sigma+1}^{\sigma'} y_{\rho t'}(x) \cdot \Delta f_{k\rho t'} \leq R_k(t') \quad (k = 1, \ldots, K; t' = 1, \ldots, T) \tag{33.10}$$

$$\begin{aligned} &\sum_{q=1}^{m} \sum_{t=1}^{t'} r_{qk(t'+1-t)} \cdot x_{qt} + \sum_{\rho=\sigma+1}^{\sigma'} y_{\rho t'}(x) \cdot \Delta f_{k\rho t'} \leq R_k(t') + (1 + \alpha_{kt'}) \times \\ &\left(R_k(t'-1) - \left(\sum_{q=1}^{m} \sum_{t=1}^{t'-1} r_{qk(t'-t)} \cdot x_{qt} + \sum_{\rho=\sigma+1}^{\sigma'} y_{\rho(t'-1)}(x) \cdot \Delta f_{k\rho(t'-1)} \right) \right) \\ &(k = K+1, \ldots, K'; \; t' = 1, \ldots, T) \end{aligned} \tag{33.11}$$

$$\begin{aligned} &\left(\sum_{q \in Q_\rho} \sum_{t=t'-p_q+1}^{t'} x_{qt} \right) - m_\rho + 1 \leq m \cdot y_{\rho t'}^{m_\rho}(x) \\ &\leq \left(\sum_{q \in Q_\rho} \sum_{t=t'-p_q+1}^{t'} x_{qt} \right) - m_\rho + m \quad (\rho = 1, 2, \ldots, \sigma; t' = 1, \ldots, T) \end{aligned} \tag{33.12}$$

$$\begin{aligned} &M_\rho - \left(\sum_{q \in Q_\rho} \sum_{t=t'-p_q+1}^{t'} x_{qt} \right) + 1 \leq m \cdot y_{\rho t'}^{M_\rho}(x) \\ &\leq M_\rho - \left(\sum_{q \in Q_\rho} \sum_{t=t'-p_q+1}^{t'} x_{qt} \right) + m \quad (\rho = 1, 2, \ldots, \sigma; t' = 1, \ldots, T) \end{aligned} \tag{33.13}$$

$$\underline{b}(t') \leq B(t') \cdot \begin{pmatrix} \sum_{t=t'-p_1+1}^{t'} x_{1t} \\ \ldots \\ \sum_{t=t'-p_m+1}^{k} x_{mt} \end{pmatrix} \leq \overline{b}(t') \qquad (t' = 1, \ldots, T) \tag{33.14}$$

$$\underline{b} \leq B \cdot \begin{pmatrix} \sum_{t=1}^{T} x_{1t} \\ \dots \\ \sum_{t=1}^{T} x_{mt} \end{pmatrix} \leq \overline{b} \tag{33.15}$$

$$a_q \leq \sum_{t=1}^{T} x_{qt} \leq 1 \qquad (q \in Q) \tag{33.16}$$

$$ES \cdot \sum_{t=1}^{T} x_{qt} \leq \sum_{t=1}^{T} t \cdot x_{qt} \leq LS \qquad (q \in Q') \tag{33.17}$$

$$\sum_{t=1}^{T} x_{qt} \geq \sum_{t=1}^{T} x_{q't} \qquad (q \in Q; \quad q' \in Q(q)) \tag{33.18}$$

$$\sum_{t=1}^{T} x_{q't} \cdot \left(\sum_{t^*=1}^{T} t \cdot x_{qt^*} + d_q^{min} \right) \leq \sum_{t=1}^{T} t \cdot x_{q't} \leq \sum_{t=1}^{T} t \cdot x_{qt} + d_q^{max} (q \in Q; q' \in Q(q)) \tag{33.19}$$

$$x_{qt} \in \{0, 1\} \qquad (q \in Q;\ t = 1, \dots, T) \tag{33.20}$$

The objective functions are defined in (33.9). This multi-objective model assumes that the organization wishes to evaluate the portfolios according to a set of attributes μ (cash-flow, sales, risk, etc.). Function $C_{\mu t'}(x)$ is composed of two terms: the first represents the added value of each of the selected projects and the second incorporates the value produced by the interrelationships between projects. It is worth indicating that both terms depend on the specific execution period at which each project q is found in period t', being t' every period of the planning horizon. Thus, we must differentiate between the specific period we are in (period t'), and the execution time of the selected project q up to that time. If project q starts at t, then the execution time of project q in period t' is $t' + 1 - t$. If $t' + 1 - t \leq 0$, the project has not been started yet, and if $t' + 1 - t > p_i$, the project has already been completed. Thus, project q will be active in t' if and only if: $\sum_{t=t'-p_q+1}^{t'} x_{qt} = 1$.

So, if project q starts at t and lasts p_q periods, then $c_{q\mu(t'+1-t)}$ represents the individual contribution of project q to f_μ in period t'.

In addition, $y_{\rho t'}(x)$ is a function that takes value 1 when synergy ρ occurs, and 0 otherwise. Thus, the second part of expression (33.9) represents the effect of positive (or negative) synergies between projects, which is similar to Stummer and Heidenberger's proposal. To consider the synergies or relationships between projects, the organization has also specified different subsets of projects $Q_\rho \subseteq Q$ with $\rho = 1, \dots, \sigma$ such that, if in period t' the portfolio contains a number of projects belonging to Q_ρ that is between m_ρ and M_ρ, there is an increase (or decrease) in value $\Delta f_{\mu\rho t'}$ (synergy ρ, $\rho = 1, \dots, \sigma$) in the attribute μ ($\mu = 1, \dots, \nu$). The technical constraints (33.12) and (33.13) are introduced to ensure that the synergies are activated properly.

On the other hand, the organization may be interested in optimizing the weighted aggregated value of some attributes ($\mu = \nu + 1, \ldots, \nu'$), at different periods. In such a case, the objective functions would be: $C_\mu(x) = \sum_{t'=1}^{T} w_{\mu t'} C_{\mu t'}(x) \quad (\mu = \nu + 1, \ldots, \nu')$ where $w_{\mu t'}$ is the weight assigned to the attribute μ in period t'. Furthermore, if some attribute μ' has an economic value and we want it to be sensitive to the interest rate to reflect different monetary values in each period, then $w_{\mu t'} = (1 + \alpha_{\mu' t'})^{-(t'-1)}$ where $\alpha_{\mu' t'}$ is the interest rate to be applied to the attribute μ' in period t'.

The feasible region is defined by Eqs. (33.10)–(33.20). The time restrictions (for each period t') are specified by the expressions (33.10)–(33.14), and the global ones (those that do not depend on the time period) are dealt with by (33.15)–(33.20).

In particular, the constraints (33.10) and (33.11) determine the availability of resources for every period. They have the same structure as the objective functions (33.9), that is, they are restrictions that deal with possible interactions that may exist between some sub-groups of projects. Symbol $R_k(t')$ denotes the total availability of resource k ($k = 1, \ldots, K$) for the time period t', and $r_{qk(t'+1-t)}$ is the amount of resource k consumed by the project q that began at t. The difference between the two expressions (33.10) and (33.11) is that in the second, reference is made to nonrenewable resources (for example, budget) ($k = K+1, \ldots, K'$) which, if not completely consumed in a period, can be transferred to the following period increased/decreased by the corresponding interest rate ($\alpha_{kt'}$).

The expressions (33.12) and (33.13) are technical restrictions that force in $y_{\rho t'}(x)$ to have a value of 1 if synergy ρ occurs in period t', and 0 otherwise.

Additional linear restrictions are reflected through conditions (33.14) and (33.15). The first one deals with the constraints that the organization imposes on the active projects that may compose the portfolio in each period t', but which do not depend on the specific progress of these projects. In this restriction, $\underline{b}(t')$ and $\bar{b}(t')$ are the lower and upper bound vectors for period t' and $B(t')$ is the coefficient matrix of linear constraints in period t'. In contrast (33.14) expresses linear restrictions that are independent of the time period. In this restriction, $\underline{b}$ and $\bar{b}$ are lower and upper bound vectors and B is the coefficient matrix of global linear constraints. This condition allows it, for example, to formulate the requirement that different versions of the same project cannot belong to the same portfolio.

Expression (33.16) is a restriction that establishes that each project, if selected, can only start once within the time horizon. In addition, this restriction allows the decision makers to establish that a project must be selected by putting a_q equal to 1. Restriction (33.16) establishes time intervals [*ES, LS*] within which certain projects must begin, being Q' a subset of Q.

The last two inequalities formalize the precedence relationships between projects. Inequality (33.18) specifies that a project q' cannot be selected unless its precursors $Q(q)$ have already been selected, and (33.19) specifies that a project q' must be started between d_q^{min} and d_q^{max} periods after starting its precursors.

To summarize, Eqs. (33.8)–(33.20) constitute a multi-objective model with binary variables and a nonlinear structure, whose solution is $\mathcal{NP}$-hard (Ehrgott and Gandibleux 2000). Especially when the number of projects or periods in the

planning horizon increases, it is difficult to reliably solve this model using classic mathematical programming techniques (Rabbani et al. 2010). The intractability of these types of multi-objective problems has driven a growing interest, in recent years, in using heuristic procedures (Coffin and Taylor 1996a; Klapka and Piños 2002; Hsieh and Liu 2004; Doerner et al. 2006; Medaglia et al. 2007) as they provide a good compromise between the quality of the solution and computational time. Evolutionary algorithms are particularly suitable in a multi-objective context, since they provide a set of efficient solutions in a single run, in contrast to the traditional techniques of mathematical programming that require several runs (Deb 2001). Among the authors who have applied these algorithms in portfolio selection are Gaytán and García (2009) and Medaglia et al. (2007).

This problem has been approached using a metaheuristic method called SS-PPS (Scatter Search for Project Portfolio Selection), which was described and empirically validated by Carazo et al. (2010). This method is an adaptation of the evolutionary method SSPMO (Scatter Search Procedure for Multiobjective Optimization, Molina et al. 2007).

33.4 Conclusions

This work has shown the process followed in the construction of a mathematical model which allows the solution of a basic problem in any organization: that of selecting and planning a project portfolio from a group of candidate projects with a multi-criteria approach and a fixed planning horizon. This model takes into account the features of each candidate project, the available resources, the multiple objectives, restrictions and characteristics of the organization, etc.

The review of the literature led us to both identify needs in the field of project portfolio selection and to analyse, review, and incorporate the advances and aspects already considered by other studies. This allowed us to justify the suitability of presenting a more general model than those that existed in the field of portfolio selection and management until now, since, in addition to incorporating all those aspects that have been collected in previous works, it has some global characteristics that render the model more complete.

The mathematical model presented is a multi-objective binary programming model with a nonlinear structure that facilitates the selection of efficient portfolios according to the set of objectives pursued by the organization, as well as their scheduling with regard to the optimum time to launch each project within the portfolio. This global and flexible model includes all those characteristics that different authors have shown to be necessary, not only through incorporation but also through omission, in an integrated manner to solve a rather general version of this kind of project selection problem. In particular:

- It incorporates different objectives pursued by the organization. Some of them can be considered per period of time, and others can be aggregated in time, according to the aims of the central decision maker.
- It incorporates temporal constraints and technical limitations that are established by the decision makers, such as pre-selected projects, precedence relationships between projects, requirements of strategy, policy, etc.
- It incorporates the value of the synergies in this selection process, both in the objectives and in some restrictions, allowing the synergy values to be different depending on the moment in time in which they are originated.
- It allows projects to start in any period of the time horizon. This makes the model flexible allowing, when the organization wishes, that one or more of the projects can finish, or not, within the considered planning horizon, so that when a new period of planning begins the decision maker incorporates his preferences in the model regarding the continuation, or not, of that project.
- It allows unused resources to be transferred from one period to the next, enabling economically measurable resources to be capitalized at the interest rate determined by the central decision maker.

To summarize, the stages of the process that we have followed and demonstrated here, consist of reviewing earlier work, identifying needs and advances, and incorporating them into what exists to produce a more global model. To conclude, we hope that our global model proposal will help in the difficult task faced by decision makers when attempting to solve problems of portfolio selection and planning, and that reflecting on the process will lead to more awareness of the complexity entailed in incorporating different socioeconomic aspects in mathematical models.

References

Archer NP, Ghasemzadeh F (1999) An integrated framework for project selection. Int J Proj Manag 17(4):207–216

Badri MA, Davis D, Davis D (2001) A comprehensive 0-1 goal programming model for project selection. Int J Proj Manag 19(4):243–252

Baker NR (1974) R&D project selection models: an assessment. IEEE Trans Eng Manag EM-21 4:165–171

Carazo AF, Gómez T, Molina J, Hernández-Díaz A, Guerrero F, Caballero R (2010) Solving a comprehensive model for multi-objective project portfolio selection. Comput Oper Res 37:630–639

Carazo AF, Contreras I, Gómez T, Pérez F (2012) A project portfolio selection problem in a group decision-making context. J Ind Manag Optim 8(1):243–261

Chien C (2002) A portfolio-evaluation framework for selecting R&D projects. R&D Manag 32(4):359–368

Chun YH (1994) Sequential decisions under uncertainty in the R&D project selection problem. IEEE Trans Eng Manag 41(4):404–413

Coffin MA, Taylor BW (1996a) R&D project selection and scheduling with a filtered beam search approach. IIE Trans 28(2):167–176

Coffin MA, Taylor BW (1996b) Multiple criteria R&D project selection and scheduling using fuzzy logic. Comput Oper Res 23(3):207–220

Cooper RG, Edgett SJ, Kleinschmidt EJ (2001) Portfolio management for new product development: results of an industry practices study. Res Technol Manag 31(4):361–380

Czajkowski AF, Jones S (1986) Selecting interrelated R&D projects in space technology planning. IEEE Trans Eng Manag EM-33 1:17–24

Deb K (2001). Multi-objective optimization using evolutionary algorithms. Wiley, Chichester

Dickinson MW, Thornton AC, Graves S (2001) Technology portfolio management: optimizing interdependent projects over multiple time periods. IEEE Trans Eng Manag EM-48 4:518–527

Doerner KF, Gutjahr WJ, Hartl RF, Strauss C, Stummer C (2004) Pareto ant colony optimization: a metaheuristic approach to multiobjective portfolio selection. Ann Oper Res 131(1–4):79–99

Doerner KF, Gutjahr WJ, Hartl RF, Strauss C, Stummer C (2006) Pareto ant colony optimization with ILP preprocessing in multiobjective portfolio selection. Eur J Oper Res 171(3):830–841

Duarte BPM, Reis A (2006) Developing a project evaluation system based on multiple attribute value theory. Comput Oper Res 33(5):1488–1504

Ehrgott M, Gandibleux X (2000) A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spektr 22:425–460

Feng B, Ma J, Fan Z-P (2011) An integrated method for collaborative R&D project selection: supporting innovative research teams. Expert Syst Appl 38:5532–5543

Fox GE, Baker NR, Bryant JL (1984) Economic models for R and D project selection in the presence of project interactions. Manag Sci 30:890–902

Gaytán J, García J (2009) Multicriteria decision on interdependent infrastructure transportation projects using an evolutionary-based framework. Appl Soft Comput 9:512–526

Gear TE, Cowie GC (1980) A note on modeling project interdependence in research and development. Decis Sci 2(4):738–748

Ghasemzadeh F, Archer N, Iyogun P (1999) A zero-one model for project portfolio selection and scheduling. J Oper Res Soc 50(7):745–755

Graves SB, Ringuest JL (2003) Models & methods for project selection: concepts from management science, finance and information technology. Kluwer Academic Publishers, Boston

Hillier FS (1967) Chance-constrained programming with 0-1 or bounded continuous decision variables. Manag Sci 14(1):34–51

Hsieh T, Liu H (2004) Genetic algorithm for optimization of infrastructure investment under time-resource constraints. Comput Aided Civil Infrastructure 19:203–212

Ireland LR (2002) Managing multiple projects in 21st century. In: Dye LD, Pennypacker JS (eds) Managing multiple projects: panning, scheduling and allocating resources for competitive advantage. Marcel Dekker Publishers, Centre for Business Practices, New York, pp 21–33

Klapka J, Piños P (2002) Decision support system for multicriterial R&D and information systems projects selection. Eur J Oper Res 140(2):434–446

Lawson CP, Longhurst PJ, Ivey PC (2006) The application of a new research and development project selection model in SMEs. Technovation 26(2):242–250

Lee JW, Kim SH (2000) Using analytic network process and goal programming for interdependent information system project selection. Comput Oper Res 27(4):367–382

Lee JW, Kim SH (2001) An integrated approach for interdependent information system project selection. Int J Proj Manag 19(2):111–118

Lorie JH, Savage LJ (1955) Three problems in rationing capital. J Bus 28:229–239

Martino JP (1995) Research and development project selection. Wiley series in engineering & technology management. Wiley, New York

Martinsuo M, Lehtonen P (2007) Role of single-project management in achieving portfolio management efficiency. Int J Proj Manag 25:56–65

Mavrotas G, Diakoulaki D, Kourentzis A (2008) Selection among ranked projects under segmentation, policy and logical constraints. Eur J Oper Res 187:177–192

Medaglia AL, Graves BS, Ringuest JL (2007) A multiobjective evolutionary approach for linearly constrained project selection under uncertainty. Eur J Oper Res 179(3):869–894

Medaglia AL, Hueth D, Mendieta JC, Sefair JA (2008) A multiobjective model for the selections and timing of public enterprise projects. Socio Econ Plan Sci 42(1):31–45

Molina J, Laguna M, Martí R, Caballero R (2007) SSPMO: a scatter tabu search procedure for non-linear multiobjective optimization. INFORMS J Comput 19(1):91–100

Moore JR, Baker NR (1969) An analytical approach to scoring model design-application to research and development project selection. IEEE Trans Eng Manag 16(3):90–98

Myers SC (1972) A note on linear programming and capital budgeting. J Financ 27:89–92

Rabbani M, Aramoon M, Khoshkhou G (2010) A multiple-objective particle swarm optimization for project selection problem. Expert Syst Appl 37(1):315–321

Reisman A (1965) Capital budgeting for interrelated projects. J Ind Eng 16(1):59–64

Reiter S (1963) Choosing an investment program among interdependent projects. Rev Econ Stud 30(1):32–36

Reiter S, Rice DB (1966) Discrete optimization solution procedures for linear and nonlinear integer programming problems. Manag Sci 12:829–850

Ringuest JL, Graves SB (1989) The linear multi-objective R&D project selection problem. IEEE Trans Eng Manag 36(1):54–57

Santhanam R, Kyparisis J (1995) A multiple criteria decision model for information system project selection. Comput Oper Res 22(8):807–818

Santhanam R, Kyparisis J (1996) A decision model for interdependent information system project selection. Eur J Oper Res 89(2):380–399

Say TE, Fusfeld AR, Parish TD (2003) Is your firm's tech portfolio aligned with its business strategy? Res Technol Manag 46(1):32–38

Schmidt RT (1993) A model for R&D project selection with combined benefit, outcome a resource interactions. IEEE Trans Eng Manag 40(4):403–410

Schniederjans MJ, Wilson RL (1991) Using the analytic hierarchy process and goal programming for information system project selection. Inf Manag 20(5):333–342

Silvola H (2006) Low-intensity R&D and capital budgeting decisions in it firms. In: Epstein MJ, Lee JY (eds) Advances in management accounting (15). Emeral Group Publishing Limited, Bingley, pp 21–49

Spradlin CT, Kutoloski DM (1999) Action-oriented portfolio management. Res Technol Manag 42(2):26–32

Stummer C, Heidenberger K (2003) Interactive R&D portfolio analysis with project interdependencies and time profiles of multiple objectives. IEEE Trans Eng Manag 50(2):175–183

Sun H, Ma T (2005) A packing-multiple-boxes model for R&D project selection and scheduling. Technovation 25(11):1355–1361

Verma D, Sinha KK (2002) Toward a theory of project interdependencies in high tech R&D environments. J Oper Manag 20(5):451–468

Weber R, Werners B, Zimmermann HJ (1990) Planning models for research and development. Eur J Oper Res 48:175–188

Weingartner HM (1963) Mathematical programming and the analysis of capital budgeting problems. Prentice-Hall, Englewood Cliffs

Weingartner HM (1966) Capital budgeting of interrelated projects: survey and synthesis. Manag Sci 12:485–516

Chapter 34
Project Portfolio Selection Under Skill Development

Walter J. Gutjahr

Abstract This chapter surveys models for project portfolio selection that incorporate the development of skills by learning and/or forgetting. Basic learning models (starting with Wright's learning curve) are recapitulated and their relations are discussed. Attention is given to simple exponential as well as to S-shaped learning curves or laws derived from inventory-like considerations. Moreover, it is shown how these models have been used by diverse authors as components of approaches to support staffing and scheduling decisions. Then, the integration of learning and forgetting within models for project selection is described in more detail by providing mathematical programming formulations, discussing approximations, and outlining numerical solution techniques. Also analytical results on optimal project portfolio selection over time are recalled. The survey discusses both, models where skill development goals are formulated as objectives, and models where they are used as constraints. Multi-objective formulations and corresponding solution techniques are outlined as well. Skill-based project selection under uncertainty is identified as a major open issue for future research.

Keywords Core competencies • Learning models • Portfolio optimization • Project portfolio selection • Skill development • Strategic management

34.1 Introduction

It is well-recognized that project selection is one of the most crucial tasks in project management, especially in a multi-project context. The job of conducting a set of projects in a best-possible way concerns the "how" of project management, but it does not yet address the "what" question. As it is rarely the case that a company or institution does not have any influence on which projects to carry out, a suitable choice of a set of projects, a so-called *project portfolio*, constitutes a substantial managerial decision (cf. also Chap. 33 in this book).

W.J. Gutjahr (✉)
Department of Statistics and Operations Research, University of Vienna, Vienna, Austria
e-mail: walter.gutjahr@univie.ac.at

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_4

Considered formally, there are several similarities between project portfolio selection and financial portfolio selection, an area that has received vast attention in the scientific literature and plays a fundamental role in practical economic decision making. On the other hand, there are also differences between both forms of portfolio choice. One of these differences is that carrying out a project is not a "memoryless" investment: Whereas the fact that an investor has put money into a financial asset in the past does not give her or him an advantage for investments in this *particular* asset in the future, the completion of a *project* by a company's own staff increases the chances that similar projects can be carried out successfully at a future time—and it decreases the costs of their execution.

To a good part, this is a consequence of *learning* effects: By working, employees improve their skills, their competencies and their knowledge, and their ability to do related work in the future grows. As a result, the company employing them gains a considerable competitive advantage, since hiring human resources in an ad hoc fashion at the time they are needed is usually expensive, causes frictional losses in project organization, and is sometimes infeasible at all. For some especially innovative strategic directions, the availability of suitably skilled in-house personnel is even an indispensable precondition. Thus, we may say that one of the distinguishing features of *project* portfolio selection, compared to other forms of portfolio selection, is that it faces learning-dependent returns on investment.

Just as skills can grow, they can also diminish. This can be caused simply by *forgetting* as a consequence of a lack of continuous training. Another possible cause of skill reduction is *knowledge depreciation*: Some specific form of knowledge or information may become obsolete because of new technological developments, and a person who is not actively working in the corresponding area may miss the opportunity to update her or his state of knowledge.

The present chapter surveys quantitative models for optimal project selection, taking the aspect of skill development (may it be upwards or downwards) into account. We shall focus on skill changes as an immediate result of carrying out (or not carrying out) some type of work, i.e., on "learning by doing". Because of its complex internal dynamics, this form of skill development poses the greatest challenge for quantitative modeling. Of course, in practice, also skill improvements by external training (courses etc.) play an important role. We can omit this issue here since in most of the outlined modeling approaches, a course equipping somebody with knowledge or increasing her or his skills can be represented as the special case of a project with negative financial return, but favorable learning opportunities.

The endeavor to develop skills on the one hand and the aim of making short-term revenues on the other hand often appear as (possibly conflicting) different objectives, even if from a long-term perspective, they serve the same purpose, since skill development and capacity building is not an end in itself. Some of the models reviewed here take a bi-objective optimization stance to deal with these two concerns. In any case, combining the analysis of returns from projects with that of skill development sheds light on the interesting question how (short-term) monetary goals and (long-term) strategic goals can be balanced. Evidently, the strategic vision of a company and its plans concerning skill development are closely connected.

Ideas such as that of "core competencies" or of the "innovator's dilemma" have had a tremendous impact on the practice of management in the last years, but the related phenomena seem to be still rather poorly understood at the level of theoretical analysis. By incorporating skill development in project selection models, they may become amenable to quantitative modeling.

The organization of this chapter is as follows: Sect. 34.2 provides learning-curve-based modeling approaches to skill evolution and their application in the context of staffing and scheduling. Section 34.3 formulates a model for project selection under skill development integrating the staffing/scheduling aspect. In Sect. 34.4, a slightly different modeling framework is used to investigate optimal project portfolio management over time on an analytical level. Section 34.5 outlines model extensions and more recent developments. Concluding remarks can be found in Sect. 34.6.

34.2 Skill Development: General Models and Applications in Staffing and Scheduling

34.2.1 Learning Curves in Management Science

Mathematical laws for learning and forgetting have been investigated early in psychology. The first paper applying theoretical learning curves in the area of management seems to be Wright (1936). Wright studied the amount to which increased experience of workers in the aircraft industry reduced the time required by them for manufacturing an airplane. His basic law, the *Wright learning curve*, reads

$$y(s) = a\, s^{-b} \tag{34.1}$$

where a is the time needed to produce the first unit, s is the variable representing the number of units, $y(s)$ is the time needed to produce the sth unit, and $b > 0$ is a parameter controlling the speed of learning. Sometimes (34.1) is written in its logarithmic form, $\log y(s) = \log a - b \log s$.

Epple et al. (1991) write the learning curve (34.1) in the form $d_t/u_t = a s_{t-1}^{-b}$, where t is the time period, d_t denotes hours worked, u_t denotes output, and

$$s_t = \sum_{\tau=1}^{t} u_\tau$$

is cumulative output. Then the authors invert both sides, which gives

$$u_t/d_t = \hat{a}\, s_{t-1}^{b} \tag{34.2}$$

with $\hat{a} = 1/a$. Taking logarithms on both sides, we get a linear statistical model enabling the estimation of the parameters $\log \hat{a}$ and b. Note that u_t/d_t is the *production rate*, a measure of the efficiency of work; we shall return to this issue later.

A similar approach is followed in Darr et al. (1995). However, this latter article also includes the possibility of *knowledge depreciation* in the model. Knowledge depreciation (or alternatively *forgetting*, which has the same consequence) acts in the opposite direction as learning: skills that are not exerted decrease. To take account of this effect, cumulative output s_t is replaced in the model equation by a variable $\hat{s}_t$ updated according to $\hat{s}_t = \lambda\, \hat{s}_{t-1} + u_t$ with some (positive) depreciation factor $\lambda \leq 1$. Evidently, the boundary case $\lambda = 1$ reproduces the model without forgetting.

Fioretti (2007) develops a model with the aim to explain the pace and extent at which, by organizational learning, the production time decreases with growing number of produced units. Improved flows between organizational units are seen as a main factor of learning, and consequently the model builds on the changes of these flows.

In Pendharkar and Subramanian (2007), the envisaged application is software development using integrated computer-aided software engineering (ICASE) tools. The authors replace Wright's learning curve by the equation

$$y = a\, e^{-bx} \tag{34.3}$$

with y denoting effort, x denoting tool experience, and constants $a > 0$, $b > 0$. Obviously, by setting $s = e^x$, we get back to (34.1), so on the assumption that tool experience grows as the logarithm of cumulative output, Wright's law is obtained. An essential contribution of the article consists in the extension of the learning curve to *group* learning: A team of K programmers is considered, where each programmer has a learning curve of identical form, but differing experience parameters $x_1, \ldots, x_K$. The values x_k $(k = 1, \ldots, K)$ are assumed as independent identically distributed random variables. The form of the overall team learning curve is derived, and it is shown how the parameters of this curve can estimated by means of an artificial neural network.

Before proceeding to the next articles on learning models, a drawback of Wright's formula should be noted: As $s \to \infty$, the time $y(s)$ to produce a unit tends to zero (or, in other words, the production rate tends to infinity), which may be considered unrealistic. The models described in the following two publications avoid this shortcoming.

Ngwenyama et al. (2007) address the question at which time an IT manager should upgrade the firm's software. If upgrades are done too early, the users have not yet reached the stage where they can make best use of the currently applied software or technology, which reduces average productivity. By too late upgrades, on the other hand, productivity gains offered by the new software or technology may be missed. For modeling the underlying learning process, the authors suggest a learning curve of the following type:

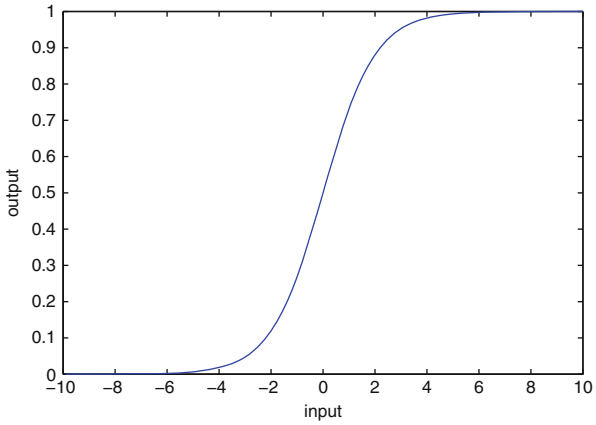


Fig. 34.1 Logistic function for $a = b = 1$

$$r(t) = \frac{r_\infty}{1 + a\, e^{-bt}} \tag{34.4}$$

Therein, $r(t)$ is the value drawn from the currently applied technology (due to benefits or cost reductions) at time t, the constant r_∞ is the upper limit of this value, b is the rate of learning, and a is the curve initialization factor. The function in (34.4) is a *logistic function*. It is increasing, has an S-shape and tends to its upper limit r_∞ as t tends to infinity. In the leftmost part, growth is slow, which expresses the empirical observation that in an early phase of learning, it takes time until skills are improved. In the middle part, the function has larger slope, which represents fast learning. In the rightmost part, the curve flattens again, which takes account of the effect that for a person who already has a high skill level, the marginal skill improvements by additional experience diminish. As the authors emphasize, the added value achieved by learning the technology approaches a plateau, it is not unlimited. In Fig. 34.1, a special logistic function is shown. With t on the input axis and $r(t)$ on the output axis, this curve represents $r(t)$ for $r_\infty = 1$ and $a = b = 1$; the time scale is chosen in such a way that the point where the curve has its maximum slope corresponds to time $t = 0$.

In its main part, the article (Ngwenyama et al. 2007) deals with the decision problem of finding points in time where the technology update should take place. We omit the details since our interest is here rather in the learning model.

Armbruster et al. (2007) present a learning model which is used within the context of the organization of a bucket brigade production system. As an effect of training and growing experience, workers improve their velocity in performing assigned jobs. The growth is modeled by the following formula: At time t, the velocity is given as

$$v(t) = v_\ell + (v_u - v_\ell)(1 - e^{-t/\tau}) \tag{34.5}$$

where v_ℓ and v_u are lower and upper bounds on the velocity, respectively, and τ measures the ease of learning. Since velocity is measured as output per time unit, this quantity can be considered as the production rate, corresponding to the ratio u_t/d_t in (34.2). However, contrary to (34.2), where the right hand side is unbounded as $t \to \infty$, the velocity $v(t)$ given by (34.5) approaches a plateau which it cannot exceed. This is similar as in the model of Ngwenyama et al. (2007), although the function (34.5) is not a logistic function. As an alternative, the authors also discuss the learning curve $v(s) = v_\ell\, s^b$ with s denoting the number of times the worker has performed a task. Obviously, this model corresponds to (34.2). The authors observed that in their simulations of a self-organizing bucket brigade system, both learning models produced similar results.

34.2.2 Integration of Learning Models in Staffing and Scheduling Decisions

In this subsection we shortly recapitulate three articles where skill learning models are incorporated as components of optimization models for decisions on task scheduling or staff assignment. The three papers consider the project(s) to be executed as given; the more complex case where also project portfolio decisions have to be made will be addressed in Sect. 34.3.

Chen and Edgington (2005) view organizational learning as a knowledge creation process. It is clear that in the area of knowledge management, learning on the one hand and knowledge depreciation on the other hand are particularly important issues for planning. The model proposed in Chen and Edgington (2005) is a rather comprehensive optimization model with either the sum or the net present value of different types of benefits as the objective function. Decisions are made about the portion of work each worker devotes in each time period to each task and to each knowledge creation (KC) process. For each task, a required competence is defined, and it is assumed that the degree of competence of each worker for performing each task can be measured. Over time, workers loose competence at a fixed rate by knowledge depreciation, but they can (over-)compensate for that by participating in KC processes, which produce competence increments in proportion to the intensity of the KC process, to the value of a logistic function applied to the time invested in the KC process, and to the value of another logistic function applied to the existing competence. The relation between the required competence to complete a task and the actual competence of the workers assigned to the task is quantified and influences the objective function. By a simulation framework, the authors show typical results from the described model in diverse scenarios.

Wu and Sun (2006) deal with a multi-project R&D environment and address the issues of task scheduling and staff assignment. Their model does not consider forgetting or knowledge depreciation. The learning component of the model is based on a Wright-type learning curve, the basic formula being $\bar{E}_p = \bar{E}_1 p^b$, where

$\bar{E}_p$ denotes the average efficiency, p denotes the total number of time periods spent by the staff on the task under consideration, and b is a parameter. This is similar to Eq. (34.2). However, a difference should be noted: Whereas in (34.2), the variable s stands for the output, the variable p refers to real work time, which is only proportional to the output as long as the efficiency remains constant. We shall examine this difference in more detail in Sect. 34.3. With E_t denoting the efficiency in time period t, we have

$$\bar{E}_p = \frac{1}{p}\sum_{t=1}^{p} E_t \quad \text{and thus} \quad E_p = p\bar{E}_p - (p-1)\bar{E}_{p-1}$$

The authors consider $n = |Q|$ projects, each of them involving one or several tasks. The staff allocation decision consists in the assignment of staff to tasks for each period t. The task scheduling decision consists in determining the workload of each task in each period t. The authors formulate a mixed-integer nonlinear program to minimize outsourcing costs, where it is assumed that each task that cannot be completed before its due date will be subject to outsourcing. Scheduling decisions are represented by real-valued variables, whereas staff allocations and outsourcing decisions are represented by integer variables. A genetic algorithm is used for solving the mathematical model heuristically.

Süer and Tummaluri (2008) propose an operator assignment model considering learning and forgetting for a cellular manufacturing application. In the third phase of a three-phase approach, operators are to be assigned to operations based on their skills. Nine different skill levels are distinguished, and a stochastic model for switches between these levels is described, where the essential criterion for a change of the level is the number of weeks an operator has continuously performed an operation (in the case of a level improvement) or not performed an operation (in the case of level deterioration). Two strategies for the assignment are compared: The first strategy consists in repeatedly solving linear assignment problems with the skill levels as input data. This strategy has the disadvantage that it usually leads to a high degree of specialization by assigning operators to operations at which they are already good, but letting their skills at other operations deteriorate. On the long run, this reduces the overall productivity. The second strategy identifies bottleneck operations and assigns very skilled operators to them, but gives the other operators the opportunity to improve their skills at non-bottleneck operations.

34.3 Project Portfolio Selection Under Skill Development

In this section, we turn to the application of skill development models in the area of project selection. In particular, an optimization model introduced in Gutjahr et al. (2008) under the name *Project Selection, Scheduling and Staffing with Learning Problem* (PSSSLP) will be recapitulated in somewhat more detail. For general

techniques of project portfolio selection and for skill-based project scheduling, the reader is referred to Chaps. 26 and 27 in the first volume of this handbook and Chap. 33 of this volume.

Suppose a set Q of *candidate projects* is given. The upper decision level of the PSSSLP consists then in the selection of a subset of projects, a so-called *project portfolio*, from the set Q. Let $y_q = 1$ if project q is selected, and $y_q = 0$ otherwise.

The (fixed) planning time interval is discretized into T periods $t = 1, \ldots, T$. On the lower decision level, human resources from a set $\mathscr{R}$ have to be assigned to the selected projects over time. As in other models for multi-skilled resources, it is supposed that each human resource $k \in \mathscr{R}$ can have a different skill level z_{kl} in each element l of a set $\mathscr{L}$ of skills (competencies). The PSSSLP model assumes that skills increase by exertion and decrease by non-exertion. Thus, more precisely, the skill levels are real numbers z_{klt} also depending on t as they evolve over time (later, we shall describe how).

Each project $q \in Q$ contains a set $V_q \subseteq V$ of activities, where $V = \bigcup_{q \in Q} V_q$ is the set of all activities occurring in one of the candidate projects. The sets V_q are assumed as disjoint, i.e., projects do not "overlap". Let $a_{iq} = 1$, if $i \in V_q$ and $a_{iq} = 0$ otherwise. For an activity i, its ready time (earliest start time) ES_i and its due date d_i are given in terms of period indices. Furthermore, to each project $q \in Q$, a real number v_q denoting the return gained from its execution is given.

Considered from a (long-term) strategic point of view, the single skills $l \in \mathscr{L}$ may have different values for the company. This can be captured by assigning a weight $w_l > 0$ to each skill $l \in \mathscr{L}$.

From the skill level z_{klt}, an *efficiency* value γ_{klt} of resource k in skill l during period t is obtained by applying some nondecreasing function φ that maps the set of reals into the interval $[0, 1]$. In this survey, we shall call φ the *efficiency function*. Efficiency in skill l is the output of a resource in an activity requiring only skill l, produced during the same time in which a resource *perfectly* skilled in skill l would produce one unit of output. In other words, efficiency is relative production rate in relation to a perfectly skilled worker. For example, if $\varphi(1) = 0.3$ and $\varphi(2) = 0.9$, then a person with skill level 1 and a person with skill level 2 will (during the same time) perform 30 and 90 %, respectively, of the amount of work of a person with perfect skill. It is seen that upper bounds for skill levels need not to exist; nevertheless, by the efficiency function φ mapping skill levels to efficiencies, the marginal efficiency gain will diminish when already high skill levels are further increased, since efficiencies cannot exceed the value 1.

A suitable shape of the efficiency function φ can be determined empirically by restricting the set of possible functions to a parameterized class and estimating the parameters from data. In Gutjahr et al. (2008), the class of logistic functions has been chosen, based on the ideas in Chen and Edgington (2005) and Ngwenyama et al. (2007) (see Sect. 34.2.2). Using this choice, $\varphi(z)$ is of the form

$$\varphi(z) = \left[1 + a\, e^{-bz}\right]^{-1} \tag{34.6}$$

with real-valued parameters $a > 0$ and $b > 0$. For the special case $a = b = 1$, refer again to Fig. 34.1; now the input axis represents the skill level z and the output axis represents the efficiency $\varphi(z)$.

The part of activity i requiring skill l is called the *work package* (i, l). It is assumed that each work package (i, l) consumes a so-called *effective work time* p_{il} $(i \in V,\ l \in \mathscr{L})$. The effective work time p_{il} is defined as the time needed by a person with maximum efficiency $\gamma_{klt} \equiv 1$ for completing work package (i, l). The work time unit is the maximum possible work time in one period. Since efficiencies are typically smaller than one, *real work times* have to be computed: For a person with efficiency $\gamma_{klt} \equiv \gamma < 1$, the real work time needed for performing work package (i, l) is p_{il}/γ, since per period, only the fraction γ of the effective work of one period is performed.

In period t, human resource k has a free *capacity* of $R_k(t) \in [0, 1]$ $(k \in \mathscr{R},\ t = 1, \ldots, T)$, expressed in units of (real) work time. Both the effective work times associated with the work packages and the free capacities of the human resources are assumed as given.

Now let us turn to the dynamics of the skill levels z_{klt}. By learning, the value z_{klt} increases in each period where human resource k works in an activity requiring skill l, and by forgetting (or by knowledge depreciation), the value z_{klt} decreases in each period where human resource k does not use skill l. The initial values z_{kl1} of the skill levels are considered as known. It is assumed that the skill level of person k in skill l increases in each period where person k has worked during an amount x of time in skill l by an increment of size $\eta_l \cdot x$, where the "learning factor" η_l is a proportionality constant that can depend on l. Similarly, it is assumed that the skill level of a person k in skill l is reduced by the amount β_l in each period by forgetting or knowledge depreciation, where β_l is a "forgetting factor" that can depend on l as well. The assumption $\eta_l > \beta_l$ ensures that learning can over-compensate forgetting.

Whereas the upper level-decision on projects to be selected is described by the binary decision variables y_q, the lower-level decision on personnel assignment and work distribution over time (respecting ready times and due dates of projects) is described by real-valued decision variables $x_{iklt} \in [0, 1]$ $(i \in V,\ l \in \mathscr{L},\ k \in \mathscr{R},\ t = 1, \ldots, T)$. The number x_{iklt} gives the planned amount of real work time spent on work package (i, l) by human resource k in period t. Time unit is again the maximum possible work time in one period, so that x_{iklt} cannot exceed the value 1.

There may be constraints requiring that in each period t, the effective work time invested in work package (i, l) must not exceed a value b_{il} $(i \in V,\ l \in \mathscr{L})$. In particular, such constraints also allow it to formulate the requirement that the work on certain activities has to be distributed evenly over time, if this is desired.

In total, the PSSSLP can be formulated as the mixed-integer mathematical program (34.7)–(34.15) below. In view of the nonlinearity of φ, it is evident that this mathematical program is nonlinear.

The objective function of (34.7)–(34.15) is a weighted sum of the total return from the selected projects and the strategic benefits resulting from the improvements of the efficiencies γ_{klt}, aggregated over all persons k of the staff of the company. The improvement is measured by the difference of the efficiencies in the period $T + 1$

following the last planning period T, and the levels at the beginning (period 1). As it is seen, the numbers w_l express the strategic importance of the single skills from the viewpoint of the management. By choosing these values, the management defines the goals concerning the future orientation of the firm with respect to competencies. It is clear that a tradeoff between immediate financial returns and strategic benefits may occur. In such a situation, the management can decide rather to invest in a promising competence development, which comes at a certain price, or rather to put competence development goals aside for the sake of high immediate returns. If the weights w_l can be quantified, the optimization model above determines the optimal compromise, taking the current competencies of the human resources of the company into account.

$$\text{Max.} \sum_{q \in Q} v_q y_q + \sum_{l \in \mathscr{L}} w_l \sum_{k \in \mathscr{R}} (\gamma_{k,l,T+1} - \gamma_{kl1}) \tag{34.7}$$

$$\text{s.t.} \quad \gamma_{klt} = \varphi(z_{klt}) \quad (k \in \mathscr{R};\ l \in \mathscr{L};\ t = 1, \ldots, T) \tag{34.8}$$

$$z_{klt} = z_{kl1} - \beta_l\,(t-1) + \eta_l \sum_{i \in V} \sum_{s=1}^{t-1} x_{ikls} \quad (k \in \mathscr{R};\ l \in \mathscr{L};\ t = 1, \ldots, T) \tag{34.9}$$

$$\sum_{i \in V} \sum_{l \in \mathscr{L}} x_{iklt} \le R_k(t) \quad (k \in \mathscr{R};\ t = 1, \ldots, T) \tag{34.10}$$

$$\sum_{t=ES_i}^{d_i} \sum_{k \in \mathscr{R}} \gamma_{klt}\, x_{iklt} = p_{il} \sum_{q \in Q} a_{iq}\, y_q \quad (i \in V;\ l \in \mathscr{L}) \tag{34.11}$$

$$\sum_{k \in \mathscr{R}} \gamma_{klt}\, x_{iklt} \le b_{il} \quad (i \in V;\ l \in \mathscr{L}; t = 1, \ldots, T) \tag{34.12}$$

$$x_{iklt} = 0 \quad \text{if } t \notin \{ES_i, ES_i + 1, \ldots, d_i\} \quad (i \in V;\ k \in \mathscr{R};\ l \in \mathscr{L};\ t = 1, \ldots, T) \tag{34.13}$$

$$x_{iklt} \ge 0 \quad (i \in V;\ k \in \mathscr{R};\ l \in \mathscr{L};\ t = 1, \ldots, T) \tag{34.14}$$

$$y_q \in \{0, 1\} \quad (q \in Q) \tag{34.15}$$

The first constraint of (34.7)–(34.15) derives efficiencies from skill levels. The second constraint describes the dynamics of skill level evolution by learning and forgetting: note that forgetting causes a fixed decrement β_l per period, whereas the increment by learning is proportional to the invested real work time. The third constraint ensures that the capacity limits of the human resources are respected, and the fourth constraint guarantees that the effective work time required for each work package of a selected project is covered by the invested real work of the human resources, weighed by efficiencies. The sum on the right hand side is 1 exactly if activity i belongs to a selected project and zero otherwise. The fifth constraint

takes upper bounds on the effective work time per period into account. The sixth constraint forbids work in a project before its ready time and after its due date, and the remaining constraints specify the range of the used decision variables.

Because of its nonlinearity, the problem (34.7)–(34.15) is difficult to solve numerically. A considerable simplification is achievable if the learning factor and the forgetting factor can be considered as small, which gives a reasonable approximation for the case of a comparably short planning period.

Mathematically, the assumption of small learning and forgetting factors η_l and β_l, respectively, is represented by setting $\eta_l = \hat{\eta}_l \cdot \epsilon$ and $\beta_l = \hat{\beta}_l \cdot \epsilon$, where the numbers $\hat{\eta}_r$ and $\hat{\beta}_r$ are constants and $\epsilon \ll 1$. This would also make the second (skill-related) term in the objective function of (34.7)–(34.15) proportional to ϵ, if the w_l values would be considered as constants. In order not to fade out the influence of the skill term, it is reasonable to set $w_l = \hat{w}_l/\epsilon$. After performing these substitutions, a Taylor expansion at $\epsilon = 0$, neglecting terms of order $\mathscr{O}(\epsilon^2)$, can be carried out. Subtracting the constant term $-\hat{\beta}_l T$ from the resulting asymptotic approximation of the objective function of (34.7)–(34.15), one obtains

$$\text{Max.} \sum_{q \in Q} v_q y_q + \sum_{l \in \mathscr{L}} \hat{w}_l \hat{\eta}_l \sum_{k \in \mathscr{R}} \varphi'(z_{kl1}) \sum_{i \in V} \sum_{s=1}^{T} x_{ikls} \tag{34.16}$$

with φ' denoting the first derivative of φ. This objective function is linear in the decision variables x_{iklt} and y_q.

Also the constraints simplify in a first order-approximation near $\epsilon = 0$: One gets

$$\sum_{k \in \mathscr{R}} \gamma_{klt}\, x_{iklt} \approx \sum_{k \in \mathscr{R}} \gamma_{kl1}\, x_{iklt} \quad (\epsilon \to 0) \tag{34.17}$$

which is linear in the variables x_{iklt} again. Note that the first two constraints are not required anymore.

Computational Solution. The resulting mixed-integer LP still poses computational challenges if the number $n = |Q|$ of candidate projects is large. For these cases, Gutjahr et al. (2008) propose solve the problem by a metaheuristic. Experiments with ant colony optimization as well a with a genetic algorithm are reported, based on synthetic test instances and on instances from a real-world project management application (E-Commerce Competence Center Austria).

Model Extensions. Model (34.7)–(34.15) and its asymptotic approximation can be extended by additional constraints. The formulations preserve the linear structure of the model version obtained from the asymptotic approximation.

1. *Maximum number of human resources per activity.* The effective work required by an activity can be covered by cooperation of several human resources. However, it is not realistic to allow a too large number of people to contribute to an activity, otherwise the communication overhead could become

counter-productive. It is possible to introduce linear constraints that limit the number of workers assigned to the activities to pre-defined bounds.

2. *Expert constraint.* In order to prevent solutions where the competency required for a work package is covered numerically by cumulating small contributions of a larger number of human resources with low skill level each, a constraint can be formulated ensuring that at least one assigned worker serves as an "expert" for the work package in the sense that s/he contributes a certain minimum amount of effective work.
3. *Minimum and maximum number of projects from predefined sets.* A constraint of this type makes sure that from a given subset of Q, a certain minimum and/or maximum number of projects is selected.
4. *Precedence relations between activities.* As customary in project scheduling, precedence relations between certain activities can be formulated.
5. *Avoiding project interruption.* The underlying scheduling model of (34.7)–(34.15) is preemptive, but if necessary, a constraint can be defined ensuring that once work in a project is started, it does not end (or drop below some minimum activity level) before the project is terminated.

Connection to Classical Learning Models. In its standard version, the learning model recalled in the beginning of Sect. 34.3 adopts the logistic function as the efficiency function φ and continues in this way those prior works that apply S-shaped learning curves. However, the model can also be used to cover learning models of the type of Wright's law (34.1). Below we shortly outline the idea, approximating the time-discrete by a time-continuous learning process (for a formally precise derivation, see Appendix B in Gutjahr 2011). With a parameter $\alpha > 0$, choose the efficiency function as $\varphi(z) = z^\alpha$, restricted to a suitable interval for z, and set $\beta_l = 0$ (no forgetting). Assume work of constant amount x_0 (real work time) per time unit. Then, omitting indices k and l in model (34.7)–(34.15), writing time t as a continuous argument instead of a discrete index, and choosing a suitable initial value $z(1)$, we get $z(t) = c\,t$ with some constant c, and hence $\gamma(t) = c^\alpha\, t^\alpha$. Let $s(t)$ denote the cumulated *effective* work time up to time t. By definition of the efficiency γ,

$$\frac{ds}{dt} = x_0 \cdot \gamma(t) = x_0\, c^\alpha\, t^\alpha$$

Integration gives $s = C\, t^{\alpha+1}$ with some constant C. Thus $s^{\alpha/(\alpha+1)}$ is proportional to $\gamma(t)$, that is, $1/\gamma(t)$ is proportional to s^{-b} with $b = \alpha/(\alpha+1)$. However, the reciprocal $1/\gamma(t)$ of the efficiency corresponds to the time needed for producing one unit at time t, and the cumulated effective work s corresponds to the number of units produced so far. Thus, within the considered interval of skill levels, Wright's law is obtained.

34.4 Analytical Results on Project Portfolio Investment over Time

Whereas Gutjahr et al. (2008) assume that 0–1 decisions on whether or not to include a project in the portfolio have to be made, the model of project portfolio selection with learning proposed in Gutjahr (2011) supposes that projects can also be partially funded. A particular focus of the last-mentioned article is to derive *analytical* results in order to obtain a deeper understanding of some of the mechanisms and features of multi-period project portfolio management under skill development, compared to financial portfolio management or to project portfolio management with skills considered as fixed. Financial portfolios typically use the effect of *diversification* as a means to hedge against risk. The model in Gutjahr (2011) shows that under skill development, the aim of *concentration* becomes important and has to be balanced against the risk-reducing effect of diversification. This provides a theoretical argument for management styles oriented on core competencies.

A basic and an extended variant of the model are presented. In both variants, a set Q is assumed to be given, where $q \in Q$ is now considered rather as a *project class* than as a single project. The decision maker can invest in a project class q by an arbitrary amount of funding. This amount can also vary over time. Time is discretized into periods $t = 1, \ldots, T$.

The human resources available to the company are now viewed as *homogeneous* with respect to skills, i.e., the skill level z_{lt} in skill l at time t is seen as a property of the entire staff rather than as an attribute of an individual worker.

Basic Variant. The basic model variant supposes that to each project class q, a unique particular skill required by this project class is assigned. Different project classes require different skills, so the skill assigned to q can be denoted by q again. This is obviously a rather strong assumption, though it is not uncommon in the literature, cf. Chen and Edgington (2005). It is relaxed in the extended variant of the model which will be described later; therein, each project class $q \in Q$ can require one or more skills from a set $\mathscr{L}$ of skills to a certain pre-defined amount.

By the managerial decision, work time is assigned to project classes for each period t, and similarly as in the model recapitulated in Sect. 34.3, it is assumed that assigning an amount x of work time to project class q (requiring skill q) during a period t increases the skill level of the staff in this skill by an increment proportional to x. The proportionality constant is denoted by η_q. On the other hand, the effect of forgetting reduces the skill level in each skill by a fixed decrement β_q in each period. Thus, if a skill is exerted to a sufficient extent, its level increases, and otherwise it decreases.

As in the model presented in Gutjahr et al. (2008), the efficiency $\gamma_{qt} \in [0, 1]$ in the skill corresponding to project class q in period t is computed from the skill level z_{qt} by the application of a nondecreasing efficiency function $\varphi : \mathbb{R} \to [0, 1]$. If a team with perfect skill in q needs d units of real work time to complete a project

from project class q, then a team with efficiency $\gamma_{qt} \leq 1$ needs $d/\gamma_{qt} \geq d$ units of real work time to complete the same project. In both cases, the *effective* work time is d. A standard choice for φ is again the logistic function (34.6), see Sect. 34.3.

It is assumed that the time capacity of the entire staff remains constant during all periods. Taking the capacity of the staff during one period as the work time unit, the amount of real work time of the staff that is invested into project class q during period t is denoted by x_{qt}. These variables are the decision variables. By the choice of the work time unit, we have $\sum_{q \in Q} x_{qt} = 1$ for all t. The overall *effective* work time in project class q computes as $y_q = \sum_{t=1}^{T} \gamma_{qt}\, x_{qt}$. Note that contrary to the model (34.7)–(34.15), binary decision variables (denoted there by y_q) are not needed anymore; the amount of engagement into a project class q is expressed now by the *continuous* quantities y_q.

To describe the returns from projects, it is assumed that each unit of effective work time invested into any project from project class q yields a return of $v_q > 0$. The values v_q represent the market situation. The model assumes that these values remain constant over the planning horizon.

After substituting for variables as far as possible, we arrive at the following nonlinear mathematical program:

$$\text{Max.} \sum_{q \in Q} v_q \sum_{t=1}^{T} x_{qt}\, \varphi \left(z_{q1} - \beta_q (t-1) + \eta_q \sum_{s=1}^{t-1} x_{qs} \right) \tag{34.18}$$

$$\text{s. t. } \mathbf{x}_t \in \mathbb{S}_n \quad (t = 1, \ldots, T) \tag{34.19}$$

where $\mathbf{x}_t$ is the vector of the variables x_{qt} $(q \in Q)$, and $\mathbb{S}_n = \{\mathbf{x} \in \mathbb{R}^n \mid \sum_{q=1}^{n} x_q = 1\}$ is the standard simplex in the space $\mathbb{R}^n$ with $n = |Q|$.

The following results are proven. We present the statements of the theorems in informal terms; for the formal statements, the reader is referred to the original article.

Theorem 34.1. *If, for each project class, the effort invested in it is not allowed to change over time, then it is optimal to put all efforts into one of the project classes and none into the other project classes.*

Theorem 34.2. *Suppose that parameters and initial values are such that during all periods, the current skill levels remain in an interval where the efficiency function φ is convex. Then there is an optimal policy investing in each period in only one project class. The optimal project class can change from period to period.*

Comment: This may be viewed as a theoretical argument for concentration on "core competencies". However, three points should be noted. First, the condition concerning the local convexity of φ is essential for the result. Considering, e.g., a logistic efficiency function (cf. Fig. 34.1), we see that it has a convex part for smaller skill levels and a concave part for higher levels. Thus, by the result, a "newcomer" lacking expert-level competencies is recommended to invest in a narrow portfolio (which may be changed over time) in all periods, but for a company with expertise

in one or several areas, it may be better to go through periods of diversification. Second, the optimal project class for investment is not defined by the current skill resources alone, but by the relation between skill resources and market. Third, the model is deterministic. In the presence of uncertainty, the advantage with respect to skill development achievable by a narrow portfolio may be outweighed by the disadvantage of higher business risk.

Theorem 34.3. *If the efficiency function φ is strictly increasing and forgetting does not take place ($\beta_q = 0$), an optimal policy under the additional constraint that in each period, investment in only one project class is allowed, consists of identical investment decisions for each period.*

Comment: In other words, the theorem says that a manager who has decided to restrict herself/himself to a focused strategy of portfolio selection only faces a *static* decision problem: s/he has to find out the most profitable project class and can then stick to it as long as the market conditions remain unchanged. (However, except e.g. in the situation given by the conditions of Theorem 2, the mentioned restriction to one project class per period may be suboptimal itself!)

Theorem 34.4. *There is an optimal policy with the following property: In the last period T, it invests only in a single project class. In the second-last period $T-1$, it can invest in this class and in at most one additional project class. In period $T-2$, it can invest in these two classes and in at most one additional project class, etc.*

Comment: Of course, the assumption of a fixed time horizon T is an idealization, so the result cannot be immediately transferred to practice. Its essence is the following: It may happen that a company starts with a broad project portfolio. The theorem recommends then that the management should successively (but in general not abruptly!) restrict the set of active project classes until only one single (optimal) project class survives. Again, it should be noted that the presence of uncertainty or market changes can make this recommendation invalid.

Extended Variant. In applications, a project class often requires several different skills instead of only one. Moreover, also the assumption that the available work capacity is the same for each period may be violated. Therefore, Gutjahr (2011) presents an extended model variant overcoming these two restrictions. The share by which project class $q \in Q$ requires skill $l \in \mathscr{L}$ is represented by a number p_{ql}. As shares, the values p_{ql} are normalized by $\sum_{l \in \mathscr{L}} p_{ql} = 1 \ \forall q \in Q$. If the manager decides to invest an effort y_q in project class q, this generates a demand of effective work of size $p_{ql}\, y_q$ in work package (q, l). Real work times have now triple indices: x_{qlt} denotes the amount of real work in work package (q, l) during period t. The available real work time capacity $R(t)$ in each period t is considered as given. Using these additional parameters, a generalization of model (34.18)–(34.19) can immediately be defined. To give the model more realism with respect to skill investment over time, it is possible to add a homogeneity constraint requiring that the ratio $\gamma_{lt}\, x_{qlt} \,/\, \gamma_{l't}\, x_{ql't}$ between the effective work times in two work packages (q, l) and (q, l') of the same project class remains the same in each period t.

The article discusses numerical solution methods for this extended optimization model and addresses also the case where the degree y_q of investment in project class q is limited by a pre-defined upper bound B_q ($q \in Q$).

In summary, the model recapitulated in this section shows that learning effects on the supply side introduce a strong force towards portfolio concentration into optimal project management. To a smaller or larger extent, this force is counteracted by the force towards diversification caused by the effect of uncertainty on the demand side. Although successful practical project portfolio management evidently depends on a suitable balance between both, an integrated theoretical model for project portfolio selection with learning *under uncertainty* seems not yet to be available.

34.5 Other Recent Literature

Several recent articles and working papers deal with project management under skill development in a similar vein as the models recapitulated in the last two sections. Project selection decisions are not addressed explicitly in all of these paper, but also where this is not the case, the proposed models could be extended to a project portfolio selection framework in a rather straightforward way. Let us outline some of these approaches.

The Model by Heimerl and Kolisch. The article Heimerl and Kolisch (2010) proposes an "inventory-type" model for skill development. It is based on a learning curve of the general form $y(z_{kl}) = f_{kl}(z_{kl})$, where z_{kl} is the experience of human resource $k \in \mathscr{R}$ in skill $l \in \mathscr{L}$, and $y(z_{kl})$ is the unit production time as in (34.1). In the experiments, the function f_{kl} is chosen as

$$f_{kl}(z_{kl}) = a_{kl}e^{-\lambda_{kl}z_{kl}} + b_{kl}$$

which extends the Pendharkar's and Subramanian's formula (34.3) by the addition of a positive constant term b_{kl}, denoting the steady state unit production time. In this way, the problem of production times tending to zero is avoided. Internal and external human resources are distinguished. Work can also be assigned to external resources ("outsourcing"), but typically with higher cost rates.

Time is discretized into periods $t = 1, \ldots, T$. In period t, to human resource k, an amount s_{klt} of work related to skill l is assigned. Experience develops according to the inventory equation

$$z_{klt} = z_{k,l,t-1} - \beta_{klt} + s_{klt} \tag{34.20}$$

where β_{klt} denotes the loss of experience caused by forgetting or knowledge depreciation. By integrating the function f_{kl}, the (real) work time x_{klt} spent by human resource k to perform the assigned s_{klt} units of (effective) work in skill l during period t can be computed. Multiplying x_{klt} by cost rates c_{kt} and summing

over all l, k, and t, the overall cost is obtained. It forms the objective function of the model.

Skill development targets are represented within the model as *constraints*. For each skill $l \in \mathscr{L}$, the firm defines a skill target value ϕ_l, to be reached at the end of the last period T. This gives the constraints

$$\sum_{k \in \mathscr{R}_l^i} \frac{1}{f_{kl}(z_{klT})} \geq \phi_l \quad (l \in \mathscr{L}) \tag{34.21}$$

where $\mathscr{R}_l^i \subseteq \mathscr{R}$ denotes the *internal* human resources having skill l (clearly, skill development goals only refer to the internal personnel). Note that the quotient on the left hand side of Eq. (34.21) gives the production rate of human resource k in skill l at the end of the time horizon.

By adding resource availability constraints $\sum_l x_{klt} \leq R_k(t)$ and the constraints $\sum_k s_{klt} \geq p_{lt}$ indicating that in each period t, the overall required work p_{lt} in skill category l has to covered, a nonlinear mathematical program is obtained. (Nonlinearity results from the property that the function representing the variables x_{klt} by the variables z_{klt} and s_{klt} is a difference of convex functions.) The authors solve the model for special instances by using a primal-dual interior point filter line-search algorithm with repeated choice of random starting points. An interesting managerial insight from these instances is that for a steeper learning curve, a higher degree of specialization among individual resources resulted.

Multi-Objective Project Selection. In the practice of project management, one frequently observes a tradeoff between economic objectives as cost reduction or (immediate) profit maximization on the one hand, strategic goals concerning competence development (which may pay off in the future) on the other hand. The model in Heimerl and Kolisch (2010) takes this tradeoff into account by formulating competence- or skill-related targets as constraints and letting cost represent the objective function. Alternatively, however, one may also follow the well-established approach in multi-objective optimization to define two or more objective functions, to determine the efficient frontier (also known as Pareto frontier) with respect to them, and to leave the final choice from the set of efficient solutions to the decision maker.

In Gutjahr et al. (2010), a multi-objective extension of the project selection model (34.7)–(34.15) is introduced, and numerical solution techniques are investigated. The problem reads now

$$\text{Max.} \{f_1(y), \ldots, f_\nu(y), g_1(x), \ldots, g_{\nu'}(x)\} \tag{34.22}$$

with

$$f_\mu(y) = \sum_{q \in Q} v_q^{(\mu)} y_q \quad (\mu = 1, \ldots, \nu) \tag{34.23}$$

and

$$g_\mu(x) = \sum_{l \in \mathscr{L}} w_l^{(\mu)} \sum_{k \in \mathscr{R}} (\gamma_{k,l,T+1} - \gamma_{kl1}) \quad (\mu = 1, \ldots, \nu') \tag{34.24}$$

The constraints are essentially the same as in (34.7)–(34.15). The set (34.23) of objective functions represents the economic benefits from the selected projects. Therein, the objective function $f_\mu(y)$ represents the μ-th economic benefit measure computed from the values $v_q^{(\mu)}$ assigned to the single projects. Observe that it is possible to evaluate a project according to more than one economic criterion. For instance, $v_q^{(1)}$ could denote return, and $v_q^{(2)}$ could denote turnover of project q. The set (34.24) of objective functions represents the skill-related benefits obtained from the increments of the efficiencies γ_{klt} over the planning horizon. Therein, the objective function $g_\mu(x)$ measures the total increment of weighted efficiencies, cumulated over employees, where the efficiency value corresponding to skill l is weighted by the importance value $w_l^{(\mu)}$. Again, more than one criterion could be used. For instance, the weights $w_l^{(1)}$ could focus on skills needed in a state-of-the-art production technology, whereas the weights $w_l^{(2)}$ could put emphasis on skills required by a promising new alternative technology. Since the efficiencies γ_{klt} are determined by their initial values γ_{kl1} and the assigned real work times x_{ikls} up to time $s = t$, the objectives g_μ are functions of the vector x of the variables x_{iklt}.

As (34.7)–(34.15), the problem above can be simplified by a linear asymptotic approximation, but remains computationally hard. For the upper decision level (determination of the binary variables y_q), the authors investigate two multi-objective metaheuristics, namely the multi-objective genetic algorithm NSGA-II and the swarm-intelligence-based technique P-ACO. The lower-level problem (determination of the continuous variables x_{iklt}) is solved by means of the Linear Programming solver of CPLEX.

If in the multi-objective optimization model (34.22)–(34.24) above, more than two objective functions are included, it may become difficult to present the efficient solutions to the decision maker in a transparent form. Moreover, with growing number of objectives, the number of efficient solutions may become too large for evaluation by direct comparison. The article Stummer et al. (2009) proposes an interactive decision support system enabling the user to explore the solutions of model (34.22)–(34.24). Graphical representation tools are offered to facilitate the analysis, and special mechanisms allow a gradual interactive restriction of the currently considered set of candidate solutions by introducing or strengthening aspiration levels for certain objective functions. All these mechanisms can be computationally fast since as soon as the basic set of efficient solutions is determined, the decision support system can work directly on this set instead of having to solve further optimization problems. Some practical experiences with the application of the method and the decision support system are discussed in Stummer et al. (2012).

Cross-Training Effects. The main contribution of the article Olivella et al. (2013) is that it introduces the effect of *cross training* into the skill development model. It is a frequent observation that learning-by-doing does not only improve the skill immediately needed for the activity under consideration, but, at least to some extent, also *related* skills. The model presented by the authors is similar to model (34.7)–(34.15). As in several other papers, a one-to-one relation between skills and tasks is assumed. Contrary to (34.7)–(34.15), the efficiency function φ does not have the real work time, but the effective work time as its argument. As a consequence, the equation for the update of experience levels looks a little bit different as in (34.7)–(34.15). What is more important, however, is that the efficiency γ_{klt} of human resource k in task (skill) l during period t is not made dependent on the experience in skill l alone, but also on the experience in related skills: The authors assume that

$$\gamma_{klt} = \varphi_l(e_{klt} + e'_{klt})$$

where e_{klt} is the experience of human resource k in task (skill) l at the beginning of period t, and e'_{klt} is the equivalent of the experience of human resource k in task (skill) l at the beginning of period t, obtained by doing *other* tasks. Both e_{klt} and e'_{klt} are measured in units of effective work time.

The objective function considered in Olivella et al. (2013) is total (effective) work performed at the end of the planning horizon. Some tasks may have due dates, others not. For those tasks i that have an assigned due date d_i, a constraint requires that they are completed within this due date. The other jobs can be completed within the planning horizon, but they need not to be completed. Cross-training goals are introduced in the form of constraints (this is similar as in the model in Heimerl and Kolisch 2010): For each skill for which a cross-training goal exists, a minimum level of experience to be reached at the end of the planning horizon is defined.

For the numerical solution, the authors relax one of the constraints and replace the resulting problem by a convex piecewise linear approximation. In a second phase, the solution of the relaxed problem is taken as a starting solution for the construction of a feasible solution to the original problem.

Other Approches. Chacosky et al. (2012) build on a model for learning and forgetting developed in Nembhard and Norman (2007). Only task assignment is addressed, no project portfolio decisions are made. Tasks and skills are considered equivalent. The skill development model is given by

$$\gamma_{kl\tau} = I_{kl} + K_{kl}\left[1 - \exp\left(-\frac{\sum_{t=1}^{\tau} x_{klt}}{L_{kl}}\right)\right] \exp\left(\frac{\sum_{t=1}^{\tau} x_{klt} - \tau}{F_{kl}}\right) \tag{34.25}$$

where $\gamma_{kl\tau}$ is the production rate (efficiency) of individual k in period τ, x_{klt} is a binary variable indicating whether or not individual k is assigned to task (skill) l in period t, L_{kl} and F_{kl} are the parameters governing the speed of learning and forgetting, respectively, and I_{kl} and K_{kl} are initial and steady state efficiency of

individual k in skill l, respectively. The task assignment model assumes an ordered sequence of tasks that have to be passed sequentially. In each period, each individual can be assigned to at most one task, and each task can be performed by at most one individual. The objective function represents the overall output produced after the last task has been completed. The output of an individual k assigned to task l in period t is equal to γ_{klt}. The authors used the binary structure of the decision variables to derive a reformulation of the task assignment problem that can be solved efficiently by standard optimization software.

In Ollila (2009), the decision maker is a research funding agency distributing funding among research groups. The specific feature of the model presented in this paper is that it assumes skill development not to depend on the time of exertion of a skill, but rather on received funding which increases the competence level of the research group.

For further related recent literature, see, for example, Fikri et al. (2011) or Attia et al. (2012). Both papers provide models without project selection decisions, but with consideration of staff assignment and skill development during the execution of multi-period projects.

34.6 Conclusions

Taking account of skill learning and skill depreciation in project selection models is not simply a refinement of basic skill-based modeling approaches. It is a more substantial attempt insofar as competence development lies at the core of the fundamental issue of *strategy* in project management. The "resource-based view of the firm" (see Peteraf 1993) has underlined the observation that typically, companies are heterogeneous in terms of their resources, and that competitive advantage results if distinctive organizational competencies are developed in agreement with market opportunities. To map this feature in formal models for strategic project management, some quantitative representation of competence development is needed. As we have seen, the awareness of this point has grown during the last years, documented by the emerging literature on skill learning in project management.

Concerning opportunities for future research, let us restrict ourselves here to one aspect that deserves particular attention. As generally in the quantitative project management literature, also in the area of project selection under skill development, most articles started with *deterministic* models. However, the presence of uncertainty is notorious in project management, so it would be highly desirable to extend existing optimization or decision support models by incorporating suitable representations of uncertainty, either by stochastic models or by other approaches as robust optimization, fuzzy logic etc. Publications on skill-based project selection under uncertainty are very rare at the moment. As far as strategic objective functions are concerned, competence-driven project selection under uncertainty has been addressed in Gutjahr and Reiter (2010), but one would like to have more comprehensive stochastic models where skill development is also represented within

the constraints. Works on project portfolio selection of this type will inevitably get closer to the area of financial portfolio selection in which stochastic models have a long tradition. A good starting point for investigations may be Liesiö and Salo (2012) where portfolio selection models from financial engineering are adopted to the field of project management. It is an open question how such approaches can be extended to the consideration of skill development.

Acknowledgements The author wants to thank Christian Stummer for useful hints.

References

Armbruster D, Gel ES, Murakami J (2007) Bucket brigades with worker learning. Eur J Oper Res 176:264–274

Attia E-A, Duquenne P, Le-Lann J-M (2012) Decision-based genetic algorithms for solving multi-period project scheduling with dynamically experienced workforce. In: Proceedings of the 9th international conference on modeling, optimization and simulation (MOSIM'12), Bordeaux

Chacosky A, Grasman SE, Hewitt M, Thomas BW (2012) A reformulation technique for solving non-linear workforce planning models with learning. Available at http://myweb.uiowa.edu/bthoa/iowa/Research_files/reformulation_article_Oct2012.pdf. Cited 30 Dec 2013

Chen ANK, Edgington TM (2005) Assessing value in organizational knowledge creation: considerations for knowledge workers. MIS Q 29:279–309

Darr ED, Argote L, Epple D (1995) The acquisition, transfer, and depreciation of knowledge in service organizations: productivity of franchises. Manag Sci 440:1750–1762

Epple D, Argote L, Devadas R (1991) Organizational learning curves: a method for investigating intra-plant transfer of knowledge acquired through learning by doing. Organ Sci 2:58–69

Fikri M, Alaoui AEH, Khomssi ME (2011) Assignment staff with dynamic competencies in multi-projects and multi-periods: modelling and solving by a hybridization of ant colony optimization. Int J Comp Sci and Netw Secur 140:227–240.

Fioretti G (2007) The organizational learning curve. Eur J Oper Res 177:1375–1384

Gutjahr WJ (2011) Optimal dynamic portfolio selection for projects under a competence development model. OR Spectr 33:173–206

Gutjahr WJ, Reiter P (2010) Bi-objective project portfolio selection and staff assignment under uncertainty. Optim Method Softw 59:417–445

Gutjahr WJ, Katzensteiner S, Reiter P, Stummer C, Denk M (2008) Competence-driven project portfolio selection, scheduling and staff assignment. Cent Eur J Oper Res 16:281–306

Gutjahr WJ, Katzensteiner S, Reiter P, Stummer C, Denk M (2010) Multi-objective decision analysis for competence-oriented project portfolio selection. Eur J Oper Res 205:670–679

Heimerl C, Kolisch R (2010) Work assignment to and qualification of multi-skilled human resources under knowledge depreciation and company skill level targets. Int J Prod Res 48:3759–3781

Liesiö J, Salo A (2012) Scenario-based portfolio selection of investment projects with incomplete probability and utility information. Eur J Oper Res 217:162–172

Nembhard DA, Norman BA (2007). Cross training in production systems with human learning and forgetting. In: Nembhard DA (ed) Workforce cross training, Chap. 4. CRC Press, Boca Raton, pp 111–129

Ngwenyama O, Guergachi A, McLaren T (2007) Using the learning curve to maximize IT productivity: a decision analysis model for timing software upgrades. Int J Prod Econ 105:524–535

Olivella J, Corominas A, Pastor R (2013) Task assignment considering cross-training goals and due dates. Int J Prod Res 540:952–962

Ollila M (2009) Dynamic impacts of competence-based competitive research funding. Available at http://salserver.org.aalto.fi/vanhat_sivut/Opinnot/Mat-2.4108/pdf-files/eoll09b.pdf. Cited 30 Dec 2013

Pendharkar PC, Subramanian GH (2007) An empirical study of ICASE learning curves and probability bounds for software development. Eur J Oper Res 183:1086–1096

Peteraf MA (1993) The cornerstones of competitive advantage: a resource-based view. Strateg Manag J 14:179–191

Stummer C, Kiesling E, Gutjahr WJ (2009) A multicriteria decision support system for competence-driven project portfolio selection. Int J Inf Tech Decis 8:379–401

Stummer C, Gutjahr WJ, Denk M, Riedmann H, Froeschl KA (2012) Training on the project: a quantifying approach to competence development. Knowl Manag Res Pract 10:64–78

Süer GA, Tummaluri R (2008) Multi-period operator assignment considering skills, learning and forgetting in labor-intensive cells. Int J Prod Res 46:469–493

Wright TP (1936). Factors affecting the cost of airplanes. J Aeronaut Sci 3:122–128

Wu M-C, Sun S-H (2006) A project scheduling and staff assignment model considering learning effect. Int J Adv Manuf Technol 28:1190–1195

Part XII
Stochastic Project Scheduling

Chapter 35
The Stochastic Time-Constrained Net Present Value Problem

Wolfram Wiesemann and Daniel Kuhn

Abstract The successful management of capital-intensive development and engineering projects requires a careful timing of the involved cash in- and outflows. To this end, the project management literature proposes to schedule the project activities so as to maximize their net present value (NPV), that is, the sum of all discounted cash flows. Traditionally, the literature on NPV maximization ignores the uncertainty inherent in the activity durations and cash flows. In this survey, we argue that this uncertainty should be accounted for explicitly, and we investigate the computational challenges involved in doing so. We then review the two major strands of literature on stochastic NPV maximization. The first set of papers provides optimal solutions under the assumption that the activity durations follow independent exponential distributions. The second strand of literature allows for generic distributions but focuses on suboptimal solutions. We conclude with a list of research questions that we believe deserve further attention.

Keywords Net present value • Project scheduling • Stochastic scheduling • Uncertain cash flows • Uncertain durations

35.1 Introduction

Traditionally, the project scheduling literature has investigated models and algorithms for timing a project's activities so as to minimize the *project makespan*, that is, the time required to complete all project activities, subject to various types of precedence and resource constraints. Despite its popularity, the project makespan largely ignores the financial aspects of a project. While this may be acceptable

W. Wiesemann (✉)
Imperial College Business School, Imperial College London, London, UK
e-mail: ww@imperial.ac.uk

D. Kuhn
Risk Analytics and Optimization Chair, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
e-mail: daniel.kuhn@epfl.ch

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_5

for small or highly time-critical projects, the prudent coordination of cash in- and outflows is crucial for the viability of large and capital-intensive projects found in the construction, IT, and manufacturing industries. In such projects, cash inflows arise from progress payments for completed work, whereas cash outflows account for salaries and payments to contractors as well as investment and operating costs.

The financial viability of a project is typically measured by its *net present value* (NPV), which results from discounting all cash flows to the start time of the project. For example, if a project gives rise to the cash flows $c_1^F, \ldots, c_n^F \in \mathbb{R}$ at the times $t_1, \ldots, t_n$, respectively, then the project's NPV evaluates to

$$\text{NPV} = \sum_{i=1}^{n} \beta^{t_i} \cdot c_i^F$$

where $\beta = (1 + r)^{-1}$ denotes the discount rate associated with the interest rate r per unit time (e.g., $r = 0.1$ for 10 %). By convention, positive cash flows denote cash inflows, while negative cash flows represent expenditures. The NPV can be interpreted as the "cash equivalent" of undertaking the project (Luenberger 1997).

The inclusion of economic considerations in project scheduling dates back to the 1960 of the last century, see Kelley (1961). Project scheduling models with NPV objective were first investigated by Russell (1970). The most elementary formulation disregards resource constraints and assumes that all model parameters are known. More precisely, we consider a project that is defined on an acyclic directed graph $G = (V, E)$, where the nodes $V = \{1, \ldots, n\}$ represent the project activities and the arcs $E \subset V \times V$ denote the precedence relations of finish-start type. Activity $i \in V$ requires $p_i \in \mathbb{R}_{\geq 0}$ time units to finish and gives rise to a cash flow of size c_i^F when it is started. Without loss of generality, we assume that 1 and n are the unique source and sink of the network, respectively, and that activity 1 is to be started at time zero. For a given discount rate $\beta \in (0, 1)$, the problem can then be formulated as follows.

$$\begin{array}{lll} \text{Max.} & \displaystyle\sum_{i \in V} c_i^F \cdot \beta^{S_i} & \\ \text{s. t.} & S \in \mathbb{R}^n & \\ & S_j \geq S_i + p_i & ((i, j) \in E) \\ & S_1 = 0 & \end{array}$$

In this formulation, which we henceforth refer to as *NPV*, the decision variable S_i denotes the start time of activity $i \in V$. The constraints ensure nonnegativity of the project schedule and satisfaction of the precedence constraints. A project deadline of $\overline{d}$ can be imposed by adding the constraint $S_n + p_n \leq \overline{d}$. We say that problem *NPV* is *deterministic* since it assumes that the network structure G, the activity durations p_i and cash flows c_i^F, as well as the discount rate β are known. Problem *NPV* is called *time-constrained* as it only accounts for precedence relations but disregards

resource constraints. Note that the NPV maximization problem *NPV* generalizes the makespan minimization problem, which we recover by setting $c_i^F = 0$, $i = 1, \ldots, n-1$, and $c_n^F = 1$.

Although the objective function in *NPV* is nonconvex, the variable substitution $y_i := \beta^{S_i}$ allows us to equivalently reformulate the problem as a linear program:

$$\begin{array}{lll} \text{Max.} & \sum_{i \in V} c_i^F \cdot y_i & \\ \text{s. t.} & y \in \mathbb{R}^n_{\geq 0} & \\ & y_j \leq \beta^{p_i} \cdot y_i & ((i,j) \in E) \\ & y_1 = 1 & \end{array}$$

This reformulation, which we henceforth refer to as *NPV'*, has been discovered by Grinold (1972). A project deadline of $\overline{d}$ can again be enforced by adding the constraint $y_n \geq \beta^{\overline{d}-p_n}$. The resulting problem can be solved with a network simplex algorithm. More efficient solution schemes have been proposed by Elmaghraby and Herroelen (1990), Neumann and Zimmermann (2000), and Schwindt and Zimmermann (2001).

Over the last decades, numerous authors have extended the deterministic time-constrained NPV maximization problem *NPV* to accommodate renewable, nonrenewable, and doubly-constrained resources as well as time-dependent cash flows. We refer the interested reader to Herroelen et al. (1997), Kimms (2001), Demeulemeester and Herroelen (2002) and Chap. 14 in the first volume of this book for extensive reviews of the literature. Perhaps surprisingly, the stochastic version of the time-constrained NPV maximization problem has received much less attention. This chapter aims to give an overview of the available literature and point out interesting avenues for future research. While we focus on the temporal aspects of the problem and disregard phenomena related to resource usage, we provide references to the literature on the resource-constrained formulations where applicable. In terms of the classification scheme proposed by Brucker et al. (1999), we thus study the problems $PS|p_j = sto, prec|\sum c_j^{\mathrm{F}} \beta_j^{\mathrm{C}_j}$ and $PS|p_j = sto, temp|\sum c_j^{\mathrm{F}} \beta_j^{\mathrm{C}_j}$, which correspond to the problems $\circ|cpm, cont, c_j|E[npv]$ and $\circ|gpr, cont, c_j|E[npv]$ in the classification scheme by Herroelen et al. (1999).

The literature on NPV maximization in stochastic project *selection* is reviewed by Kavadias and Loch (2004). NPV maximization problems in stochastic machine scheduling and process systems engineering are discussed by Slotnick (2011), Verderame et al. (2010), Ierapetritou and Li (2009), Li and Ierapetritou (2008), while NPV maximization in new product development is studied, amongst others, by Colvin and Maravelias (2010, 2011), De Reyck et al. (2007), and De Reyck and Leus (2008), see Chap. 57 in this book. The estimation of activity durations and cash flows is studied by Lock (2007), Maylor (2010), and Meredith and Mantel (2006). Herroelen et al. (1997) illuminate the relation between stochastic NPV maximization and real options theory (Dixit and Pindyck 1994). For the literature on the payment scheduling problem, where the cash flows are themselves decision

variables that can be optimized by the project manager or the client, we refer to Dayanand and Padman (1999, 2001), He et al. (2012), and Kolisch and Padman (2001).

The remainder of this chapter proceeds as follows. The next section highlights the importance of stochastic project scheduling and its associated computational challenges. Section 35.3 reviews the literature on the stochastic time-constrained NPV problem with independent and exponentially distributed activity durations, whereas Sect. 35.4 surveys the available solution methods when the activity durations are modeled as random variables that follow generic distributions. We conclude in Sect. 35.5 with some open questions that we believe deserve further scrutiny.

35.2 Stochastic Project Scheduling: Importance and Challenges

A key characteristic of projects is their one-off nature (Project Management Institute 2013), which implies that the parameters in problem *NPV* are typically uncertain at the planning stage. In particular, some of the activities $i \in V$ required to complete the project may be unknown, the activity durations p_i and cash flows c_i^F may be uncertain, and the interest rate r underlying the discount rate β can fluctuate subject to the financial situation of the company as well as macroeconomic developments. Adopting the consensus view in the project scheduling literature, we assume that the network structure G and the discount rate β are known, and we study the consequences of uncertain activity durations p and cash flows c^F. For a survey on project scheduling with stochastic network structure, see Neumann (1999).

The stochastic NPV maximization problem models the activity durations and cash flows as random variables. In the following, we designate random objects by the tilde sign, that is, the uncertain activity durations and cash flows are denoted by $\tilde{p}$ and $\tilde{c}^F$, whereas their realizations are written as p and c^F. To simplify the exposition, we assume that the cash flow $\tilde{c}_i^F$ arises when activity i is *started*, whereas the duration $\tilde{p}_i$ of activity i is known once i is *completed*. The models can readily be adapted to accommodate cash flows that arise when the activities are completed. In analogy to the deterministic NPV maximization problem, our objective is to select activity start times that maximize the project's NPV. However, the stochastic variant of the problem raises several new questions. Firstly, how can we choose activity start times when we do not know the activity durations? After all, assigning a deterministic start time to an activity $j \in V$ may violate the precedence constraints if a predecessor of j finishes later than anticipated. Secondly, how can we compare the NPVs of different project schedules when the NPVs are themselves random variables? In fact, being the discounted sum of the uncertain activity cash flows, the project's NPV is itself no longer deterministic.

In view of the first question, we can think of the project as a stochastic process that evolves in continuous time. At any point in time t, some of the activities are

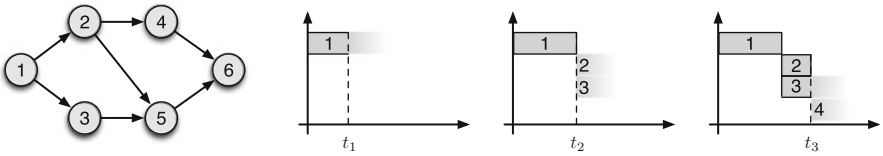


Fig. 35.1 For the project graph on the *left*, the Gantt charts on the *right* illustrate the project evolution at three time points $t_1 < t_2 < t_3$. At time t_1, the project manager only knows the realization of $\tilde{c}_1^F$ since the first activity is still active. At time t_2, the first activity has finished. Knowing the realizations of $\tilde{c}_1^F$ and $\tilde{p}_1$, the project manager decides to start activities 2 and 3. At time t_3, the second activity has finished. Under the knowledge of $\tilde{c}_1^F$, $\tilde{c}_2^F$, $\tilde{c}_3^F$, $\tilde{p}_1$, and $\tilde{p}_2$, the project manager initiates activity 4

completed, some may be active, and others are idle, that is, they have not yet been initiated. The project manager can then decide which of the idle activities are started at time t, given (1) her knowledge of the durations of the activities that are completed by time t, (2) the information that the durations of the active activities must exceed their current "lifetimes", and (3) the knowledge of all the cash flows of the activities that have been started by time t. This is the *non-anticipativity requirement* in stochastic programming (Kall and Wallace 1994; Ruszczyński and Shapiro 2003): the decisions selected at any point in time must only depend on the available information, that is, we are not allowed to "look into the future". Contrary to the deterministic NPV maximization problem, the activity start times are therefore no longer deterministic decision variables, but they represent non-anticipative functions of the available information. Mathematically speaking, we say that the activity start times have to be stopping times. The situation is complicated by the fact that the times at which we observe the realizations of the random variables $\tilde{p}_i$ and $\tilde{c}_i^F$ themselves depend on our scheduling decisions, that is, the information structure of our problem is decision-dependent. Problems of this type are typically very challenging to solve, and they have received fairly limited attention in the stochastic programming literature. Figure 35.1 illustrates the situation.

How can we compare the NPVs of two project schedules if those NPVs are random variables? A preference order over random variables is commonly obtained via *risk measures*, which assign each random variable a real number. An elementary risk measure that is used extensively in the stochastic project scheduling literature is the expected value, which can be interpreted as the "average outcome" that we would expect to observe if we repeated the same project many times. Although the expected value is attractive from a computational viewpoint, it does not account for the risk aversion of the decision maker. Indeed, most projects are unique undertakings that involve substantial financial investments. As such, the project manager might be much more concerned about particularly undesirable project outcomes than about the average performance. Risk measures that account for the decision maker's risk aversion include the variance, the value-at-risk (VaR), and the conditional value-at-risk (CVaR), see Pflug (2000) and Rockafellar and

Uryasev (2000). Omitting some technical details, the ε-VaR of a random variable $\tilde{\xi}$ represents the ε-quantile of the distribution of $\tilde{\xi}$, whereas the ε-CVaR denotes the expected value of $\tilde{\xi}$ under the assumption that $\tilde{\xi}$ takes on one its $\varepsilon \cdot 100\,\%$ "worst" values. Thus, using VaR or CVaR gives the undesirable outcomes more weight in the objective than the average outcomes.

Our discussion so far indicates that the stochastic NPV maximization problem is significantly more involved than its deterministic counterpart. As such, the question naturally arises whether it may be sufficient to approximate the stochastic problem with a *nominal model* in which we replace the uncertain parameters with deterministic quantities (such as their expected or most likely values). To answer this question, we follow the reasoning by Möhring (2001) and consider a project $G = (V, E)$ with activities $V = \{1, \ldots, n\}$ and precedences $E = \{(1, i) : i = 2, \ldots, n-1\} \cup \{(i, n) : i = 2, \ldots, n-1\}$. The cash flows satisfy $\tilde{c}_i^F = 0$, $i = 1, \ldots, n-1$, and $\tilde{c}_n^F = 1$. The durations $\tilde{p}_2, \ldots, \tilde{p}_{n-1}$ follow independent uniform distributions with support $[0, 10]$, whereas $\tilde{p}_1 = \tilde{p}_n = 0$ almost surely. Figure 35.2 (left) illustrates the project network for $n = 6$. If we replace the uncertain activity durations with their expected values, then the nominal model amounts to a deterministic time-constrained NPV problem with activity durations $p_1 = p_n = 0$ and $p_i = 5, i = 2, \ldots, n-1$, and cash flows $c_i^F = 0, i = 1, \ldots, n-1$, and $c_n^F = 1$. For a fixed discount rate $\beta \in (0, 1]$, the deterministic model is optimized by the early start schedule ("start each activity as soon as all of its precessors are completed") with an NPV of β^5. Let us now investigate the optimal objective value of the stochastic NPV maximization problem. Since n is the only activity with a nonzero cash flow and $\tilde{c}_n^F > 0$ almost surely, the early start policy also maximizes the NPV in the stochastic setting. Note that the early start policy is non-anticipative by construction. The probability that the NPV in the stochastic problem exceeds $z \in \mathbb{R}_{\geq 0}$ evaluates to

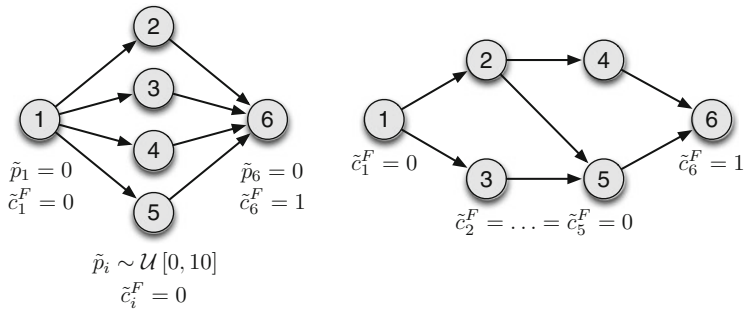


Fig. 35.2 Two example projects. In the *left project*, the durations of the first and the last activity as well as all cash flows are deterministic, whereas the other durations follow independent uniform distributions. In the *right project* all cash flows are deterministic, and the durations follow independent distributions with known density functions f_i and cumulative distribution functions F_i

$$\begin{aligned} P\left(\beta^{\max\{\tilde{p}_2,\ldots,\tilde{p}_{n-1}\}} > z\right) &= P\left(\min\left\{\beta^{\tilde{p}_2},\ldots,\beta^{\tilde{p}_{n-1}}\right\} > z\right) \\ &= \prod_{i=2}^{n-1} P\left(\beta^{\tilde{p}_i} > z\right) = \prod_{i=2}^{n-1} P\left(\tilde{p}_i < \log_\beta z\right) \\ &= \left(\frac{\log_\beta z}{10}\right)^{n-2} \end{aligned}$$

where the second identity follows from the independence of the activity durations. For fixed β, this quantity goes to zero for all $z > \beta^{10}$ when the number of activities n approaches infinity. Since $\beta^5 > \beta^{10}$, we thus conclude that the nominal problem provides a poor approximation if the number of activities n is large. For further details, see Jørgensen and Wallace (2000) and Elmaghraby (2005).

In view of the disappointing approximation quality of the nominal problem there is a need to solve the stochastic NPV maximization problem. We tackle this challenging task from two angles. First, we discuss how we can characterize the NPV if the optimal start time policy is known. Afterwards, we comment on the structural properties of the optimal start time policy.

Consider the project shown in Fig. 35.2 (right), which is an adaptation of an example presented by Demeulemeester and Herroelen (2002). We assume that the activity durations $\tilde{p}_i$ follow independent probability distributions with density functions f_i and cumulative distribution functions F_i. For ease of exposition, we assume that the cash flows satisfy $\tilde{c}_1^F = \ldots = \tilde{c}_5^F = 0$ and $\tilde{c}_6^F = 1$ almost surely. Note that the non-anticipative early start policy is again optimal since n is the only activity with a nonzero cash flow and $\tilde{c}_n^F > 0$ almost surely. We aim to characterize the cumulative distribution function of the project's NPV under the early start policy. If we denote the cumulative distribution function of activity i's earliest start time by G_i, then we obtain for the first three activities that

$$G_1(t) = \begin{cases} 1 & \text{if } t \geq 0, \\ 0 & \text{otherwise;} \end{cases} \qquad \text{and} \qquad G_2(t) = G_3(t) = P\left(\tilde{p}_1 \leq t\right) = F_1(t)$$

The distribution of the earliest start time of activity 4 is

$$G_4(t) = P\left(\tilde{p}_1 + \tilde{p}_2 \leq t\right) = \left(F_1 * f_2\right)(t)$$

where $(\chi_1 * \chi_2)(t) := \int_{\tau\in\mathbb{R}} \chi_1(\tau)\cdot\chi_2(t-\tau)\,\mathrm{d}\tau$ denotes the convolution of two functions χ_1 and χ_2. The earliest start time of activity 5 depends on the maximum of two independent random variables:

$$\begin{aligned} G_5(t) &= P\left(\max\{\tilde{p}_1 + \tilde{p}_2,\ \tilde{p}_1 + \tilde{p}_3\} \leq t\right) = P\left(\tilde{p}_1 + \max\{\tilde{p}_2, \tilde{p}_3\} \leq t\right) \\ &= (f_1 * \Gamma)(t) = \left(f_1 * [F_2 \cdot F_3]\right)(t) \end{aligned}$$

where

$$\begin{aligned}\Gamma(t) &:= P\left(\max\{\tilde{p}_2, \tilde{p}_3\} \le t\right) &&= P\left(\tilde{p}_2 \le t,\ \tilde{p}_3 \le t\right)\\ &= P\left(\tilde{p}_2 \le t\right) \cdot P\left(\tilde{p}_3 \le t\right) &&= F_2(t) \cdot F_3(t)\end{aligned}$$

Here, we use the notation $(\chi_1 \cdot \chi_2)(t) := \chi_1(t) \cdot \chi_2(t)$. Calculating G_6 is more involved as it depends on the maximum of *dependent* random variables:

$$\begin{aligned}G_6(t) &= P\left(\max\{\tilde{p}_1 + \tilde{p}_2 + \tilde{p}_4,\ \tilde{p}_1 + \tilde{p}_2 + \tilde{p}_5,\ \tilde{p}_1 + \tilde{p}_3 + \tilde{p}_5\} \le t\right)\\ &= P\left(\tilde{p}_1 + \max\{\tilde{p}_2 + \tilde{p}_4,\ \tilde{p}_2 + \tilde{p}_5,\ \tilde{p}_3 + \tilde{p}_5\} \le t\right)\\ &= (f_1 * \Gamma')(t)\end{aligned}$$

where

$$\begin{aligned}&\Gamma'(t) := P\left(\max\{\tilde{p}_2 + \tilde{p}_4,\ \tilde{p}_2 + \tilde{p}_5,\ \tilde{p}_3 + \tilde{p}_5\} \le t\right)\\ &= \int_{\delta_2,\delta_5 \ge 0} P\left(\delta_2 + \tilde{p}_4 \le t\right) \cdot P(\delta_2 + \delta_5 \le t) \cdot P\left(\tilde{p}_3 + \delta_5 \le t\right) \cdot f_2(\delta_2) \cdot f_5(\delta_5)\, \mathrm{d}\delta_2\, \mathrm{d}\delta_5\\ &= \int_{\substack{\delta_2,\delta_5 \ge 0,\\ \delta_2+\delta_5 \le t}} P\left(\tilde{p}_4 \le t - \delta_2\right) \cdot P\left(\tilde{p}_3 \le t - \delta_5\right) \cdot f_2(\delta_2) \cdot f_5(\delta_5)\, \mathrm{d}\delta_2\, \mathrm{d}\delta_5\\ &= \int_{\substack{\delta_2,\delta_5 \ge 0,\\ \delta_2+\delta_5 \le t}} F_4(t - \delta_2) \cdot F_3(t - \delta_5) \cdot f_2(\delta_2) \cdot f_5(\delta_5)\, \mathrm{d}\delta_2\, \mathrm{d}\delta_5\end{aligned}$$

Hence, the probability that the project's NPV exceeds $z \in \mathbb{R}_{\ge 0}$ is given by

$$P\left(\beta^{\tilde{p}_6} > z\right) = P\left(\tilde{p}_6 < \log_\beta z\right) = G_6(\log_\beta z)$$

Estimating a project's NPV in this way can be done algorithmically (Ghomi and Hashemin 1999; Schmidt and Grossmann 2000), but the approach becomes impractical for large networks. In fact, we cannot expect that there is an algorithm that efficiently determines the cumulative distribution function of a project's NPV. It follows from Hagstrom (1988) that even if all cash flows are deterministic and nonnegative and the activity durations are independent Bernoulli random variables, the calculation of any pre-specified quantile of a project's NPV under the early start policy is #PSPACE-hard.

So far, we have only considered NPV maximization problems where all cash flows were nonnegative. Such problems are optimized by the (non-anticipative) early start policy. However, it is well-known that the NPV is a *non-regular objective*, that is, it is not optimized by the early start schedule in general. This is illustrated by the deterministic project in Fig. 35.3 (left). If we solve the associated deterministic NPV maximization problem *NPV* for a sufficiently large discount rate β (i.e., $\beta > \sqrt[3]{1/2}$), then we obtain the optimal activity start times $S_1^\star = S_2^\star = 0$, $S_3^\star = 2$

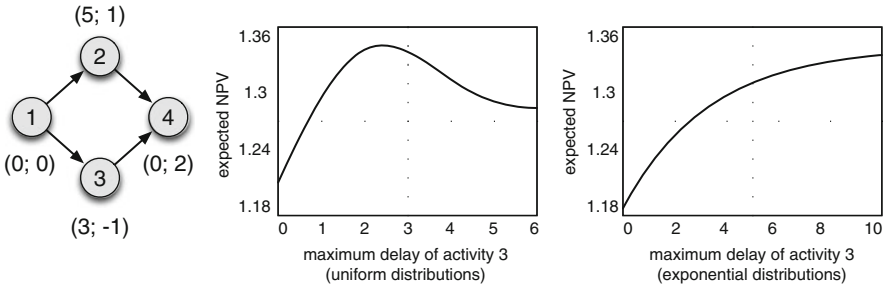


Fig. 35.3 The *left graph* shows an example project with nominal activity durations (first value) and cash flows (second value). The charts on the *right* show the optimal non-anticipative start time policies for activity 3 if $\tilde{p}_2$ and $\tilde{p}_3$ follow independent uniform (*middle*) and exponential (*right*) distributions

and $S_4^\star = 5$. Thus, the activities 1, 2, and 4 are started as early as possible, whereas activity 3 is started as late as possible (without affecting its successor's start time). More generally, it follows from the extreme point optimality of problem *NPV'* that there is always an optimal schedule for the deterministic time-constrained NPV problem in which every activity starts immediately after all of its predecessors are completed, or it is completed at the time when one of its successors is started, see Neumann and Zimmermann (2000).

Unfortunately, this property does not carry over to the stochastic problem formulation, even if we assume that all cash flows are deterministic and that the activity durations follow independent probability distributions. This is due to the fact that activities in stochastic project scheduling problems typically have no slack, that is, delaying any activity beyond its earliest start time may affect the start time of other activities. To illustrate this, assume that the durations in Fig. 35.3 (left) satisfy $\tilde{p}_1 = \tilde{p}_4 = 0$ almost surely, whereas $\tilde{p}_2$ and $\tilde{p}_3$ follow independent uniform distributions with supports $[4, 6]$ and $[2, 4]$, respectively. By construction, any optimal schedule will start the activities 1, 2, and 4 as early as possible. To find an optimal start time policy, we therefore only need to determine the optimal start time for activity 3. Clearly, we should never start activity 3 later than the completion of activity 2 (if $\beta > \sqrt[4]{1/2}$). Since no further cash flow is observed between the start and the completion of activity 2, the optimal start time of activity 3 must be determined by the minimum of a deterministic time t and the random completion time of activity 2. The dependence of the project's expected NPV on t is shown in Fig. 35.3 (middle). The optimal value is $t^\star \approx 2.42$. Note that at time $t^\star$, the expected duration of activity 3 (which is 3) exceeds the expected residual duration of activity 2 (which is 2.58). The situation changes fundamentally if we assume that $\tilde{p}_2$ and $\tilde{p}_3$ follow independent exponential distributions with expected values $E[\tilde{p}_2] = 5$ and $E[\tilde{p}_3] = 3$. In this case, Fig. 35.3 (right) shows that $t^\star = \infty$, that is, activity 3 should be started only once activity 2 is completed. This is no coincidence. In fact, under the assumption that all activity durations follow independent exponential distributions, the next section shows that there is always

an optimal start time schedule which only starts activities when other activities terminate.

In conclusion, the stochastic time-constrained NPV problem poses at least two challenges. On one hand, evaluating a risk measure of the project's NPV under any fixed policy becomes difficult if more than just a few activities are involved. On the other hand, the optimal activity start times are difficult to characterize as they lack the structural properties known from the deterministic problem. In order to facilitate a numerical solution, the existing methods make some simplifying assumptions. In the next two sections, we discuss the two most common simplifications, namely the assumption of independent and exponentially distributed activity durations (Sect. 35.3) and the restriction to suboptimal start time policies (Sect. 35.4).

35.3 Stochastic NPV Maximization: Exponential Activity Distributions

We consider again stochastic NPV maximization problems where the project graph $G = (V, E)$ and the discount rate β are known, whereas the activity durations $\tilde{p}_i$ and cash flows $\tilde{c}_i^F$ are uncertain. We assume that the activity durations $\tilde{p}_i$ follow independent exponential distributions with known rates $\lambda_i > 0$, so that the expected durations are $E[\tilde{p}_i] = 1/\lambda_i < \infty$. The cash flows $\tilde{c}_i^F$ can follow any distributions as long as they are mutually independent and also independent of the activity durations. We assume that the cash flows arise when the respective activities are initiated. As usual, node 1 represents the unique source of the project graph, and we assume that this activity is to be started at time zero. At the end of this section, we will comment on relaxations of these assumptions.

Exponential distributions are *memoryless*. This means that the time $s \in \mathbb{R}_{\geq 0}$ already spent on activity i does not provide any information about the remaining time $t \in \mathbb{R}_{\geq 0}$ still to be spent on i, or mathematically: $P(\tilde{p}_i > s + t \mid \tilde{p}_i > s) = P(\tilde{p}_i > t)$. In other words, the probability that activity i takes 5 more weeks to finish is not affected by the fact that we have already spent 10 weeks on i; it would be the same if we had just started the activity. This assumption is not very realistic in project scheduling, but it will be crucial for the algorithmic developments in this section.

Motivated by the discussion of the previous section, we describe a project's state at any time $t \geq 0$ through a partition $s_t = (\mathscr{I}_t, \mathscr{A}_t, \mathscr{C}_t)$ of the project activities $i \in V$ into the sets $\mathscr{I}_t$ of *idle activities* (activities that have not yet been started), $\mathscr{A}_t$ of *processed activities* (activities that are currently being processed) and $\mathscr{C}_t$ of *completed activities*. The project starts at time $t = 0$ in state $s_0 = (V \setminus \{1\}, \{1\}, \emptyset)$ where activity 1 is being processed and all other activities are idle. The project is finished when it reaches the state $(\emptyset, \emptyset, V)$ where all activities are completed. At any time $t \geq 0$, the project manager takes an action $a_t \subseteq \mathscr{I}_t$ that prescribes which of the idle activities $j \in \mathscr{I}_t$ are started (if any). The precedence constraints

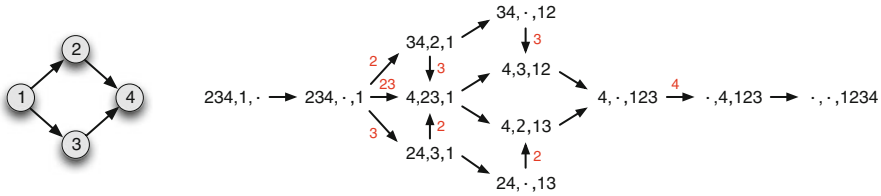


Fig. 35.4 For the project graph on the *left*, the diagram on the *right* visualizes the project states (nodes) and the admissible actions (arcs). From *left to right*, each node label lists the idle, active, and completed activities (with "·" encoding the empty set). The arc labels indicate which activities are started by the associated action (absent labels imply that no activities are started). We omit the unlabeled self loops that are associated with each node

stipulate that only those activities $j \in \mathscr{I}_t$ can be started that satisfy $\{i \in V : (i, j) \in E\} \subseteq \mathscr{C}_t$, that is, an activity can only be started if all of its immediate predecessors are completed. At time t, the project manager takes the decision a_t under the knowledge of the entire history (s_τ, a_τ), $\tau \in [0, t]$, of the past states and actions. The objective is to find a policy, that is, a non-anticipative mapping from state-action histories to actions that maximizes the project's expected NPV. The project evolution is illustrated in Fig. 35.4.

Under the assumption of independent and exponentially distributed activity durations, our stochastic NPV maximization problem has an equivalent representation as a continuous-time Markov decision process. This was first discovered by Kulkarni and Adlakha (1986) for stochastic makespan minimization problems. The result was later extended by Buss and Rosenblatt (1997) to analyze stochastic NPV problems where up to two activities can be delayed. To our knowledge, Sobel et al. (2009) were the first to propose a generic reformulation of the stochastic NPV maximization problem as a continuous-time Markov decision process. This reformulation allows us to apply a classical result for uniformizable continuous-time Markov decision processes with the following implication:

> Whenever the project enters a new state $s_t = (\mathscr{I}_t, \mathscr{A}_t, \mathscr{C}_t)$ at some time $t \geq 0$, it is optimal to either immediately start some idle activities $i \in \mathscr{I}_t$ or to wait until at least one of the active activities $i \in \mathscr{A}_t$ is completed.

We have already observed this phenomenon in an example in the previous section.

The fact that we only need to start activities when the project state changes allows us to drastically simplify the problem. In particular, we can subdivide the continuous-time project evolution into discrete *decision epochs* $t \in \mathbb{Z}_{\geq 0}$ within each of which the project state remains unchanged. By slight abuse of notation, we denote the state in epoch t by $s_t = (\mathscr{I}_t, \mathscr{A}_t, \mathscr{C}_t)$. The project starts in the epoch 0 in state $s_0 = (V \setminus \{1\}, \{1\}, \emptyset)$ where all but the first activity are idle. In any epoch $t \in \mathbb{Z}_{\geq 0}$ and state $s_t = (\mathscr{I}_t, \mathscr{A}_t, \mathscr{C}_t)$, the project manager selects a subset a_t of the idle activities $\mathscr{I}_t$ that are to be started immediately ($a_t = \emptyset$ is allowed).

The project then transitions into an intermediate epoch $s'_t = (\mathscr{I}'_t, \mathscr{A}'_t, \mathscr{C}'_t)$ where $\mathscr{I}'_t = \mathscr{I}_t \setminus a_t$, $\mathscr{A}'_t = \mathscr{A}_t \cup a_t$ and $\mathscr{C}'_t = \mathscr{C}_t$. Afterwards, the project manager waits until one of the active activities $i \in \mathscr{A}'_t$ is completed. If activity $i \in \mathscr{A}'_t$ is the first activity to be completed, then the project transitions to epoch $t+1$ and state $s_{t+1} = (\mathscr{I}'_t, \mathscr{A}'_t \setminus \{i\}, \mathscr{C}'_t \cup \{i\})$.

In order to truthfully reproduce the dynamics of the continuous-time project evolution, we need to specify the random time spent in each intermediate epoch corresponding to state s'_t and the random successor state s_{t+1}. Let us first determine the distribution of the time $\min_{j \in \mathscr{A}'_t} \tilde{p}_j$ spent in the intermediate epoch $s'_t = (\mathscr{I}'_t, \mathscr{A}'_t, \mathscr{C}'_t)$:

$$\begin{aligned} P\left(\min_{j \in \mathscr{A}'_t} \tilde{p}_j < \delta\right) &= 1 - P\left(\tilde{p}_j \geq \delta \;\; (j \in \mathscr{A}'_t)\right) \\ &= 1 - \prod_{j \in \mathscr{A}'_t} \mathrm{e}^{-\lambda_j \cdot \delta} \;=\; 1 - \mathrm{e}^{-\delta \cdot \sum_{j \in \mathscr{A}'_t} \lambda_j} \end{aligned} \tag{35.1}$$

Here, the second identity is due the fact that the activity durations follow independent exponential distributions. From the expression on the right-hand side we see that the time spent in the intermediate state s'_t is described by an exponentially distributed random variable with rate $\sum_{j \in \mathscr{A}'_t} \lambda_j$.

Activity $i \in \mathscr{A}'_t$ is completed first among the activities in $\mathscr{A}'_t$ with probability

$$\begin{aligned} P\left(\tilde{p}_i < \tilde{p}_j \;\; (j \in \mathscr{A}'_t \setminus \{i\})\right) &= \int_{\delta \in \mathbb{R}_{\geq 0}} P\left(\delta \leq \tilde{p}_i \leq \delta + \mathrm{d}\delta \wedge \tilde{p}_j > \delta \;\; (j \in \mathscr{A}'_t \setminus \{i\})\right) \\ &= \int_{\delta \in \mathbb{R}_{\geq 0}} \lambda_i \cdot \mathrm{e}^{-\lambda_i \cdot \delta} \prod_{j \in \mathscr{A}'_t \setminus \{i\}} \mathrm{e}^{-\lambda_j \cdot \delta} \, \mathrm{d}\delta \;=\; \frac{\lambda_i}{\sum_{j \in \mathscr{A}'_t} \lambda_j} \end{aligned} \tag{35.2}$$

Here, the second identity holds because the activity durations follow independent exponential distributions, while the last identity follows from elementary algebraic manipulations. One can show that the time spent in s'_t and the activity that finishes first in s'_t are independent random variables (Bertsekas 2007; Puterman 1994).

The subdivision of the project evolution into decision epochs allows us to reformulate the stochastic NPV maximization problem as a discrete-time Markov decision process with state- and action-dependent discount rates. To this end, we define a *discrete-time Markov decision process* through

- a state space $\mathbb{S}$ with designated start state $s_0 \in \mathbb{S}$,
- an action space $\mathbb{A}_s$ associated with each state $s \in \mathbb{S}$, with $\mathbb{A} = \bigcup_s \mathbb{A}_s$,
- transition probabilities $\pi : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \mapsto [0, 1]$ so that $\sum_{s'} \pi(s'|s, a) = 1 \;\; (s, a)$,
- a reward function $r : \mathbb{S} \times \mathbb{A} \mapsto \mathbb{R}$, and
- state- and action-dependent discount factors $\gamma : \mathbb{S} \times \mathbb{A} \mapsto [0, 1)$.

A discrete-time Markov decision process starts in state s_0. At any time $t \in \mathbb{Z}_{\geq 0}$, the process occupies a state $s_t \in \mathbb{S}$, and we select a decision $a \in \mathbb{A}_s$ under the knowledge of the entire state-action history (s_τ, a_τ), $\tau \in \{0, \ldots, t\}$. We then obtain an immediate reward of $r(s, a)$, and the Markov decision process stochastically transitions to a successor state $s_{t+1} \in \mathbb{S}$ according to the transition probabilities $\pi(s_{t+1}|s_t, a_t)$. Note that the transition probabilities do not depend on the history of past states and actions $s_0, a_0, \ldots, s_{t-1}, a_{t-1}$; this is the *Markov property*. The goal is to determine a non-anticipative policy, that is, a mapping from state-action histories to actions, that maximizes the expected total discounted reward. The discount factor between periods t and $t+1$ is given by $\gamma(s_t, a_t)$, which depends on both the state s_t in period t and the action a_t taken in period t.

We define the discrete-time Markov decision process associated with our NPV maximization problem as follows. We identify the state space $\mathbb{S}$ with the set of all partitions $(\mathscr{I}, \mathscr{A}, \mathscr{C})$ of V with the property that all immediate predecessors of active or completed activities are completed, that is, $\{i \in V : (i, j) \in E\} \subseteq \mathscr{C}$ for all $j \in \mathscr{A} \cup \mathscr{C}$. Note that this state space typically grows exponentially with the number n of project activities. For each state $s = (\mathscr{I}, \mathscr{A}, \mathscr{C}) \in \mathbb{S}$, the set of dormant activities $\{j \in \mathscr{I} : \{i \in V : (i, j) \in E\} \subseteq \mathscr{C}\}$ contains all idle activities that have only completed predecessors. Moreover, we identify the action space $\mathbb{A}_s$ with the set of all subsets of the dormant activities, including the empty set.

For the state $s = (\mathscr{I}, \mathscr{A}, \mathscr{C}) \in \mathbb{S}$ and the action $a \in \mathbb{A}_s$, the probability to transition to the state $s' = (\mathscr{I}', \mathscr{A}', \mathscr{C}') \in \mathbb{S}$ is given by

$$\pi(s'|s, a) = \begin{cases} \dfrac{\lambda_i}{\sum\limits_{j \in \mathscr{A} \cup a} \lambda_j} & \text{if } \mathscr{I}' = \mathscr{I} \setminus a,\ \mathscr{A}' = \mathscr{A} \cup a \setminus \{i\} \text{ and } \mathscr{C}' = \mathscr{C} \cup \{i\}, \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, $\pi(s'|s, a)$ is nonzero if and only if the state s' emerges from state s by initiating the activities in a and afterwards waiting until the first activity $i \in \mathscr{A} \cup \{a\}$ terminates. We have derived the probability of this event in (35.2).

For any state $s \in \mathbb{S}$ and action $a \in \mathbb{A}_s$, we define the reward $r(s, a) = \sum_{j \in a} E[\tilde{c}_j^F]$ as the sum of expected cash flows of those activities that we start in state s. Thus, the expected value is all we need to know about the uncertain cash flows.

Consider next the discount factor $\gamma(s, a)$ associated with state $s = (\mathscr{I}, \mathscr{A}, \mathscr{C}) \in \mathbb{S}$ and action $a \in \mathbb{A}_s$. We discount the cash flows arising after state s by $E[\beta^{\tilde{\delta}}]$, where $\tilde{\delta}$ denotes the random time spent in the intermediate state $s' = (\mathscr{I} \setminus a, \mathscr{A} \cup a, \mathscr{C})$:

$$\gamma(s, a) \;=\; E\left[\beta^{\tilde{\delta}}\right] \;=\; \int_{\delta \in \mathbb{R}_{\geq 0}} \beta^{\delta} \cdot \lambda \cdot \mathrm{e}^{-\delta\lambda}\, \mathrm{d}\delta \;=\; \frac{\lambda}{\lambda - \log \beta}$$

where $\lambda = \sum_{j \in \mathscr{A} \cup a} \lambda_j$. Here, the second identity follows from (35.1).

Sobel et al. (2009) show that the state- and action-dependent discount rate can be absorbed in the transition probabilities. The resulting problem is a discrete-time Markov decision process with a constant discount rate for which a variety of solution methods exist (Bertsekas 2007; Puterman 1994). Note that every state of the project is visited at most once, that is, active activities never become idle again, and completed activities remain completed. This allows us to determine the optimal policy with a standard backward recursion. Using such a backward recursion, Sobel et al. (2009) solve networks with up to 25 activities within a few minutes.

The major computational bottleneck of the backward recursion is the need to simultaneously store information about all states $s \in \mathbb{S}$ in memory. Creemers et al. (2010b) propose a memory-efficient variant of the backward recursion. The algorithm is based on a partition of the state space induced by the maximal antichains of the project graph. A maximal antichain is defined as a maximal set of project activities $i \in V$ that can be processed simultaneously. The authors show that only a fraction of the maximal antichains and their associated states need to be stored in memory at any point in time. Numerical experiments show that their algorithm can solve projects with up to 120 activities within a few hours.

We close this section with extensions that have been proposed in the literature.

Sobel et al. (2009) relax the assumption of exponentially distributed activity durations and consider activity durations that follow independent phase-type distributions. To this end, they model each activity $i \in V$ as a subproject that consists of a network of exponentially distributed subactivities. Since phase-type distributions are dense in the set of all positive-valued distributions, we can approximate any activity duration arbitrarily well at the expense of augmenting the state space $\mathbb{S}$. Sobel et al. (2009) report on numerical results where they solve projects with up to ten activities, each of which consists of up to five subactivities. Note that even with this extension, the activity durations must still be stochastically independent. Also, activities can only be started whenever an entire activity $i \in V$ is completed because the newly introduced subactivities have no counterpart in the real project. This implies, however, that the determined policies are no longer guaranteed to be optimal. Benmansour et al. (2010) discuss how probability distributions can be approximated by phase-type distributions in an economical way.

Sobel et al. (2009) also show how to incorporate the option to abandon the project during its execution. This is achieved by amending the action spaces $\mathbb{A}_s$, $s \in \mathbb{S}$, by an "abandonment decision" that deterministically leads to the state $(\emptyset, \emptyset, V)$.

Creemers et al. (2009, 2010a) study project scheduling problems in which any of the activities can fail, and failure of an activity results in the abandonment of the entire project. Such failures can be accounted for by introducing stochastic transitions from each state-action pair (s, a), $s \in \mathbb{S}$ and $a \in \mathbb{A}_s$, to an auxiliary "project failure" state. The authors also consider projects where intermediate project milestones can be achieved by pursuing either of several alternative strategies. Both extensions are motivated in the context of research-and-development projects.

The models in this section can readily accommodate multiple renewable resources (such as labor and machinery) if we restrict the action spaces $\mathbb{A}_s$.

In particular, in state $s_t = (\mathscr{I}_t, \mathscr{A}_t, \mathscr{C}_t)$ we restrict $\mathbb{A}_{s_t}$ to those actions $a_t \subseteq \mathscr{A}_t$ that satisfy all precedences *and* that lead to intermediate states $s'_t = (\mathscr{I}_t \setminus a_t, \mathscr{A}_t \cup a_t, \mathscr{C}_t)$ where the active activities $\mathscr{A}_t \cup a_t$ satisfy the resource constraints. If the resource allocation also affects the activity durations, then we face a multi-mode problem. When there are finitely many alternative execution modes for each activity, all of which result in exponentially distributed activity durations, then we can solve the multi-mode problem by augmenting the state space. To this end, we need to record for each project state $s_t = (\mathscr{I}_t, \mathscr{A}_t, \mathscr{C}_t)$ the execution modes of the active activities $i \in \mathscr{A}_t$. Such discrete multi-mode problems have been studied in the context of weighted cost/tardiness project scheduling problems, see Tereso et al. (2004, 2008, 2010). These problems are very difficult to solve, and projects with 18 activities already require several days of computing time (Tereso et al. 2003). If the resource allocation can vary continuously, these reformulations are no longer possible. Rudolph and Elmaghraby (2009) and Nadjafi and Kolyaei (2010) solve stochastic project scheduling problems with continuous resource allocations through heuristics that alternate between computing the optimal project schedule for a fixed resource allocation and modifying the resource allocation. Azaron et al. (2006) and Azaron and Tavakkoli-Moghaddam (2006, 2007) solve multi-objective variants of the stochastic project scheduling problem with continuous resource allocations through time discretization. The resulting multi-objective nonlinear optimization problems are solved using the surrogate worth trade-off, the STEM, or the goal attainment method. We can also incorporate nonrenewable resources (such as capital) if we record the remaining resource budget in each state. Again, this comes at the expense of a significantly increased state space. For further information, we refer to Chap. 38 in this book.

We finally mention the paper by Gutin et al. (2013) that studies two-player stochastic interdiction games. In these games, a proliferator aims to minimize the expected duration of a nuclear weapons development project, whereas an interdictor endeavors to maximize the project duration by delaying some of the project activities. Using similar techniques as in this section, the authors reformulate the interdiction game as a discrete-time Markov decision process. The assumption of independent and exponentially distributed activity durations is again crucial for their reformulation.

35.4 Stochastic NPV Maximization: Generic Activity Distributions

We now relax the assumption that the activity durations follow independent exponential distributions. Instead, we model the activity durations and cash flows as random variables that are described by a finite set of scenarios $\{(p_i^\sigma, c_i^{F\sigma}) : \sigma \in \Sigma\}$ with associated occurrence probabilities π_σ, $\sigma \in \Sigma$. The assumption of a discrete distribution may seem restrictive since the activity durations are often described via

independent Beta distributions (Malcolm et al. 1959). In those cases, we can replace the continuous probability distribution with a discrete approximation using scenario reduction techniques (Heitsch and Römisch 2003; Henrion et al. 2009). Note that we allow the activity durations and cash flows to exhibit stochastic dependencies even across different activities. This is meaningful as longer activity durations often imply higher expenditures, which in turn result in lower cash flows. To simplify the exposition, we again assume that nodes 1 and n are the unique source and sink of the network, respectively, the discount rate β is deterministic, all precedences are of finish-start type, and we aim to maximize the project's expected NPV. At the end of this section, we comment on generalizations.

We consider the following generic stochastic NPV maximization problem.

$$\begin{array}{ll} \text{Max.} & \displaystyle\sum_{\sigma\in\Sigma} \pi_\sigma \cdot \sum_{i\in V} c_i^{F\sigma} \cdot \beta^{S_i^\sigma} \\ \text{s. t.} & S \in \mathbb{R}^{|\Sigma|n}; \quad y \in \Pi \\ & \left.\begin{array}{ll} S_j^\sigma \geq S_i^\sigma + p_i^\sigma & ((i,j)\in E) \\ S_1^\sigma = 0 & \\ S^\sigma = f^\sigma(y) & \end{array}\right\} \quad (\sigma \in \Sigma) \end{array}$$

In this problem, which we henceforth refer to as *SNPV*, the decision variable S_i^σ denotes the factual start time of activity $i \in V$ in scenario $\sigma \in \Sigma$. The decision vector y encodes a start time policy that is selected from the set of policy parameters Π. In analogy to the deterministic NPV maximization problem *NPV*, the first two constraints impose nonnegativity of the activity start times and satisfaction of the precedence constraints in all scenarios $\sigma \in \Sigma$. The last constraint requires the activity start times S^σ in scenario $\sigma \in \Sigma$ to be generated by the non-anticipative start time policy y. For any fixed policy encoded by $y \in \Pi$, the functions $f^\sigma(y)$ characterize the unique activity start times for scenario σ that result from implementing policy y.

As we have discussed in Sect. 35.2, the space of non-anticipative start time policies is huge. It contains all functions that for any time point $t \geq 0$ and any possible constellation of completed, active, and idle activities prescribe which activities (if any) are to be started. Moreover, in absence of independent and exponentially distributed activity durations, it is no longer optimal to start activities only when other activities finish. In the following, we study subsets of the space of all non-anticipative policies that allow us to formulate and solve optimization problems. Thus, we are interested in suboptimal but implementable (that is, non-anticipative) start time policies.

A particularly simple class of start time policies is given by the *rigid start time policies* (Wiesemann et al. 2010). In this case, we set $\Pi = \mathbb{R}^n$ and

$$S^\sigma = f^\sigma(y) \quad \Leftrightarrow \quad S_j^\sigma = y_j \quad (j \in V)$$

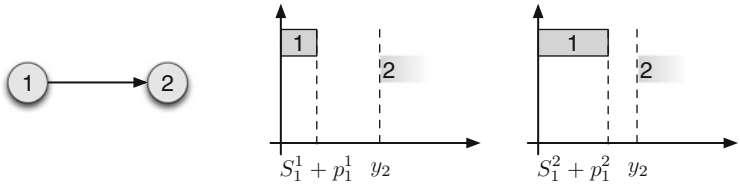


Fig. 35.5 For the project graph on the *left*, the Gantt charts on the *right* illustrate a rigid start time policy in two scenarios. For y to be a feasible policy, it needs to satisfy $y_2 \geq S_1^\sigma + p_1^\sigma$ for all $\sigma \in \Sigma$

Rigid start time policies start all activities $j \in V$ at scenario-independent times $y_j \in \mathbb{R}$, see Fig. 35.5. Since the start times do not depend on the information revealed over time, they trivially satisfy the non-anticipativity requirement. Under the variable transformations $S_i^\sigma := \beta^{S_i^\sigma}$ and $y_i := \beta^{y_i}$, the stochastic NPV maximization problem *SNPV* with rigid start time policies can be reformulated as a linear program:

$$\begin{array}{lll} \text{Max.} & \displaystyle\sum_{\sigma \in \Sigma} \pi_\sigma \cdot \sum_{i \in V} c_i^{F\sigma} \cdot S_i^\sigma & \\ \text{s. t.} & S \in \mathbb{R}_{\geq 0}^{|\Sigma| n}; \quad y \in \mathbb{R}_{\geq 0}^n & \\ & \left.\begin{array}{ll} S_j^\sigma \leq \beta^{p_i^\sigma} \cdot S_i^\sigma & ((i,j) \in E) \\ S_1^\sigma = 1 & \\ S_i^\sigma = y_i & (i \in V) \end{array}\right\} & (\sigma \in \Sigma) \end{array}$$

This problem can either be solved with standard linear programming software or using decomposition methods from stochastic programming (Kall and Wallace 1994; Ruszczyński and Shapiro 2003). Note that the activity start times S_j^σ have to satisfy the precedence constraints in all scenarios $\sigma \in \Sigma$. Since the activity durations vary across the scenarios, we can expect scenario-independent activity start times to provide overly conservative solutions to the stochastic NPV maximization problem *SNPV*. This is confirmed by numerical tests, see Wiesemann et al. (2010).

A richer class of start time policies is given by the *target processing time policies*. In this case, we set $\Pi = \mathbb{R}^n$ and

$$x = f^\sigma(y) \quad \Leftrightarrow \quad S_j^\sigma = \max \left\{ \sup_{i \in V} \{S_i^\sigma + p_i^\sigma \,:\, (i,j) \in E\}, \; y_j \right\} \qquad (j \in V)$$

Here, the inner supremum over the predecessor activities of $j \in V$ evaluates either to minus infinity (if j has no predecessor activities, that is, if $j = 1$) or to the earliest start time of activity j in scenario σ. The outer maximum requires activity j to start either at its earliest start time or at its (scenario-independent)

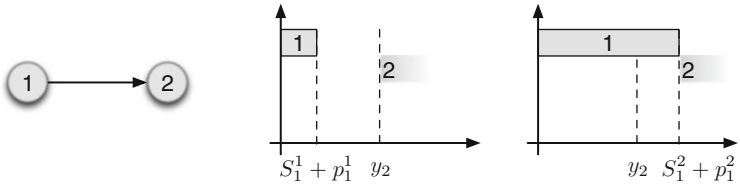


Fig. 35.6 For the project graph on the *left*, the Gantt charts on the *right* illustrate a target processing time policy in two scenarios. Activity 2 is started either at its target processing time (first chart) or at its earliest possible start time (second chart)

target processing time y_j, depending on which value is larger. Thus, under the target processing time policy y, the activity $j \in V$ is started as early as possible but never before its target processing time y_j, see Fig. 35.6. By construction, target processing time policies are non-anticipative, that is, they do not require the project manager to "look into the future". We can also see that any feasible rigid start time policy y constitutes an admissible target processing time policy that results in the same activity start times but not vice versa. Thus, target processing time policies are more general than rigid start time policies. Target processing time policies can be traced back to the stochastic machine scheduling literature where they are used to determine the latest job release times that achieve prespecified due dates with a high probability (Elmaghraby et al. 2000; Elmaghraby 2001). More recently, they were applied to stochastic project scheduling problems to determine stable project plans that minimize earliness and tardiness penalties, see Trietsch (2005, 2006) and Bendavid and Golany (2011a,b).

Under the variable transformations $S_i^\sigma := \beta^{S_i^\sigma}$ and $y_i := \beta^{y_i}$, the stochastic NPV maximization problem *SNPV* with target processing time policies can be reformulated as the following *nonlinear* program:

$$
\begin{array}{lll}
\text{Max.} & \displaystyle\sum_{\sigma\in\Sigma} \pi_\sigma \cdot \sum_{i\in V} c_i^{F\sigma} \cdot S_i^\sigma & \\
\text{s. t.} & S \in \mathbb{R}_{\geq 0}^{|\Sigma|n}; \;\; y \in \mathbb{R}_{\geq 0}^{n} & \\
& \left.\begin{array}{ll}
S_j^\sigma \leq \beta^{p_i^\sigma} \cdot S_i^\sigma & ((i,j) \in E) \\
S_1^\sigma = 1 & \\
S_j^\sigma = \min\left\{ \inf\limits_{i\in V} \left\{ \beta^{p_i^\sigma} \cdot S_i^\sigma \;:\; (i,j) \in E \right\},\; y_j \right\} & (j \in V)
\end{array}\right\} \; (\sigma \in \Sigma)
\end{array}
\tag{35.3}
$$

Here, the inequality constraint is redundant because it is implied by the last equality constraint. Since the last equality constraint in (35.3) has a nonlinear right-hand side, the problem is nonconvex. Moreover, we cannot relax the equality to a less than or equal constraint without potentially violating non-anticipativity. However, we can eliminate the nonlinearity at the cost of introducing auxiliary binary variables:

$$
\begin{array}{lll}
\text{Max.} & \displaystyle\sum_{\sigma\in\Sigma} \pi_\sigma \cdot \sum_{i\in V} c_i^{F\sigma} \cdot S_i^\sigma & \\
\text{s. t.} & S \in \mathbb{R}_{\geq 0}^{|\Sigma|n}; \quad y \in \mathbb{R}_{\geq 0}^{n} & \\
& z_{ij}^\sigma \in \{0,1\} \; ((i,j) \in E; \sigma \in \Sigma) & \\
& z_i^\sigma \in \{0,1\} \; (i \in V; \sigma \in \Sigma) & \\
& \left.\begin{array}{ll}
S_1^\sigma = 1 & \\
S_j^\sigma \leq \beta^{p_i^\sigma} \cdot S_i^\sigma & ((i,j) \in E) \\
S_j^\sigma \leq y_j & (j \in V) \\
S_j^\sigma \geq \beta^{p_i^\sigma} \cdot S_i^\sigma - (1 - z_{ij}^\sigma) & ((i,j) \in E) \\
S_j^\sigma \geq y_j - (1 - z_j^\sigma) & (j \in V) \\
z_j^\sigma + \displaystyle\sum_{(i,j)\in E} z_{ij}^\sigma = 1 & (j \in V)
\end{array}\right\} \; (\sigma \in \Sigma)
\end{array}
\tag{35.4}
$$

Note that the factual activity start times satisfy $S_j^\sigma \in [0,1]$. Thus, the third and fourth inequality constraint ensure that $S_j^\sigma = \beta^{p_i^\sigma} \cdot S_i^\sigma$ if $z_{ij}^\sigma = 1$ and $S_j^\sigma = y_j$ if $z_j^\sigma = 1$, whereas they are redundant whenever $z_{ij}^\sigma = 0$ or $z_j^\sigma = 0$, respectively. The last constraint ensures that all but one of these constraints are redundant for each j and σ. This implies that there is always an optimal solution in which the factual activity start times S_j^σ satisfy the nonlinear equality constraint from model (35.3).

Problem (35.4) constitutes a mixed-integer linear program that can be solved with off-the-shelf optimization software. Wiesemann et al. (2010) report that problem instances with up to 20 activities and 10 scenarios can be solved reliably with CPLEX 11.2 within a time limit of 10 min.[1] For larger problem instances, they propose a problem-specific branch-and-bound scheme. The basic idea of the algorithm is as follows. The root node of the branch-and-bound tree considers a relaxation of problem (35.3) that replaces the nonlinear equality constraints with less than or equal constraints. This relaxation has an equivalent reformulation as a deterministic NPV maximization problem and can thus be solved very efficiently. The algorithm then branches on violations of the relaxed equality constraints. If a relaxation of problem (35.3) violates the nonlinear equality constraint associated with activity j and scenario σ, for example, the procedure adds $|\{i \in V : (i,j) \in E\}| + 1$ child nodes to the branch-and-bound tree, each of which contains one additional constraint of the form $S_j^\sigma = \beta^{p_i^\sigma} \cdot S_i^\sigma$, $(i,j) \in E$, or $S_j^\sigma = y_j$. The relaxations associated with the child nodes again constitute deterministic NPV maximization problems, and they can be solved very efficiently using warm-starting techniques. The branch-and-bound algorithm can solve instances with up to 50 activities within a time limit of 10 min.

In the related literature, Tavares et al. (1998) suggest a generic solution method for stochastic project scheduling problems that is based on *floating factor policies*. To this end, they define the total float of an activity $i \in V$ as the difference between

[1] CPLEX homepage: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer.

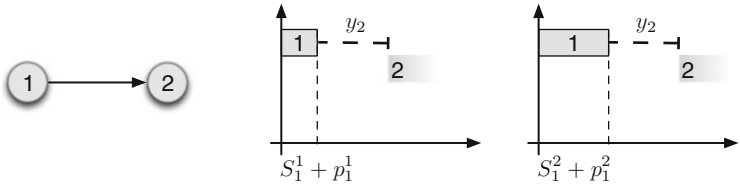


Fig. 35.7 For the project graph on the *left*, the Gantt charts on the *right* illustrate an activity delay policy in two scenarios. Activity 2 is always started y_2 time units after its earliest possible start time

its latest start time LS_i and earliest start time ES_i, given some project deadline and average activity durations. A floating factory policy $\theta \in [0, 1]$ then assigns a target processing time of $ES_i + \theta \cdot (LS_i - ES_i)$ to each activity $i \in V$. The authors suggest to determine optimal floating factor policies through simulation.

A different class of start time policies is given by the *activity delay policies*. Here, we set $\Pi = \mathbb{R}^n_{\geq 0}$ and

$$S = f^\sigma(y) \quad \Leftrightarrow \quad S_j^\sigma = \begin{cases} \max\limits_{i \in V} \{S_i^\sigma + p_i^\sigma \ : \ (i, j) \in E\} + y_j & \text{if } j \neq 1, \\ y_1 & \text{otherwise} \end{cases} \qquad (j \in V)$$

Under the activity delay policy y, activity $j \in V$ is started y_j time units after its earliest possible start time in each scenario $\sigma \in \Sigma$, see Fig. 35.7. One can show that the classes of target processing time policies and activity delay policies are incomparable, that is, neither one is a subset of the other. Note that the classes of rigid start time policies and activity delay policies are incomparable as well.

Under the variable transformations $S_i^\sigma := \beta^{S_i^\sigma}$ and $y_i := \beta^{y_i}$, we can reformulate the stochastic NPV maximization problem with activity delay policies as a nonlinear program. However, the resulting formulation contains bilinear terms of the form $\beta^{p_i^\sigma} \cdot S_i^\sigma \cdot y_j$ which can no longer be linearized with binary variables. Nevertheless, one can adapt the branch-and-bound scheme from Wiesemann et al. (2010) to this policy class. To our knowledge, this has not been attempted so far.

Activity delay policies date back to the eighties of the previous century. Beginning with the early start schedule, Shtub (1986) proposes a heuristic that iteratively delays activities with negative cash flows until a probabilistic deadline constraint is violated. The activities are considered in order of ascending cash flows, that is, the activity with the largest negative cash flow is delayed first. During its execution, the algorithm generates a frontier of approximately Pareto-efficient solutions that trade off the net present value against the probability to complete the project on time. Satisfaction of the deadline constraint is verified using Monte Carlo sampling.

Buss (1995) determines activity delay policies using a stochastic gradient descend method. He concludes that the problem is very challenging since the

objective function is highly variable but almost flat near the suspected optimum. The author reports solutions for projects with up to 21 activities.

Benati (2006) proposes a two-stage heuristic for the stochastic NPV maximization problem with activity delay policies. In the first stage, optimal "average" activity start times are determined through a variant of the deterministic NPV maximization problem *NPV* that uses average activity durations and cash flows:

$$\begin{aligned}
\text{Max.} \quad & \sum_{\sigma \in \Sigma} \pi_\sigma \cdot \sum_{i \in V} c_i^{F\sigma} \cdot \beta^{S_i} \\
\text{s. t.} \quad & S \in \mathbb{R}^n \\
& S_j \geq \sum_{\sigma \in \Sigma} \pi_\sigma \cdot \max\{S_i + p_i^\sigma \,:\, (i,j) \in E\} \qquad (j \in V \setminus \{1\}) \\
& S_1 = 0
\end{aligned}$$

In the second stage, an activity delay policy y is determined by setting $y_1 = S_1^\star$ and

$$y_j = S_j^\star - \sum_{\sigma \in \Sigma} \pi_\sigma \cdot \max_{i \in V} \{S_i^\star + p_i^\sigma \,:\, (i,j) \in E\} \qquad (j \in V \setminus \{1\})$$

where $S^\star$ denotes the optimal activity start times from the first stage. Intuitively, the activity delay policy thus found aims to start the activities close to their "average" optimal starting times. In numerical experiments, the activity delay policies determined by this two-stage procedure are outperformed by the target processing time policies presented earlier, see Wiesemann et al. (2010). It is unclear whether this is caused by the suboptimality of the two-stage procedure or whether target processing time policies generically perform better than activity delay policies. Another attempt to determine optimal activity delay policies is reported by Wang et al. (2000), who develop a simulated annealing heuristic for the problem.

We close with some remarks about possible generalizations of the models considered in this section. It is straightforward to extend the models to account for uncertainty in the interest rate r underlying the discount rate β. More interestingly, the models allow the network structure G to be uncertain as well. Uncertain precedence constraints, for example, can be accommodated by choosing a different arc set $A^s \subset V \times V$ for each scenario $\sigma \in \Sigma$. As shown by Wiesemann et al. (2010), the models can be extended to generalized precedence constraints such as project deadlines and maximum time lags (Elmaghraby and Kamburowski 1990, 1992). Moreover, it is possible to formulate models that combine target processing time policies with activity delay policies. For example, one could envision policies that schedule each activity either according to a target processing time or according to an activity delay, or possibly according to the earlier or later time specified by them. To our knowledge, this has not been considered so far. Finally, the models considered here can be extended to cater for the decision maker's risk aversion. For example, a variant of the stochastic NPV maximization problem *SNPV* that

maximizes the ε-CVaR of the project's NPV (see Sect. 35.2) can be formulated as follows.

$$
\begin{array}{ll}
\text{Max.} & \alpha - \dfrac{1}{\varepsilon} \cdot \displaystyle\sum_{\sigma \in \Sigma} \pi_\sigma \cdot \gamma^\sigma \\
\text{s. t.} & S \in \mathbb{R}^{|\Sigma| n}; \;\; y \in \Pi; \;\; \alpha \in \mathbb{R}; \;\; \gamma \in \mathbb{R}^{|\Sigma|}_{\geq 0} \\
& \left.\begin{array}{ll}
S_j^\sigma \geq S_i^\sigma + p_i^\sigma & ((i,j) \in E) \\
S_1^\sigma = 0 & \\
S^\sigma \in f^\sigma(y) & \\
\gamma^\sigma \geq \alpha - \displaystyle\sum_{i \in V} c_i^{F\sigma} \cdot \beta^{S_i^\sigma} &
\end{array}\right\} \; (\sigma \in \Sigma)
\end{array}
$$

In this model, α and γ are auxiliary decision variables that are required to evaluate the CVaR, see Pflug (2000) and Rockafellar and Uryasev (2000). One readily verifies that the discussed methods for rigid start time policies, target processing time policies, and activity delay policies extend to this formulation. To our knowledge, this CVaR formulation has not yet been explored in the literature.

Ke and Liu (2005) study resource-constrained variants of the stochastic NPV maximization problem that optimize the expected value and the value-at-risk of a project's NPV. The authors develop a genetic algorithm that aims to determine an optimal target processing time policy. The algorithm is extended to fuzzy activity durations in Ke and Liu (2007, 2010). Chen and Zhang (2012) consider a multi-mode resource-constrained variant of the stochastic NPV maximization problem and solve it via ant colony optimization and Monte Carlo sampling. The authors determine a static selection of execution modes for each activity such that the early-start policy results in a high expected NPV. Özdamar (1998) and Özdamar and Dündar (1997) study a resource-constrained multi-mode variant of the stochastic NPV maximization problem with capital as a single nonrenewable resource. Capital is randomly replenished through project revenues and can be temporally acquired at given costs. The authors propose an online scheduling heuristic that aims to maximize the project's NPV while satisfying a specified deadline as a soft constraint.

Uçal and Kuchta (2011) aim to maximize a project's fuzzy NPV under resource constraints. It is assumed that the activity durations are known, whereas the cash flows are described by fuzzy numbers. The authors propose a heuristic that schedules activities according to the sum of their NPV and the NPVs of their immediate successor activities.

Shavandi et al. (2012) maximize a project's fuzzy NPV in the reverse situation where the cash flows are known but the activity durations are fuzzy. The authors reformulate the problem of finding the optimal fuzzy activity start times as a multi-objective nonlinear program, which they solve via Taylor approximations.

The start time policy y in the stochastic NPV maximization problem *SNPV* bears some similarity to the baseline schedules considered in robust project scheduling. A *baseline policy* is defined as a set of deterministic activity start times that are

selected in the planning stage and that should be followed as closely as possible during project implementation. Robust project scheduling assumes that the activity durations are uncertain, and it determines baseline policies that minimize the expected sum of weighted differences between the (uncertainty-affected) factual activity start times and the start times in the baseline schedule. For reviews of the literature on robust project scheduling, we refer the interested reader to Herroelen and Leus (2004, 2005) and Chaps. 39 and 40 in this book.

We remark that in practice, the models presented in this chapter would be solved again whenever new information becomes available. For a detailed discussion of this point, we refer to Jørgensen and Wallace (2000) and Bidot et al. (2009).

35.5 Conclusions

Stochastic project scheduling has a long and distinguished history that dates back to the early days of operations research (Fazar 1959; Malcolm et al. 1959). Nonetheless, the literature on stochastic NPV maximization is still in its infancy. In the light of recent theoretical and practical advances in project management under uncertainty (Goldratt 1997; Herroelen and Leus 2005), we expect that stochastic NPV maximization will receive renewed interest in the future. To this end, we identify several fruitful directions for further research.

First and foremost, the current models make very restrictive assumptions about the probability distributions governing the uncertain problem parameters. We therefore advocate to study variants of the dynamic programming-based models of Sect. 35.3 that can accommodate non-exponentially distributed activity durations. Such problems are likely to resist exact solution, but they could be optimized using approximate dynamic programming techniques (Bertsekas and Tsitsiklis 1996; Powell 2011). First attempts in this direction are reported by Herbots et al. (2007). Similarly, the informational requirements of the stochastic programming-based models of Sect. 35.4 could be relaxed to allow for partial knowledge of the probability distributions. Using modern robust optimization techniques (Ben-Tal et al. 2009), this could also lead to scenario-free problem formulations that scale gracefully with problem size. First results of this type have been reported by Goh and Hall (2013), Wiesemann (2012) and Wiesemann et al. (2012b), but we are not aware of any contributions to the NPV maximization problem.

A second promising area for future research is the modeling of risk aversion. As we discussed in Sect. 35.2, managers are likely to give greater weight to undesirable outcomes than the average project performance. To date, this is not reflected in the project scheduling literature, which predominantly studies expected value problems. The models in Sect. 35.3 can cater for risk aversion through the inclusion of higher-order moments (Kulkarni and Adlakha 1986), whereas the formulations in Sect. 35.4 can be extended to optimize risk measures such as the conditional value-at-risk. Risk-averse project scheduling has been explored recently

in the context of makespan minimization by Goh and Hall (2013), Wiesemann (2012) and Wiesemann et al. (2012a).

Another interesting research direction is the integration of resource constraints. While this has been largely accomplished in the models of Sect. 35.3, it is yet to be investigated for the formulations in Sect. 35.4. A possible starting point would be to optimize over modified versions of the policy classes proposed by Yang et al. (1995) for the resource-constrained deterministic NPV maximization problem.

NPV maximization bears similarity to the payment scheduling problem, where both the activity start times and the cash flows can be chosen by the decision maker (Dayanand and Padman 1999, 2001; He et al. 2012; Kolisch and Padman 2001). It would be instructive to study the effects of uncertainty and risk aversion in the context of this problem, which to our knowledge has not been attempted.

Finally, we remark that the stochastic NPV maximization problem is closely related to real options theory (Dixit and Pindyck 1994). Research in that area has shown that the NPV criterion may lead to incorrect advice if applied to investment decisions that can be delayed (i.e., that are not of the "now-or-never" type). Although the potential impact of real options theory on project scheduling has been recognized for some time (Herroelen et al. 1997), the NPV maximization literature has largely disregarded these results.

References

Azaron A, Tavakkoli-Moghaddam R (2006) A multi-objective resource allocation problem in dynamic PERT networks. Appl Math Comput 181(1):163–174

Azaron A, Tavakkoli-Moghaddam R (2007) Multi-objective time-cost trade-off in dynamic PERT networks using an interactive approach. Eur J Oper Res 180(3):1186–1200

Azaron A, Katagiri H, Sakawa M, Kato K, Memariani A (2006) A multi-objective resource allocation problem in PERT networks. Eur J Oper Res 172(3):838–854

Benati S (2006) An optimization model for stochastic project networks with cash flows. Comput Manag Sci 3(4):271–284

Bendavid I, Golany B (2011) Predetermined intervals for start times of activities in the stochastic project scheduling problem. Ann Oper Res 186(1):429–442

Bendavid I, Golany B (2011) Optimal feeding buffers for projects or batch supply chains by an exact generalization of the newsvendor result. Ann Oper Res 189(1):25–42

Benmansour R, Elmaghraby SE, Allaoui H, Artiba A (2010) The approximation of distributions by phase-type distributions. In: Proceedings of the 3rd international conference on informations systems, logistics and supply chain, Casablanca

Ben-Tal A, El Ghaoui L, Nemirovski A (2009) Robust optimization. Princeton University Press, Princeton

Bertsekas DP (2007) Dynamic programming and optimal control, vol 2. Athena Scientic, Nashua

Bertsekas DP, Tsitsiklis JN (1996) Neuro-dynamic programming. Athena Scientific, Nashua

Bidot J, Vidal T, Laborie P, Beck JC (2009) A theoretic and practical framework for scheduling in a stochastic environment. J Sched 12(3):315–344

Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999) Resource-constrained project scheduling: notation, classication, models, and methods. Eur J Oper Res 112(1):3–41

Buss AH (1995) Sequential experimentation for estimating the optimal delay of activities in PERT networks. In: Proceedings of the 1995 winter simulation conference, Arlington

Buss AH, Rosenblatt MJ (1997) Activity delay in stochastic project networks. Oper Res 45(1):126–139

Chen W-N, Zhang J (2012) Scheduling multi-mode projects under uncertainty to optimize cash flows: a Monte Carlo ant colony system approach. J Comput Sci Technol 27(5):950–965

Colvin M, Maravelias CT (2010) Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. Eur J Oper Res 203(1):205–215

Colvin M, Maravelias CT (2011) R&D pipeline management: task interdependencies and risk management. Eur J Oper Res 215(3):616–628

Creemers S, De Reyck B, Leus R (2009) R&D project planning with multiple trials in uncertain environments. In: Proceedings of the IEEE international conference on industrial engineering and engineering management, Macau

Creemers S, Leus R, De Reyck B (2010) Project scheduling with alternative technologies: incorporating varying activity duration variability. In: Proceedings of the IEEE international conference on industrial engineering and engineering Management, Macau

Creemers S, Leus R, Lambrecht M (2010) Scheduling Markovian PERT networks to maximize the net present value. Oper Res Lett 38(1):51–56

Dayanand N, Padman R (1999) On payment schedules in contractor client negotiations in projects: an overview of the problem and research issues. In: Węglarz J (ed) Project scheduling: recent models, algorithms and applications. Kluwer Academic, Dordrecht, pp 477–508

Dayanand N, Padman R (2001) Project contracts and payment schedules: the client's problem. Manag Sci 47(12):1654–1667

Demeulemeester EL, Herroelen WS (2002) Project scheduling: a research handbook. Kluwer Academic, Dordrecht

De Reyck B, Leus R (2008) R&D-project scheduling when activities may fail. IIE Trans 40(4):367–384

De Reyck B, Grushka-Cockayne Y, Leus R (2007) A new challenge in project scheduling: the incorporation of activity failures. Rev Bus Econ LII(3):411–434

Dixit AK, Pindyck RS (1994) Investment under uncertainty. Princeton University Press, Princeton

Elmaghraby SE (2001) On the optimal release time of jobs with random processing times, with extensions to other criteria. Int J Prod Econ 74(1–3):103–113

Elmaghraby SE (2005) On the fallacy of averages in project risk management. Eur J Oper Res 165(2):307–313.

Elmaghraby SE, Herroelen WS (1992) The scheduling of activities to maximize the net present value of projects. Eur J Oper Res 49(1):35–49

Elmaghraby SE, Kamburowski J (1990) On project representation and activity floats. Arab J Sci Eng 15(4B):626–637

Elmaghraby SE, Kamburowski J (1992) The analaysis of activity networks under generalized precedence relations (GPRs). Manag Sci 38(9):1245–1263

Elmaghraby SE, Ferreira AA, Tavares LV (2000) Optimal start times under stochastic activity durations. Int J Prod Econ 64(1–3):153–164

Fazar W (1959) Program evaluation and review technique. Am Stat 13(2):10

Ghomi SMTF, Hashemin SS (1999) A new analytical algorithm and generation of Gaussian quadrature formula for stochastic network. Eur J Oper Res 114(3):610–625

Goh J, Hall NG (2013) Total cost control in project management via satisficing. Manag Sci 59(6):1354–1372

Goldratt EM (1997) Critical chain. The North River Press, Great Barrington

Grinold RC (1972) The payment scheduling problem. Nav Res Logist Q 19(1):123–136

Gutin E, Kuhn D, Wiesemann W (2013) Interdiction games on Markovian PERT networks. Available on Optimization Online, Forthcoming, Manag Sci

Hagstrom JN (1988) Computational complexity of PERT problems. Networks 18(2):139–147

He Z, Wang N, Li P (2012) Simulated annealing for financing cost distribution based project payment scheduling from a joint perspective. Ann Oper Res (published online before print)

Heitsch H and Römisch W (2003) Scenario reduction algorithms in stochastic programming. Comput Optim Appl 24(2–3):187–206

Henrion R, Küchler C, Römisch W (2009) Scenario reduction in stochastic programming with respect to discrepancy distances. Comput Optim Appl 43(1):67–93

Herbots J, Herroelen WS, Leus R (2007) Efficient and effective solution procedures for order acceptance and capacity planning. Working paper, Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, doi:10.2139/ssrn.1089359

Herroelen WS, Leus R (2004) Robust and reactive project scheduling: a review and classification of procedures. Int J Prod Res 42(8):1599–1620

Herroelen WS, Leus R (2005) Project scheduling under uncertainty: survey and research potentials. Eur J Oper Res 165(2):289–306

Herroelen WS, Van Dommelen P, Demeulemeester EL (1997) Project network models with discounted cash flows: a guided tour through recent developments. Eur J Oper Res 100(1):97–121

Herroelen WS, Demeulemeester E, De Reyck B (1999) A classification scheme for project scheduling. In: Węglarz J (ed) Project scheduling: recent models, algorithms and applications. Kluwer Academic, Dordrecht, pp 1–26

Ierapetritou M, Li Z (2009) Modeling and managing uncertainty in process planning and scheduling. In: Chaovalitongse W, Furman KC, Pardalos PM (eds) Optimization and logistics challenges in the enterprise. Springer, Berlin, pp 97–144

Jørgensen T, Wallace SW (2000) Improving project cost estimation by taking into account managerial flexibility. Eur J Oper Res 127(2):239–251

Kall P, Wallace SW (1994) Stochastic programming. Wiley, Hoboken

Kavadias S, Loch CH (2004) Project selection under uncertainty: dynamically allocating resources to maximize value. Kluwer Academic, Dordrecht

Ke H, Liu B (2005) Project scheduling problem with stochastic activity duration times. Appl Math Comput 168(1):342–353

Ke H, Liu B (2007) Project scheduling problem with mixed uncertainty of randomness and fuzziness. Eur J Oper Res 183(1):135–147

Ke H, Liu B (2010) Fuzzy project scheduling problem and its hybrid intelligent algorithm. Appl Math Model 34(2):301–308

Kelley JE Jr (1961) Critical path planning and scheduling: mathematical basis. Oper Res 9(3):296–320

Kimms A (2001) Mathematical programming and financial objectives for scheduling projects. Kluwer Academic, Dordrecht

Kolisch R, Padman R (2001) An integrated survey of deterministic project scheduling. Omega-Int J Manag S 29(3):249–272

Kulkarni VG, Adlakha VG (1986) Markov and Markov-regenerative PERT networks. Oper Res 34(5):769–781

Li Z, Ierapetritou M (2008) Process scheduling under uncertainty: review and challenges. Comput Chem Eng 32(4–5):715–727

Lock D (2007) Project management, 9th edn. Gower Publishing, Burlington

Luenberger DG (1997) Investment science. Oxford University Press, Oxford

Malcolm DG, Roseboom JH, Clark CE, Fazar W (1959) Application of a technique for research and development program evaluation. Oper Res 7(5):646–669

Maylor H (2010) Project Management, 4th ed. Prentice Hall, Upper Saddle River, NJ

Meredith JR, Mantel SJ (2006) Project management: a managerial approach. Wiley, Hoboken

Möhring RH (2001) Scheduling under uncertainty: Bounding the makespan distribution. In: Alt H (ed) Computational discrete mathematics: advanced lectures. Springer, Berlin, pp 79–97

Nadjafi BA, Kolyaei S (2010) A multi objective Fibonacci search based algorithm for resource allocation in PERT networks. J Opt Ind Eng 3(6):13–23

Neumann K (1999) Scheduling of projects with stochastic evolution structure. In: Węglarz J (ed) Project scheduling: recent models, algorithms and applications. Kluwer Academic, Dordrecht, pp 309–332

Neumann K, Zimmermann J (2000) Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. Eur J Oper Res 127(2):425–443

Özdamar L (1998) On scheduling project activities with variable expenditure rates. IIE Trans 30(8):695–704

Özdamar L, Dündar H (1997) A flexible heuristic for a multi-mode capital constrained project scheduling problem with probabilistic cash inflows. Comp Oper Res 24(12):1187–1200

Pflug G (2000) Some remarks on the value-at-risk and the conditional value-at-risk. In: Uryasev S (ed) Probabilistic constrained optimization: methodology and applications. Kluwer Academic, Dordrecht, pp 272–281

Powell WP (2011) Approximate dynamic programming: solving the curses of dimensionality, 2nd edn. Wiley, Hoboken

Project Management Institute (2013) A guide to the project management body of knowledge (PMBOK guide), 5th edn. Project Management Institute, Newtown Square

Puterman ML (1994) Markov decision processes: discrete stochastic dynamic programming. Wiley, Hoboken

Rockafellar RT, Uryasev S (2000) Optimization of conditional value-at-risk. J Risk 2(3):21–41

Rudolph AJ, Elmaghraby SE (2009) The optimal resource allocation in stochastic activity networks via continuous time Markov chains. In: Proceedings of the international conference on industrial engineering and systems management, Montreal

Russell AH (1970) Cash flows in networks. Manag Sci 16(5):357–373

Ruszczyński A, Shapiro A (2003) Stochastic programming. Elsevier, Amsterdam

Schmidt CW, Grossmann IE (2000) The exact overall time distribution of a project with uncertain task durations. Eur J Oper Res 126(3):614–636

Schwindt C, Zimmermann J (2001) A steepest ascent approach to maximizing the net present value of projects. Math Method Oper Res 53(3):435–450

Shavandi H, Najafi AA, Moslehirad A (2012) Fuzzy project scheduling with discounted cash flows. Econ Comput Econ Cyb 46(1):219–232

Shtub A (1986) The trade-off between the net present cost of a project and the probability to complete it on schedule. J Oper Manag 6(4):461–470

Slotnick SA (2011) Order acceptance and scheduling: a taxonomy and review. Eur J Oper Res 212(1):1–11

Sobel MJ, Szmerekovsky JG, Tilson V (2009) Scheduling projects with stochastic activity duration to maximize expected net present value. Eur J Oper Res 198(3):697–705

Tavares LV, Ferreira JAA, Coelho JS (1998) On the optimal management of project risk. Eur J Oper Res 107(2):451–469

Tereso AP, Araújo MMT, Elmaghraby SE (2003) Experimental results of an adaptive resource allocation technique to stochastic multimodal projects. In: Proceedings of the international conference on industrial engineering and production management, Porto

Tereso AP, Araújo MMT, Elmaghraby SE (2004) Adaptive resource allocation in multimodal activity networks. Int J Prod Econ 92(1):1–10

Tereso AP, Araújo MMT, Moutinho RS, Elmaghraby SE (2008) Project management: multiple resources allocation. In: Proceedings of the international conference on engineering optimization, Rio de Janeiro

Tereso AP, Barreiro D, Araújo MMT, Elmaghraby SE (2010) On the optimal resource allocation in projects considering the time value of money. In: Proceedings of the third international conference on information systems, logistics and supply chain, Casablanca

Trietsch D (2005) Economically balanced criticalities for robust project scheduling and control. Working paper, ISOM Department, University of Auckland, Auckland

Trietsch D (2006) Optimal feeding buffers for projects or batch supply chains by an exact generalization of the newsvendor result. Int J Prod Res 44(4):627–637

Uçal İ, Kuchta D (2011) Project scheduling to maximize fuzzy net present value. In: Proceedings of the world congress on engineering, London

Verderame PM, Elia JA, Li J, Floudas CA (2010) Planning and scheduling under uncertainty: a review across multiple sectors. Ind Eng Chem Res 49(9):3993–4017

Wang J, Wu Z, Sun Y (2000) Optimum activity delay in stochastic activity networks. In: Proceedings of the 7th international workshop on project management and scheduling, Osnabrück

Wiesemann W (2012) Optimization of temporal networks under uncertainty. Springer, Berlin

Wiesemann W, Kuhn D, Rustem B (2010) Maximizing the net present value of a project under uncertainty. Eur J Oper Res 202(2):356–367

Wiesemann W, Kuhn D, Rustem B (2012) Multi-resource allocation in stochastic project scheduling. Ann Oper Res 193(1):193–220

Wiesemann W, Kuhn D, Rustem B (2012) Robust resource allocations in temporal networks. Math Program 135(1–2):437–471

Yang KK, Tay LC, Sum CC (1995) A comparison of stochastic scheduling rules for maximizing project net present value. Eur J Oper Res 85(2):327–339

Chapter 36
The Stochastic Discrete Time-Cost Tradeoff Problem with Decision-Dependent Uncertainty

Evelina Klerides and Eleni Hadjiconstantinou

Abstract In this chapter we examine how managerial flexibility can be incorporated into project management techniques where project activities are executed under time-cost tradeoff settings and uncertainty. We advocate the use of a stochastic dynamic model where the activity scheduling decisions are taken dynamically over time thus offering flexibility to the decision maker of adjusting the decisions according to observations. To this end, we show that the problem is amenable to a Multi-Stage Stochastic Integer Programming approach with decision-dependent uncertainty as the decisions influence the revelation time of the random variables. We present the mathematical formulation of this problem and develop algorithmic approaches for obtaining effective lower and upper bounds. Our extensive computational results, based on a large number of test instances of varying size and degree of uncertainty, demonstrate the effectiveness of the proposed approaches in finding tight bounds for a class of non-standard stochastic programs bearing additional computational complexity.

Keywords Project scheduling • Stochastic programming • Time-cost tradeoff • Uncertain durations

36.1 Introduction

Stochastic Programming (SP) is an area of Mathematical Programming which provides a framework for modeling optimization problems involving uncertainty. Generally, the goal of SP techniques is to find a solution that is feasible for all

E. Klerides
Business School, Imperial College London, London, UK
e-mail: e.klerides05@imperial.ac.uk

E. Hadjiconstantinou (✉)
Business School, Imperial College London, London, UK

School of Engineering, Frederick University, Nicosia, Cyprus
e-mail: e.hconstantinou@imperial.ac.uk; eng.he@frederick.ac.cy

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_6

(or for some selected or representative) realizations of the random parameters while minimizing or maximizing a performance measure, usually a function of the decision variables. Depending on the decision maker, we may consider measures such as expectations, worst case performance, or a probability of attaining a predetermined target goal. Most of the work in the SP literature is devoted to the expected value model involving worst-case constraints.

For presentation purposes, assume we wish to minimize the expected value of the objective function $f(x, \tilde{\xi})$ for decisions $x \in X$ and random variables $\tilde{\xi} \in \Xi$. The SP literature defines three types of decisions, based on either a "Here-and-Now" (HN), "Wait-and-See" (WS) or "Mean-Value" (MV) approach. As their name suggests, HN decisions are taken *a priori*, i.e., before any randomness is considered, and their objective value z_{HN} is given by $z_{HN} = \min_x E(\tilde{\xi})[f(x, \tilde{\xi})]$. On the other hand, the WS solution $x_{WS} = \arg\min_x [f(x, \xi_{WS})]$ is obtained *a posteriori*, where ξ_{WS} are the realized values of the random parameters $\tilde{\xi}$, and the resulting objective value is given by z_{WS}. In this case, we assume a perfect information state and that all randomness has already been realized. Finally, the MV approach only considers expected values; its solution $x_{MV} = \arg\min_x [f(x, E(\tilde{\xi}))]$, leads to the deterministic objective value $z_{MV}^{det} = f(x_{MV}, E(\tilde{\xi}))$, and to the expected objective value $z_{MV} = E(\tilde{\xi})[f(x_{MV}, \tilde{\xi})]$, if x_{MV} is feasible with respect to all possible outcomes of the random variables.

In SP models, the stochastic process is characterized by three basic concepts, namely, the *scenario tree*, the *scenario problems* and the *non-anticipativity constraints*, Higle (2005). A scenario is one particular realization of all the stochastic elements in the problem. The scenario tree depicts the manner in which the stochastic elements evolve over time and is a structured representation of the way information is observed. Hence, each node represents a possible information state while each arc emanating from a node represents a possible transition to another information state, at a later time. Associated with this arc is a transition probability. A scenario problem is the deterministic optimization problem derived from one scenario; the latter corresponds to a path from the root node to a leaf node in the scenario tree. The probability of a scenario is a combination of all the probabilities of the arcs included in the path and is equal to the probability of reaching the corresponding leaf node from the root node.

In general, the sequence of decisions taken for stochastic programs needs to conform to the information structure of the problem, as this is defined by the scenario tree. For this purpose, we impose the *non-anticipativity* or *implementability* constraints which link decisions for different scenarios. Two scenarios are said to be *indistinguishable* at a given point in time if they are identical in the realizations for all stochastic parameters in which uncertainty has been resolved until that time. The non-anticipativity constraints state that if two scenarios are indistinguishable at a given point in time then decisions for these two scenarios at the current time should be the same. In other words, decisions taken at a given time can only be based on knowledge that is available at that point in time and not based on information that will be revealed in the future. Hence, we refer also to the above as implementability constraints as they result in solutions that are implementable.

From the modeling perspective, standard SP techniques in the literature involve assigning appropriate probability distribution functions to the stochastic problem

parameters and assume that the optimization decisions cannot influence the stochastic process. The uncertainty is then said to be of exogenous nature and the scenario tree is treated as an input into the optimization problem. However, when the underlying stochastic process depends on the optimization decisions, the scenario tree must be treated as a decision variable. In this case, the optimization decisions can influence either the transition probabilities or the revelation time of the values of the uncertain variables. The formulation of the non-anticipativity constraints thus becomes more complicated as the nodes of the scenario tree, and hence the sets of scenarios passing through each node, are not fixed. This is the case of the so-called decision-dependent or endogenous uncertainty, which may become an impeding factor in preserving tractability of the optimization model.

In this chapter, we study the Stochastic Discrete Time-Cost Tradeoff Problem (SDTCTP), in which the decision-maker is allowed to execute each activity in a project under a number of different modes, each associated to a time-cost pair, so that either the project completes within budget in minimum time or within its deadline in minimum cost. The activity durations are random variables to the problem. Klerides and Hadjiconstantinou (2010) showed that by assuming the mode selection variables to be static and fixed regardless of the realisations of the stochastic parameters, the SDTCTP automatically reduces to a two-stage SP model, where the mode selection is done at the first stage, and the scheduling process is performed during the second stage. In this paper, we investigate the modeling and algorithmic implications of relaxing the above assumption, hence, we consider the SDTCTP with Dynamic Modes. This problem is referred to in this paper as the SDTCTP with DM.

The remainder of this chapter is organized as follows. In the next section we provide an overview of the state-of-the art developments in the area of SDTCTP and decision-dependent uncertainty (Sect. 36.2). A problem description of the SDTCTP with DM is given in Sect. 36.3 and its mathematical formulation is presented in Sect. 36.4, including the complicating non-anticipativity constraints, as a result of decision-dependent uncertainty. In the same section we also present effective and efficient lower bounding techniques as well as heuristic methodologies, specifically designed to provide tight bounds to the SDTCTP with DM. Section 36.5 provides an illustrative example and the computational study presented in Sect. 36.6 demonstrates the capabilities of the proposed algorithms. The final conclusions are given in Sect. 36.7.

36.2 Literature Review

The focus of this chapter is on the budget version of the SDTCTP, where the objective is to minimize the expected project makespan subject to a pre-defined budget. In this section we provide an overview of the relevant literature on both the budget and deadline versions of the SDTCP; the latter aims to minimize the total cost subject to a pre-defined deadline.

The first part of this section provides an overview of the static and dynamic models developed for the SDTCTP. The second part describes the modeling and algorithmic advances for problems dealing with decision-dependent uncertainty.

To the best of our knowledge, the only developments in the field of optimal solution methodologies for SDTCTPs have only considered the static version of the problem. This is different to the problem considered in this chapter, which is able to dynamically make the mode selections according to different outcomes of the random variables, i.e., are scenario-dependent. This new feature exploits the revelation of uncertainty (duration of activities that have been completed) during project execution thus allowing to make more informed decisions for the activities that have not yet been started. In the described developments which follow, the proposed models assume that the optimization provides a choice of modes for each activity which is independent of the scenarios; the solutions cannot be adapted to observed outcomes.

Godinho and Branco (2012) propose an electromagnetism-based heuristic approach for dynamic multi-mode project scheduling under uncertainty. They apply a tardiness penalty whenever the project duration exceeds the pre-determined deadline and each activity's execution mode is found by comparing its starting time to a set of thresholds. Klerides and Hadjiconstantinou (2010) consider the budget SDTCTP with static modes, where, as opposed to the dynamic SDTCTP considered in this chapter, the activities are processed under pre-selected modes which cannot be adapted during the project execution according to different outcomes of the random variables. The approach assumes that the mode selection is performed at the first stage and all the uncertainty is resolved at the second stage. The authors propose a path-based two-stage SP approach for the time-cost tradeoff problem that examines different scenarios of activity durations and delivers a solution which accommodates fluctuations in the activity durations. The SP model is solved using a decomposition-based approach which allows decoupling the different scenario problems and proves to be capable of solving for the first time in the literature, many large and hard test instances in reasonable computational time using modest memory requirements. Zhu et al. (2007) also study the static two-stage SP approach for the problem of setting target finish times (due dates) for activities in a project network under time-cost tradeoffs and stochastic activity durations. The authors study the budget-constrained version of the problem using a heuristic methodology and report computational results for test instances consisting of 90 activities. For a special case of the SDTCTP, based on a two-stage formulation with recourse, a stochastic branch-and-bound approach has been suggested by Gutjahr (2000). A specified number of mode combinations is available for the project activities. The author assumes that penalty costs are incurred if the project exceeds its deadline; the proposed model minimizes the expected penalty cost plus the cost associated with the selected activity modes. The paper reports promising computational results for 33 randomly generated activity-on-arc problem instances consisting of 25, 50, and 100 nodes with beta-distributed activity durations and 10, 15, or 20 crashing modes.

The remaining developments in the literature deal with the continuous stochastic TCTP, see for example Wollmer (1985) and Laslo (2003).

Since this chapter deals with endogenous or decision-dependent uncertainty, we now review the previous work in the more general SP framework on this type of uncertainty. Due to the challenging nature of these problems, relevant previous work is limited to a few papers only. Pflug (1990) was the first to address problems with decision-dependent uncertainty. Decisions may influence the evolution of the scenario tree in at least two ways. On the one hand, the decision maker may take action to influence one possibility or outcome to become more probable than another, hence, influence the probability distributions. On the other hand, the decision maker could make decisions in order to get more accurate information by resolving some of the uncertainty and as a result, either completely disregard future (thereafter impossible) outcomes or become more certain as to which possibilities may occur.

For the first type of endogenous uncertainty, Vishwanath et al. (2004) consider a two-stage network problem, where in the first stage investment decisions may be taken to increase the probabilities of some arcs being available for traversal at the second stage. Ahmed (2000) also considers network design, server selection, and facility location problems with decision-dependent uncertainty of the first type. A 0-1 hyperbolic programming formulation as well as an exact algorithm for single stage problems with discrete decisions are presented.

Jonsbraten et al. (1998) addressed problems with the second type of endogenous uncertainty. The proposed implicit enumeration algorithm, which solves two-stage problems of this type, includes a branch-and-bound approach to determine the optimal set of decisions, each corresponding to a different scenario tree. Held and Woodruff (2005) propose a heuristic approach for the multi-stage network interdiction problem. Goel and Grossmann (2004) consider the gas field problem, which also suffers from the second type of decision-dependent uncertainty. A disjunctive formulation of the non-anticipativity constraints is presented, which is then solved using a heuristic algorithm. Goel and Grossmann (2007) generalize their approach to accommodate both exogenous and endogenous uncertainty and suggest problem reduction techniques. Their formulation is solved via a Lagrangian duality based branch-and-bound algorithm. Extensions of this work are found in Tarhan et al. (2009). Solak (2007) presents a formulation for the project portfolio optimisation problem which is amenable to scenario decomposition. The author solves sample problems using Lagrangian relaxation and lower bounding heuristics. Finally, Colvin and Maravelias (2010) develop a branch-and-cut algorithm to solve the multi-stage stochastic pharmaceutical clinical trial planning problem, where the essential non-anticipativity constraints are removed and only added when they are violated within the search tree. The model is initially reduced in size using several theoretical properties including some proposed in Goel and Grossmann (2007) for problems involving scenarios covering the entire space of possible outcomes. The algorithm is tested on six instances, with the smallest instance involving 64 scenarios and 12 stages solved within 100 CPU seconds and the largest instance with 4,608 scenarios and 6 stages solved in almost 78 CPU hours.

36.3 Problem Definition

The SDTCTP can be defined as follows: We are given an activity-on-node project network of n activities. The network is defined by an acyclic digraph $G = (V, E)$, where $V = \{0, \ldots, n+1\}$ is the set of *nodes* (activities) and E is the set of arcs (immediate precedence constraints) of the project. We assume that node 0 is the unique source node in the network and that node $n+1$ is the unique sink node; nodes 0 and $n+1$ represent dummy activities. Without loss of generality, we assume that the activities are topologically ordered, so that all the predecessors of activity i have an index which is less than i. We denote the start activities, i.e. the activities whose only predecessor is node 0 as the set V^s. Let $\mathscr{P}$ be the index set of all paths in the project network, starting from activity 0 and ending at activity $n+1$ and V_ℓ the set of activities contained in path $\ell \in \mathscr{P}$.

Each activity $i \in V$ has an index set of $\mathscr{M}_i$ possible modes of size M_i and each mode $m \in \mathscr{M}_i$ is associated to a (stochastic) duration $\tilde{p}_{im}$, and a (deterministic) cost, c_{im}. Since $\tilde{p}_{im}$ is unknown before the completion of activity i under mode m, it is represented by a vector of possible realisations. Following the classic PERT approach, we assume that the random variables $\tilde{p}_{im}$ are independent and that the corresponding individual distributions can be estimated. We also assume that $\tilde{p}_{im}$ for all $i \in V, m \in \mathscr{M}_i$ take integer values.

The uncertainty in the SDTCTP is represented using a set of discrete scenarios, Σ, in which each scenario $\sigma \in \Sigma$ is associated to a probability of occurrence, π_σ, where $\sum_{\sigma \in \Sigma} \pi_\sigma = 1$, and a realization of activity durations $p^\sigma_{im} (i \in V, m \in \mathscr{M}_i)$. In other words, each scenario $\sigma \in \Sigma$ contains one possible realisation of the vector $(\tilde{p}_{im})_{i \in V, m \in \mathscr{M}_i}$. For simplicity, we assume that the vectors $\tilde{p}_{im}$ are ordered with respect to the scenario index: $\tilde{p}_{im} = (p^1_{im}, p^2_{im}, \ldots, p^{|\Sigma|}_{im})$. Dummy activities 0 and $n+1$ have one mode with zero duration/cost under all scenarios.

It is assumed that the project's lifetime is represented by the discrete set of time periods $0, 1, \ldots, T$ where T denotes the project's horizon. If T is not known, we set it to the completion time of the project using the mode of longest duration for each activity i under each scenario σ, $T^\sigma_{max} = \max_{\ell \in \mathscr{P}} [\sum_{i \in V_\ell} \max_{m \in \mathscr{M}_i} [p^\sigma_{im}]]$, and maximize across all scenarios, i.e., $T = \max_{\sigma \in \Sigma} T^\sigma_{max}$. Similarly, the minimum completion time (feasible for all scenarios), T_{min}, is calculated via the modes of shortest duration for each activity i under scenario σ, $T^\sigma_{min} = \max_{\ell \in \mathscr{P}} [\sum_{i \in V_\ell} \min_{m \in \mathscr{M}_i} [p^\sigma_{im}]]$ and taken as the maximum across all scenarios. For the maximum (b_{max}) and minimum (b_{min}) budget we use the summation of the maximum and minimum costs for all the activities, respectively.

The notation used to describe the SDTCTP with DM and its mathematical formulation is given in Table 36.1.

This chapter deals with finding bounds for the HN solution to the SDTCTP with DM. The value of this solution is bounded from below by the WS solution value, z_{WS}, taken as the expectation across all scenarios. For each scenario problem σ, let z^σ_{WS} denote the optimal objective value for the DTCTP involving the durations of the activities defined for scenario σ. Similarly, the solution to the MV problem (z_{MV})

Table 36.1 Notation: SDTCTP with DM

Sets	
$V = \{0, \ldots, i, \ldots, n+1\}$	Set of nodes (project activities)
E	Set of arcs (immediate precedence constraints)
$\mathscr{M}_i$	Index set of modes for activity i
$M_i = \lvert\mathscr{M}_i\rvert$	Size of set $\mathscr{M}_i$
$\mathscr{P}$	Index set of all paths in the network
V^s	Set of start activities
V_ℓ	Set of activities contained in path ℓ
Σ	Index set of scenarios
Indices	
$i, j \in V$	An activity in the project
$m, m' \in \mathscr{M}_i$	A mode of activity i
$\ell \in \mathscr{P}$	A path in the project network
$t, t', \tau \in \mathbb{N}$	A time period
$\sigma, \sigma' \in \Sigma$	A scenario
Parameters	
b	Project budget
(b_{min}, b_{max})	Minimum, maximum total project cost
π_σ	Probability of scenario σ
(p^σ_{im}, c_{im})	Duration, cost of activity i under mode m under scenario σ
$(T^\sigma_{min}, T^\sigma_{max})$	Minimum, maximum project completion time under scenario σ
$T = \max_{\sigma\in\Sigma}[T^\sigma_{max}]$	Project horizon
$T_{min} = \max_{\sigma\in\Sigma}[T^\sigma_{min}]$	Minimum project completion time feasible for all scenarios
Optimization variables	
x^σ_{imt}	Whether or not activity i is started at time t under mode m under scenario σ
$z_{b-DM}(x)$	Project completion time of *b-SDTCTP* with DM

bounds z_{HN} from above. The optimal decision values under the MV scenario are denoted by x_{MV}. Using x_{MV}, the corresponding objective function value under each scenario $\sigma \in \Sigma$ is denoted as z^σ_{MV}. Table 36.2 gives the definition of these concepts, as these are applied for the SDTCTP with DM.

The HN solution to the SDTCTP with DM is to select a mode and assign a starting time for each activity under each scenario so that a given objective is achieved. Note that the focus of this chapter is on the budget version of the problem, denoted as *b-SDTCTP* with DM, where the objective is to minimize the expected project makespan subject to a pre-defined budget. In this problem, the assignment of modes is dynamic, i.e., the decision on the execution mode for an activity is only made when the activity becomes eligible to start and the selection depends

Table 36.2 Notation: standard stochastic programming for the SDTCTP with DM

z_{WS}^{σ}	Optimal WS project completion time under scenario σ found by solving the deterministic scenario problem σ
$z_{WS} = \sum_{\sigma \in \Sigma} \pi_{\sigma} z_{WS}^{\sigma}$	Expected project completion time under a WS approach
z_{HN}	Optimal expected project completion time under a HN approach. This is equivalent to the solution value of Model 1
σ_{MV}	MV scenario, that considers only expected values for random variables
x_{MV}	Optimal mode selection variables for the MV scenario
z_{MV}^{σ}	Project completion time using x_{MV} under scenario σ
$z_{MV} = \sum_{\sigma \in \Sigma} \pi_{\sigma} z_{MV}^{\sigma}$	Expected project completion time under a MV approach

on the observations made during the course of the decision process until that time. The decision process consists of a stage-wise selection of execution modes for a number of eligible activities under each scenario. Hence, an appropriate approach is to formulate the problem as a multi-stage stochastic program with recourse, in which recourse actions can be taken at each stage, after uncertainty on the durations of completed activities is revealed.

36.4 A New Multi-Stage SP Model for the *b-SDTCTP* with DM

In this section, we present a multi-stage stochastic integer formulation for the *b-SDTCTP* with DM (36.2)–(36.8) that will be used for finding its HN solution. In this problem, the time horizon is represented by a discrete set of time periods. Each stage (time period or decision point) of the problem is defined as either the beginning of the project ($t = 0$) or any point in time during project execution for which at least one activity in any scenario has been completed. The observed duration value of the completed activity reduces the number of scenarios under consideration at the current stage in the following way. Assume that an activity i was executed at a previous stage under mode m and has been completed by the current time stage. The observed duration value of the completed activity is p_{im}^{σ} for some $\sigma \in \Sigma$. It follows that all scenarios $\sigma' \in \Sigma$, such that $p_{im}^{\sigma'} \neq p_{im}^{\sigma}$, are excluded from the decision process at the current and all subsequent stages; we are certain they cannot be the true scenario. Following the completion of the activity, one or more activities may become eligible to start. The assignment of the execution modes for the eligible activities, based on realized duration values of the completed activities, should put the decision maker in the best possible position to cope with the future uncertainties captured by the scenarios which are still under consideration at the current stage. At the end of the decision process, all the activities are completed and the true scenario is revealed; the true path on the scenario tree becomes known. However, as at the beginning of the process we are completely unaware of which path/scenario will be revealed, a set of solutions for all scenarios is required; for each possible outcome an "optimal" action is necessary.

To develop a mathematical model of the *b-SDTCTP* with DM (36.2)–(36.8), we define the binary decision variables x^{σ}_{imt} (Table 36.1) for all activities $i \in V$ and $m \in \mathscr{M}_i$, at time $t \leq T$ and under all scenarios $\sigma \in \Sigma$ such that:

$$x^{\sigma}_{imt} = \begin{cases} 1, & \text{if activity } i \text{ is executed under mode } m, \text{ started at time } t, \\ & \text{under scenario } \sigma \\ 0, & \text{otherwise} \end{cases} \tag{36.1}$$

Ideally, for any common part of two or more paths in the scenario tree, we would only have a single decision variable. However, this is impossible, since the timing of the stages of the problem cannot be predicted in advance, as they depend on the activities' durations, which are the stochastic elements of the problem. To make matters worse, the durations are also dependent on the modes selected at a previous stage during the scheduling process. This means that the realization of future stochastic parameters depends on the decisions made so far, hence revealing that the problem contains decision-dependent uncertainty; the scenario tree is thus unknown.

To preserve the structure of the scenario tree, i.e., the common parts of the paths, we apply the non-anticipativity constraints (Sect. 36.1). In the case of the *b-SDTCTP* with DM, two scenarios σ and σ' ($\sigma \neq \sigma'$) are indistinguishable, if the observed duration value of any activity which has been completed before and including time t is identical under both scenarios σ and σ'. It is worth noting that, similarly to Solak (2007), in the mathematical formulation, we consider an explicit representation of the decision-dependent non-anticipativity constraints in a compact way. This enables the use of solution approaches developed in the literature for classical stochastic integer programs and also scenario decomposition methods.

As in Goel and Grossmann (2007), we define a set $\Delta(\sigma, \sigma')$ for all pairs of scenarios σ and σ', with $\sigma \neq \sigma'$ consisting of elements which distinguish between σ and σ'. The set consists of all pairs (i, m), $i \in V, m \in \mathscr{M}_i$ such that the value of the duration of activity i under mode m is different for scenarios σ and σ', i.e., $\Delta(\sigma, \sigma') = (i, m)_{i \in V, m \in \mathscr{M}_i, p^{\sigma}_{im} \neq p^{\sigma'}_{im}}$. By definition, $\Delta(\sigma, \sigma') = \Delta(\sigma', \sigma)$.

The sequence of events in the *b-SDTCTP* with DM is as follows. Observations p^{σ}_{jm} (realized duration values), for already scheduled activities j, and decisions x^{σ}_{imt}, for unscheduled activities i, are both made at the beginning of period t, with decisions being made immediately after observations. Note that the observations resulting from decisions taken at time t may be made after several stages; if decision $x^{\sigma}_{imt} = 1$ is taken, then the resolution of the uncertainty regarding activity i under mode m will be made at the beginning of time period $t' = t + p^{\sigma}_{im}$ and before new decisions are taken at time t'.

The mathematical formulation of the *b-SDTCTP* with DM is given in Eqs. (36.2)–(36.8).

Objective (36.2) minimizes the expected project completion time of the stochastic dynamic model. Note that $z_{dyn}(x)$ is a function of the decision variables x and denotes the expected project makespan of the *b-SDTCTP* with DM. Constraints

(36.3) and (36.4) ensure that the selection of modes does not exceed the available budget and that the precedence relations between the activities are satisfied, respectively. Logical constraints (36.5) impose the fact that only one mode must be assigned to each activity.

$$\text{Min. } z_{dyn}(x) = \sum_{\sigma \in \Sigma} \sum_{m \in \mathscr{M}_{n+1}} \sum_{t=0}^{T} \pi_\sigma t x^{\sigma}_{(n+1)mt} \tag{36.2}$$

$$\text{s.t.} \quad \sum_{i \in V} \sum_{m \in \mathscr{M}_i} \sum_{t=0}^{T} x^{\sigma}_{imt} c_{im} \leq b \quad (\sigma \in \Sigma) \tag{36.3}$$

$$\sum_{t=0}^{T} \sum_{m \in \mathscr{M}_j} t x^{\sigma}_{jmt} - \sum_{t=0}^{T} \sum_{m \in \mathscr{M}_i} t x^{\sigma}_{imt} - \sum_{m \in \mathscr{M}_i} \sum_{t=0}^{T} x^{\sigma}_{imt} p^{\sigma}_{im} \geq 0 \quad ((i,j) \in E;\ \sigma \in \Sigma) \tag{36.4}$$

$$\sum_{m \in \mathscr{M}_i} \sum_{t=0}^{T} x^{\sigma}_{imt} = 1 \quad (i \in V;\ \sigma \in \Sigma) \tag{36.5}$$

$$x^{\sigma}_{im0} - \sum_{\sigma' \in \Sigma} \pi_{\sigma'} x^{\sigma'}_{im0} = 0 \quad (i \in V;\ m \in \mathscr{M}_i;\ \sigma \in \Sigma) \tag{36.6}$$

$$|x^{\sigma}_{imt} - x^{\sigma'}_{imt}| - \sum_{(j,m') \in \Delta(\sigma,\sigma')} \sum_{\tau=0}^{t-\min\{p^{\sigma}_{jm'}, p^{\sigma'}_{jm'}\}} (x^{\sigma}_{jm'\tau} + x^{\sigma'}_{jm'\tau}) \leq 0$$
$$(\sigma, \sigma' \in \Sigma : \sigma < \sigma';\ i \in V;\ m \in \mathscr{M}_i;\ t = 1, 2, \ldots, T) \tag{36.7}$$

$$x^{\sigma}_{imt} \in \{0, 1\} \quad (i \in V;\ m \in \mathscr{M}_i;\ t = 0, 1, \ldots, T;\ \sigma \in \Sigma) \tag{36.8}$$

The non-anticipativity requirements are applied via constraints (36.6) and (36.7). If scenarios σ and σ' are indistinguishable at time t, then the decisions $x^{(.)}_{imt}$ should be the same for both scenarios σ and σ'. All scenarios are indistinguishable before any decision is taken at the first stage ($t = 0$) (constraint (36.6)). Constraint (36.7) imposes the non-anticipativity rule to scenarios σ and σ' for $t \geq 1$, if σ and σ' indistinguishable. To explain this, assume for now that only two scenarios, σ and σ', are under consideration. (The concepts can be easily extended to the case where more than two scenarios are considered.) Scenarios σ and σ' are said to be indistinguishable if no activity j has been scheduled under a mode m' and at time τ, such that, $(j, m') \in \Delta(\sigma, \sigma')$ and $\tau \leq t - \min\{p^{\sigma}_{jm'}, p^{\sigma'}_{jm'}\}$. Time $\tau = t - \min\{p^{\sigma}_{jm'}, p^{\sigma'}_{jm'}\}$ is the latest starting time for activity j which, scheduled under mode m', distinguishes between scenarios σ and σ' by time t. To clarify this

statement, assume without loss of generality that $p^{\sigma}_{jm'} < p^{\sigma'}_{jm'}$. Then if activity j has been scheduled under mode m' by time $\tau = t - \min\{p^{\sigma}_{jm'}, p^{\sigma'}_{jm'}\} = t - p^{\sigma}_{jm'}$, this implies that either the activity would be completed by time t, revealing that scenario σ' is not the true scenario, or it would still be in progress by time t, revealing that σ is not the true scenario. In either case, scheduling the activity under that mode by time τ would result in distinguishing between those two scenarios; constraints (36.9) are then redundant. If the activity is scheduled under that mode at any time after time τ, then scenarios σ and σ' would still be indistinguishable at time t; activity j would still be in progress, revealing that any of σ and σ' could be the true scenario. In that case, constraints (36.7) restrict all decisions taken at time t for the two scenarios into being the same; the two scenarios at time t have not yet taken a different route in the scenario tree.

Finally, constraints (36.8) define the domain of the decision variables.

Let $z^*_{dyn} = \min_x\{z_{dyn}(x)\}$ denote the optimal objective value of the model given by Eqs. (36.2)–(36.8).

36.4.1 Scenario Tree Generation: An Illustrative Example

We illustrate the concepts regarding the multi-stage nature of the *b-SDTCTP* with DM, including the evolution of the scenario tree, the non-anticipativity constraints, the sets of indistinguishable scenarios, and the scenario problems (Sect. 36.1), using the example project network given in Fig. 36.1 and the corresponding data displayed in Table 36.3.

In Fig. 36.2, we show the early stages of a possible scenario tree for the example (using specific decisions). Note that the leaf nodes of the tree correspond to a particular scenario problem, denoted at the bottom of Fig. 36.2. The tree is constructed by enlisting all the possible information states/decision points (identified by the node numbers shown in the circles), along with their probabilities (numbers shown on the arcs of the tree) in a tree structure which respects the order of the timings of the information revelations.

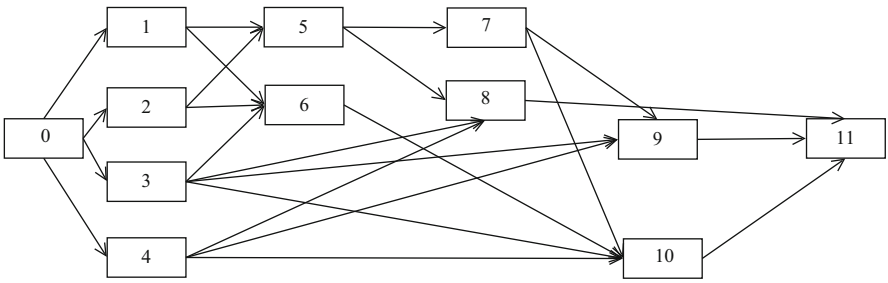


Fig. 36.1 Project network example

Table 36.3 Data for the activities of the project example in Fig. 36.1

Activity	Mode	Activity cost	Activity duration scenarios				
			1	2	3	4	5
0	1	0	0	0	0	0	0
1	1	7	10	10	11	8	11
	2	8	5	4	4	6	7
	3	9	4	2	3	3	2
	4	20	2	1	4	3	2
2	1	16	18	17	16	18	16
	2	17	15	17	14	15	14
	3	18	12	10	11	11	14
	4	20	6	8	7	5	7
3	1	10	13	14	14	15	14
	2	11	9	10	10	9	9
	3	13	8	6	9	10	7
	4	18	6	7	8	7	4
4	1	2	19	19	19	18	18
	2	3	17	15	17	17	17
	3	7	8	6	8	8	9
	4	15	5	5	4	4	4
5	1	1	18	19	20	18	18
	2	2	10	10	8	12	8
	3	9	8	8	10	10	8
	4	15	3	2	2	5	5
6	1	2	15	16	13	13	15
	2	6	12	14	12	13	11
	3	17	6	5	6	5	5
	4	20	4	3	2	5	4
7	1	3	18	18	19	20	18
	2	5	7	7	6	9	6
	3	8	6	5	4	7	6
	4	17	5	3	5	3	6
8	1	2	14	15	13	15	16
	2	6	12	13	14	13	13
	3	16	3	4	5	5	3
	4	20	2	1	4	1	4
9	1	6	16	17	15	15	16
	2	10	11	12	11	10	12
	3	16	7	8	8	9	6
	4	19	4	2	3	5	4
10	1	3	15	15	15	17	16
	2	10	8	10	7	9	9
	3	13	7	8	8	7	5
	4	18	5	3	5	6	3
11	1	0	0	0	0	0	0

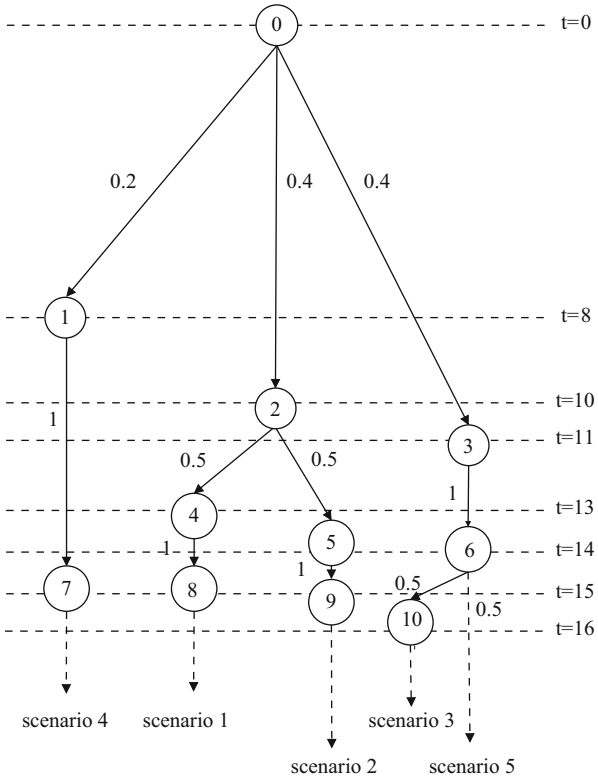


Fig. 36.2 Scenario tree for the project network of Fig. 36.1

The beginning of the project ($t = 0$) denotes the first decision point of the problem when no observations have been made yet; this decision point is associated to node 0 in Fig. 36.2. As no scenario-related information has been revealed at this point, all five scenarios are indistinguishable. To respect the non-anticipativity constraints, we require that all scheduling decisions at this stage are common to all five scenarios. Four activities are eligible to start following the dummy start activity 0, namely activities $1, 2, 3$, and 4. Let us assume that these activities are scheduled under modes $1, 2, 1$, and 2, respectively, at $t = 0$, by setting $x^{\sigma}_{im0} = 1$ for $\sigma = 1, \ldots, 5$ and $m = 1, 2, 1, 2$ for $i = 1, 2, 3, 4$, respectively. Table 36.4 shows the different possible observations (realized duration values) resulting from these decisions.

We observe that the earliest possible time for which new information may be revealed is at time $t = 8$. This occurs if activity 1 is completed by time 8, which is only true under scenario 4 (observation $p^4_{1,1} = 8$); we denote this particular state as node 1. All other activities are still in progress at node 1. As this information state is only possible under scenario 4, this particular scenario distinguishes itself from the remaining scenarios by time $t = 8$ and is the only scenario to go through node 1.

Table 36.4 Possible observations resulting from the scheduling decisions taken at $t = 0$

Activity	Mode	Activity durations for scenario σ				
i	m	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$
1	1	10	10	11	8	11
2	2	15	17	14	15	14
3	1	13	14	14	15	14
4	2	17	15	17	17	17

Decisions on scheduling eligible activities at stages/nodes originating from node 1 are independent of the remaining scenarios (the non-anticipativity constraints which include decision variables for scenario 4 become thereafter redundant). If, on the other hand, activity 1 is still in progress at time 8, then scenario 4 is considered impossible and is thus left out of the decision process at all subsequent stages. Only one of the remaining scenarios $1, 2, 3$, and 5 may be the true scenario (these scenarios are still indistinguishable at time $t = 8$).

Following the same process, we create node 2 to represent the information state where we observe the completion of activity 1 at $t = 10$ (observation $p^1_{1,1} = p^2_{1,1} = 10$) which is only true under scenarios 1 and 2; both of these scenarios are said to be indistinguishable between them at $t = 10$ but have distinguished themselves from scenarios $3, 4$, and 5. There are no scheduling decisions to be made under this state as there are no eligible activities to start at $t = 10$. Scenarios 1 and 2 are separated in the scenario tree at $t = 13$. By this time, activity 3 is either completed (observation $p^1_{3,1} = 13$), denoted by node 4, or still in progress; in the latter case, activity 3 is eventually completed at $t = 14$ with observation $p^2_{3,1} = 14$ and denoted by node 5. The non-anticipativity constraints on decisions under scenario 1 taken at stages originating from node 4 and on decisions under scenario 2 taken at stages originating from node 5 are thereafter redundant.

So far, we have considered the possibility of activity 1 ending either by time 8 or by time 10. In fact, if activity 1 is not completed by time 10, then it must end by time 11. Node 3 is associated to the state where the duration of activity 1 is equal to 11; scenarios 3 and 5 are the only scenarios to pass through this node and are therefore still indistinguishable between them at this time ($p^3_{1,1} = p^5_{1,1} = 1$). Node 6 represents the state where both activities 2 and 3 are completed at time $t = 14$, which is the case under both scenarios 3 and 5. At this stage, activity 5 is eligible to start, but the non-anticipativity constraints still apply. Therefore the selected execution mode for activity 5 has to be the same under both scenarios 3 and 5. Assume that activity 5 is to be scheduled under mode 4, meaning that $x^3_{5,4,14} = x^5_{5,4,14} = 1$. Scenarios 3 and 5 are distinguished from each other at time $t = 16$ when activity 5 ends under scenario 3 (node 10) but is still in progress under scenario 5. All the non-anticipativity constraints leave the decisions unaffected after time $t = 16$.

The above process is repeated until all activities are scheduled under all scenarios and the scenario tree is then completed; the route from the root node to a particular leaf node indicates the sequence of observations and associated scheduling decisions for the corresponding scenario.

36.4.2 Computation of Lower Bound LB Based on a Two-Stage Relaxation Algorithm

In this section, we propose an algorithm, referred to as TS_A, for solving a relaxation of (36.2)–(36.8); the obtained solution value is a valid lower bound for the *b-SDTCTP* with DM. In Sect. 36.3 we stated that the optimal solution to (36.2)–(36.8), also referred to as the HN solution, is bounded from below by z_{WS}. In this section, we prove that the proposed lower bound also bounds z_{HN} from below and is at least as strong as z_{WS}.

Although the problem involves a multi-stage decision making process, a natural simplification is a two-stage relaxation in which we assume a perfect information state once the decisions at the first-stage are made. This relaxation is expected to ease the computational burden induced by the complicating non-anticipativity constraints, which grow exponentially with the number of scenarios. The lower bound is also expected to be stronger than the WS lower bound, since it includes additional constraints on the non-anticipativity property of the decisions taken at $t = 0$, i.e., the mode selections for the activities which have no predecessors. Following the revelation of the uncertainty regarding these activities, it is expected that some scenarios will become distinct, thereby making a subset of the non-anticipativity constraints redundant. This feature is expected to contribute in delivering a tight lower bound.

36.4.2.1 Overview of Algorithm $CPA_{b-DTCTP}$

Before we describe in detail the proposed lower bound, we briefly provide an overview of Algorithm $CPA_{b-DTCTP}$ proposed in Hadjiconstantinou and Klerides (2010) for the deterministic DTCTP and which is implemented in this section. Algorithm $CPA_{b-DTCTP}$ is a delayed-constraint generation approach, specifically designed to solve the path-based formulation of the DTCTP. The algorithm begins by enumerating all the source-sink paths in the project. Each path corresponds to a *path constraint* in the path-based formulation which ensures that the project completion time is at least as large as the sum of the durations of the activities on the path. The algorithm then iteratively solves the path-based formulation of the DTCTP using a subset of the path constraints, each time adding the path constraints which are violated by the obtained solution. The algorithm ends when the predefined optimality criteria are met.

36.4.2.2 Algorithm TS_A

In the relaxation of (36.2)–(36.8), constraints (36.7), i.e., the non-anticipativity constraints for all times $t \geq 1$, are ignored. The non-anticipativity requirements are only applied for $t = 0$, the first stage of the problem. Once the decisions at $t = 0$

are made, the scenarios are decoupled as all the remaining constraints are defined separately for each scenario. Hence, at the second stage, the problem decomposes into distinct scenario problems. For this reason, the relaxed problem can be viewed as the two-stage relaxation of the *b-SDTCTP* with DM, thereafter referred to as problem *TS*.

TS_A is a two-stage algorithm which solves problem *TS*. The value of the resulting optimal solution is a lower bound to (36.2)–(36.8) and is denoted by *LB*. Algorithm TS_A involves finding the best combination of fixed modes $m^F = (m_i)_{i \in V^s}$ for the start activities, such that the expected project completion time across all scenarios is minimized, i.e.,

$$LB = \min_{m^F}\{LB(m^F)\} = \min_{m^F}\{\sum_{\sigma \in \Sigma} \pi_\sigma z_{TS}^{*\sigma}(m^F)\} \qquad \text{(TS)}$$

where $z_{TS}^{*\sigma}(m^F)$ is the optimal solution value to the specific deterministic scenario problem σ given fixed modes m^F. The latter problem is given in (36.9)–(36.14), where the binary decision variables y_{im} denote whether or not activity $i \in V$ is executed under mode $m \in \mathscr{M}_i$.

$$\text{Min. } z_{TS}^{\sigma}(m^F, y) \tag{36.9}$$

$$\text{s.t.} \quad \sum_{i \in V} \sum_{m \in \mathscr{M}_i} y_{im} c_{im} \leq b \tag{36.10}$$

$$\sum_{m \in \mathscr{M}_i} y_{im} = 1 \quad (i \in V \setminus \{V^s\}) \tag{36.11}$$

$$z_{TS}^{\sigma}(m^F, y) - \sum_{i \in V_\ell} \sum_{m \in \mathscr{M}_i} y_{im} p_{im}^{\sigma} \geq 0 \quad (\ell \in \mathscr{P}) \tag{36.12}$$

$$y_{im_i^F} \geq 1 \quad (i \in V^s) \tag{36.13}$$

$$y_{im} \in \{0, 1\} \quad (i \in V;\ m \in \mathscr{M}_i) \tag{36.14}$$

Objective (36.9) minimizes the two-stage project completion time under scenario σ when fixed modes m^F are assigned to all $i \in V^s$. Constraint (36.10) ensures that the selection of modes does not exceed the available budget and logical constraints (36.11) allow only one execution mode to be selected for each activity. Inequalities (36.12) guarantee that the project completion time is greater than or equal to the lengths of all the paths in the project network. Finally, constraints (36.13) and (36.14) impose the fixed modes for the start activities and define the domain for the decision variables, respectively.

If constraints (36.13) are ignored, the model is the same as the deterministic path-based model for the *b-DTCTP* given in Hadjiconstantinou and Klerides (2010). Equations (36.9)–(36.14) can then be solved using algorithm $CPA_{b-DTCTP}$ proposed in that paper, where constraints (36.13) are additionally included in the model of the relaxed problem at each iteration.

Clearly, the complexity of the algorithm depends on the number of scenarios as well as on the number of start activities in the project and their execution modes. In the worst-case, the $CPA_{b-DTCTP}$ algorithm is invoked a number of times equal to $|\Sigma| \prod_{i \in V^s} |\mathscr{M}_i|$. To improve the computational performance of the algorithm, we have developed a dominance rule, which is able to significantly reduce the search space for m^F. The dominance rule is applied for a given combination of modes m^F, and each time algorithm $CPA_{b-DTCTP}$ is applied to solve scenario $\sigma \in \Sigma$. The fundamental idea is that, for each scenario, a lower bound on the expected project completion time is computed. If this lower bound is greater than a known upper bound on the expected project makespan, the combination of modes is discarded. The dominance rule is described in detail below.

36.4.2.3 Dominance Rule

The dominance rule uses as input a known upper bound UB for the expected project completion time (e.g., the project horizon T) and the WS completion times for all scenarios $\sigma \in \Sigma$, z^{σ}_{WS}, the latter values being obtained using algorithm $CPA_{b-DTCTP}$. The rule is applied for each combination of m^F and each time the value $z^{\sigma *}_{TS}(m^F)$ is computed (using (36.9)–(36.14)), for any scenario $\sigma \in \Sigma$. Note that if a scenario problem turns out to be infeasible for m^F, then the choice of modes is automatically discarded.

Consider a specific combination of fixed modes for the set V^s, given by m^F and apply algorithm $CPA_{b-DTCTP}$ for all $\sigma' \leq \sigma$ to obtain optimal objective values $z^{*\sigma'}_{TS}(m^F)$ for all $\sigma' \leq \sigma$.

The dominance rule calculates the lowest possible value that can be obtained for the current combination of modes, m^F. This lower value, LB_{DR}, is calculated as

$$LB_{DR} = \sum_{\sigma' \leq \sigma} \pi_{\sigma'} z^{*\sigma'}_{TS}(m^F) + \sum_{\sigma' > \sigma} \pi_{\sigma'} z^{\sigma'}_{WS} \tag{36.15}$$

If the value LB_{DR} is greater than UB, then m^F is dominated. Clearly, the lower the upper bound used, then more effective the dominance rule. The computation of the upper bound is discussed in Sect. 36.4.3.

The complete algorithm TS_A is given in Algorithm 36.1.

Proposition 36.1. $LB \geq z_{WS}$.

Proof. The WS problem is a relaxation of problem TS, namely the two-stage *b-SDTCTP* with DM (constraints (36.6) are omitted). Therefore, the lower bound obtained from algorithm TS_A is at least as large than the one obtained from solving all scenario subproblems individually (z_{WS}). Hence, $LB \geq z_{WS}$. □

Additionally to the lower bound LB, algorithm TS_A also provides a lower bound on the project completion time under each scenario σ given by

$LB^{\sigma} = \min_{m^F}\{z_{TS}^{*s}(m^F)\}$. We now prove that the derived lower bound is stronger than the standard WS lower bound for each scenario problem σ.

Proposition 36.2. *$LB^{\sigma} \geq z_{WS}^{\sigma}$ for all $\sigma \in \Sigma$.*

Proof. z_{WS}^{σ} is the optimal solution found by solving a relaxation of (36.9)–(36.14), which ignores fixed modes for any activities. Hence, it is lower than $z_{TS}^{\sigma^*}$. In general, the following is true: $LB^{\sigma} \geq z_{WS}^{\sigma}$, *for all* $\sigma \in \Sigma$. □

Algorithm 36.1 TS_A: enumeration algorithm for problem TS

initialize
  $\mu := 1,\ m_i^F(\mu) := 1\ (i \in V^s)$
  $LB^{\sigma} := T\ (\sigma \in \Sigma)$
  $LB := T$

while $\mu < \mu^{max}$ **do**

  Step 1: Set $lb := 0$.
  while $\sigma \in \Sigma$ and $m^F(\mu)$ is not dominated **do**

    Step 2.1: Apply $CPA_{b-DTCTP}$ to solve (36.9)–(36.14) for scenario σ using fixed modes $m^F(\mu)$.
    Step 2.2: If the problem is infeasible, $m^F(\mu)$ is discarded. Otherwise, compute $z_{TS}^{\sigma^*}(m^F(\mu))$. Update $lb := lb + z_{TS}^{\sigma^*}(m^F(\mu))$.
    Step 2.3: Use the dominance rule to check if $m^F(\mu)$ is dominated.

  end while
  if $m^F(\mu)$ is not dominated **then**
    Step 3: $LB := \min\{lb, LB\}$
    while $\sigma < |\Sigma|$ **do**
      Step 4: $LB_{TS}^{\sigma} := \min\{LB^{\sigma}, z_{TS}^{\sigma^*}(m^F(\mu))\}$
    end while
  end if
  Step 5: Set $\mu := \mu + 1$ and set new combination of modes $m^F(\mu)$.
end while

return LB, LB^{σ} $(\sigma \in \Sigma)$

36.4.3 Computation of Upper Bound UB Based on a Static-Modes Policy

In this section we will present a heuristic approach for finding a good feasible solution to the *b-SDTCTP* with DM. The proposed heuristic, referred to as *HEUR*, achieves nearly optimal solutions in a fast and efficient way.

36.4.3.1 Overview of Algorithm $DA_{b-SDTCTP}$

The heuristic algorithm described in this section refers to Algorithm $DA_{b-SDTCTP}$ proposed in Klerides and Hadjiconstantinou (2010) for the two-stage SDTCTP. This algorithm was specifically designed to solve to optimality the path-based two-stage stochastic programming model for the SDTCTP with static modes. The first step of the algorithm is to solve a relaxation of the problem using delayed constraint generation, with constraints (cuts) being selected from the set of path constraints for each scenario. The solution to the relaxed problem (mode selections) enables the decomposition of the original problem into separate scenario subproblems, where each subproblem reduces to finding the project completion time under that scenario, using forward calculations and the mode values. The constraint generation procedure is repeated over a number of iterations until the termination criteria are satisfied.

36.4.3.2 Heuristic Algorithm *HEUR*

The fundamental idea behind the heuristic algorithm is building a feasible, non-anticipative solution $\hat{x}(\Pi) = (\hat{x}^{\sigma}_{imt}(\Pi))_{i \in V, m \in \mathscr{M}_i, t \leq T, \sigma \in \Sigma}$ from a static (scenario-independent) mode selection policy Π. Note that Π is a list of modes $\Pi_i \in \mathscr{M}_i$ for each activity $i \in V$ and is associated to mode selection variables y^{Π} such that:

$$y^{\Pi} = (y^{\Pi}_{im})_{i \in V, m \in \mathscr{M}_i | y^{\Pi}_{im} = 1 \text{ for } m = \Pi_i} \tag{36.16}$$

The reverse is also true, i.e., for mode selection variables y^{Π}, there exists a mode selection policy Π such that:

$$\Pi_i = \sum_{m \in \mathscr{M}_i} m y^{\Pi}_{im} \tag{36.17}$$

Once the non-anticipative solution is found, then the heuristic solution value is given by

$$UB = z_{dyn}(\hat{x}(\Pi)) \qquad (HEUR)$$

found by using $\hat{x}(\Pi)$ as a solution to (36.2)–(36.8).

The proposed heuristic algorithm *HEUR* is described below (Algorithm 36.2). It is interesting to note that we derive the static mode selection policy Π from the optimal solution to the *b-SDTCTP* with static modes by using (36.17) and the decomposition algorithm $DA_{b-SDTCTP}$ proposed in Klerides and Hadjiconstantinou (2010).

Algorithm 36.2 *HEUR*: heuristic algorithm for the *b-SDTCTP* with DM

Step 1: Use $DA_{b-SDTCTP}$ to derive optimal mode selection variables y^*_{im}. Set $\Pi_i := \sum_{m \in \mathscr{M}_i} m y^*_{im}$.
Step 2: Use $NONANT(\pi)$ (Procedure 36.1) to derive the heuristic non-anticipative solution $\hat{x}(\Pi)$.
Step 3: Using $\hat{x}(\Pi)$, set $UB := z_{dyn}(\hat{x}(\Pi))$ from (36.2).
return UB.

Step 2 of Algorithm *HEUR* involves procedure $NONANT(\Pi)$ (described in Procedure 36.1), which constructs a non-anticipative solution $\hat{x}(\Pi)$ based on a static mode selection policy Π. The procedure goes through all the decision stages of the project scheduling process. At each iteration, for given time t and decisions $\hat{x} = (\hat{x}^{\sigma}_{im\tau})_{i \in V, m \in \mathscr{M}_i, \tau < t, \sigma \in \Sigma}$, Procedure *FindIndistinguishableScenarios(t,$\hat{x}$)* (Procedure 36.2) is invoked to find all the sets of indistinguishable scenarios Σ_μ for $\mu = 1, \ldots, |IS|$ (Procedure 36.1, Step 1). For all such sets which are not singletons, all eligible activities are found using Procedure *FindEligibleActivities(t,$\hat{x}$, Σ_μ)* (Procedure 36.3); the modes prescribed by policy Π (Procedure 36.1, Step 2.2) are assigned to the eligible activities. The essence of $NONANT(\Pi)$ lies in the assignment of modes for any set Σ_μ of indistinguishable scenarios with a single member ($|\Sigma_\mu| = 1$), i.e., a scenario which has completely differentiated itself from all remaining scenarios. In such a case, the procedure allows assigning modes and starting times for all unscheduled activities independently of the policy Π, as long as this mode assignment is feasible and minimizes the project completion time of that scenario. The procedure terminates when either all of the scenarios have been distinguished or when the project horizon has been reached.

Procedure 36.2 displays the details of procedure *FindIndistinguishableScenarios(t,x)*. The procedure returns the set *IS*, which contains all sets of indistinguishable scenarios at time t and for decisions x taken so far, with $\bigcup_{\mu \le |IS|} \Sigma_\mu = \Sigma$ and $\Sigma_{\mu_1} \cap \Sigma_{\mu_2} = \emptyset$ for $\mu_1 \neq \mu_2$. The history of observations and decisions for each pair of scenarios σ and σ' ($\sigma < \sigma'$) is examined. If an activity i has been scheduled at mode $m \in \mathscr{M}_i$ with $(i, m) \in \Delta(\sigma, \sigma')$ at time $\tau \le t - \max\{p^{\sigma}_{im}, p^{\sigma'}_{im}\}$ under either of the two scenarios, then both scenarios are distinct from each other by time t. Otherwise, these scenarios are indistinguishable at time t and are therefore included in the same set.

Procedure 36.3 finds all the activities which are eligible to start at time t, using decisions x taken so far for a set of indistinguishable scenarios Σ'. Since all scenarios in set Σ' share the same history we need only perform the procedure for one of the members of the set Σ'. Let $\sigma = \Sigma'_1$ denote the first element of set Σ'. The eligible activities for scenario σ will be automatically eligible for all scenarios in Σ'. The eligible activities are identified as those having all their predecessors completed by time t. Note that the output of *FindEligibleActivities(t,x,Σ')* is a list of the eligible activities in a non-decreasing order of index.

Procedure 36.1 $NONANT(\pi)$

initialise
  $distinct\_scenario_\sigma$:= false ($\sigma \in \Sigma$).
  $t := 0$.
  $\hat{x} := 0, \ldots, 0$.

while $t \leq T$ **do**

  Step 1: $IS := FindIndistinguishableScenarios(t, \hat{x})$.
  Step 2: $\mu := 1$.

  while $\mu \leq |IS|$ **do**

    $\sigma := \Sigma_\mu$
    if $|\Sigma_\mu| = 1$ and $distinct\_scenario_\sigma = false$ **then**
      Step 2.1: $V' := \{i \in V | \hat{x}^\sigma_{im\tau} = 1$ for some $m \in \mathscr{M}_i, \tau < t\}$
      Apply $CPA_{b-DTCTP}$ with additional constraints in the relaxed problem:

$$y_{i\Pi_i} \geq 1 \ (i \in V') \tag{36.18}$$

      to get optimal values $\hat{y}^*_{im}$ for all $i \in V \backslash V'$. Use forward calculations to get the optimal starting times S_i^* for all $i \in V \backslash V'$.
      Set $\hat{x}^\sigma_{imt} := 1 \ (i \in V \backslash V'; m = \sum_{m' \in \mathscr{M}_i} m' y_{im'}; t = S_i^*)$.

    else
      Step 2.2: $\mathscr{D} = FindEligibleActivities(t, \hat{x}^\sigma, \Sigma_\mu)$
      Set $\hat{x}^\sigma_{imt} := 1 \ (\sigma \in \Sigma_\mu; \ i \in \mathscr{D}; m = \Pi_i)$.
    end if
    $\mu := \mu + 1$.
  end while
  Find the next decision point t.
end while

return $\hat{x}$.

In the worst case, the mode assignment, $\hat{x}(\Pi)$, obtained by $NONANT(\Pi)$ is exactly the same as prescribed by Π for all scenarios and no additional benefit is achieved by using the algorithm. In such a case, the solution value $z_{dyn}(\hat{x}(\Pi))$ is equal to the one obtained by solving the *b-SDTCTP* with static modes with policy Π, $z_{st}(x^\Pi)$.

Therefore, algorithm $NONANT(\Pi)$ guarantees to give a solution with expected project completion time at most as large as the one obtained from applying the solution x^Π to the *b-SDTCTP* with static modes:

$$z_{st}(x^\Pi) \geq z_{dyn}(\hat{x}(\Pi)) \tag{36.19}$$

Proposition 36.3. $UB \leq z_{MV}$.

Procedure 36.2 ***FindIndistinguishableScenarios(t,x)***

initialise
$NoOfSets := 1,\ IS := \{\emptyset\}.$

for $\sigma, \sigma' \in \Sigma,\ \sigma < \sigma'$ **do**
if $\sum_{(j,m')\in\Delta(\sigma,\sigma')} \sum_{\tau=0}^{t-\max\{p^{\sigma}_{jm'}, p^{\sigma'}_{jm'}\}} (x^{\sigma}_{jm'\tau} + x^{\sigma'}_{jm'\tau}) = 0$ **then**
if $\exists L \in \{1, NoOfSets\}$ such that $\sigma \in \Sigma_L$ **then**
$\Sigma_L := \Sigma_L \cup \{s'\}.$
else
$NoOfSets := NoOfSets + 1.$
$\Sigma_{NoOfSets} := \{\sigma, \sigma'\}.$
$IS := \{\Sigma_1, \ldots, \Sigma_{NoOfSets}\}.$
end if
end if
end for

return $IS.$

Procedure 36.3 ***FindEligibleActivities(t,x,Σ')***

initialise
$\sigma := \Sigma'_1,\ \mathscr{D} := \emptyset.$
$V' := \{i \in V | x^{\sigma}_{im\tau} = 1\ \forall \sigma \in \Sigma',$ for some $m \in \mathscr{M}_i,\ \tau < t\}.$
$\mathscr{G} := \{i \in V' | \sum_{\tau<t} (\tau + p^{\sigma}_{im}) x^{s}_{im\tau} \leq t,\ \forall \sigma \in \Sigma'\}.$

for $i \in V \backslash V'$ **do**
$pred_i := \{j \in V | (j,i) \in E\}.$
if $j \in \mathscr{G} \quad \forall\ j \in pred_i$ **then**
$\mathscr{D} := \mathscr{D} \cup \{i\}.$
end if
end for
Reorder $\mathscr{D}$ in non-decreasing order of index.

return $\mathscr{D}.$

Proof. Assume that the MV decision values, y_{MV}, represent the optimal solution to the *b-SDTCTP* with static modes. Then, $z_{MV} = z^*_{st} = z_{st}(y_{MV})$, where $z_{st}(y)$ is the solution value of the SDTCTP with static modes when values y are used and $z^*_{st} = \min_y\{z_{st}(y)\}$.

Now suppose that y_{MV} is not the optimal policy obtained from solving the *b-SDTCTP* with static modes. This means that there exists a mode selection $y^{\Pi} \neq y_{MV}$ such that $z_{st}(y^{\Pi}) \leq z_{MV}$. Note that y^{Π} is associated to a policy Π such that $\Pi_i = m$ if $y^{\Pi}_{im} = 1$ from (36.16). We also have that $z_{b-DM}(\hat{y}(\Pi)) \leq z_{st}(y^{\Pi})$ from (36.19) and the proof follows. Therefore, in general we have that:

$$UB \leq z_{MV} \tag{36.20}$$

□

The complexity of *HEUR* is mostly affected by two procedures, namely algorithms $CPA_{b-DTCTP}$ and $DA_{b-SDTCTP}$, implemented in Step 2.1 of Algorithm 36.1 and Step 1 of Algorithm 36.2, respectively. These two algorithms solve to optimality $\mathscr{NP}$-hard problems and are therefore of exponential complexity. However, the algorithms are not essential to the implementation of the fundamental idea of the heuristic and can be easily replaced by any other optimal or heuristic approach. In the absence of the exact solution approaches, the computational effort of the heuristic is expected to be mostly spent on Procedure 36.2, which performs a number of operations of order $T|\Sigma|^2 \sum_{i \in V} \mathscr{M}_i$. In the computational study which follows we show that experimentally the two exact solution approaches are quite fast and converge within reasonable computational times.

36.5 Illustrative Example

In this section, we will demonstrate the value obtained from considering different models to estimate the expected project completion time of a project example. Consider the activity-on-node project network shown in Fig. 36.1 and assume, for illustrative purposes, only five equiprobable scenarios given in Table 36.4 are used and the budget is equal to $b = 117$.

If we solve the deterministic model assuming all the random variables take their expected values (Mean-Value (MV) approach associated to scenario 1), the deterministic optimum project completion time is given by $z_{MV}^{det} = 23$. Using $x_{MV} = (1, 2, 4, 3, 3, 4, 3, 2, 1, 3, 3, 1)$, the expected project completion time, found by applying the solution to all five scenarios, is found to be $z_{MV} = 25.6$. Furthermore, the optimal static expected objective value is $z_{st}^* = 24.8$ obtained using the algorithm in Klerides and Hadjiconstantinou (2010).

Assuming dynamic modes for the example problem, we apply TS_A (Algorithm 36.1), which gives a lower bound equal to $LB = 24.2$ and *HEUR* (Algorithm 36.2) producing a heuristic solution value equal to $UB = 24.2$, thus proving optimality ($z_{dyn}^* = 24.2$). Full details of the algorithm are shown below. These results confirm that estimating the expected project completion time of the example without taking into account managerial flexibility leads to poor approximations. On the one hand, the expected objective value associated to x_{MV} overestimates the stochastic dynamic solution by at least 5.8 %. Similarly, the stochastic static model solution overestimates the corresponding dynamic solution by at least 2.5 %. As a result we can claim that the bound *UB* is a better approximation to the optimum stochastic dynamic objective value and provides an improvement over the optimum static value. In fact, in this case it provides the optimal solution.

Step 1: Use $DA_{b-SDTCTP}$ to derive optimal policy $\Pi = (1, 3, 4, 3, 3, 4, 3, 3, 1, 3, 2, 1)$ for activities $0, 1, \ldots, 11$.

Step 2: Use $NONANT(\Pi)$ to derive the heuristic non-anticipative solution $\hat{x}(\Pi)$ in the following way.
Initialize $distinct\_scenario_\sigma := \text{false}\ (\sigma \in \Sigma; \hat{x} := (0, \ldots, 0))$.

- $t = 0$. Apply procedure *FindIndistinguishableScenarios*$(0, \hat{x})$ to obtain set of indistinguishable scenarios $IS = \{\{1, 2, 3, 4, 5\}\}$.
- $n = 1$ and $\sigma = 1$.Apply procedure *EligibleActivities*$(0, \hat{x}^1, \{1, 2, 3, 4, 5\})$ to find the eligible set of start activities $\mathscr{D} = \{1, 2, 3, 4\}$. Set $\hat{x}^\sigma_{imt} := 1\ (\sigma = 1, \ldots, 5; i = 1, 2, 3, 4; m = 3, 4, 3, 3)$.
- $t = 5$ is the next decision point (activity 5 is eligible to start for $\sigma = 4$). Apply procedure *FindIndistinguishableScenarios*$(5, \hat{x})$ to obtain $IS = \{\{1\}, \{2, 5\}, \{3\}, \{4\}\}$.
- $n = 1$ and $\sigma = 1$. Since $|\Sigma_1| = 1$ and $distinct\_scenario_1 = false$, set the set of scheduled activities as $V' := 1, 2, 3, 4$ and apply $CPA_{b-DTCTP}$ with additional constraints in the relaxed problem:

$$y_{im} \geq 1\ (i = 1, 2, 3, 4; m = 3, 4, 3, 3) \tag{36.21}$$

  to obtain optimal modes $m_i = 4, 3, 3, 1, 3, 2, 1$ and optimal starting times $S_i^* = 6, 8, 9, 9, 15, 15, 23$ for unscheduled activities $i = 5, 6, 7, 8, 9, 10, 11$. Update $\hat{x}^1$ and set $distinct\_scenario_1 := true$.
- $n = 2$ and $\sigma = 2$. Apply procedure *FindEligibleActivities*$(5, \hat{x}^2, \{2, 5\})$, to find that there are no eligible activities at $t = 5$ for $\sigma = 2, 5$, i.e., $\mathscr{D} = \emptyset$.
- $n = 3$ and $\sigma = 3$. Since $|\Sigma_3| = 1$ and $distinct\_scenario_3 = false$, set $V' = 1, 2, 3, 4$ and apply $CPA_{b-DTCTP}$ with additional constraints (36.21) to get optimal modes $m_i = 4, 3, 3, 1, 3, 2, 1$ and optimal starting times $S_i^* = 7, 9, 9, 9, 13, 15, 22$ for unscheduled activities $i = 5, 6, 7, 8, 9, 10, 11$. Update $\hat{x}^3$ and set $distinct\_scenario_3 := true$.
- $n = 4$ and $\sigma = 4$. Following the same procedure as for $n = 1$ and $n = 3$, get optimal modes $m_i = 4, 3, 3, 1, 3, 2, 1$ and optimal starting times $S_i^* = 5, 10, 10, 10, 17, 17, 26$ for unscheduled activities $i = 5, 6, 7, 8, 9, 10, 11$. We update $\hat{x}^4$ and set $distinct\_scenario_4 := true$.
- $t = 6$ is the next decision point (activity 6 is eligible to start for $\sigma = 2$).
- Apply procedure *FindIndistinguishableScenarios*$(6, \hat{x})$ to obtain the set of indistinguishable scenarios $IS = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$.
- For $\sigma = 1, 3, 4$ we have that $distinct\_scenario_\sigma = true$.
- $n = 2$ and $\sigma = 2$.
- Apply $CPA_{b-DTCTP}$ with additional constraints (36.21) to obtain optimal modes $m_i = 4, 2, 2, 1, 4, 4, 1$ and optimal starting times $S_i^* = 8, 8, 10, 10, 17, 22, 25$ for unscheduled activities $i = 5, 6, 7, 8, 9, 10, 11$. Update $\hat{x}^2$ and set $distinct\_scenario_2 := true$.
- $n = 5$ and $\sigma = 5$.

- Apply $CPA_{b-DTCTP}$ with additional constraints (36.21) to obtain optimal modes $m_i = 4, 2, 2, 2, 3, 3, 1$ and optimal starting times $S_i^* = 7, 7, 12, 12, 18, 18, 25$ for unscheduled activities $i = 5, 6, 7, 8, 9, 10, 11$. Update $\hat{x}^5$ and set $distinct\_scenario_5 := true$.
- Since $distinct\_scenario_\sigma = true\, \sigma = 1, \ldots, 5$, exit procedure $NONANT(\Pi)$.

Step 3: Solve (36.2)–(36.8) using $\hat{x}(\Pi)$ and obtain $UB = 24.2$.

36.6 Computational Results

The study presented in this section addresses the computational aspects of implementing the algorithms proposed in this chapter and solving SDTCTPs with DM. In the first part of the study, we discuss the test set for the computational experiments, including the project network characteristics and the stochastic parameters of the test instances. In the second part, the performance of the lower and upper bounding techniques is evaluated with respect to several performance measures.

36.6.1 Test Problem Generation

The computational study presented in this section is based on a set of stochastic DTCTPs of varying network complexity and levels of uncertainty. We follow the same test problem generation procedure used in the computational study of Klerides and Hadjiconstantinou (2010).

The preliminary results of several computational experiments we performed led us to the following selection of representative deterministic benchmark instances. The deterministic instances, which are to be extended to stochastic instances, are taken from the test set used in Demeulemeester et al. (1998). Networks of 10 and 20 activities and 2, 4, and 6 modes per activity were selected. The activity durations/costs are randomly selected from two intervals $[1, 20]$ and $[1, 100]$, the network complexity assumes the relatively most difficult Coefficient of Network Complexity (CNC) (namely, CNC $= 2.1$) and the budget $b = b_{min} + \theta(b_{max} - b_{min})$ of the problem is obtained from two values for $\theta \in \{0.25, 0.5\}$.

For the scenario generation procedure, we consider a uniform probability distribution function, four values for the variance $(1, 5, 9, 13)$ and 5, 10, 20, or 50 scenarios.

Table 36.5 displays the complete set of problem parameters used in our computational study. Each combination of the parameters forms a problem class and ten instances were randomly generated in each class.

Table 36.5 Problem characteristics and stochastic parameters for the stochastic instances originating from Demeulemeester et al. (1998)

Parameter	Values
No. of activities	10, 20
No. of modes	2, 4, 6
Scale	[1, 20], [1, 100]
CNC	2.1
θ	0.25, 0.5
Distribution	*uniform*
Variance	1, 5, 9, 13
No. Of scenarios	5, 10, 20, 50
Allowed CPU time	1,000 s

Table 36.6 Computational results for TS_A and *HEUR*: effect of the variance

Variance	No. of problem instances	p_{opt}	$\Delta_{LB}^{\emptyset}$
1	120	55 %	0.19 %
5	120	51 %	0.33 %
9	120	46 %	0.59 %
13	120	48 %	0.47 %
All instances	Overall averages	50 %	0.39 %

36.6.2 Results

Algorithms TS_A and *HEUR* have been coded in Microsoft Visual Studio C++ and run on an Intel Core 2 processor (2.5 GHz with 3.5 GB of RAM, Windows Operating System) using CPLEX v11.1. Note that the imposed time limit was set to 1,000 CPU seconds.

Tables 36.6, 36.7, 36.8 provide information on the following performance indicators:

- p_{opt}: Percentage of problems for which the resulting lower bound *LB* equals the upper bound *UB*, thus finding the optimal solution;
- $\Delta_{LB}^{\emptyset}$: Average percentage deviation of the resulting upper bound *UB* from the lower bound *LB*, calculated as $100\% \cdot \frac{(UB-LB)}{LB}$;

36.6.2.1 Impact of the Variance

We first report on the results obtained from the computational implementation of TS_A and *HEUR* with respect to the variance. Table 36.6 displays the effect of the choice of variance on the computational performance of the proposed algorithms. Note that the results are based on the instances with ten scenarios; we additionally fix θ to take the value 0.5. Hence our results are based on solving $2\times3\times2\times10 = 120$ instances per value of variance.

We observe that the lower and upper bounds obtained seem be of better quality for the lowest value of the variance (1); the bounds obtained prove optimality for

55 % of the instances. The average percentage deviation of the upper bound from the lower bound is only 0.19 % when the variance equals 1 and it is on average 0.39 % over all instances solved.

36.6.2.2 Impact of Number of Scenarios

The number of scenarios to be considered by the mathematical model is expected to have a significant effect on the computational performance of the solution algorithm. This is due to the fact that the model size increases exponentially with the number of scenarios, as a result of the non-anticipativity constraints (36.6)–(36.7). For this reason, it is extremely important for the model to feature a number of scenarios which not only provides a good quality solution (good estimation of the true optimum) but also allows a solution to be obtained in reasonable computational time.

Table 36.7 shows the effect of the number of scenarios on the computational performance of TS_A and *HEUR*. Note that the instances assume a variance equal to 1 and $\theta = 0.5$. Hence, the results represent averages over $2 \times 3 \times 2 \times 10 = 120$ instances for each value chosen for a given number of scenarios.

As it can be observed from Table 36.7, the gap between the lower and upper bounds obtained becomes smaller as the number of scenarios increases. However, the lower bound equals the upper bound more often (71 % of the instances) for models with five scenarios, compared to 55 %, 40 % and 29 % of the instances with 10, 20, and 50 scenarios, respectively.

36.6.2.3 Impact of Network Characteristics

Overall, the lower bound obtained in Sect. 36.4.2 equals the upper bound proposed in Sect. 36.4.3 for 58 % of the instances (Table 36.8). On average, the obtained upper bound deviates by 0.26 % from the lower bound. The greatest effect on the bounding techniques is noted by the choice of the scale of the activity durations and costs. Increasing the scale from $[1, 20]$ to $[1, 100]$, significantly reduces $\Delta_{LB}^{\emptyset}$, the average percentage deviation of the upper bound from the lower bound. Our bounding techniques perform particularly well for larger scales of the activity durations,

Table 36.7 Computational results for TS_A and *HEUR*: effect of the number of scenarios

No. of scenarios	No. of problem instances	p_{opt}	$\Delta_{LB}^{\emptyset}$
5	120	71 %	0.24 %
10	120	55 %	0.19 %
20	120	40 %	0.13 %
50	120	29 %	0.11 %
All instances	Overall averages	49 %	0.19 %

Table 36.8 Computational results for TS_A and *HEUR*: effect of the network characteristics

Problem Parameters	No. of problem Instances	p_{opt}	$\Delta_{LB}^{\emptyset}$
Activities			
10	120	68 %	0.24 %
20	120	47 %	0.29 %
Modes			
2	80	78 %	0.16 %
4	80	63 %	0.20 %
6	80	33 %	0.45 %
Activity durations/costs			
[1,20]	120	42 %	0.49 %
[1,100]	120	73 %	0.05 %
θ			
0.25	120	60 %	0.24 %
0.5	120	55 %	0.29 %
All instances	Overall averages	58 %	0.26 %

proving optimality for 73 % of the instances; optimality is proven for only 42 % of instances with scales ranging between $[1, 20]$.

36.7 Conclusions

In this chapter, we have introduced the Stochastic DTCTP with Dynamic Modes, SDTCTP with DM. As opposed to the SDTCTP with Static Modes, this problem offers flexibility to the project manager to adjust the mode selection variables according to observations. This flexibility, however, comes at a cost in terms of model complexity. The model suffers from decision-dependent uncertainty, which makes even the mere formulation of the non-anticipativity constraints rather cumbersome.

The chapter describes the lower and upper bounding techniques that we have developed for the SDTCTP with DM. Our extensive computational results on a large number of test instances of varying size and degree of uncertainty show the effectiveness of the bounds. It is expected that including such strong bounds would bring significant improvements to the performance of exact solution methodologies. We are currently in the process of developing a branch-and-bound methodology that uses these bounds in order to solve to optimality SDTCTPs with DM.

References

Ahmed S (2000) Strategic planning under uncertainty: stochastic programming approaches. Ph.D. dissertation, University of Illinois, Urbana-Champaign, IL, USA

Colvin M, Maravelias CT (2010) Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. Eur J Oper Res 203:205–215

Demeulemeester EL, De Reyck B, Foubert B, Herroelen WS, Vanhoucke M (1998) New computational results on the discrete time/cost trade-off problem in project networks. J Oper Res Soc 49(11):1153–1163

Godinho P, Branco F (2012) Adaptive policies for multi-mode project scheduling under uncertainty. Eur J Oper Res 216:553–562

Goel V, Grossmann IE (2004) A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. Comput Chem Eng 28(8):1409–1429

Goel V, Grossmann IE (2007) A class of stochastic programs with decision dependent uncertainty. Math Program Ser B 108:355–394

Gutjahr W (2000) A stochastic branch-and-bound approach to activity crashing in project management. INFORMS J Comput 12(2):125

Hadjiconstantinou E, Klerides E (2010) A new path-based cutting plane approach for the discrete time-cost tradeoff problem. Comput Manag Sci 7(3):313–336

Held H, Woodruff D (2005) Heuristics for multi-stage interdiction of stochastic networks. J Heuristics 11(5–6):483–500

Higle JL (2005) Stochastic programming: optimization when uncertainty matters. INFORMS, New Orleans

Jonsbraten TW, Wets RJB, Woodruff DL (1998) A class of stochastic programs with decision dependent random elements. Ann Oper Res 82:83–106

Klerides E, Hadjiconstantinou E (2010) A decomposition-based stochastic programming approach for the project scheduling problem under time/cost trade-off settings and uncertain durations. Comp Oper Res 37(12):2131–2140

Laslo Z (2003) Activity time-cost tradeoffs under time and cost chance constraints. Comput Ind Eng 44(3):365–384

Pflug G (1990) On-line optimization of simulated markovian processes. Math Oper Res 15:381–395

Solak S (2007) Efficient solution procedures for multistage stochastic formulations of two problem classes. Ph.D. dissertation, Georgia Institute of Technology, Atlanta, Georgia, USA

Tarhan B, Grossmann IE, Goel V (2009) Stochastic programming approach for the planning of offshore oil or gas field infrastructure under decision-dependent uncertainty. Ind Eng Chem Res 48:3078–3097

Vishwanath K, Peeta S, Salman S (2004) Investing in the links of a stochastic network to minimize expected shortest path length, working paper

Wollmer RD (1985) Critical path planning under uncertainty. Math Program Stud 25:164–171

Zhu G, Bard JF, Yu G (2007) A two-stage stochastic programming approach for planning with uncertain activity durations. J Sched 10:167–180

Chapter 37
The Stochastic Resource-Constrained Project Scheduling Problem

Maria Elena Bruni, Patrizia Beraldi, and Francesca Guerriero

Abstract Resource-constrained project scheduling has been widely investigated in the academic literature, but the issue of the incorporation of uncertainty in project scheduling has received a growing research attention only in the last 15 years. This chapter gives an overview of models and methods for the resource-constrained project scheduling under uncertainty. The case of known deterministic renewable resource requirements and random activity durations with a known probability distribution function is studied in detail. In particular, we show how, through the use of joint probabilistic constraints, a feasible baseline schedule with minimum makespan can be built, which is able to tolerate a certain degree of uncertainty and to absorb dynamic variations in activity durations. The use of joint probabilistic constraints, within the stochastic scheduling problem, represents an innovative element in the literature and enables the relaxation of the common assumption that only one activity at a time disturbs the starting time of a successor activity, rather limiting the joint probability of disruption of the preceding activities to a given probability level. The results obtained with the proposed heuristics are discussed and compared with two well known heuristics taken from the literature on a set of randomly generated project instances. A practical application concerning a real project for construction of students' apartments at the University of Calabria, Italy, is also illustrated. Based on the analysis of the various researches discussed in this chapter, avenues for future research will be also outlined.

Keywords Chance constraints • Project scheduling • Random variables • Stochastic RCPSP • Stochastic scheduling

M.E. Bruni (✉) • P. Beraldi • F. Guerriero
Department of Mechanical, Energy and Management Engineering, University of Calabria, Rende (CS), Italy
e-mail: mariaelena.bruni@unical.it; patrizia.beraldi@unical.it; francesca.guerriero@unical.it

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_7

37.1 Introduction

Most of the literature that studies the resource-constrained project scheduling problem RCPSP assumes complete information on the resource usage and activity durations, and determines a feasible baseline schedule, i.e., a list of activity starting times, minimizing the makespan value (see Chap. 1 in the first volume of this handbook). Interesting extensions of the RCPSP involve the introduction into the problem of randomness that is present to a some extent in all projects. In a construction project, for example, a plethora of adverse events (worker sickness, weather, delayed parts delivery, unforeseen difficulty of tasks) may produce a schedule shifting and a raise in costs.

The stochastic RCPSP (SRCPSP, for short) tries to incorporate in the formulation itself the uncertainty by expressing the random parameters of the problem as random variables. The resulting stochastic problem is challenging from both a theoretical and computational point of view. The first issue concerns the genuine interpretation of the RCPSP under uncertainty and the way this uncertainty is tackled. The problem can be viewed as a stochastic dynamic optimization problem, where decisions are made each time new information becomes available. We observe that in the terminology used in the robust project scheduling literature (see Chap. 40) this can be intended as a purely reactive approach.

For some projects, a baseline plan should be agreed long before the project starts. In these cases, if we interpret the RCPSP under uncertainty as a problem where a tentative plan—which can be changed during project execution—is determined before knowing the realization of uncertainty, we have in fact a standard stochastic program with recourse. This means that before the project is started, we can make investments and scheduling decisions and, afterward, we shall interfere in order to fix shortcomings. This can be thought of a proactive-reactive approach, in which a robust initial schedule is built, by following a proactive scheduling strategy whereas reactive procedures revise or re-optimize the baseline schedule.

If, on the contrary, the project manager has little opportunity to change the baseline plan (milestone dates are referenced in contracts and penalty clauses applied if missed), the activity schedule should be as reliable as possible and deviations should be avoided to the maximum extent. In this case, the chance constraints framework can be a useful mathematical toolkit to address the problem, which becomes purely proactive. Whereas recourse models transform the randomness into a risk neutral probability measure, namely the expectation, chance-constrained models deal more explicitly with the distribution itself.

The last approach, which will not be treated in this chapter, is related to what is known as the wait-and-see solution in the field of stochastic programming. It represents a situation where all uncertainty will be resolved before the project starts and decisions can be postponed until this point in time.

Unfortunately, there is no generally valid justification to state that any one of these approaches is more appropriate. It depends on the applicative context at hand and on the project manager's risk aversion. Although very different, and regardless

the specific paradigm adopted, all the approaches presented cannot overlook the importance of an adequate treatment of uncertainty.

A significant amount of theoretical work has been done to conceptualize and measure uncertainty in project scheduling (Jaafari 2001; Perminova et al. 2008), but even the most proficient managers have difficulty handling it. They use decision milestones to anticipate outcomes, risk management to prevent disasters but too often the project ends up with an overrun schedule or overflowing budget. As a matter of fact, whilst project risk management has become a well established and widely used project management method (see Chap. 49), uncertainty has been seldom if ever, dutifully incorporated into a scheduling/project management system providing insight on preventive and contingent actions to hedge against uncertainty. Project managers are familiar with risk management thank also to a large repertoire of tool supporting important analytical activities. However, the same tools do not explicitly address uncertainty in the construction of a schedule. The lack of practitioner's knowledge on adequate answers to uncertainty in RCPSP is surrounded by a relatively sparse scientific interest for this hot topic. In effect, whilst RCPSPs have been widely investigated in the academic literature, the issue of the incorporation of uncertainty in project scheduling has received a growing research attention only in the last 15 years.

We should mention that, while probability theory can be a powerful tool in most circumstances, sometimes the type of uncertainty encountered in particular classes of projects does not fit the axiomatic basis of probability theory simply because uncertainty is related to fuzziness rather than to randomness. In this chapter we will review only the former type of uncertainty, referring the reader to Chaps. 41 and 42 of this book for other kind of approaches.

In the next section we shall review the literature on the SRCPSP. The Sect. 37.3 is devoted to the presentation of a chance-constrained heuristic for the SRCPSP, whereas in Sect. 37.4 an application of the SRCPSP to construction projects is presented. Finally, Sect. 37.5 conclusions are given and avenues for future research discussed.

37.2 Literature Review

The literature concerning the SRCPSP has almost neglected the full range of sources of uncertainty associated with a project (Atkinson et al. 2006), rather focusing on the most obvious aspect concerning the estimates of potential variability of activities duration. Besides this, uncertainty might be related to resources availability, for possible breakdowns or simply temporary shortage and/or resource consumption. Furthermore, new activities may be incorporated in the project or other activities may be even deleted (Lambrechts et al. 2007).

In this section, we will review only the approaches that consider activity duration uncertainty, bearing in mind that different sources of uncertainty may affect the RCPSP and leaving the discussion to the Sect. 37.5 of this chapter.

In methodologies for stochastic project scheduling with uncertain durations, the scheduling problem is mostly viewed as a stochastic dynamic optimization problem with the expected makespan being the most studied objective. Starting times for the activities are defined through a multi-stage decision process also called policies. A policy is a dynamic decision process that defines, at completion of some activity, appropriate actions concerning the choice of a set of activities that should be executed next. The activities that can be started at decision points are such that neither precedence nor resource constraints are violated. In deterministic scheduling, this strategy is well-known as parallel list scheduling scheme (see Chap. 1 in the first volume of this handbook). Any policy can be viewed as a dynamic decision process (as we explained before) or as a function. In fact, a policy transforms a combinatorial object (represented by an ordered list of activities) into a schedule. Radermacher (1981) used this view to formally define a priority policy, which schedules activities according to a given priority order. At any decision point, the maximum number of activities, amongst the set of activities not yet started and with all its predecessors completed, are scheduled following the priority given. While priority policies are easy to define and implement, they have been neglected since they may lead to anomalies (Graham 1966) related to the possibility of increasing project duration due to decreasing activity durations.

To circumvent this drawback, preselective policies have been studied by Igelmund and Radermacher (1983a,b). Germane to these policies is the concept of minimal forbidden set defined as the minimum cardinality set of activities, without precedence constraints, whose total resource consumption exceeds the resource availability. A preselective policy defines for each minimal forbidden set a (preselected) activity to be postponed in order to solve potential resource conflicts. This activity is subject to a waiting condition since it is not executed until at least one activity in the forbidden set has been completed. A preselective policy can be then represented by a collection of partially ordered sets, which extends the partial order of precedence constraints inducing a digraph which has a node for each activity and for each waiting condition.

Möhring and Stork (2000) following Igelmund and Radermacher (1983a) studied the so-called linear preselective policies, which combine priority and preselective policies. In effect, an activity is selected to be delayed and the choice respects the order imposed by a priority list in such a way that, in a forbidden set, the activity dominating (with respect to the prespecified order) all the activities belonging to the same set is chosen. Since each linear preselective policy is also a preselective policy, linear preselective policies inherit the analytic properties of being monotone and continuous. These properties were exploited by Stork (2001) to develop a branch-and-bound procedure equipped with dominance rules and different branching schemes to efficiently compute an optimal preselective policy. The lower bound is computed by approximating the expected makespan with the help of simulation techniques. The branch-and-bound was also tested on the class of so called activity-based policies, that have the nice feature of avoiding the handling of the minimal forbidden sets. In particular, an activity-based policy is represented by a priority list of the activities and starts each activity in the order imposed in the

list as early as possible, considering the side constraint that the starting time of any activity should be greater than the starting time of any dominated (with respect to the ordering criterion selected) activity. Since these activity-based policies perform activity incrementation rather than time incrementation, they are the stochastic counterpart of the serial schedule-generation scheme described in Chap. 1 in the first volume of this book.

More recently, a hybrid proactive-reactive approach has been proposed in Ashtiani et al. (2011), where a new class of policies, called preprocessor policies, is defined, which make a-priori sequencing decisions resolving some, but not necessarily all, resource conflicts in a preprocessing phase, while the remaining conflicts are dynamically resolved during project execution. In particular, a preprocessor policy is defined by a set of activity pairs (which adds extra precedence relations between activities) and an ordered list used by a priority based policy to solve conflicts during project execution. In order to find a good list and also good activities pairs, genetic algorithms are applied, able to obtain competitive results for the SRCPSP.

Policies have also been used for determining predictive activity starting times, with the objective of minimizing costs related to positive and negative deviations of actual starting times, from the predicted ones, and to penalties/bonuses associated with late/early project completion. For this problem, Deblaere et al. (2007) proposed a solution procedure that generates an initial policy then improved through a descent procedures. The vector of predictive starting times is then determined by a methodology combining insights from the newsvendor problem and simulation. The whole procedure heavily relies on simulation.

Golenko-Ginzburg et al. (1997) presented a heuristic algorithm for the SRCPSP with random activities duration and the expected makespan as objective. The heuristic shares, with the scheduling policies reviewed above, the use of decision points at which starting times are determined. If, at a certain point of time, a resource conflict arises, amongst the set of activities that can be feasibly started, a competition among the activities is carried out by solving a zero-one integer programming problem. The problem aims at maximizing the total contribution of the accepted activities to the expected project duration, where such contribution is defined as the product of the average duration of the activity and its probability of being on the critical path, calculated via simulation.

The same authors (Golenko-Ginzburg et al. 1998) extended the case by considering random durations dependent on the resource allocation. Their purpose was to find both the activities starting times and the optimal allocation of resources in order to minimize expected project duration. The problem is more involved than before, since the probability density functions of the random durations do not remain unchanged in the course of the project realization, but they depend on the resource allocation. Four heuristics are then proposed in order to circumvent this issue and both exact and heuristic algorithms devised to address the knapsack resource allocation problem set up to solve resource conflicts. Starting from the concept of critical chain introduced by Goldratt (1997), in Rabbani et al. (2007) a new heuristic was presented, implementing backward pass scheduling for feeding-in resources, with the objective of minimizing the expected project duration and its

variance. Similarly to the work of Golenko-Ginzburg et al. (1997), the solution of a zero-one integer programming approach is suggested to allocate the resources, considering that the activities with the greatest probability to be on the critical chain and the greatest correlation with the project variance are fed-in first. After the resource allocation has been completed, the most critical chain and its standard deviation are calculated on the basis of criticality probability of each chain. Finally, a project buffer, proportional to the standard deviation of the project duration, is added to the end of the most critical chain.

Tsai and Gemmil (1998) proposed a tabu search based heuristic, which uses multiple tabu lists, randomized short-term memory, and multi start diversification mechanism. Later, Ballestín (2007) developed regret-based biased random sampling procedures then embedded into a genetic algorithm, whereas Ballestín and Leus (2009) proposed a greedy randomized adaptive search procedure capable of outperforming other heuristic algorithms in the literature.

For a special case involving only one renewable resource (the budget) a two-stage integer linear stochastic program has been proposed in Zhu et al. (2007), where target times are determined in the first stage, whereas in the second stage a deviation from this target is allowed. The objective is to balance the cost of project completion as a function of activity target times with the expected penalty related to possible deviations from the target.

The authors of this chapter, in Bruni et al. (2011a), proposed a chance-constrained based heuristic aiming at building a baseline schedule which is protected against possible disruptions. This work brings its own originality for the fact that it has as a focal point the machinery of chance constraints and for the proactive point of view, rather unusual in the stochastic scheduling literature. The next section will be devoted to a detailed description of this method and to a presentation of a summary of the computational results obtained.

37.3 A Chance-Constrained Based Heuristic

The method proposed in this section separates the dynamic from the stochastic aspects of the problem, considering the project as a sequence of stages where uncertainty is resolved by means of an anticipative static policy. According to this approach, the stochastic programming framework, in the form of probabilistic constraints, is used to take anticipative decisions in the planning phase and not during project execution.

To the best of our knowledge, none of the methods proposed in the literature considers joint probabilistic constraints. Our work differs from the cited papers in some important aspects. First of all, we consider the stochastic programming framework and, in particular, the probabilistic paradigm in the form of joint probabilistic constraints. This powerful tool allows us to relax the assumption, common in the literature, that only one activity at a time disturbs the starting time of a successor activity, rather limiting the joint probability of disruption of

the preceding activities to a given probability level. Secondly, our point of view is rather unusual in the literature on stochastic project scheduling since our aim is the construction of a proactive schedule under uncertainty, partially bridging the gap between the stochastic scheduling literature and the robust one.

We assume that the activity network of a SRCPSP is given by a directed acyclic graph $G = (V, E)$.

We assume the presence of a set $\mathscr{R}$ of K renewable resources with a per-period availability R_k. Each activity $i \in V$ has to be processed without interruptions, requiring a constant amount of resource r_{ik} for each renewable resource type $k \in \mathscr{R}$. We assume that activities durations are represented by random variables $\tilde{p}_i, \forall i \in V$, defined on a given probability space Ω equipped with an algebra $\mathscr{F}$ and with a probability measure P. The random vector of starting times is denoted as $\tilde{S} = (\tilde{S}_0, \ldots, \tilde{S}_{n+1})$.

In uncertain environments, especially from a practical point of view, project managers are mainly interested in the generation of a proactive schedule (i.e., a vector of proactive starting times S and proactive completion times C) with a quality that does not degrade during execution with respect to future perturbations. In fact, during the execution of a project, the realized starting time of an activity may be different from its predictive starting time. The corresponding deviation is still a random variable defined for a generic activity i as follows:

$$\tilde{\Delta}_i = \begin{cases} \tilde{S}_i - S_i, & \text{if } \tilde{S}_i - S_i > 0 \\ 0, & \text{otherwise} \end{cases}$$

A natural question is how to construct a schedule, with associated vectors of predictive starting and completion times, that attempts to limit the risk of such deviation. Two classical approaches can be used to deal with random deviations. Unit penalty costs can be assigned for each individual deviation, and the resulting expected penalty cost can be minimized, or alternatively, one may specify a model in which the risk of deviations is accepted with a certain probability.

From a mathematical standpoint, risk averse constraints can be formulated by using the theory of joint probabilistic constraints as in the sequel:

$$P(\tilde{\Delta}_i > 0, \quad (i \in V)) \leq \epsilon \tag{37.1}$$

This probabilistic constraint limits from above the probability of a schedule disruption by the parameter $\epsilon \in [0, 1]$.

Though appealing, determining a vector of predictive starting times such that the makespan of the schedule is minimized and constraint (37.1) is fulfilled is cumbersome from both a theoretical and a computational point of view. Hence, rather than tackling the full complex stochastic dynamic problem presented above, the problem is decoupled and the dynamic and the stochastic aspects handled separately, reconciling their benefits through the use of stochastic information. The ultimate objective is still the minimization of the project makespan.

Decisions to be made concern which activities to start at certain decision points by taking into account the resources availability. A decision point occurs either at the beginning of the project, or when at least one of the running activities is completed, until the last activity is scheduled. At each decision point, a resource-feasible partial schedule is built and suitable proactive starting and completion times are set by means of an anticipative stochastic model that accounts for future uncertainty. Since the policy we use can be viewed as a stochastic dynamic version of the parallel schedule-generation scheme, it is easy to verify that the partial schedule constructed is feasible with respect to precedence and resource constraints.

A detailed description of the stochastic dynamic generation scheme (SDGS) heuristic is given in what follows. Let us introduce the required notation:

- μ the iteration counter;
- t_μ the decision time associated with iteration μ;
- $\mathscr{C}_\mu$ the set of activities already scheduled and completed up to t_μ;
- $\mathscr{A}_\mu$ the set of activities, which are active at t_μ;
- $\mathscr{D}_\mu$ the set of activities whose predecessors have been completed at time t_μ;
- $\mathscr{D}_\mu^{'}$ a subset of $\mathscr{D}_\mu$ containing activities that will start at time t_μ;
- $R_k(t_\mu)$ the residual resource availability at time t_μ;
- ρ a priority rule.

An algorithmic description of the SDGS heuristic is given below.

Algorithm 37.1 Scheme of the SDGS Heuristic

Initialization
Set $\mu := 0$, $t_0 := 0$, $\mathscr{C}_0 := \emptyset$, $\mathscr{D}_0 := \emptyset$, $\mathscr{A}_0 := \{0\}$, $\mathscr{D}_0^{'} := \{0\}$, $C_0 = t_0$, $S_0 = t_0$
$R_k(t_0) := R_k, k \in \mathscr{R}$
Choose a priority rule ρ
repeat
  $\mu := \mu + 1$
  $t_\mu := min_{j \in \mathscr{A}_{\mu-1}} C_j$
  $\mathscr{C}_\mu := \mathscr{C}_{\mu-1} \cup \{i \in \mathscr{A}_{\mu-1} | C_i \leq t_\mu\}$
  $\mathscr{A}_\mu := \{i \in V | C_i > t_\mu\}$
  $\mathscr{D}_\mu := \{i \in V \setminus (\mathscr{C}_\mu \cup \mathscr{A}_\mu) | Pred(i) \subseteq \mathscr{C}_\mu\}$
  $\mathscr{D}_\mu^{'} := \emptyset$
  $R_k(t_\mu) := R_k - \sum_{i \in \mathscr{A}_\mu} r_{ik}$
  while $\mathscr{D}_\mu \neq \emptyset$ **do**
    Use the priority rule ρ to select a new activity $i \in \mathscr{D}_\mu$ to be scheduled
    $\mathscr{D}_\mu := \mathscr{D}_\mu \setminus \{i\}$
    if i is such that $R_k(t_\mu) - r_{ik} \geq 0$ **then**
      $\mathscr{D}_\mu^{'} := \mathscr{D}_\mu^{'} \cup \{i\}$
      $\mathscr{A}_\mu := \mathscr{A}_\mu \cup \{i\}$
      $R_k(t_\mu) := R_k(t_\mu) - r_{ik}$
      $S_i := t_\mu$
    end if
  end while
  Determine the proactive completion times C_i, $\forall i \in \mathscr{D}_\mu^{'}$
until $|\mathscr{C}_\mu \cup \mathscr{A}_\mu| = n + 1$

Decisions concerning the appropriate selection of activities to start $\mathscr{D}'_\mu$ and their proactive completion times should ensure the satisfaction of the stability constraint (37.1) accounting for the potential impact of a delay. At a generic instant $t_\mu \leq t \leq t_{\mu+1}$, the probability of not causing a disruption in the schedule in the future is the probability that for any activity i currently under execution the condition $C_i \geq \tilde{C}_i$ is verified. In order to enforce this condition, at each decision point t_μ the following problem with joint chance constraints is solved:

$$\text{Min. } C^\mu_{max} \tag{37.2}$$

$$\text{s.t.} \quad C^\mu_{max} \geq C_i \quad (i \in \mathscr{D}'_\mu) \tag{37.3}$$

$$P\left(C_i \geq t_\mu + \tilde{p}_i \quad (i \in \mathscr{D}'_\mu)\right) \geq (1-\epsilon) \tag{37.4}$$

where C^μ_{max} represents the makespan of the partial schedule built considering activities $i \in \mathscr{D}'_\mu$ and $t_\mu + \tilde{p}_i = \tilde{C}_i \ \forall i \in \mathscr{D}'_\mu$. Here joint chance constraints are imposed to set the completion time of the activities, at each decision point, in such a way that the probability of not disrupting the schedule in the future is at least $(1-\epsilon)$ (i.e., the risk of disruption is at most ϵ).

We should point out that if, at least in principle, separate chance constraints can be used to deal with uncertain durations, the solution provided by the corresponding model may in some context be considered inappropriate. In fact, imposing a small probability of disruption for each activity $i \in \mathscr{D}'_\mu$ does not assure a small joint probability for all $i \in \mathscr{D}'_\mu$.

It is worth observing that, although at each decision point we accept the risk of a disruption with probability ϵ, we cannot impose a limit on the probability of not completing the whole project on time. However, we may express the timely project completion probability as a function of the number of decision points performed by the algorithm. A crude lower bound for the probability of project to be completed on time is $(1-\epsilon)^\nu$ where ν is an upper bound on the number of iterations. In order to show how the heuristic works, we report an illustrative example with only five activities plus the two dummy activities 0 and 6 (see Fig. 37.1). Only one resource

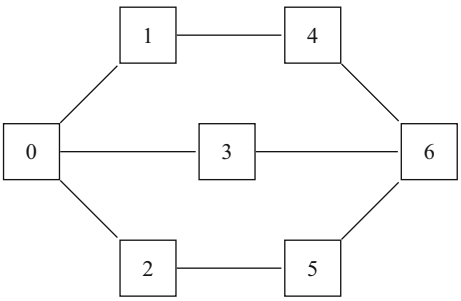


Fig. 37.1 Toy example

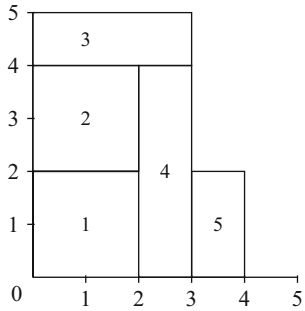


Fig. 37.2 Resulting schedule

is required to execute the activities (i.e., $K = 1$) and the resource consumptions are: $r_{11} = 2$, $r_{21} = 2$, $r_{31} = 1$, $r_{41} = 1$, $r_{51} = 2$, and $R_1 = 5$. In addition, activities durations follow a Poisson distribution with mean 1.5 for activities $1, 2, 3$, and 1 for activities 4 and 5. The value of ϵ is fixed to 0.45.

The activities have been ordered by the rule ρ as follows: $(1, 2, 3, 4, 5)$.

After the dummy activity 0 has been scheduled, activities $1, 2, 3$ can be scheduled at time $t_1 = 0$ ($\mathscr{D}_1^{'} = \{1, 2, 3\}$) since they do not cause any resource conflict. Therefore $S_1 = S_2 = S_3 = 0$. Problem (37.2)–(37.4) is then invoked and completion times C_1, C_2, and C_3 are appropriately set. The next decision point is $t_2 = 2$. At that time, activity 4 is an eligible activity and there are sufficient resources units available, so it is started. For activity 5 no sufficient resource units are available. Therefore $\mathscr{D}_2^{'} = \{4\}$, and problem (37.2)–(37.4) reduces to a problem with a single chance constraint. The next decision point is $t_3 = 3$. At this time activity 5 is started.

The resulting feasible schedule is depicted in Fig. 37.2. We note that the disruption probability of activity 4 depends on the disruption probability of activities 1 and 2, whereas it is not influenced by the completion time of activity 3. By imposing a threshold risk parameter of $\epsilon = 0.45$ our heuristic set completion times of activities $1, 2, 3$ in such a way that

$$P \left\{ \begin{array}{l} C_1 \geq \tilde{C}_1 \\ C_2 \geq \tilde{C}_2 \\ C_3 \geq \tilde{C}_3 \end{array} \right\} \geq 0.55$$

thus limiting the disruption probability of activity 4.

37.3.1 Solving the Joint Probabilistically Constrained Problem

The SDGS heuristic involves the repeated solution of model (37.2)–(37.4). In the following, we show how to derive a deterministic equivalent formulation, in the case of independent random variables.

Under the independence assumption among the random variables $\tilde{p}_i$, the probabilistic constraints (37.4) can be rewritten as

$$\prod_{i \in \mathscr{D}'_\mu} P\left(C_i \geq t_\mu + \tilde{p}_i\right) \geq (1 - \epsilon) \tag{37.5}$$

Denoting with F_i the marginal probability distribution function of the random variable $\tilde{p}_i$, and with a variable substitution $x_i = C_i - t_\mu$, constraints (37.5) can be stated equivalently as

$$\prod_{i \in \mathscr{D}'_\mu} F_i(x_i) \geq (1 - \epsilon) \tag{37.6}$$

and by taking logarithms:

$$\sum_{i \in \mathscr{D}'_\mu} \ln F_i(x_i) \geq \ln(1 - \epsilon) \tag{37.7}$$

(see, for example, Miller and Wagner 1965; Jagannathan 1974). Since the logarithm is an increasing function and $0 < F_i \leq 1$, this transformation leads to an equivalent condition. Furthermore, for log-concave distribution functions (including several commonly used probability distributions as for example the Uniform, Normal, Exponential, and many others, see Prékopa 1995 and Dentcheva et al. 1998) convexity of the constraints is preserved. We observe that also in the case of discrete distributions, problems with joint probabilistic constraints can be reduced to deterministic equivalent problems. For more details, the interested readers are referred to Dentcheva et al. (1998). Therefore, depending on the continuous or discrete nature of the random variables involved in the problem, the deterministic equivalent problem takes the form either of a nonlinear continuous problem or a linear integer problem.

We will show how deterministic equivalents are built for the example network of Fig. 37.1. Let us first consider the case of a discrete distribution. From Eq. (37.7) if $l_i + q_i$ is a known upper bound, where l_i represents the $(1 - \epsilon)$ quantile of the marginal distribution F_i, that is, the smallest integer value such that $F_i(x) \geq (1-\epsilon)$, it is evident that x_i can be written as $x_i = l_i + \sum_{q=1,\ldots,q_i} x_{iq}$ where x_{iq} are binary variables. Therefore, the probabilistic constraints can be rewritten as

$$\sum_{i \in \mathscr{D}'_\mu} \sum_{q=1,\ldots,q_i} a_{iq} x_{iq} \geq b \tag{37.8}$$

where $a_{iq} = \ln F_i(l_i + q) - \ln F_i(l_i + q - 1)$ and $b = \ln(1 - \epsilon) - \ln F(l)$.

In the considered case:

$$x_1 = l_1 + \sum_{q=1}^{5} x_{1q} = 1 + x_{11} + x_{12} + x_{13} + x_{14} + x_{15}$$

$$x_2 = l_2 + \sum_{q=1}^{5} x_{2q} = 1 + x_{21} + x_{22} + x_{23} + x_{24} + x_{25}$$

$$x_3 = l_3 + \sum_{q=1}^{5} x_{3q} = 1 + x_{31} + x_{32} + x_{33} + x_{34} + x_{35}$$

Therefore, the joint probabilistic constraints can be transformed in the following mixed-integer problem.

$$\begin{aligned}
&\text{Min. } C^1_{max} \\
&\text{s.t. } C^1_{max} \geq 1 + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} \\
&\quad C^1_{max} \geq 1 + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} \\
&\quad C^1_{max} \geq 1 + x_{31} + x_{32} + x_{33} + x_{34} + x_{35} \\
&\quad a_{11}x_{11} + a_{12}x_{12} + a_{13}x_{13} + a_{14}x_{14} + a_{15}x_{15} + a_{21}x_{21} + a_{22}x_{22} + a_{23}x_{23} + a_{24}x_{24} + \\
&\quad + a_{25}x_{25} + a_{31}x_{31} + a_{32}x_{32} + a_{33}x_{33} + a_{34}x_{34} + a_{35}x_{35} \geq \ln(1-\epsilon) - \ln((1-\epsilon)^3) \\
&\quad x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{31}, x_{32}, x_{33}, x_{34}, x_{35} \in \{0, 1\} \\
&\quad C_{max} \geq 0
\end{aligned}$$

If we instead suppose that $\tilde{p}_1$, $\tilde{p}_2$, and $\tilde{p}_3$ follow a Uniform distribution $U_1[a_1, b_1]$, $U_2[a_2, b_2]$, and $U_3[a_3, b_3]$, respectively, the joint probabilistic constraints can be transformed in the following nonlinear problem.

$$\begin{aligned}
&\text{Min. } C^1_{max} \\
&\text{s.t. } C^1_{max} \geq C_1 \\
&\quad C^1_{max} \geq C_2 \\
&\quad C^1_{max} \geq C_3 \\
&\quad \ln F_1(C_1) + \ln F_2(C_2) + \ln F_3(C_3) \geq \ln(1-\epsilon)
\end{aligned}$$

or equivalently

$$\begin{aligned}
&\ln\left(\frac{\chi_1 - a_1}{b_1 - a_1}\right) + \ln\left(\frac{\chi_2 - a_2}{b_2 - a_2}\right) + \ln\left(\frac{\chi_3 - a_3}{b_3 - a_3}\right) \geq \ln(1-\epsilon) \\
&C_1, C_2, C_3, C_{max} \geq 0
\end{aligned}$$

37.3.2 Computational Experiments

In this section, with the aim of assessing the performance of the SDGS heuristic, some computational experience will be presented together with a comparative evaluation with a set of benchmark heuristics. These are modifications of both parallel schedule generation schemes (PSGS) and serial schedule generation schemes (SSGS), in which the deterministic durations are replaced by their $(1-\epsilon)$-quantile counterparts ($p_{1-\epsilon}$). We should remark that this is equivalent of using separate chance constraints within classical schedule generation heuristics for the deterministic RCPSP. Different static priority rules for generating the priority list were tested:

1. (MaxC) The MaxC rule orders the activities $j \in V$ by decreasing value of their total resource requirement $r(j) = \sum_{k=1}^{K} r_{jk}$.
2. (MinC) The MinC rule orders the activities $j \in V$ by increasing value of their total resource requirement $r(j) = \sum_{k=1}^{K} r_{jk}$.
3. (MaxD*C) The MaxD*C rule orders the activities by decreasing value of $F_j^{-1}(1-\epsilon) * r(j)$ with $r(j)$ defined as above.
4. (MinD) The MinD rule orders the activities by increasing value of $F_j^{-1}(1-\epsilon)$.
5. (LST) The LST rule orders the activities by increasing value of their latest starting time as described in Kolisch and Hartmann (1999).
6. (LFT) The LFT orders the activities by increasing value of their latest finish time as described in Davis and Patterson (1975).
7. (MTS) The MTS orders the activities by decreasing value of the number of their successors as described in Alvarez-Valdes and Tamarit (1989).

Whilst the last three rules have been taken from the literature Kolisch and Hartmann (1999), the other rules have been proposed by the authors in Bruni et al. (2011a), who consider also the STC (Van de Vonder et al. 2008) and the RFDFF heuristic (Van De Vonder et al. 2006).

Only a summary of the computational experience is reported; the interested readers are referred to Sect. 3.2 in Bruni et al. (2011a) for the complete set of results.

The computational experiments have been carried out on a set of benchmark problems selected from the project scheduling problem library PSPLIB (Kolisch and Sprecher 1997), available at http://129.187.106.231/psplib/, including 30, 60 and 90 nodes, leading to a total of 2,550 runs.

For all the instances, two types of distribution have been tested in order to asses the effectiveness of the proposed approach with both continuous and discrete distributions. In particular, for the continuous case, real activity duration is assumed to be a uniform random variable $U(0.75p, 2.85p)$, where p equals the deterministic duration. For the discrete case it has been considered a Poisson distribution with mean p. Activity durations are assumed to be independent.

For every network instance, 1,000 scenarios have been simulated by drawing different actual activity durations from the described distribution functions. Using these simulated activity durations, the realized schedule is constructed by applying

the following reactive procedure. An activity list is obtained by ordering the activities in increasing order of their starting times in the proactive schedule. Ties are broken by increasing activity number. Relying on this activity list, a parallel schedule generation scheme builds a schedule based on the actual activity durations. We opted for the railway execution mode never starting activities earlier than their prescheduled start time in the baseline schedule (Deblaere et al. 2007). Actually, this type of constraint is inherent to course scheduling, sports timetabling, railway and airline scheduling, or when activity execution cannot start before the necessary resources have been delivered. The quality has been evaluated by the following a posteriori measures: average tardiness (Tavg), average timely project completion probability (TPCP), average disruption probability over all networks and executions (Davg). Also the predictive makespan (Mak) has been reported enabling a fair comparison amongst the different algorithms.

Computational times are rather low and do not constitute a bottleneck for the algorithms execution notwithstanding at each decision point either a MIP for discrete random variables case and or a nonlinear continuous model for continuous random variables is solved.

A first set of numerical experiments has been carried out with the aim of assessing the variation of the performance measures of our SDGS algorithm as a function of the risk level (measured by ϵ) for different priority rules. Figures 37.3 and 37.4 report the Tavg and the TPCP for different ϵ values and priority rules from 1 to 4, for the 30 nodes test problems. As we can observe in Fig. 37.3, the average tardiness decreases with ϵ. This is an expected result since for decreasing value of ϵ, we impose a more prudent project manager's position imposing a higher risk aversion level. Mathematically speaking, as the value of ϵ decreases, probabilistic constraints are somehow more binding and the schedule is more robust, since it is less exposed to disruptions. The opposite trend can be observed in Fig. 37.4 for the TPCP which increases for decreasing ϵ values.

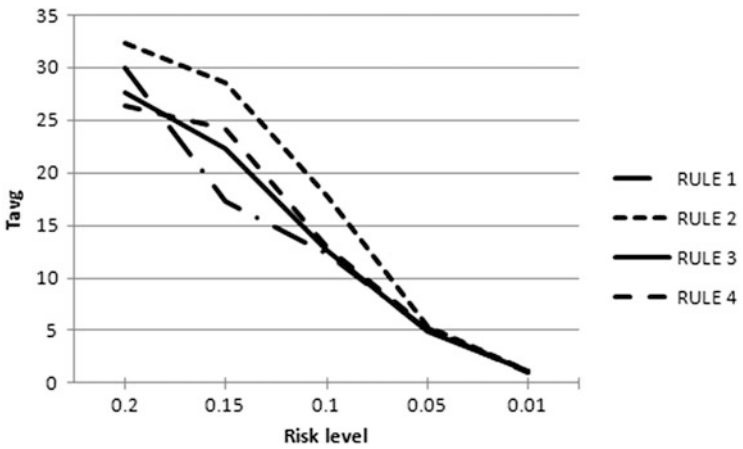


Fig. 37.3 ϵ values versus Tavg

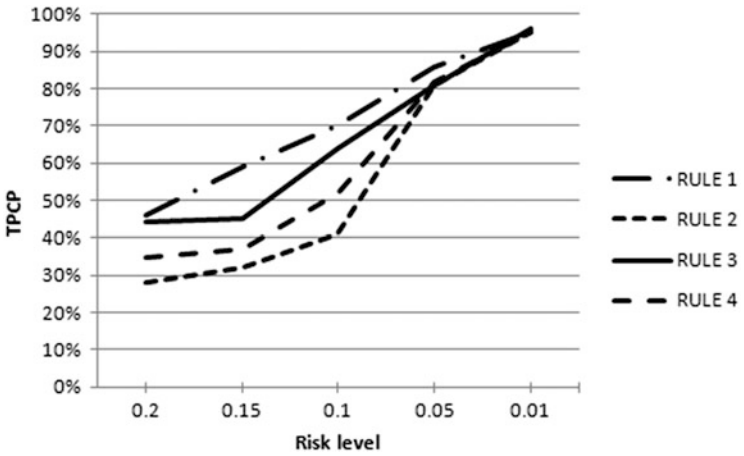


Fig. 37.4 ϵ values versus TPCP

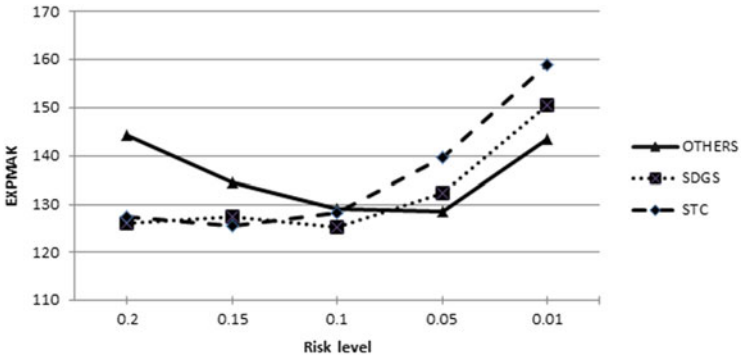


Fig. 37.5 Expected makespan for varying ϵ values: 30 nodes, discrete case

A second set of experiments has been carried out to compare the performance of the SDGS with respect to the benchmark approaches. In particular, we shall present hereafter a graphical comparison on the basis of the expected makespan EXPMAK, (obtained as the sum of the predictive makespan Mak plus the expected tardiness Tavg) and the average probability of disruption Davg.

Figures 37.5, 37.6, and 37.7 show the EXPMAK for the 30, 60, and 90 nodes networks, respectively. Average values have been reported for the SDGS and all the benchmarks considered (named OTHERS). The STC heuristic has been included in the graph, whereas the RFDFF heuristic has been excluded from the comparison since it is always largely outperformed.

It is immediately clear from Figs. 37.5, 37.6, and 37.7 that the expected makespan of SDGS is in general smaller than that of STC, but the same is not thoroughly true for the benchmarks. Observing the intersection between the continuous line

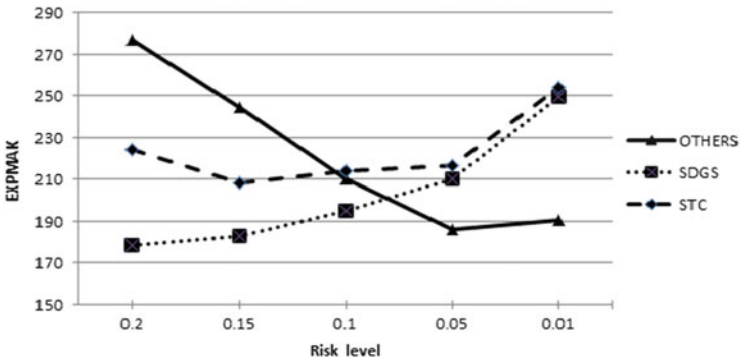


Fig. 37.6 Expected makespan for varying ϵ values: 60 nodes, discrete case

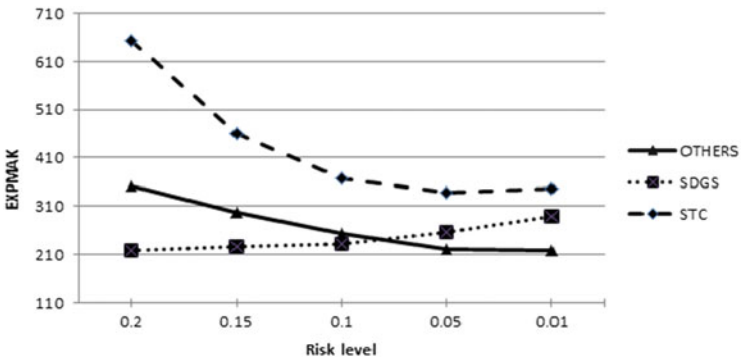


Fig. 37.7 Expected makespan for varying ϵ values: 90 nodes, discrete case

and the dashed line for ϵ between 0.05 and 0.1 we may conclude that for relevant risk levels ($\epsilon \in [0.05, 0.2]$) SDGS should be preferred in terms of EXPMAK. An opposite behaviour emerges for lower risk levels. In practice, risk-averse project managers, for budget restrictions, may accept to bear some risk to avoid unnecessary extra costs. Therefore, the range $[0.05, 0.2]$ constitutes a meaningful choice for moderately risk averse project managers.

The reader may notice a seemingly strange trend in the expected makespan, that seems to have a non-monotone behaviour, with a decreasing slope up to the minimum and an increasing or constant slope afterwards. This behaviour is more evident for the 30 nodes networks, and in general it is relevant for the benchmark algorithms (OTHERS) and for the STC in the case with 90 nodes. This unforeseen descendant behaviour of the expected makespan is due to the influence of two opposing forces that are in effect. As depicted in Fig. 37.8, on the one hand there is a predictive makespan (Mak) whose value increases as ϵ decreases, and on the other hand the expected tardiness (Tavg) that drastically reduces as long as the risk we are willing to bear decreases.

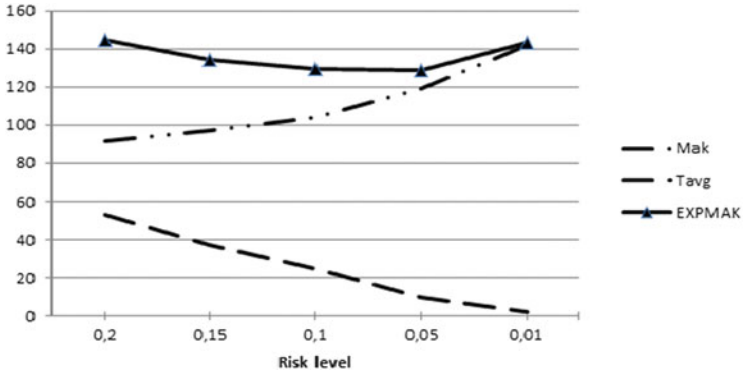


Fig. 37.8 Expected makespan components for varying ϵ values: 30 nodes, discrete case

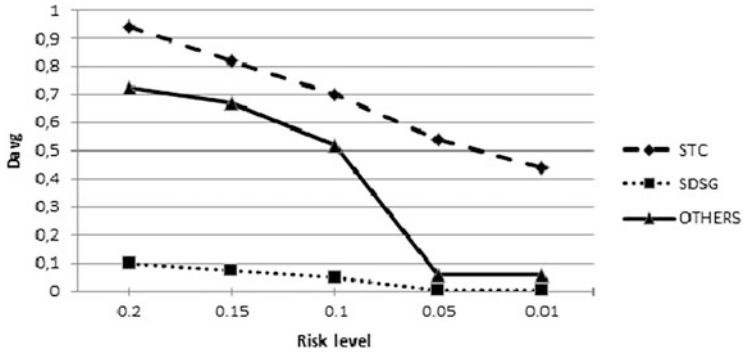


Fig. 37.9 Davg for varying ϵ values: 90 nodes, discrete case

Investigating the comparative performance of the algorithms in terms of Davg, we observed that SDGS exhibits the best performance with very low Davg, especially for large networks, as evident from Fig. 37.9. This claim is supported by the consideration that the Davg gap between SDGS and STC algorithms increases with the dimension of the network. With respect to the comparison between these two algorithms, there is some evidence on the superiority of SDGS over STC for the stability measures considered up to this point. This superiority is also supported by the Tavg values, which can be unacceptably high for both the STC heuristic and the others benchmark heuristics considered. In effect, an apposite behaviour can be observed for the TPCP for which the superiority of the STC is evident over all the algorithms considered. We would like to remark that in the worst case, the TPCP of the STC heuristic doubles the TPCP of the SDGS.

If solution stability is deemed of utmost importance, the best choice seems to be the SDGS heuristic. This heuristic guarantees very good stability performance in terms of disruption probability. If, on the contrary, the sensitivity of the schedule performance in terms of the objective value is the criterion to pursuit, we observe

that nice results are obtained for $\epsilon \geq 0.01$ by the STC heuristic with high TPCP and also acceptable stability indicators. When the project manager is very conservative and risk averse ($\epsilon \leq 0.05$), an attractive alternative especially for large instances can be constituted by the benchmark procedures that offer a good comprise between computational time and solution quality. However, above this risk level they fall inevitably in solutions of substantially lower quality.

As far as the RFDFF is concerned, we observe that notwithstanding the unbuffered schedule fed into RFDFF depends on the ϵ value considered, the results obtained are almost the same whatever the risk aversion of the decision maker is. This behaviour can be due to the right-justification mechanism, which inserts buffers in front of the activities in order to make the schedule solution robust.

The general trend of the continuous distribution function is similar to the one observed for the discrete case, albeit with some differences. We notice that the performance in terms of EXPMAK of the benchmark heuristics (excluding as before the RFDFF) is now comparable to the performance of the SDGS, at least for the 30 and 60 nodes networks. As already observed in the discrete case, also in this case the benchmark procedures outperform SDGS and STC in the expected makespan for ϵ value between 0.1 and 0.15. We further observe that in this case, STC outperforms SDGS for ϵ values above 0.1 for the 30 nodes network and above 0.15 for the 60 nodes network. The EXPMAK of the STC for the network with 90 nodes is on the contrary quite high. This worsening in the EXPMAK is compensated by a higher TPCP for the SDGS, as evident from Fig. 37.10.

As a byproduct, we observe that the STC heuristic seems to be less sensible to the variation of the risk value, with Davg quite high, especially if compared with the SDGS values. It is also worth noting that the Davg of the SDGS heuristic is very low, falling down to zero for small ϵ values.

In conclusion, the heuristic presented is able to build a proactive baseline schedule with a satisfactory behavior under uncertainty. Since the scheduling decisions anticipate the uncertainty, by embedding it into the mathematical program with joint probabilistic constraints, the resulting plan will be hedged against possible

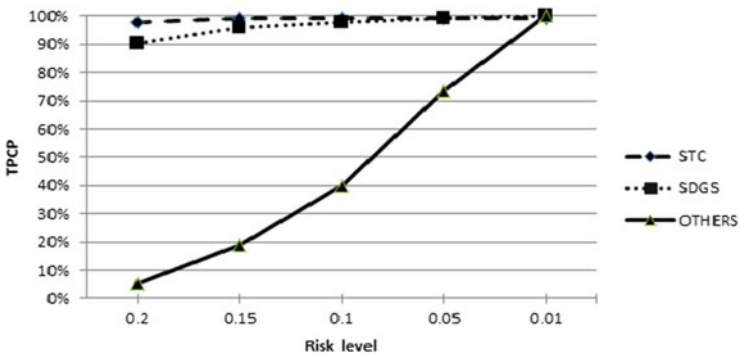


Fig. 37.10 TPCP for varying ϵ values: 60 nodes, continuous case

disruptions and only a minimum adaptation effort will be required during the project execution in case of disruptions. This is a nice feature, especially for some projects for which a baseline schedule is not only compellingly important but also absolutely necessary. The next section presents an example of such projects concerning the interesting applicative context of the construction industry.

37.4 The SRCPSP in Construction Projects

Construction projects are usually characterized by high complexity. Several factors determine this feature: a great number of activities has to be performed in order to achieve project completion, a variety of resources, both material and human, are necessary to perform activities, and therefore great capital investments have to be managed. An efficient scheduling phase is crucial in order to ensure that the project is completed on time and within budget. In this respect, a detailed baseline project schedule plays a crucial role: as widely recognized in Mehta and Uzsoy (1998), it supports project managers in monitoring the work progress, facilitating resource allocation, and providing a basis for managing external activities, such as relations with contractors. In construction industry, baseline schedule generation is usually performed by using different scheduling techniques, like, for instance, PERT (Malcolm et al. 1959) embedded in most computer software packages developed for construction project management. The main drawback of these time-oriented scheduling techniques is the assumption of unlimited availability of resources for each project activity (Nkasu 1994). In real construction projects, many problems arise when activities require resources that are available only in limited quantities making resource allocation indispensable in the generation of realistic baseline project schedules (Kim and Garza 2005). As a matter of fact, ignoring resource considerations in the scheduling phase of the project will lead to extremely poor schedule performance. Moreover, the complex dynamic and uncertain environment in which construction projects have to be performed highlights the need for effective planning and scheduling tools.

The objective of this section is to describe an application of SRCPSP on a real project for construction of students' apartments at the University of Calabria, Italy, where the need of a proactive approach and the generation of a baseline schedule which is protected against disruptions is of utmost importance. A user friendly tool for project scheduling under uncertainty was developed and used as a risk response method that the project team could easily use, after a proper risk assessment program has quantified the impact of potential risks involved in the project at hand on individual activities duration. The only action required to managers is to define the project breakdown structure and store project data such as activity number, ID, resource requirement, precedence relations among activities, and resource availability. Furthermore, the manager has the possibility to choose the value of reliability parameter $1 - \epsilon$ controlling the probability of the project to finish on time. Once the data are uploaded on the system, the baseline schedule is

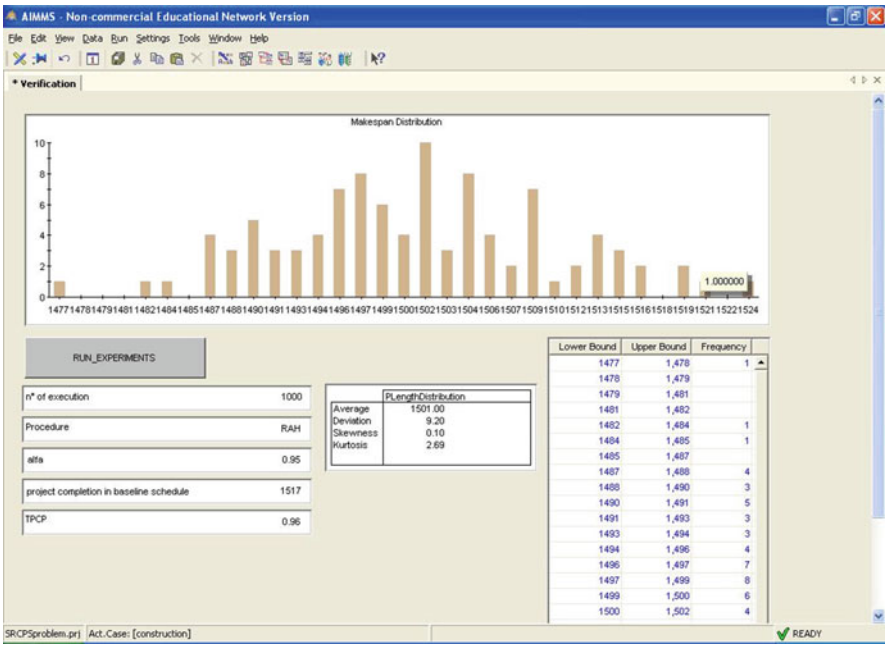


Fig. 37.11 Screenshot of the simulation experiments

automatically generated, and if desired, the simulation phase performed. A screenshot of the graphical interface developed for our tool is shown in Fig. 37.11.

The project consists of 43 activities; the first and the last one are dummy activities representing the starting and the ending of the project, respectively. The project network is displayed in Fig. 37.12, while ID, number, expected duration, and labour requirement of activities are listed in Table 37.1.

The project managers considered two risks to be important: errors in execution and poor weather conditions. Estimates of probability and the impact of different case scenarios as well as the overall frequency of risk occurrence were obtained from the project management team during an interview session and compared with the historical data of similar construction projects completed at the University of Calabria. Using this information, a protected schedule for the UNICAL project was built by using the scheduling mechanism described in Bruni et al. (2011b), in which the probability level for the on-time completion of the project was set to 0.95. On the other hand, an unprotected deterministic schedule was generated by the project managers on the basis of their own experience and with the support of a deterministic quantitative tools for the solution of resource-constrained project scheduling problems. Managers estimated project completion time taking into account external/internal critical factors such as weather conditions, manpower, and

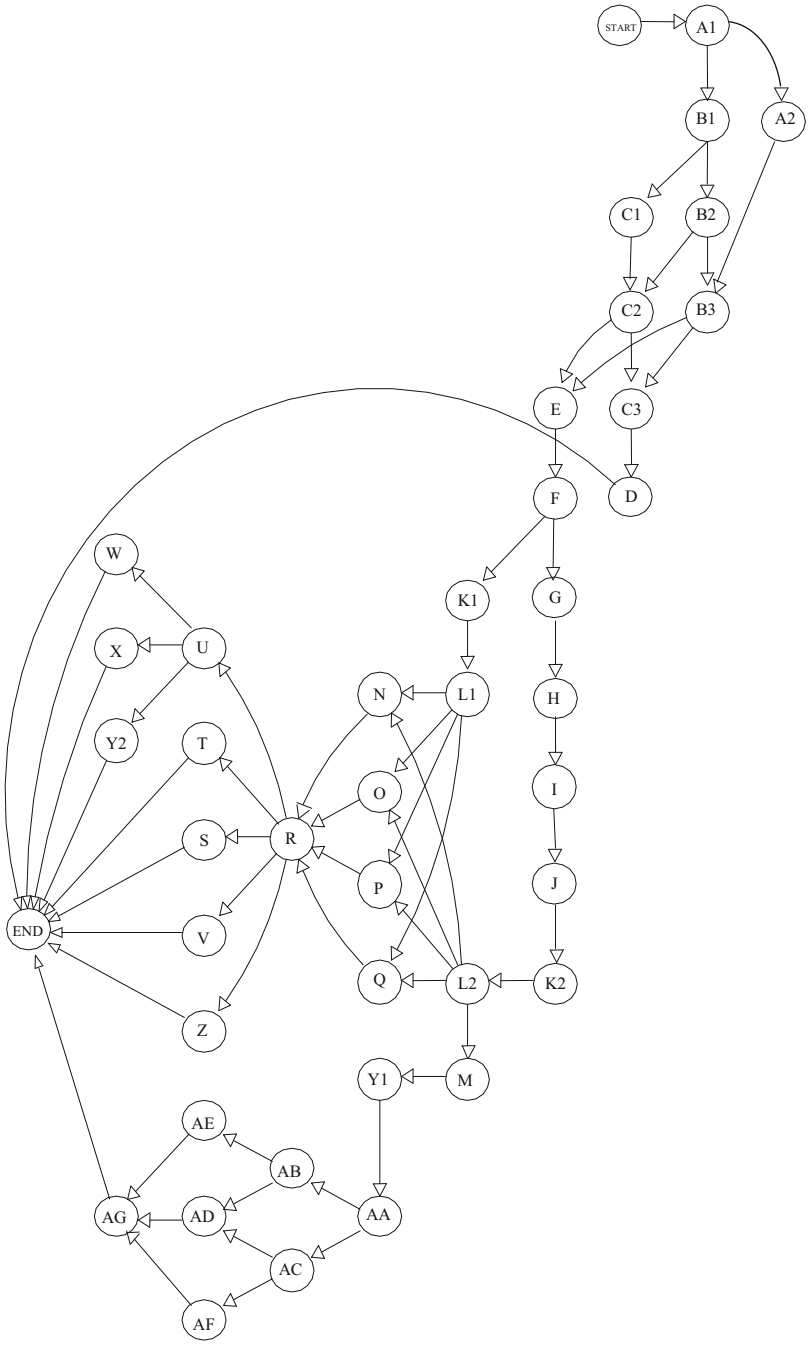


Fig. 37.12 Construction project network

Table 37.1 Project description

Activity ID	Activity description	Expected duration (days)	Resource requirement
START	DUMMY START	0	0
A1	Building yard delimitation	10	7
A2	Building yard resource preparation	20	7
B1	Excavation works	16	5
B2	Grading	16	5
B3	Site preparation	18	5
C1	Basement foundations	16	6
C2	Footings	16	6
C3	Foundation walls	18	6
D	Crawl space	50	6
E	First floor	100	6
F	Second floor	75	6
G	Third floor	75	6
H	Fourth floor	50	6
I	Fifth floor	50	6
J	Roofing	45	23
K1	Exterior wall	90	6
K2	Exterior wall	30	6
L1	Interior wall	70	6
L2	Interior wall	30	6
M	Wall and ceiling finishes	50	15
N	Electric installation	60	12
O	Other system	30	12
P	Waterworks and plumbing	60	12
Q	Heating system	60	9
R	Interior plaster	100	15
S	Finish electrical	30	12
T	Finish other system	30	12
U	Tiling	120	7
V	Finish heating system	30	9
Z	External door and window frames	100	5
W	Internal door and window frames	100	5
X	Finish waterworks plumbing	30	12
Y1 4	Exterior sidings	30	4
Y2	Interior sidings and painting	90	4
AA	Services connections	45	14
AB	Exterior lighting system	45	3
AC	Fire fighting system	30	6
AD1	Driveway and parking	60	2
AE	Pedestrian crossing	76	3
AF	Disable crossing	20	4
AG	Garden	30	6
Z	DUMMY END	0	0

resources availability and most-likely durations of activities. In order to perform an a posteriori analysis we tested the two schedules in a simulation phase, in which a number of possible project realizations, called scenarios, were simulated and a reactive scheduling procedure was applied for each scenario, opting for never starting activities earlier than their prescheduled start time in the baseline schedule. The schedule generated by managers sets a completely unrealistic planned project delivery date of about 1,250 days, with a probability around 50 % to be exceeded. This performance can be very unsatisfactory especially for construction projects for which very high penalties are usually associated to heavy due date violations and schedule breakages. Such observation underlines the crucial value of an accurate planning phase and, as a byproduct, the inadequacy of traditional scheduling procedures in facing uncertainty. Furthermore, managers can consider a more realistic delivery date when take part in a call, rather than a tentative date, for project completion, that will not be respected with a high probability.

The schedule generated by the SDGS method results in a planned project delivery date with a probability (estimated in the simulation phase) of being completed on time equal to 0.96. In Fig. 37.13 the tardiness for different ϵ values is reported. The tardiness represents the difference between its actual completion time and the planned one in the baseline schedule. It is evident that if a penalty is due for each extra period required to execute an activity, tardiness represents an important measure of performance for scheduling in construction project. As confirmed by the results, higher probability values lead to better solutions in term of risk adverseness. The proper calibration of the reliability value is up to decision maker, who could find the best tradeoff on the basis of his experience.

This application can be seen as the first attempt to provide managers with a robust analytical tool with a graphical interface and very easy to use, capable to quantify the risk associated to a baseline schedule and to support their experience in the planning phase of complex projects.

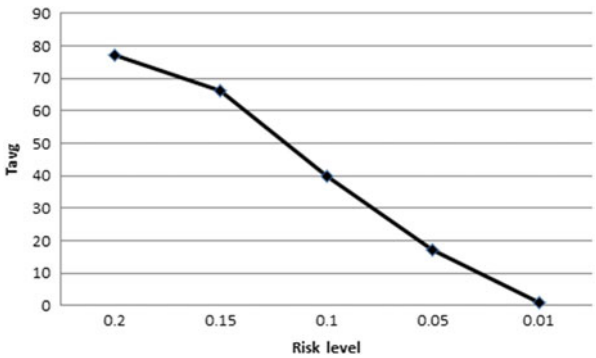


Fig. 37.13 Tardiness for varying ϵ values

37.5 Conclusions

The SRCPSP is a very challenging problem both from a computational and a theoretical point of view. It poses interesting modeling and computational questions, yet far to be resolved. Nonetheless, relevant scientific contributions have shown the tractability of the SRCPSP and its importance in practical applications. Notwithstanding the encouraging results obtained, some remarks are in order. The results of the researches reviewed are not valid for dealing with uncertainty in a general setting. Potential uncertainties may stem from unavailability of resources, changes in ready times and due dates, incorporation or dropping of new activities, etc. The appropriate treatment of these kinds of uncertainty is an interesting area of future research.

Potential for further research lies also in generalized RCPSP stochastic variants and solutions based on combined proactive-reactive approaches into the framework of two-stage stochastic programming. Moreover, the research could look into the potential for creation of proactive methods for the resource allocation problem under uncertainty.

References

Alvarez-Valdes R, Tamarit JM (1989) Heuristic algorithms for resource constrained project scheduling. A review and an empirical analysis. In: Słowiński R, Węglarz J (eds) Advances in project scheduling. Elsevier, Amsterdam, pp 113–134

Ashtiani B, Leus R, Aryanezhad, MB (2011) New competitive results for the stochastic resource constrained project scheduling problem: exploring the benefits of preprocessing. J Sched 14:157–171

Atkinson R, Crawford L, Ward S (2006) Fundamental uncertainties in projects and the scope of project management. Int J Proj Manag 24:687–698

Ballestín F (2007) When it is worthwhile to work with the stochastic RCPSP? J Sched 10:153–166

Ballestín F, Leus R (2009) Resource-constrained project scheduling for timely project completion with stochastic activity durations. Prod Oper Manag 18:459–474

Bruni ME, Beraldi P, Guerriero F, Pinto E (2011a) A heuristic approach for resource constrained project scheduling with uncertain activity durations. Comput Oper Res 38:1305–1318

Bruni ME, Beraldi P, Guerriero F, Pinto E (2011b) A scheduling methodology for dealing with uncertainty in construction projects. Eng Comput 28(8):1064–1078

Davis E, Patterson J (1975) A comparison of heuristic and optimum solutions in resource constrained project scheduling. Manag Sci 21:944–955

Deblaere F, Demeulemeester E, Herroelen W, Van de Vonder S (2007) Robust resource allocation decisions in resource-constrained projects. Decis Sci 38:1–37

Dentcheva D, Prékopa A, Ruszczyński A (1998) On stochastic integer programming under probabilistic constraints. Research Report RRR29-98, RUTCOR Rutgers Center for Operations Research, Rutgers University, Newark

Goldratt EM (1997) Critical chain. The North River Press, Great Barrington

Golenko-Ginzburg D, Gonik A (1997) Stochastic network project scheduling with non-consumable limited resources. Int J Prod Econ 48:29–37

Golenko-Ginzburg D, Gonik A (1998) A heuristic for network project scheduling with random activity durations depending on the resource allocation. Int J Prod Econ 55:149–162

Graham RL (1966) Bounds on multiprocessing timing anomalies. Bell Syst Tech J 45:1563–1581
Igelmund G, Radermacher FJ (1983a) Algorithmic approaches to preselective strategies for stochastic scheduling problems. Networks 13:29–48
Igelmund G, Radermacher FJ (1983b) Preselective strategies for the optimization of stochastic project networks under resource constraints. Networks 13:1–28
Jaafari A (2001) Management of risks, uncertainties and opportunities on projects: time for a fundamental shift. Int J of Proj Manag 19(2):89–101
Jagannathan R (1974) Chance-constrained programming with joint constraints. Oper Res 22(2):358–372
Kim K, Garza MJ (2005) Evaluation of the resource-constrained critical path method algorithms. J Constr Eng Manag 131(5):522–532
Kolisch R, Hartmann S (1999) Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. In: Węglarz J (ed) Project scheduling: recent models, algorithms and applications. Kluwer, Dordrecht, pp 147–178
Kolisch R, Sprecher A (1997) PSPLIB a project scheduling problem library. Eur J Oper Res 96:205–216
Lambrechts O, Demeulemeester E, Herroelen W (2007) A tabu search procedure for developing robust predictive project schedules. Int J Prod Econ 111(2):496–508
Malcolm DG, Roseboom JH, Clark CE, Fazar W (1959) Application of a technique for research and development program evaluation. Oper Res 7(5):656–669
Mehta SV, Uzsoy RM (1998) Predictable scheduling of a job shop subject to break-downs. IEEE T Roboti Autom 14(3):365–378
Miller BL, Wagner HM (1965) Chance constrained programming with joint constraints. Oper Res 13(6):930–945
Möhring RH, Stork F (2000) Linear preselective strategies for stochastic project scheduling. Math Method Oper Res 52(3):501–515
Nkasu MM (1994) COMSARS: a computer-sequencing approach to multiresource-constrained scheduling – Part 1: deterministic networks. Int J Proj Manag 12(3):183–192
Perminova O, Gustafsson M, Wikstrom K (2008) Defining uncertainty in projects a new perspective. Int J Proj Manag 26:73–79
Prékopa, A (1995) Stochastic programming. Kluwer Academic, Boston
Rabbani M, Fatemi Ghomi SMT, Jolai F, Lahiji NS (2007) A new heuristic for resource-constrained project scheduling in stochastic networks using critical chain concept. Eur J Oper Res 176:794–808
Radermacher FJ (1981) Optimale Strategien für stochastische Scheduling-Probleme. Habilitation thesis, RWTH Aachen, Aachen, Germany. In: Schriften zur Informatik und angewandten Mathematik 98, RWTH Aachen
Stork F (2001) Stochastic resource-constrained project scheduling. Ph.D. dissertation, Technische Universität Berlin, Berlin
Tsai YW, Gemmil DD (1998) Using tabu search to schedule activities of stochastic resource-constrained projects. Eur J Oper Res 111:129–141
Van De Vonder S, Demeulemeester E, Herroelen W, Leus R (2006) The trade-off between stability and makespan in resource-constrained project scheduling. Int J Prod Res 44(2):215–236
Van de Vonder S, Demeulemeester E, Herroelen W (2008) Proactive heuristic procedures for robust project scheduling: an experimental analysis. Eur J Oper Res 189:723–733
Zhu G, Bard JF, Yu G (2007) A two-stage stochastic programming approach for project planning with uncertain activity durations. J Sched 10:167–180

Chapter 38
The Markovian Multi-Criteria Multi-Project Resource-Constrained Project Scheduling Problem

Saeed Yaghoubi, Siamak Noori, and Amir Azaron

Abstract This chapter develops a Markovian multi-objective mathematical programming model for the resource allocation problem in dynamic PERT networks with a finite capacity of concurrent projects. It is assumed that new projects are generated according to a Poisson process and activity durations are independent random variables with exponential distributions. This system is represented as a queueing network with finite concurrent projects, where each activity of a project is operated at a dedicated service station with one server located in a node of the network. In this investigation, not only activity durations, but also operating costs of service stations per period are all considered as independent random variables. This problem is formulated as a multi-objective model using continuous-time Markov processes with three conflicting objectives to optimally control the resources allocated to service stations. It is impossible to solve this problem optimally in a reasonable time, and consequently we apply a particle swarm optimization (PSO) method to solve this multi-objective continuous-time problem using a goal attainment technique. Finally, to show the effectiveness of the proposed PSO, we compare the results of a discrete-time approximation of the original optimal control problem with the results obtained by the proposed PSO.

Keywords Dynamic scheduling • Multi-criteria scheduling • Multi-project scheduling • Resource constraints

S. Yaghoubi (✉) • S. Noori
Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran
e-mail: yaghoubi@iust.ac.ir; snoori@iust.ac.ir

A. Azaron
Department of Industrial Engineering, Istanbul Sehir University, Istanbul, Turkey
e-mail: amirazaron@sehir.edu.tr

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_8

38.1 Introduction

Nowadays, multi-project scheduling is extensively applied because all projects in the context of an organization are considered as one system, while constrained resources are allocated among multiple projects. Furthermore, in some organizations, the project-oriented approach is adopted as a primary approach. Therefore, multi-project management is of significant interest in project management and scheduling, whereas traditional project management concentrates on individual projects.

A Multi-Project Resource-Constrained Scheduling Problem (MPRCSP) in static and deterministic conditions is one of the major topics in the literature of multi-project scheduling. In general, two types of approaches for analyzing the MPRCSP exist.

To analyze this problem, one approach is to link all projects of an organization synthetically together into a large single project, while the other approaches consider the projects as independent components subject to resource constraints. In addition, these approaches use an objective function considering all projects.

Wiest (1967) and Pritsker et al. (1969) first studied the MPRCSP and presented a zero-one programming approach and a heuristic model for analyzing this problem, respectively. Next, Kurtulus and Davis (1982) and Kurtulus and Narula (1985) studied this problem by employing the priority rules and describing measures. Furthermore, some studies have been focused on the MPRCSP by using multi-objective and multi-criteria approaches. For instance, Chen (1994) developed a zero-one goal programming model for the MPRCSP with maintenance of mineral processing, and Lova et al. (2000) presented a multi-criteria model for analyzing the MPRCSP. Kanagasabapathi et al. (2009) analyzed the MPRCSP in a static environment by defining performance measures including the maximum tardiness and mean tardiness of projects. In addition, Tsubakitani and Deckro (1990) and Lova and Tormos (2001) presented heuristic methods for solving the MPRCSP. In addition, Krüger and Scholl (2008) developed the MPRCSP by considering transfer times and their costs.

Meta-heuristic methods, such as particle swarm optimization, genetic algorithm, simulated annealing, artificial neural networks, and their hybrids, are used in various fields, while these methods have been rarely applied in multi-project scheduling (Chen and Shahandashti 2009). Kumanan et al. (2006) and Gonçalves et al. (2008) proposed a genetic algorithm for the resource-constrained multi-project scheduling problem. Ying et al. (2009) presented a hybrid genetic algorithm for the MPRCSP. In addition, Chen and Shahandashti (2009) developed a hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints.

In all of the mentioned studies, the MPRCSP has been studied in static and deterministic environments, while a few investigations have been focused on multi-project scheduling under uncertainty conditions. Fatemi Ghomi and Ashjari (2002) described a simulation model for multi-project resource allocation with

stochastic activity durations by considering a multi-channel queueing. A mixed-integer nonlinear programming model was also extended by Nozick et al. (2004) to optimally allocate the resources, while the probability distributions of the activity duration and allocated resources are dependent. Furthermore, Kao et al. (2006) and Byali and Kannan (2008) proposed an event-driven approach and Critical Chain Project Management (CCPM) approach for overcoming uncertainty in multi-project environment, respectively.

Clearly, multi-project scheduling is more complex than single project scheduling and the problem is even more difficult when the task durations are stochastic. On the other hand, in many organizations, not only the tasks durations are uncertain, but also new projects emerge dynamically over the time horizon. In such occasion, organization is faced with a multi-project system denominated as "dynamic PERT network", in which its scheduling procedure is more elaborate than before. Adler et al. (1995) first presented a process-based system for a dynamic PERT network by using simulation and considering the organization as a stochastic processing network. They supposed that the organization comprises a collection of "service stations" (work stations) or "resources", where one or more identical parallel servers can be located. Thus, the dynamic PERT network can be considered as a queueing network and is appropriate and attractive for organizations that have similar projects, such as maintenance projects. Anavi-Isakow and Golany (2003) then applied the concept of CONWIP (constant work-in-process) in dynamic PERT networks by using a simulation study. They explained two control mechanisms, CONPIP (COnstant Number of Projects In Process) and CONTIP (CONstant Time of projects In Process). The CONPIP mechanism restricts the number of projects, while the CONTIP mechanism limits the total processing time by all the projects that are active in the system.

For the resource allocation problem in dynamic PERT network, two general approaches exist. In the first approach, the number of servers in every service station is fixed and allocated resources affect the mean of service times. We call this approach "resources affecting servers". Azaron and Tavakkoli-Moghaddam (2006, 2007) as the first pioneering researchers presented multi-objective models using continuous-time Markov processes for the resource allocation problem in dynamic PERT networks, where new projects are generated according to a Poisson process and the activity durations are exponentially distributed random variables. They also assumed that the capacity of the system is infinite, the number of servers in every service station is either one or infinity, the discipline of queues is First Come First Served (FCFS), and the allocated resources affect the mean activity durations. Yaghoubi et al. (2011a) introduced an analytical multi-objective model using continuous-time Markov processes for the resource allocation problem in a dynamic PERT network, where the capacity of the system is finite and projects are generated according to a Poisson process. Yaghoubi et al. (2014) also developed a simulated annealing algorithm to solve the multi-class version of the same problem when projects from different classes are different in their precedence networks and the durations of the activities.

The second approach was proposed by Cohen et al. (2005, 2007), where resources may work in parallel, i.e., the number of servers and resources allocated in every service station are equal (e.g., electrical work station with electricians, mechanical work station with mechanics, etc.) and the amount of resources available to be allocated to all service stations is constant. We refer to this approach as "resources as servers". Cohen et al. (2005, 2007) achieved nearly optimally allocated resources to the entities that perform the projects in CONPIP system by using Cross Entropy (CE) based on simulation. Yaghoubi et al. (2011b) then extended an analytical multi-objective model using continuous-time Markov processes to optimally control resources allocated to the activities in a multi-server dynamic PERT network, for this approach.

In all mentioned papers, it is assumed that the direct cost of the service station is deterministic and increases when we allocate more resources to that particular service station. In this chapter, we relax this assumption and assume that the direct cost of the service station is stochastic, which is the main difference between this chapter and the previous mentioned papers.

Moreover, the multi-objective continuous-time stochastic programming problems of previous mentioned papers are impossible to solve optimally for large-sized problems in a reasonable time. Therefore, the continuous-time problem is discretized, which means that the differential equations are transformed into difference equations and the integral terms into summation terms. Li and Wang (2009) also proposed a multi-objective risk-time-cost tradeoff problem in a dynamic PERT network by using a general project risk element transmission theory. They transformed the continuous-time model into a discrete-time model and used radial basis function (RBF) neural networks to solve this discrete model of the problem.

Reviewing the above mentioned studies shows that no meta-heuristic method has already been proposed to solve the resource allocation (time-cost tradeoff) problem in dynamic PERT networks with the finite capacity (CONPIP) and this problem was only solved in discrete-time approximation form. In this chapter, we introduce a particle swarm optimization (PSO) method to solve this multi-objective continuous-time problem using a goal attainment technique, which is another contribution of this chapter over the existing literature.

It is worth mentioning that Azaron et al. (2005) developed a multi-objective model for the time-cost tradeoff problem in classical PERT networks with generalized Erlang distributions of activity durations, using a genetic algorithm. They also compared the results obtained by a genetic algorithm against the results of a discrete-time approximation method for solving the original optimal control problem. Moreover, Azaron et al. (2006, 2007) developed some multi-objective models for the time-cost tradeoff problem in classical PERT networks with different assumptions on the distributions of activity durations (exponential in one and generalized Erlang in the other paper), different objective functions, and also different solution techniques (goal programming and goal attainment in one and the interactive SWT technique in the other one).

In this chapter, we consider a multi-project system with similar projects and finite capacity. It is assumed that the new projects, including all their activities,

arrive in the system according to a Poisson process. Such a system is represented as a queueing network with finitely many concurrent projects, while each activity of any project is performed at a dedicated service station settled in a node of the network based on the FCFS discipline. It is also assumed that there is only one server in each service station, while the activity durations (i.e., service times) are independent random variables with exponential distributions. On the other hand, not only activity durations, but also direct costs of service stations per period are all considered as independent random variables. Therefore, the total direct cost, which is the summation of direct costs of service stations, will also be a random variable. Moreover, the mean time spent in each service station is decreased and the mean direct cost of the service station is increased when we allocate more resources to that particular service station. It means that the mean time spent in each service station and the mean direct cost of the service station are, respectively, non-increasing and non-decreasing functions of the amount of resources allocated to that service station.

In this chapter, we propose a multi-objective model using continuous-time Markov processes for the resource allocation problem in dynamic PERT networks with the finite capacity of concurrent projects (CONPIP) for the resources affecting servers approach, and solve this model by a particle swarm optimization algorithm (PSO).

The problem is formulated as a multi-objective model with three conflicting objectives to optimally control the resources allocated to service stations. In this model, the first objective is to maximize the probability that the total direct cost of service stations per period does not exceed a certain cost level or budget. The mean project completion time in the steady state is also considered as the second objective, which should be minimized. Moreover, the probability that the system becomes empty in the steady state is considered as the third objective function, which should be minimized as well.

Since the resulting mathematical model is continuous-time, it is too complicated to be solved optimally. Therefore, we develop a particle swarm optimization (PSO) approach to solve it using a goal attainment technique. The reason is that PSO gets better results in a faster, cheaper way compared with other methods. In addition, PSO needs few parameter settings and is also an efficient global optimizer for continuous variables, while it does not require that the optimization problem is differentiable. Finally, to show the effectiveness of the proposed PSO, we compare the results of the discrete-time approximation of the original optimal control problem with the results obtained by the proposed PSO based on the computational time of the discrete-time approximation technique as its stopping criterion.

In the remainder of this chapter, firstly, the dynamic PERT network with the finite capacity of concurrent projects is modeled as finite-state continuous-time Markov processes. Secondly, a multi-objective model to optimally control the resources allocated to the servers is developed, where the direct costs of the service stations are stochastic. Thirdly, we propose a particle swarm optimization algorithm for solving the resulting multi-objective problem, and finally draw the conclusions.

38.2 Modeling of Markov Dynamic PERT Networks with CONPIP

In this section, we model the dynamic PERT networks with finite capacity of concurrent projects (CONPIP) using continuous-time Markov processes. For this purpose, we extend the method of Kulkarni and Adlakha (1986) in this step, because this method is an analytical one, simple, easy to implement on a computer, and computationally stable. It is assumed that a project is represented as an activity-on-node (AoN) graph, and also the new projects including all their activities are generated according to a Poisson process with the rate of λ. Each activity is processed at a dedicated service station located in a node of the network. The activities associated with successive projects contend for resources on the FCFS basis. This dynamic PERT network can be represented as a network of queues, where the service times represent the durations of the corresponding activities and the arrival stream to each node follows a Poisson process with the rate of λ. It is also assumed that the number of servers in each service station equals one, while the service times (i.e., activity durations are independent random variables with exponential distributions).

To model dynamic PERT networks with the CONPIP, we transform the dynamic PERT network, represented as an activity-on-node (AoN) network, to a classic PERT network represented as an activity-on-arc (AoA) network as follows by replacing any node in the AoN network with a stochastic activity:

- The activity-on-arc network is denoted by $G = (V, E)$ with node set V and arc set E. Note that the words "activity" and "arc" are used synonymously in what follows.
- The indices of the nodes (i.e., the events) are i and j: $i, j \in V$.
- Each activity corresponds to an arc $a = (i, j) \in E$.
- The set of activities with start event i coincides with the set of arcs E_i^+ emanating from node i, and the set of activities with end event j coincides with the set of arcs E_j^- leading into node j.
- The start and end nodes of activity $a = (i, j)$ are designated by $s(a) = i$ and $e(a) = j$.

Any node with a service station is substituted by an activity (i, j), whose length is equal to the waiting time in the service station. The indicated process is the opposite of absorbing an edge e in a graph G in graph theory, see Azaron and Modarres (2005) for more details. Then, all arcs with zero length are eliminated.

Let s and t be the source and sink nodes, respectively. The length of arc $a \in E$ is an exponentially distributed random variable with parameter μ_a.

Definition 38.1. For $V_1 \subset V$ such that $s \in V_1$ and $t \in V_2 = V \backslash V_1$, an (s, t)-cut is defined as follows:

$$(V_1, V_2) = \{a \in E : s(a) \in V_1, e(a) \in V_2\} \tag{38.1}$$

An (s,t)-cut (V_1, V_2) is denominated a uniformly directed cut (UDC), if $(V_2, V_1) = \emptyset$ (i.e., there are no two arcs in the cut belonging to the same path in the project network). Each UDC is clearly a set of arcs, in which the starting node of each arc belongs to V_1 and the ending node of that arc belongs to V_2.

Definition 38.2. An (E_1, E_2, E_3), E_1, E_2, and E_3 being subsets of E is defined as admissible three-partition of a uniformly directed cut D if $D = E_1 \cup E_2 \cup E_3$ and $E_1 \cap E_2 = E_1 \cap E_3 = E_2 \cap E_3 = \emptyset$, and also $E^-_{e(a)} \not\subset E_2$ for any $a \in E_2$.

Definition 38.3. During the project's execution at time t, each activity can be in one and only one of active, dormant, in queue or idle states, which are defined as follows:

(i) *Active:* an activity a is active at time t, if it is being performed at time t.
(ii) *Dormant:* an activity a is called dormant at time t, if it has been completed but there is at least one unfinished activity in $E^-_{e(a)}$ at time t.
(iii) *In queue:* activity a is in queue at time t, if all preceding activities of activity a have been completed, but service station a is serving another project.
(iv) *Idle:* an activity a is idle at time t, if it is neither active, nor dormant, nor in queue at time t.

Definition 38.4. The state of project q at time t is $X_q(t) = (A_q(t), D_q(t), Q_q(t))$, where $A_q(t)$, $D_q(t)$, and $Q_q(t)$ are defined as follows:

$A_q(t)$ = set of active activities in project q at time t
$D_q(t)$ = set of dormant activities in project q at time t
$Q_q(t)$ = set of in queue activities in project q at time t

Let N be the maximum number of projects in the system, the state of the system at time t is defined as follows:

$$X(t) = [(A_1(t), D_1(t), Q_1(t)), (A_2(t), D_2(t), Q_2(t)), \ldots, (A_N(t), D_N(t), Q_N(t))] \tag{38.2}$$

Also, the admissible three-partition cut of the system is denoted by:

$$[E_1, E_2, E_3] = [(E_{1_1}, E_{2_1}, E_{3_1}), (E_{1_2}, E_{2_2}, E_{3_2}), \ldots, (E_{1_N}, E_{2_N}, E_{3_N})] \tag{38.3}$$

where $(E_{1q}, E_{2q}, E_{3_q})$ can be any admissible three-partition cut of project q or $(\emptyset, \emptyset, \emptyset)$.

Let $\mathscr{E}^q_3$ be the set containing $(\emptyset, \emptyset, \emptyset)$ and all admissible three-partition UDCs of project q. Note that the same admissible three-partition cuts of different projects cannot occur simultaneously. Also, the project that has been entered to the system earlier than the other projects, is regarded as "forward project" (project 1). The activities of forward project do not wait in queue.

The state of the system at the time zero is $X(0) = [(\emptyset, \emptyset, \emptyset), (\emptyset, \emptyset, \emptyset), \ldots, (\emptyset, \emptyset, \emptyset)]$ and obviously $\{X(t), t \geq 0\}$ is a finite-state continuous-time Markov process, where all the states are transient and there is no absorbing state. Thus, the components of the infinitesimal generator matrix of this process, denoted by $G = \left[g\left\{(E_1, E_2, E_3), (E_1', E_2', E_3')\right\}\right]$, are calculated as follows:

1. Transition 1: In this transition, the transition rate is μ_a

if $a \in E_{1_q}, E_{e(a)}^{-} \not\subset E_{2_q} \cup \{a\}$ **then**
Begin:
$E_{1_q}' := E_{1_q} \setminus \{a\}$,
$E_{2_q}' := E_{2_q} \cup \{a\}$,
for $j = q + 1$ to N do
if $a \in E_{3_j}$ **then** $\left\{E_{1_j}' := E_{1_j} \cup \{a\}, E_{3_j}' := E_{3_j} \setminus \{a\}\right\}$,
end

2. Transition 2: In this transition, the transition rate is μ_a

if $a \in E_{1_q}, E_{e(a)}^{-} \subset E_{2_q} \cup \{a\}$ **then**
Begin:
if $q = 1$ **then**
Begin:
$E_{1_1}' := (E_{1_1} \setminus \{a\}) \cup E_{e(a)}^{+}$,
$E_{2_1}' := E_{2_1} \setminus E_{e(a)}^{-}$,
for $j = 2$ to N do
if $a \in E_{3_j}$ **then**
$\left\{E_{1_j}' := E_{1_j} \cup \{a\}, E_{3_j}' := E_{3_j} \setminus \{a\}\right\}$,
end
if $q \neq 1$**then**
Begin:
$H := \emptyset$,
for $j = 1$ to $q - 1$ do
$H := H \cup (E_{e(a)}^{+} \cap E_{1_j})$,
$E_{1_q}' := (E_{1_q} \setminus \{a\}) \cup (E_{e(a)}^{+} \setminus H)$,
$E_{2_q}' := E_{2_q} \setminus E_{e(a)}^{-}$,
$E_{3_q}' := E_{3_q} \cup H$,
for $j = q + 1$ to N do
if $a \in E_{3_j}$ **then**
$\left\{E_{1_j}' := E_{1_j} \cup \{a\}, E_{3_j}' := E_{3_j} \setminus \{a\}\right\}$,
end
end

3. Transition 3: In this transition, the transition rate is λ

if $E_{1_q} = E_{2_q} = E_{3_q} = \emptyset$ and $(E_{1_{q-1}} \neq \emptyset \; or \; E_{3_{q-1}} \neq \emptyset \; or \; E_{2_{q-1}} \neq \emptyset)$
then

Begin:

$H := \emptyset,$
for $j = 1$ to $q - 1$ do
$H := H \cup (E_s^+ \cap E_{1_j}),$
$E'_{3_j} := H,$
$E'_{1_j} := E_s^+ \backslash H,$
end

As mentioned before, $\{X(t), t \geq 0\}$ is a finite-state continuous-time Markov process, when all states are transient and there is no absorbing state. Representing Σ as the set of system states, whose members are numbered as $1, 2, \ldots, K = |\Sigma|$, where state 1 is the initial state $X(0) = [(\emptyset, \emptyset, \emptyset), (\emptyset, \emptyset, \emptyset), \ldots., (\emptyset, \emptyset, \emptyset)]$. Clearly, the number of system states grows exponentially with the number of UDCs and the capacity of the system, in which there is a limit on the number of projects in the system. Also, we define

$$\pi_i(t) = P(X(t) = i \,|X(0) = 1) \quad i = 1, 2, \ldots, K \tag{38.4}$$

According to Chapman–Kolmogorov forward equations, the system of linear differential equations for the vector $\pi(t) = [\pi_1(t), \pi_2(t), \ldots, \pi_k(t)]$ is given by:

$$\begin{aligned} \pi'(t) &= \tfrac{d\pi(t)}{dt} = \pi(t) \cdot G \\ \pi(0) &= [1, 0, \ldots, 0] \end{aligned} \tag{38.5}$$

where $\pi'(t)$ denotes the derivative of the state vector $\pi(t)$, and G is the infinitesimal generator matrix of the stochastic process $\{X(t), t \geq 0\}$.

38.3 Resource Allocation Problem

In this section, we propose a Markovian multi-objective model in order to optimally control the resources allocated to the servers in a dynamic PERT network with finite capacities, where such a system is represented as a queueing network. In this investigation, not only the service times but also the direct costs of service stations per period are independent random variables. Therefore, the total direct cost, which is the summation of direct costs of service stations, will also be a random variable. Moreover, it is assumed that the mean time spent in each service station is decreased and the mean direct cost of the service station is increased when we allocate more resources to that particular service station. It means that the mean time spent in each

service station and the mean direct cost of the service station are non-increasing and non-decreasing functions of the amount of resource allocated to that service station, respectively.

In our model, the first objective is to maximize the probability that the total direct cost of service stations per period does not exceed a certain cost level or budget. The mean project completion time in the steady state is considered as the second objective, which should be minimized. The last objective is to minimize the probability that the system becomes empty in the steady state. This objective is equivalent to maximizing the utilization factor of the system, because the utilization factor is the probability that the system is busy. The first and the second objectives are in conflict with each other, because if we allocate more resources to service stations, then the probability that the total direct cost of service stations per period do not exceed a certain budget will be decreased and therefore the mean project completion time will be increased.

Let x_a be the amount of resources allocated to service station a and $\tilde{c}$ be the random variable representing the total direct cost of service stations per period. Moreover, let $C_a(x_a)$ and $V_a(x_a)$ be, respectively, the mean and the variance of direct cost of service station a per period. We also define b as a certain cost level or budget value, which the total direct cost should not exceed. It is assumed that $C_a(x_a)$ is a non-decreasing function of amount of the resources x_a allocated to it.

According to the Central Limit Theorem, the summation of a sufficiently large number of independent random variables, each with a well-defined mean and well-defined variance, will be approximately normally distributed. Moreover, in real projects, the number of activities is sufficiently large. Therefore, the distribution of the total direct cost converges to a normal distribution. Note that the distribution of $Z = (\tilde{c} - \sum_{a\in E} C_a(x_a))/(\sqrt{\sum_{a\in E} V_a(x_a)})$ converges to the standard normal distribution.

Max. $P\ (\tilde{c} \leq b)$ is considered as the first objective, which is equivalent to

$$\text{Max.} \quad P\left(Z = \frac{\tilde{c} - \sum\limits_{a\in E} C_a(x_a)}{\sqrt{\sum\limits_{a\in E} V_a(x_a)}} \leq \frac{b - \sum\limits_{a\in E} C_a(x_a)}{\sqrt{\sum\limits_{a\in E} V_a(x_a)}}\right) \tag{38.6}$$

On the other hand, considering the bell shape of the normal distribution, this objective will also be equivalent to

$$\text{Max.} \quad \frac{b - \sum\limits_{a\in E} C_a(x_a)}{\sqrt{\sum\limits_{a\in E} V_a(x_a)}} \Rightarrow \text{Max.} \quad \frac{\left(b - \sum\limits_{a\in E} C_a(x_a)\right)^2}{\sum\limits_{a\in E} V_a(x_a)} \tag{38.7}$$

Consequently, in our model, we consider (38.7) as the first objective.

In addition, let Σ^p be the subset of Σ in that the system has p projects in processing, i.e., the system has $N-p$ capacity for accepting new projects. Let L and W be the average number of projects in the system and the mean project completion time in the steady state, respectively. Therefore, according to Little's Law, we have $L = \lambda' W$, where λ', the effective arrival rate, is equal to $\lim_{t\to\infty} \lambda \sum_{i\in\Sigma-\Sigma^N} \pi_i(t)$, and L is given by:

$$L = \lim_{t\to\infty} \sum_{j=1}^{N} \sum_{i\in\Sigma^j} j\pi_i(t) \tag{38.8}$$

Consequently, the second objective to be minimized is as follows:

$$W = \lim_{t\to\infty} \frac{\sum_{j=1}^{N} \sum_{i\in\Sigma^j} j\pi_i(t)}{\lambda \sum_{i\in\Sigma-\Sigma^N} \pi_i(t)} \tag{38.9}$$

Finally, the third objective to be minimized is equal to $\lim_{t\to\infty} \pi_1(t)$.

Moreover, the mean service time in the service station a is a non-increasing function $p_a(x_a)$ of the amount of resource x_a allocated to it. Thus, the mean service time in the service station a is equal to $1/\mu_a(= p_a(x_a))$, because there is one server located in it. Let R_a^{max} denote the maximum amount of resource available to be allocated to the service station a, R_a^{min} denote the minimum amount of resource needed to get the activity a done, $x = (x_a)^{\mathrm{T}}_{a\in E}$, and R represent the amount of resource available to be allocated to all service stations. In practice, $C_a(x_a)$, $V_a(x_a)$, and $p_a(x_a)$ can be obtained using linear regression by referring to the previous similar activities or the judgments of experts in this area.

Finally, we have a multi-objective stochastic programming problem in that the objective functions are given by:

1. Maximizing the probability that the total direct cost of service stations per period does not exceed the budget b:

$$\text{Max.} f_1(x) = \frac{\left(b - \sum_{a\in E} C_a(x_a)\right)^2}{\sum_{a\in E} V_a(x_a)} \tag{38.10}$$

2. Minimizing the mean project completion time in the steady state:

$$\text{Min.} f_2(x) = \lim_{t\to\infty} \frac{\sum_{j=1}^{N} \sum_{i\in\Sigma^j} j\pi_i(t)}{\lambda \sum_{i\in\Sigma-\Sigma^N} \pi_i(t)} \tag{38.11}$$

3. Minimizing the probability that the system becomes empty in the steady state:

$$\text{Min. } f_3(x) = \lim_{t\to\infty} \pi_1(t) \tag{38.12}$$

The infinitesimal generator matrix G is a function of the control vector $x = (x_a)^{\mathrm{T}}_{a\in E}$. Therefore, the nonlinear dynamic model is represented by

$$\pi'(t) = \pi(t)\cdot G(\mu) \tag{38.13}$$

$$\begin{aligned} &\pi_i(0) = 0 \quad (i = 2,\ldots,K) \\ &\pi_1(0) = 1 \end{aligned} \tag{38.14}$$

The following constraint should also be considered to guarantee having a response in the steady state:

$$\frac{\lambda \lim\limits_{t\to\infty} \sum\limits_{i\in\Sigma-\Sigma^N} \pi_i(t)}{\mu_a} < 1 \Rightarrow \mu_a - \lambda \lim_{t\to\infty} \sum_{i\in\Sigma-\Sigma^N} \pi_i(t) > 0 \quad (a \in E) \tag{38.15}$$

In any optimization model, such constraints cannot be used. Hence, it is replaced it by

$$\mu_a - \lambda \lim_{t\to\infty} \sum_{i\in\Sigma-\Sigma^N} \pi_i(t) \geq \varepsilon \quad (a \in E) \tag{38.16}$$

Consequently, the resulting multi-objective optimal control problem (OCP) will be:

$$\begin{aligned} &\text{Max.} && f_1(x) = \frac{\left(b - \sum\limits_{a\in E} C_a(x_a)\right)^2}{\sum\limits_{a\in E} V_a(x_a)} \\ &\text{Min.} && f_2(x) = \lim_{t\to\infty} \frac{\sum\limits_{j=1}^{N} \sum\limits_{i\in\Sigma^j} j\,\pi_i(t)}{\lambda \sum\limits_{i\in\Sigma-\Sigma^N} \pi_i(t)} \\ &\text{Min.} && f_3(x) = \lim_{t\to\infty} \pi_1(t) \\ &\text{s.t.} && \pi'(t) = \pi(t)\cdot G(\mu) \\ & && \pi_i(0) = 0 \quad (i = 2,\ldots,K) \\ & && \pi_1(0) = 1 \\ & && \pi_i(t) \leq 1 \quad (i = 1,2,\ldots,K) \\ & && p_a(x_a) = \frac{1}{\mu_a} \quad (a \in E) \end{aligned} \tag{38.17}$$

$$\mu_a - \lambda \lim_{t\to\infty} \sum_{i\in\Sigma-\Sigma^N} \pi_i(t) \geq \varepsilon \quad (a \in E)$$
$$x_a \geq R_a^{min} \quad (a \in E)$$
$$x_a \leq R_a^{max} \quad (a \in E)$$
$$\sum_{a\in E} x_a \leq R$$

38.3.1 Goal Attainment Method

We now need to use a multi-objective method to solve (OCP). We actually use a goal attainment technique for this purpose, because it is simple and computationally efficient. The goal attainment method needs to determinate a goal, g_μ, and a weight, w_μ, for each objective function. w_μ's represent the importance of the μ-th objective, where, if an objective has the smallest w_μ, then it will be the most important objective. w_μ's ($\mu = 1, 2, 3$) are normalized such that $\sum_{\mu=1}^{3} w_\mu = 1$. To determine g_μ for the μ-th objective, we have to solve the corresponding single objective problem first and then set the value of g_μ very close to the optimal single objective value. The goal attainment method is actually a variation of the goal programming method intending to minimize the maximum weighted deviation from the goals.

Since the goal attainment method has fewer variables to work with, compared to other simple and interactive multi-objective methods, it will be computationally faster and more suitable to solve the complex optimization problem (OCP). The resulting goal attainment formulation of the resource allocation problem (RAP) is given by

$$\begin{aligned}
&\text{Min.} \quad z \\
&\text{s.t.} \quad \frac{\left(b - \sum_{a\in E} C_a(x_a)\right)^2}{\sum_{a\in E} V_a(x_a)} + w_1 z \geq g_1 \qquad (38.18)\\
&\lim_{t\to\infty} \frac{\sum_{j=1}^{N} \sum_{i\in\Sigma^j} j\pi_i(t)}{\lambda \sum_{i\in\Sigma-\Sigma^N} \pi_i(t)} - w_2 z \leq g_2 \\
&\lim_{t\to\infty} \pi_1(t) - w_3 z \leq g_3 \\
&\pi'(t) = \pi(t) \cdot G(\mu) \\
&\pi_i(0) = 0 \quad (i = 2, \ldots, K) \\
&\pi_1(0) = 1 \\
&\pi_i(t) \leq 1 \quad (i = 1, 2, \ldots, K)
\end{aligned}$$

$$p_a(x_a) = \frac{1}{\mu_a} \quad (a \in E)$$

$$\mu_a - \lambda \lim_{t \to \infty} \sum_{i \in \Sigma - \Sigma^N} \pi_i(t) \geq \varepsilon \quad (a \in E)$$

$$x_a \geq R_a^{min} \quad (a \in E)$$

$$x_a \leq R_a^{max} \quad (a \in E)$$

$$\sum_{a \in E} x_a \leq R$$

38.4 Particle Swarm Optimization Algorithm

The continuous-time stochastic programming problem (RAP) is impossible to solve in this form (see Azaron and Tavakkoli-Moghaddam 2007 for more details). Therefore, we apply a particle swarm optimization (PSO) approach to solve (RAP), using a goal attainment formulation. PSO is a population-based stochastic optimization technique proposed by Kennedy and Eberhart (1995), inspired by social behavior of bird flocking or fish schooling. Eberhart and Kennedy (1995) and Eberhart et al. (1996) soon extended the PSO into a powerful optimization method. In the PSO algorithm, a number of simple particles are randomly selected in the search space of some problems or functions, and each evaluates the objective function at its current position. Each particle's movement is determined by considering some aspects of the history of its own current and best (best-fitness) positions with those of one or more components of the swarm, with some random perturbations. The next iteration occurs after all particles have been moved and it is expected to move the swarm toward the better solutions.

Each individual in the particle swarm consists of three m-dimensional vectors, namely, the current position *pos(p)*, the previous best position $pos^{best}(p)$, and the velocity *vel(p)*, where m is the dimensionality of the search space. Moreover, the current location *pos(p)* can be assumed as a set of coordinates describing a point in the space. On each iteration of the PSO algorithm, the current location is evaluated as a problem solution. If the current location is better than any that has been obtained so far, then the coordinates are saved in $pos^{best}(p)$. For comparison with next iterations, the value of the best function result so far is also saved in variable $\varphi^{best}(p)$. Indeed, the aim is to find better locations and updating $pos^{best}(p)$ and $\varphi^{best}(p)$. New positions are obtained by adding *vel(p)* coordinates to *pos(p)*, and the algorithm is controlled by regulating *vel(p)*.

A particle swarm by itself has almost no ability to solve any problem and improvement takes place only when the particles interact together. PSO is a population-wide phenomenon, while populations are formed based on some sort of communication structure or topology. In the particle swarm optimization process, each particle interacts with some other points and is influenced by the best particle

found by any member of its topological neighborhood. The velocity of each particle is also iteratively regulated so that the particle randomly fluctuates around $pos^{best}(p)$ and pos^{best} positions, where pos^{best} is the best known position that has been found so far.

In order to have the simple form of (RAP) and to prepare it for implementing the proposed PSO algorithm, we reformulate it as follows (RAP'):

$$
\begin{aligned}
\text{Min.} \quad & z = \max\left\{ \frac{g_1 - \frac{\left(b - \sum_{a\in E} C_a(x_a)\right)^2}{\sum_{a\in E} V_a(x_a)}}{w_1}, \frac{\lim_{t\to\infty} \frac{\sum_{j=1}^{N} \sum_{i\in\Sigma^j} j\pi_i(t)}{\lambda \sum_{i\in\Sigma-\Sigma^N} \pi_i(t)} - g_2}{w_2}, \frac{\lim_{t\to\infty} \pi_1(t) - g_3}{w_3} \right\} \\
\text{s.t.} \quad & \pi'(t) = \pi(t)\cdot G(\mu), \quad \pi(0) = [1 \quad 0 \quad \ldots. \quad 0] && \text{(a)} \\
& p_a(x_a) = \frac{1}{\mu_a} \quad (a \in E) && \text{(b)} \\
& \mu_a - \lambda \lim_{t\to\infty} \sum_{i\in\Sigma-\Sigma^N} \pi_i(t) \geq \varepsilon \quad (a \in E) && \text{(c)} \\
& \sum_{a\in E} x_a \leq R && \text{(d)} \\
& x_a \in \left[R_a^{min}, R_a^{max}\right] \quad (a \in E)
\end{aligned}
\tag{38.19}
$$

Moreover, to determine the fitness function of the PSO algorithm, we use a Lagrangian function. The Lagrangian function consists of objective function plus the sum of penalty terms corresponding to the constraints of the model. With regard to the objective function of the model (RAP'), the fitness function of the PSO algorithm, $\varphi(x)$, is given in Eq. (38.20). This equation is the original objective function plus the penalty terms corresponding to the violation of constraints (c) and (d) in RAP'. Parameters γ_1 and γ_2 are penalty coefficients, which should be relatively large.

$$
\begin{aligned}
\varphi(x) = \max\left\{ \frac{g_1 - \frac{\left(b - \sum_{a\in E} C_a(x_a)\right)^2}{\sum_{a\in E} V_a(x_a)}}{w_1}, \frac{\lim_{t\to\infty} \frac{\sum_{j=1}^{N} \sum_{i\in\Sigma^j} j\pi_i(t)}{\lambda \sum_{i\in\Sigma-\Sigma^N} \pi_i(t)} - g_2}{w_2}, \frac{\lim_{t\to\infty} \pi_1(t) - g_3}{w_3} \right\} + \\
\gamma_1 \sum_{a\in E} \max\left\{ \varepsilon - \mu_a + \lambda \lim_{t\to\infty} \sum_{i\in\Sigma-\Sigma^N} \pi_i(t), 0 \right\} + \gamma_2 \max\left\{ \sum_{a\in E} x_a - R, 0 \right\}
\end{aligned}
\tag{38.20}
$$

Let σ_{pop} and n_{iter}^{max} be the size of the population and the maximum number of iterations, respectively. The population size is often determined empirically on the basis of the dimensionality and perceived difficulty of the problem. However, values

in the range 20–50 are quite common for the population size. In this chapter, the number of n_{iter}^{max} iterations is considered as stopping criterion. Note that usually a sufficiently good fitness or a maximum number of iterations is determined as the stopping criterion in the PSO algorithm. To present the solution representation scheme in PSO, we also use $pos_k(p) = (x_{p,a}^k)_{a\in E}^{\mathrm{T}}$ as the designator for the position of particle p at the k-th iteration, i.e., the amount of resources allocated to service stations for particle p at the k-th iteration, where $x_{p,a}^k \in \left[R_a^{min}, R_a^{max}\right] \quad (a \in E)$, $p = 1, \ldots, \sigma_{pop}$, $k = 1, \ldots, n_{iter}^{max}$. Let $pos_k^{best}(p)$ and pos_k^{best} be the best position of particle p and the best known swarm position until the k-th iteration, respectively.

It is also assumed that $vel_k(p)$ is the velocity of particle p at the k-th iteration, where each component of $vel_k(p)$ is kept within the range $[-vel^{max}, vel^{max}]$. Note that determining the optimal value of vel^{max} is problem-specific in PSO; however, no reasonable rule of thumb is known. In addition, let $\varphi_k^{best}(p)$ and φ_k^{best} be the best fitness function of particle p and the best known swarm fitness function until the k-th iteration, respectively.

PSO needs proper acceleration coefficients α_1 and α_2, and also inertia weight ω to warrant that the algorithm converges to a good solution. The treatment of a PSO changes fundamentally with the value of α_1 and α_2, while these parameters specify the magnitude of the random forces in the direction of personal best $pos_k^{best}(p)$ and neighborhood best pos_k^{best} in each iteration. The values $\alpha_1 = \alpha_2 = 2.0$ are almost ubiquitously adopted in early PSO research (for more details see Shi and Eberhart 1998; Poli et al. 2007). In addition, we consider the linear inertia weight reduction of PSO as $\omega_{k+1} = \zeta \cdot \omega_k$, where ω_k is the inertia weight of the k-th iteration and ζ is the decrement factor.

Consequently, the proposed PSO algorithm to solve the problem (RAP') is presented as follows:

Initial stage:

- Determine the values of the population size σ_{pop}, the number of iteration n_{iter}^{max}, acceleration coefficients α_1 and α_2, the inertia weight ω_1, the penalty coefficients γ_1 and γ_2, and ε.
- **for** $p = 1, \ldots, \sigma_{pop}$
  Randomly initialize particle positions $pos_1(p) = (x_{p,a}^1)_{a\in E}^{\mathrm{T}}$, where $x_{p,a}^1 \in \left[R_a^{min}, R_a^{max}\right] \quad (a \in E)$.
- **for** $p = 1, \ldots, \sigma_{pop}$
  Randomly initialize particle velocities $vel_1(p)$, where each component of $vel_1(p)$ is kept within the range of $[-vel^{max}, vel^{max}]$.
- **for** $p = 1, \ldots, \sigma_{pop}$
  $\varphi_1^{best}(p) := M$,
  and also $\varphi_1^{best} := M$, where M is a large value.
- Set the counter $k := 1$.

Optimizing loop (Repeat):

- Obtain μ_a's $(a \in E)$ based on the constraint (b) in (RAP'), $p_a(x_{p,a}^k) = 1/\mu_a$, and obtain the matrix G for each particle p $(p = 1, \ldots, \sigma_{pop})$.

- Solve the system of differential equations based on constraint (a) in (RAP'), and compute $\pi_i(T')$ $(i = 1, \ldots, K)(\lim_{t\to\infty} \pi_i(t))$ for each particle p $(p = 1, \ldots, \sigma_{pop})$, where T' is a large constant.
- Calculate the fitness function, $\varphi_p^k(pos_k(p))$, according to (38.20) for each particle p $(p = 1, \ldots, \sigma_{pop})$.
- **for** $p = 1, \ldots, \sigma_{pop}$
  if $\varphi_p^k(pos_k(p)) < \varphi_k^{best}(p)$ **then**
  $pos_k^{best}(p) := pos_k(p)$
  $\varphi_k^{best}(p) := \varphi_p^k(pos_k(p))$
- **for** $p = 1, \ldots, \sigma_{pop}$
  if $\varphi_p^k(pos_k(p)) < \varphi_k^{best}$ **then**
  $pos_k^{best} := pos_k(p)$
  $\varphi_k^{best} := \varphi_p^k(pos_k(p))$
- **for** $p = 1, \ldots, \sigma_{pop}$
  $vel_{k+1}(p) := \omega_k vel_k(p) + \alpha_1 u_1(pos_k^{best}(p) - pos_k(p)) + \alpha_2 u_2(pos_k^{best}(p) - pos_k(p))$, where u_1 and u_2 represent vectors of random numbers uniformly distributed in [0, 1]. Also, each component of $vel_{k+1}(p)$ is kept within the range of $[-vel^{max}, vel^{max}]$.
- **for** $p = 1, \ldots, \sigma_{pop}$
  $pos_{k+1}(p) := pos_k(p) + vel_{k+1}(p)$, where $x_{p,a}^{k+1} \in \left[R_a^{min}, R_a^{max}\right]$ $(a \in E)$.
- $\omega_{k+1} := \zeta \cdot \omega_k$,where ζ is the decrement factor.
- $k := k + 1$

Until the stopping criterion is met $(k > n_{iter}^{max})$.

- Display pos_k^{best} and φ_k^{best} as best solution.

End.

To show the effectiveness of the proposed PSO approach, we also compare the associated results against the results of a discrete-time approximation of the problem (RAP), where the differential equations are converted into difference equations. Let T' be the time from which on the system is in the steady state, which we divide it into $\tau(= T'/\Delta t)$ equal portions with the length of Δt. Consequently, the corresponding discrete model (DRAP) (see Azaron and Tavakkoli-Moghaddam 2007) is as follows:

$$\text{Min.} \quad z$$

$$\text{s.t.} \quad \frac{\left(b - \sum_{a\in E} C_a(x_a)\right)^2}{\sum_{a\in E} V_a(x_a)} - w_1 z \le g_1$$

$$\frac{\sum_{j=1}^{N} \sum_{i\in\Sigma^j} j\pi_i(\tau)}{\lambda \sum_{i\in\Sigma-\Sigma^N} \pi_i(\tau)} - w_2 z \le g_2$$

$$\pi_1(\tau) - w_3 z \le g_3$$

$$
\begin{aligned}
&\pi(r+1) = \pi(r) + \pi(r) \cdot G(\mu)\Delta t \quad (r = 0, 1, 2, \ldots, \tau - 1) \\
&\pi_i(0) = 0 \quad (i = 2, \ldots, K) \\
&\pi_1(0) = 1 \\
&\pi_i(r) \leq 1 \quad (i = 1, \ldots, K), \quad (r = 1, 2, \ldots, \tau) \\
&p_a(x_a) = \frac{1}{\mu_a} \quad (a \in E) \\
&\mu_a - \lambda \sum_{i \in \Sigma - \Sigma^N} \pi_i(\tau) \geq \varepsilon \quad (a \in E) \\
&x_a \geq R_a^{min} \quad (a \in E) \\
&x_a \leq R_a^{max} \quad (a \in E) \\
&\sum_{a \in E} x_a \leq R
\end{aligned}
\tag{38.21}
$$

38.5 Numerical Example

To illustrate the proposed PSO algorithm, we consider the network depicted in Fig. 38.1 taken from Yaghoubi et al. (2011a). It is assumed that we have a system with the capacity of two concurrent projects and six service stations depicted as the AoN graph in Fig. 38.1.
The assumptions are:

- The new projects including all their activities are generated according to a Poisson process with the rate of $\lambda = 3$ per year.
- The activity durations (service times) are independent random variables with exponential distributions.
- There is one server in every service station located in the nodes.
- The capacity of the system is two projects.
- The direct costs of service stations per period are independent random variable.
- The amount of resource available to be allocated to all service stations is 18.
- The value of ε is equal 0.01.

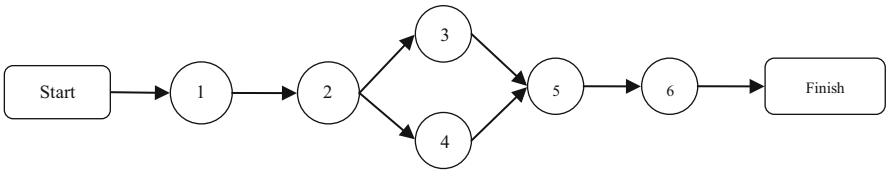


Fig. 38.1 AoN network

Table 38.1 Characteristics of the activities

Activity (a)	$C_a(x_a)$	$V_a(x_a)$	$p_a(x_a)$	R_a^{min}	R_a^{max}
1	$2x_1^2 + 1$	x_1	$0.5 - 0.05x_1$	1	5
2	x_2	$x_2/4$	$0.6 - 0.1x_2$	1	4
3	$3x_3 + 4$	$x_3/2$	$0.7 - 0.12x_3$	1	4
4	$x_4 + 3$	$x_4/3$	$0.8 - 0.08x_4$	1	6
5	$2x_5$	$x_5/4$	$0.4 - 0.03x_5$	1	5
6	$x_6 + 2$	$x_6/3$	$0.5 - 0.06x_6$	1	5

Table 38.2 Admissible three-partition cuts

1. $[\emptyset, \emptyset]$	8. $[(3, 4), 1]$	15. $[6, \emptyset]$	22. $[(3^*, 4), (3, 4^q)]$	29. $[5, (3^*, 4)]$
2. $[1, \emptyset]$	9. $[2, 2^q]$	16. $[5, 1]$	23. $[(3, 4^*), (3^q, 4)]$	30. $[5, (3, 4^*)]$
3. $[2, \emptyset]$	10. $[(3, 4^*), \emptyset]$	17. $[(3^*, 4), 2]$	24. $[6, 2]$	31. $[6, (3^*, 4)]$
4. $[1, 1^q]$	11. $[5, \emptyset]$	18. $[(3, 4^*), 2]$	25. $[5, (3, 4)]$	32. $[5, 5^q]$
5. $[(3, 4), \emptyset]$	12. $[(3^*, 4), 1]$	19. $[(3, 4), (3^q, 4^q)]$	26. $[(3^*, 4), (3^*, 4^q)]$	33. $[6, (3, 4^*)]$
6. $[2, 1]$	13. $[(3, 4), 2]$	20. $[6, 1]$	27. $[(3, 4^*), (3^q, 4^*)]$	34. $[6, 5]$
7. $[(3^*, 4), \emptyset]$	14. $[(3, 4^*), 1]$	21. $[5, 2]$	28. $[6, (3, 4)]$	35. $[6, 6^q]$

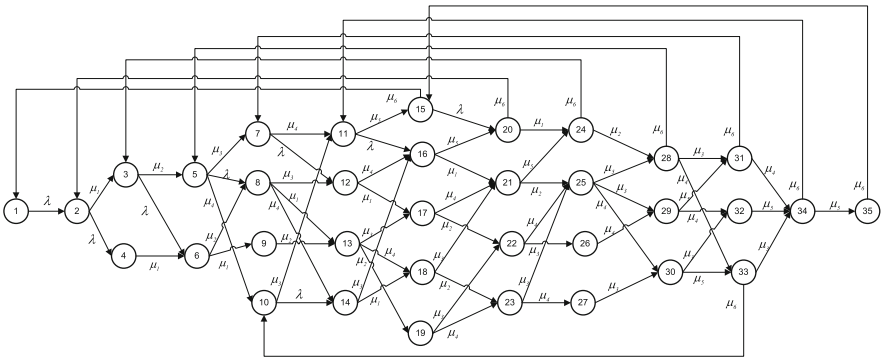


Fig. 38.2 Rate diagram for the continuous-time Markov chain

Table 38.1 shows the characteristics of the activities, where the time unit and the cost unit are, respectively, in year and in thousand dollars.

In Table 38.2, all admissible three-partition cuts of the network of Fig. 38.1 are presented, where we use superscript star and q to denote "dormant" and "in queue" activities, respectively. Figure 38.2 shows the rate diagram, where the nodes represent the states of the system.

After determining the system states and transition rates depicted in Table 38.2 and Fig. 38.2, we obtain the infinitesimal generator matrix $G(\mu)$. Table 38.3 shows the infinitesimal generator matrix $G(\mu)$, where diagonal components are equal to the negative sum of the other components at the same row.

Table 38.3 Matrix $G(\mu)$

State	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
1		λ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0		μ_1	λ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0		0	μ_2	λ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0		0	μ_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0		0	μ_3	λ	0	μ_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0		0	μ_2	μ_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0		0	0	0	μ_4	λ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0		0	0	0	μ_3	μ_1	μ_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0		0	0	0	μ_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0		μ_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0		0	0	0	μ_5	λ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0		0	0	0	μ_4	μ_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	μ_3	μ_4	μ_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0		0	μ_3	0	μ_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	μ_6	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	λ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	μ_5	μ_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	μ_4	μ_2	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	μ_5	0	μ_2	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	μ_3	μ_4	0	0	0	0	0	0	0	0	0	0	0	0
20	0	μ_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	μ_1	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	μ_5	μ_2	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	μ_4	μ_3	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	μ_3	0	μ_4	0	0	0	0	0	0	0	0
24	0	0	μ_6		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	μ_2	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	μ_5	μ_3	μ_4	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	μ_2	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	μ_3	0	0	0	0	0
28	0	0	0	0	μ_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	μ_3	0	μ_4	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	μ_5	μ_4	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	μ_3	μ_5	0	0
31	0	0	0	0	0	0	μ_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	μ_4	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	μ_4	0
33	0	0	0	0	0	0	0	0	0	μ_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		μ_3	0
34	0	0	0	0	0	0	0	0	0	0	μ_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		μ_5
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	μ_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 38.4 Classification of states

Number of projects	State	Σ^p
0	1	Σ^0
1	2,3,5,7,10,11,15	Σ^1
2	4, 6, 8, 9, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35	Σ^2

To illustrate the second objective, consider Table 38.4. The second objective for this example is

$$\text{Min.} \lim_{t\to\infty} \frac{\left[\pi_2(t)+\pi_3(t)+\pi_5(t)+\pi_7(t)+\sum_{i=10}^{11}\pi_i(t)+\pi_{15}(t)\right]}{\lambda\left[\pi_1(t)+\pi_2(t)+\pi_3(t)+\pi_5(t)+\pi_7(t)+\sum_{i=10}^{11}\pi_i(t)+\pi_{15}(t)\right]}$$

$$+\frac{2\left[\pi_4(t)+\pi_6(t)+\sum_{i=8}^{9}\pi_i(t)+\sum_{i=12}^{14}\pi_i(t)+\sum_{i=16}^{35}\pi_i(t)\right]}{\lambda\left[\pi_1(t)+\pi_2(t)+\pi_3(t)+\pi_5(t)+\pi_7(t)+\sum_{i=10}^{11}\pi_i(t)+\pi_{15}(t)\right]}$$

The aim is to obtain the optimal resource allocation to the different activities. For this purpose, we consider the goals, $g_1 = 1.65^2$, $g_2 = 1.3$, $g_3 = 0.05$, and the following sets of weighted parameters c for the three objectives to generate a set of Pareto-optimal solutions according to the goal attainment formulation (RAP):

Set 1: $w_1 = 0.3,\ w_2 = 0.1,\ w_3 = 0.6,$
Set 2: $w_1 = 0.2, w_2 = 0.1,\ w_3 = 0.7,$
Set 3: $w_1 = 0.2,\ w_2 = 0.2,\ w_3 = 0.6,$

According to proposed PSO algorithm, we set the penalty coefficients to be $\gamma_1 = 10$, $\gamma_2 = 20$, the maximum number of iterations to be $n_{iter}^{max} = 100$ or $n_{iter}^{max} = 200$, and acceleration coefficients to be $\alpha_1 = \alpha_2 = 2.0$. Moreover, we set the inertia weight to be $\omega_1 = 1$, and the decrement factor to be $\zeta = 0.99$. For this example, we consider $vel^{max} = 0.1(R_a^{max} - R_a^{min})$ for every service station and all PSO experiments are replicated five times using different random initial solutions. To do so, we use MATLAB 7 on a PC Pentium 4, CPU 3 GHz.

The optimal allocated resources according to PSO algorithm, the computational times, t_{cpu} (mm:ss), and also the values of all objectives functions for the different combinations of T', σ_{pop}, and n_{iter}^{max} for the first set of weights ($w_1 = 0.3, w_2 = 0.1$, $w_3 = 0.6$) are shown in Table 38.5.

So, the optimal allocated resources for set 1 are $x_1 = 1.667$, $x_2 = 2.940$, $x_3 = 2.862$, $x_4 = 5.272$, $x_5 = 1.036$, $x_6 = 4.222$, and the objective function values are: $f_1 = 0.178$, $f_2 = 2.148$, $f_3 = 0.039$ ($z = 8.483$). Based on Table 38.5,

Table 38.5 The computational results of the PSO algorithm for set 1 ($w_1 = 0.3, w_2 = 0.1, w_3 = 0.6$)

T'	σ_{pop}	$n_{iter}^{\max}$	x_1	x_2	x_3	x_4	x_5	x_6	z	f_1	f_2	f_3	t_{cpu}
18.0	30	100	1.519	2.672	2.290	4.208	2.945	4.366	9.134	0.192	2.213	0.057	$00' : 17''$
18.0	30	200	2.081	2.643	2.498	4.513	1.406	4.859	8.766	0.093	2.177	0.056	$00' : 19''$
18.0	50	100	1.714	3.019	2.547	4.801	1.709	4.210	8.724	0.105	2.172	0.044	$00' : 22''$
18.0	50	200	1.640	2.849	2.880	5.272	1.260	4.098	8.576	0.149	2.158	0.040	$00' : 28''$
19.2	30	100	2.942	3.007	2.145	4.411	1.900	3.595	9.287	6.711	2.229	0.046	$00' : 15''$
19.2	30	200	1.586	2.667	2.464	4.396	2.038	4.849	8.845	0.251	2.185	0.065	$00' : 19''$
19.2	50	100	1.621	2.699	2.962	5.315	1.326	4.077	8.669	0.122	2.167	0.040	$00' : 26''$
19.2	50	200	1.601	2.693	3.079	5.488	1.061	4.076	8.566	0.153	2.157	0.036	$00' : 30''$
21.6	30	100	2.031	2.278	2.004	4.671	3.477	3.522	9.714	0.278	2.271	0.081	$00' : 18''$
21.6	30	200	2.182	2.827	2.435	4.618	1.696	4.241	8.876	0.348	2.188	0.047	$00' : 22''$
21.6	50	100	1.616	2.764	2.893	5.264	1.495	3.968	8.672	0.121	2.167	0.042	$00' : 25''$
21.6	50	200	1.667	2.940	2.862	5.272	1.036	4.222	8.483	0.178	2.148	0.039	$00' : 31''$
25.2	30	100	1.202	2.082	1.999	4.251	3.562	4.904	9.248	0.784	2.225	0.062	$00' : 17''$
25.2	30	200	1.736	2.728	2.558	4.615	1.548	4.814	8.720	0.106	2.172	0.055	$00' : 21''$
25.2	50	100	1.560	2.543	3.089	5.448	1.557	3.804	8.774	0.902	2.177	0.041	$00' : 25''$
25.2	50	200	1.818	3.548	2.523	5.102	1.015	3.993	8.648	0.128	2.165	0.042	$00' : 32''$

Table 38.6 Pareto-optimal solutions obtained by the PSO algorithm

w	x_1	x_2	x_3	x_4	x_5	x_6	z	f_1	f_2	f_3	t_{cpu}
Set 1	1.667	2.940	2.862	5.272	1.036	4.222	8.483	0.178	2.148	0.039	$00' : 31''$
Set 2	1.276	3.130	2.813	5.294	1.184	4.303	8.438	1.035	2.144	0.039	$00' : 29''$
Set 3	1.078	3.396	2.701	5.268	1.049	4.508	4.166	1.889	2.133	0.038	$00' : 31''$

Table 38.7 Pareto-optimal solutions obtained by the discrete-time approximation technique

w	x_1	x_2	x_3	x_4	x_5	x_6	z	f_1	f_2	f_3	t_{cpu}
Set 1	1	3.552	3.184	5.265	1	3.999	8.257	1.343	2.126	0.044	$10' : 02''$
Set 2	2.293	3.587	3.039	5.378	1	2.702	8.493	1.024	2.149	0.042	$14' : 26''$
Set 3	1	3.714	2.849	4.677	1	4.761	4.116	1.899	2.123	0.052	$11' : 43''$

if the values of σ_{pop} and $n_{iter}^{\max}$ are increased, for any specific T', the quality of solution is increased. However, the computational times are also increased, which is undesirable.

Table 38.6 summarizes the results obtained by the PSO algorithm for the three sets of weight parameter combinations.

Now, we consider various combinations of T', τ, and Δt. To do so, we use LINGO 8 on a PC Pentium 4, CPU 3 GHz. The optimal allocated resources, the computational time, t_{cpu}, and also the values of all objectives functions for the three sets of parameter combinations w are given in Table 38.7.

Table 38.8 Comparing the PSO results against the discrete-time approximation technique with the same t_{cpu} for the parameters of set 1

No.	T'	Discrete-time approximation technique				PSO algorithm		$\left\lvert z^{\emptyset}_{PSO} - z^{\emptyset}_{Dis} \right\rvert$	Rank
		τ	Δt	$z^{\emptyset}_{Dis}$	t_{opu}	$z^{\emptyset}_{PSO}$	t_{opu}		
1	18.0	100	0.18	8.904	$03' : 15''$	8.276	$03' : 15''$	0.628	14
2	18.0	120	0.15	8.626	$05' : 31''$	8.175	$05' : 31''$	0.451	10
3	18.0	150	0.12	8.423	$06' : 52''$	8.159	$06' : 52''$	0.264	4
4	18.0	180	0.10	8.318	$08' : 34''$	8.146	$08' : 34''$	0.172	1
5	19.2	107	0.18	8.939	$04' : 48''$	8.261	$04' : 48''$	0.678	16
6	19.2	128	0.15	8.713	$06' : 09''$	8.163	$06' : 09''$	0.550	12
7	19.2	160	0.12	8.572	$08' : 11''$	8.072	$08' : 11''$	0.500	11
8	19.2	192	0.10	8.310	$09' : 26''$	8.048	$09' : 26''$	0.262	2
9	21.6	120	0.18	8.894	$03' : 54''$	8.305	$03' : 54''$	0.589	13
10	21.6	144	0.15	8.527	$06' : 49''$	8.152	$06' : 49''$	0.375	9
11	21.6	180	0.12	8.450	$08' : 31''$	8.116	$08' : 31''$	0.334	7
12	21.6	216	0.10	8.297	$10' : 02''$	7.995	$10' : 02''$	0.302	6
13	25.2	140	0.18	8.863	$05' : 27''$	8.231	$05' : 27''$	0.632	15
14	25.2	168	0.15	8.604	$04' : 45''$	8.268	$04' : 45''$	0.336	8
15	25.2	210	0.12	8.338	$08' : 50''$	8.075	$08' : 50''$	0.263	3
16	25.2	252	0.10	8.261	$11' : 43''$	7.989	$11' : 43''$	0.272	5

As we noted, solving the goal attainment formulation (RAP) optimally and comparing the PSO results with the optimal results is impossible. Therefore, we try to compare the PSO results with the results of the discrete-time model (DRAP). As stopping criterion for the PSO algorithm we use the computational times of the LINGO solver, which is applied to the discrete-time approximation. The population size is chosen to be $\sigma_{pop} = 50$.

Table 38.8 presents the results of the discrete-time approximation technique for different combinations of the parameters and the results of the PSO algorithm based on the computational time of the discrete-time approximation technique as its stopping criterion. A paired sample Wilcoxon signed-rank test analysis with $\alpha = 0.05$ is utilized to investigate whether solutions obtained by solving the discrete-time approximation differ from the PSO algorithm with same t_{cpu} or not. The paired sample Wilcoxon signed-rank test is an alternative non-parametric method for the t-test. When the normality assumption is not satisfied or the sample size is too small, the t-test is not valid (for more details see Siegel 1956), and the paired sample Wilcoxon signed-rank test is then used. Let n be the sample size, i.e., the number of pairs. For $\ell = 1, \ldots, n$, let z^{ℓ}_{Dis} and z^{ℓ}_{PSO} be the objective function values obtained by the discrete-time approximation algorithm and the PSO algorithm with the same amount of t_{cpu}, respectively. Moreover, let $z^{\emptyset}_{Dis}$ and $z^{\emptyset}_{PSO}$ be the mean of the objective function values obtained by the discrete-time approximation algorithm and the PSO algorithm with the same amount of t_{cpu}, respectively. The null hypothesis (H_0) is

defined as $z_{Dis}^{\emptyset} - z_{PSO}^{\emptyset} = 0$, while the alternate hypothesis (H_1) is $z_{Dis}^{\emptyset} - z_{PSO}^{\emptyset} > 0$. Consequently, the test procedure will be as follows:

for $\ell = 1, \ldots, n$

calculate $\left|z_{Dis}^{\ell} - z_{PSO}^{\ell}\right|$ and $sgn\left(z_{Dis}^{\ell} - z_{PSO}^{\ell}\right)$.

Exclude pairs with $\left|z_{Dis}^{\ell} - z_{PSO}^{\ell}\right| = 0$.

Order the remaining n' pairs from the smallest value of $\left|z_{Dis}^{\ell} - z_{PSO}^{\ell}\right|$ to the largest.

Rank the pairs and let R_ℓ denote the rank.

Calculate $W = \left|\sum_{\ell=1}^{n} \left(sgn\left(z_{Dis}^{\ell} - z_{PSO}^{\ell}\right) R_\ell\right)\right|$.

When n' increases, the distribution of W converges to normal. In this case, for $n' \geq 10$, a z_W–score can be computed as $z_W = (W - 0.5)/\sigma_W$, in which $\sigma_W = \sqrt{[n'(n'+1)(2n'+1)]/6}$.

if $z_W > z_{1-\alpha}$, **then** reject H_0

if ($n' < 10$ and $W \geq W_{1-\alpha}(n')$), **then** reject H_0.

Based on the results shown in Table 38.8, $z_W = 3.503$, while $z_{1-\alpha}$ is 1.65. Therefore, H_0 is rejected against H_1. Consequently, the quality of the optimal solution obtained by solving the PSO algorithm is better than the discrete-time approximation technique. This shows that the performance of the PSO algorithm is better than the discrete-time approximation technique, considering the same amount of t_{cpu}. PSO is a very good algorithm for obtaining near-optimal allocated resources. Note that similar results are obtained for the parameters of sets 2 and 3.

38.6 Conclusions

In this chapter, we developed a multi-objective model using continuous-time Markov processes for the resource allocation problem in dynamic PERT networks with a finite capacity of concurrent projects, a control mechanism called COnstant Number of Projects In Process (CONPIP). It was assumed that the new projects are generated according to a Poisson process and activity durations are independent random variables with exponential distributions. The system was represented as a queueing network with a finite number of concurrent projects, in which each activity of a project is processed at a dedicated service station with one server located in a node of the network serving activities according to the FCFS discipline. In addition, it was assumed that the direct costs of service stations per period are independent random variables.

The number of system states grows exponentially with the number of UDCs of the network and the system capacity, which is the main drawback of the proposed analytical model. However, this is a major drawback in many analytical approaches. The problem was formulated as a multi-objective model using continuous-time Markov processes with three conflicting objectives to optimally control the resources allocated to service stations. Since it is impossible to solve the original continuous-time problem to optimality in a reasonable time, we developed a PSO algorithm to solve it. Finally, to show the effectiveness of the proposed PSO, we compared the results of the discrete-time approximation of the original optimal control problem with the results obtained by the proposed PSO based on the computational times of the discrete-time approximation technique.

References

Adler PS, Mandelbaum A, Nguyen V, Schwerer E (1995) From project to process management: an empirically-based framework for analyzing product development time. Manag Sci 41:458–484

Anavi-Isakow S, Golany B (2003) Managing multi-project environments through constant work-in-process. Int J Proj Manag 21:9–18

Azaron A, Modarres M (2005) Distribution function of the shortest path in networks of queues. OR Spectr 27:123–144

Azaron A, Tavakkoli-Moghaddam R (2006) A multi-objective resource allocation problem in dynamic PERT networks. Appl Math Comput 181:163–174

Azaron A, Tavakkoli-Moghaddam R (2007) Multi-objective time–cost trade-off in dynamic PERT networks using an interactive approach. Eur J Oper Res 180:1186–1200

Azaron A, Perkgoz C, Sakawa M (2005) A genetic algorithm approach for the time-cost trade-off in PERT networks. Appl Math Comput 168:1317–1339

Azaron A, Katagiri H, Sakawa M, Kato K, Memariani A (2006) A multi-objective resource allocation problem in PERT networks. Eur J Oper Res 172:838–854

Azaron A, Katagiri H, Sakawa M (2007) Time-cost trade-off via optimal control theory in Markov PERT networks. Ann Oper Res 150:47–64

Byali RP, Kannan MV (2008) Critical chain project management: a new project management philosophy for multi project environment. J Spacecr Technol 18:30–36

Chen V (1994) A 0-1 goal programming-model for scheduling multiple maintenance projects at a copper mine. Eur J Oper Res 6:176–191

Chen PH, Shahandashti SM (2009) Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. Automat Constr 18:434–443

Cohen I, Golany B, Shtub A (2005) Managing stochastic, finite capacity, multi-project systems through the Cross Entropy methodology. Ann Oper Res 134:183–199

Cohen I, Golany B, Shtub A (2007) Resource allocation in stochastic, finite-capacity, multi-project systems through the cross entropy methodology. J Sched 10:181–193

Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the IEEE international symposium on micro machine and human science XI, Nagoya, pp 39–43

Eberhart RC, Simpson PK, Dobbins RW (1996) Computational intelligence PC tools. Academic, Boston

Fatemi Ghomi SMT, Ashjari B (2002) A simulation model for multi-project resource allocation. Int J Proj Manag 20:127–130

Gonçalves JF, Mendes JJM, Resende MGC (2008) A genetic algorithm for the resource constrained multi-project scheduling problem. Eur J Oper Res 189:1171–1190

Kanagasabapathi B, Rajendran C, Ananthanarayanan K (2009) Performance analysis of scheduling rules in resource-constrained multiple projects. Int J Ind Syst Eng 4:502–535

Kao HP, Hsieh B, Yeh Y (2006) A Petri-net based approach for scheduling and rescheduling resource-constrained multiple projects. J Chin Inst Ind Eng 23:468–477

Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks IV, pp 1942–1948

Krüger D, Scholl A (2008) Managing and modelling general resource transfers in (multi-) project scheduling. OR Spectr 32:369–394

Kulkarni V, Adlakha V (1986) Markov and Markov-regenerative PERT networks. Oper Res 34:769–781

Kumanan S, Jegan JG, Raja K (2006) Multi-project scheduling using a heuristic and a genetic algorithm. Int J Adv Manuf Tech 31:360–366

Kurtulus IS, Davis EW (1982) Multi-project scheduling: categorization of heuristic rules performance. Manag Sci 28:161–172

Kurtulus IS, Narula SC (1985) Multi-project scheduling: analysis of project performance. IIE Trans 17:58–66

Li C, Wang K (2009) The risk element transmission theory research of multi-objective risk-time-cost trade-off. Comput Math Appl 57:1792–1799

Lova A, Tormos P (2001) Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling. Ann Oper Res 102:263–286

Lova A, Maroto C, Tormos P (2000) A multicriteria heuristic method to improve resource allocation in multiproject scheduling. Eur J Oper Res 127:408–424

Nozick LK, Turnquist MA, Xu N (2004) Managing portfolios of projects under uncertainty. Ann Oper Res 132:243–256

Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. Swarm Int 1:33–57

Pritsker AAB, Watters LJ, Wolfe PM (1969) Multiproject scheduling with limited resources: a zero-one programming approach. Manag Sci 16:93–108

Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: Proceedings of the IEEE international conference on evolutionary computation, pp 69–73

Siegel S (1956) Non-parametric statistics for the behavioral sciences. McGraw-Hill, New York

Tsubakitani S, Deckro RF (1990) A heuristic for multi-project scheduling with limited resources in the housing industry. Eur J Oper Res 49:80–91

Wiest JD (1967) A heuristic model for scheduling large projects with limited resources. Manag Sci 13:359–377

Yaghoubi S, Noori S, Azaron A, Tavakkoli-Moghaddam R (2011a) Resource allocation in dynamic PERT networks with finite capacity. Eur J Oper Res 215:670–678

Yaghoubi S, Noori S, Bagherpour M (2011b) Resource allocation in multi-server dynamic PERT networks using multi-objective programming and Markov process. Iran J Sci Technol 35: 131–147

Yaghoubi S, Noori S, Azaron A, Fynes B (2014) Resource allocation in multi-class dynamic PERT networks with finite capacity. Eur J Oper Res (submitted)

Ying Y, Shou Y, Li M (2009) Hybrid genetic algorithm for resource constrained multi-project scheduling problem. J Zhejiang Univ (Eng Sci) 43:23–27

Part XIII
Robust Project Scheduling

Chapter 39
Robust Optimization for the Discrete Time-Cost Tradeoff Problem with Cost Uncertainty

Öncü Hazır, Mohamed Haouari, and Erdal Erel

Abstract Projects are subject to various sources of uncertainty that hamper reaching project targets; hence, it is crucial importance to use effective approaches to generate robust project schedules, which are less vulnerable to disruptions caused by uncontrollable factors. In this vein, this chapter examines analytical models and algorithms of robust multi-mode project scheduling, specifically, the robust discrete time-cost tradeoff problem (DTCTP). The models and algorithms presented in this chapter can support project managers from a wide range of industries in scheduling activities to minimize deviations from project goals. Furthermore, some surrogate measures that aim at providing an accurate estimate of the schedule robustness are developed and related experimental results are presented. Finally, some potential research areas are proposed and discussed.

Keywords Project management • Robust optimization • Scheduling • Uncertain cost

39.1 Introduction

In order to better control and organize activities and reach organizational targets, today, organizations are becoming more and more project-driven. As a consequence, developing effective management techniques and applying them efficiently in projects attract the attention of both practitioners and academics. In this regard,

Ö. Hazır (✉)
Faculty of Economics and Administrative Sciences, TED University, Ankara, Turkey
e-mail: oncu.hazir@tedu.edu.tr

M. Haouari
Department of Mechanical and Industrial Engineering, College of Engineering, Qatar University, Doha, Qatar
e-mail: mohamed.haouari@qu.edu.qa

E. Erel
Faculty of Business Administration, Bilkent University, Ankara, Turkey
e-mail: erel@bilkent.edu.tr

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_9

we focus on multi-mode project scheduling methods; specifically a well-known problem, the *discrete time-cost tradeoff problem* (DTCTP). A *mode* corresponds to a processing alternative of an activity, i.e., a different technology or a different resource assignment. These finite number of processing alternatives accommodate a compromise; faster ones being more costly. We examine two versions of the problem: deadline (DTCTP-D) and budget (DTCTP-B). In the deadline problem, total cost is minimized while terminating the project within the given deadline; whereas, the budget problem consists in minimizing the project duration without surpassing the given budget. Both of these multi-mode scheduling problems have practical application areas since they tackle the important time-cost tradeoff in processing activities.

DTCTP is a special case of multi-mode project scheduling. In single and multi-mode project scheduling literature, the majority of the studies assume knowledge of complete information. A detailed review of these studies can be found in Kolisch and Padman (2001), Herroelen (2005), Węglarz et al. (2011) and in Chap. 21 in the first volume of this handbook. Specializing on the problem, De et al. (1995, 1997) have respectively presented a survey on the problem and shown that the DTCTP is strongly NP-hard. For solution to optimality, Demeulemeester et al. (1996) provided a branch-and-bound algorithm; whereas Hazır et al. (2010a) developed a Benders decomposition based algorithm. On the other hand, to solve large instances, Akkan et al. (2005) and Vanhoucke and Debels (2007) proposed approximate approaches (see also Chap. 30 in the first volume of this handbook).

Although the majority of the studies in the field rely on the deterministic approaches, projects face various sources of uncertainty: some activities take more time than expected or some resources become unavailable due to reasons such as machine breakdowns. In order to reach project targets, hedging against uncertainty and proactive planning of the activities have become important. Stochastic programming, robust optimization, sensitivity analysis, parametric programming, and fuzzy programming are the main candidate optimization approaches to model and integrate project uncertainty in planning (Herroelen and Leus 2005).

In this chapter, we address robust scheduling of the DTCTP. In Sect. 39.2 we present and explain some robust optimization models, application areas. Later, in Sect. 39.3 we introduce relevant robustness measures. Finally, we discuss the rewarding topics and directions in Sect. 39.4.

39.2 Robust Optimization and Project Management

Robust optimization aims to build solutions that are insensitive to data uncertainty. As the worst-case performance of the system is focused (Kouvelis and Yu 1997), some degree of pessimism is inherent in this approach. The most common approaches involve minmax cost and minmax regret objectives (Gabrel and Murat 2010). Respectively, maximum cost and maximum regret, which is the difference between the cost of the solution and optimal one across all scenarios, are minimized. They have been applied to well known combinatorial optimization problems

(Aissi et al. 2009) and used in various application areas such as engineering and finance (Bertsimas et al. 2011).

In project management, following a scenario based approach, Yamashita et al. (2007) proposed robust optimization models for the resource availability cost problem, which consists in minimizing a cost function of resource availability. Each scenario corresponds to a realization of the uncertain durations. Differently, Cohen et al. (2007) used interval uncertainty in their recent robust scheduling study. Their study examines the effects of uncertainty on the continuous time-cost tradeoff problem. Similarly, Hazır et al. (2011) assumed interval uncertainty; but, focused on the discrete problem. For this problem, Klerides and Hadjiconstantinou (2010) addressed uncertainty in durations and used stochastic programming (see Chap. 36 of this book); whereas, Xu et al. (2012) applied fuzzy logic and presented a case study for an extension of the problem (see Chap. 59 of this book). Recently, Artigues et al. (2013) investigated minmax regret policies to model the robust version of the well-known resource-constrained project scheduling problem (see also Chap. 40 of this book).

When accurate records of past data are available to estimate probability distributions appropriately, stochastic programming has the advantage of incorporating this available information. However, as each project is unique, usually it is difficult to obtain reliable data. Therefore, in that sense, robust optimization is advantageous; furthermore, the project performance remains under control even in the worst-case conditions. In addition, the robust approach is distinctively different from sensitivity analysis, since it is proactive; it addresses uncertainty in the modeling phase.

Next, we summarize the modeling and solution approach of Hazır et al. (2011) for the robust DTCTP in the following section.

39.2.1 Robust DTCTP

Consider a project with a set of n activities and a precedence graph in AoN (activity-on-node) representation, $G = (V, E)$, where V, the node set, contains n activities and two dummy nodes, 0 and $n + 1$, that signify project start and termination. $E \subseteq V \times V$, the arc set corresponds to precedence relationships among activities. Each activity $j \in V$ can be carried out in one of the $|\mathscr{M}_j|$ modes, where each one, $m \in \mathscr{M}_j$, is defined by duration p_{jm} and a cost interval $\left[\underline{c}_{jm}, \overline{c}_{jm}\right]$. We define c_{jm} and Δc_{jm} as the most likely cost estimate and maximum cost increase from the nominal value; $\Delta c_{jm} = \overline{c}_{jm} - c_{jm}$. Using this notation, a mixed-integer programming formulation of the robust DTCTP-D can be given as follows:

$$\text{Min.} \sum_{j=1}^{n} \sum_{m \in \mathscr{M}_j} c_{jm} x_{jm} + g(x) \tag{39.1}$$

$$\text{s.t.} \sum_{m \in \mathscr{M}_j} x_{jm} = 1 \quad (j = 1, \ldots, n) \tag{39.2}$$

$$C_j - C_i - \sum_{m \in \mathscr{M}_j} p_{jm} x_{jm} \geq 0 \quad ((i, j) \in E) \tag{39.3}$$

$$C_{n+1} \leq \overline{d} \tag{39.4}$$

$$g(x) = \text{Max.} \left\{ \sum_{j \in V} \sum_{m \in \mathscr{M}_j} \Delta c_{jm} x_{jm} u_j : \sum_{j \in V} u_j \leq \Gamma, u_j \in \{0, 1\} \right\} \tag{39.5}$$

In the formulation, the continuous decision variable, C_j, represents the completion time of activity j, whereas the binary variable x_{jm} assigns mode $m \in \mathscr{M}_j$ for activity j, i.e., $x_{jm} = 1$. With the objective of minimizing the total cost (Eq. 39.1), a unique mode should be chosen for each activity (Eq. 39.2), while precedence constraints (Eq. 39.3) must be satisfied and the deadline, notated with $\overline{d}$, must not be exceeded (Eq. 39.4).

Note that maximum cost deviation is formulated with function $g(x)$ (Eq. 39.5). In this function, binary vector u is used to determine the activities (at most Γ) that will be processed with the worst-case cost values, i.e., $\{j : u_j = 1\}$. Note that if $\Gamma = 0$, the deterministic DTCTP-D is obtained, whereas high values of this parameter represent risk-averse decision making behavior.

Based on the formulation given above, Hazır et al. (2011) examined three robust versions of DTCTP-D. Model formulations, exact and approximate solution approaches, and computational results were presented.

The first version assumes that uncertainty could be defined as a *cardinality constrained set* (Bertsimas and Sim 2003; Bertsimas et al. 2011); that is, only a subset of coefficients (Γ of them) reach the worst case values. For solution, Benders decomposition, which has been widely used in various combinatorial optimization problems including project and airline scheduling, was used (Erenguc et al. 1993; Li and Womer 2009; Sherali et al. 2010; Chap. 27 in the first volume of this handbook).

Regarding robustness of project schedules, an important factor to be investigated is total slack, which is the amount of time by which the activity completion could be delayed without delaying the project termination (see Garaix et al. 2013 and Chap. 41 of this book for calculating the slack values in case of interval uncertainty for activity durations). Note that larger slacks provide flexibility in scheduling and resource allocation. Moreover, activity costs and durations are interdependent, as they both depend on the amount of resource allocation. During project execution, non-critical activities could be started later or less amount of resources could be allocated to these activities. Therefore, in achieving the cost targets, these activities constitute less risk when compared to the critical ones. Hence, assigning the worst-case costs to activities with ample slacks could be unrealistically pessimistic. On the other hand, in case of disruptions, for finishing critical activities on time, more monetary resources might have to be allocated.

In this regard, in the second version, Hazır et al. (2011) considered only the *potentially critical* activities to define the possible cost deviations (*criticality-based robust model*). These activities are identified based on the total slack/activity

duration ratio. The smaller the ratio, the more critical the activity. Lastly, a third model that gives priority to critical activities over non-critical ones in calculating cost deviations was also developed. As could be expected, the second and third models are more difficult to solve; hence high quality approximate solutions were sought using tabu search.

When these approaches are compared, the first approach is the most pessimistic one and performs well under extreme scenarios. However, the second one allocates larger project buffers at the end, which has been shown to be efficient in case of variations in activity durations. The third one results in total slacks distributed more evenly among activities.

39.2.2 An Application Area

As an application area, Hazır et al. (2011) cite *Build-Operate-Transfer (BOT)* projects, in which a public service or an infrastructure investment is undertaken and operated for a specific period of time by a private firm and then transferred to a public institution. The BOT contracts offer several mutual advantages. For the public sector, they offer an alternative financing mechanism to carry out large investment projects and encourage foreign investment inflow. On the other hand, as the client is usually the government, they reduce demand and credit risks of the private sector. To have more information about the characteristics of these contracts, we refer to the paper of McCarthy and Tiong (1991).

One of the application areas of BOT model are airport and construction projects. The private enterprise constructs the airport or harbor and operates it for a predefined period of time and then transfers the right to operate to the public. The enterprise can extend the operating period via completing the construction earlier. Early completion could be highly profitable. For that reason, in case of deviations from the baseline plan, extra resources are commonly used to speed up the activities. However, these additional allocations create cost uncertainty. Protection against deviations in total cost and project duration then becomes the key concern of project managers and in this regard, decision support systems that involve robust planning approaches are valuable. In addition, considering multiple alternatives for processing activities fit the characteristics of these types of contracts.

39.3 Robustness Measures

In addition to algorithm design, another important issue in robust scheduling is to evaluate the robustness of schedules and the comparison of scheduling algorithms based on the robustness of the designs produced (example studies are: Goren and Sabuncuoglu 2008 for machine scheduling, and Chtourou and Haouari 2008 for resource-constrained project scheduling problem). For this well-known

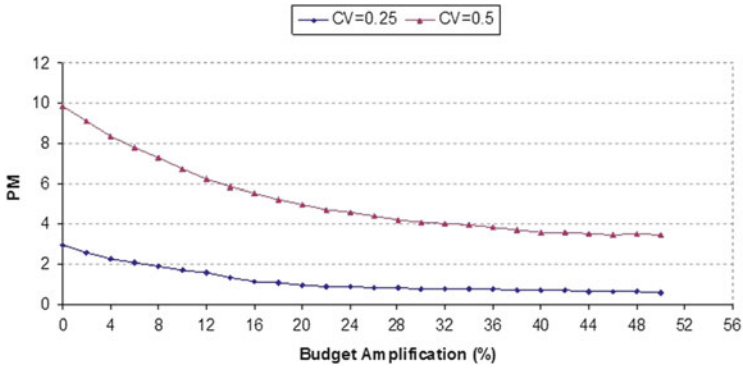


Fig. 39.1 The relationship between budget amplification and performance measure

project management problem, Al-Fawzan and Haouari (2005) followed a bi-objective approach by optimizing both robustness and project makespan. Regarding DTCTP, Hazır et al. (2010b) proposed nine time-based measures, which could be categorized as:

1. Average Slack
2. Weighted Slack
3. Slack Utility Function
4. Dispersion of Slacks
5. Percentage of Potentially Critical Activities
6. Project Buffer Size

In order to evaluate these measures, a simulation study was carried out for a large number project networks. As a result of the simulation, project buffer size and weighted slack (weights referring to the number of immediate successors of the activity) were found to be best estimates of schedule robustness.

Based on these results, a two phase scheduling algorithm to place a project buffer with minimal cost was proposed (inserting project buffers instead of augmenting the activity durations is one of the important characteristics of critical chain project scheduling). To accomplish that, an inflated budget is used; the initial budget is augmented so that the scheduling algorithm produces a schedule with a shorter termination time. Then, the effect of this budget amplification is experimentally tested using a subset of instances of Akkan et al. (2005). One of the performance measures (PM) is the average delay in the project completion time as percentage of the project deadline. That is, using this measure, schedules that exhibit less delays in case of disruptions are considered to be more robust.

In the simulation based analysis, first, a budget amplification rate is set and a baseline schedule with a project buffer is generated using this new budget value. Then, for each instance of the test bed, perturbations in processing times were randomly generated and the resulting project termination time is noted. Figure 39.1 shows some of the results. The coefficient of variation (CV) values 0.25 and 0.5

characterize small and moderate variability in activity durations. In case of low variability, PM could be significantly improved with small budget increases. For instance, with a 6 % budget increase ($\eta = 0.06$), the average delay decreased from about 3 to 2 %. Further increasing the budget accompanies a decrease in the marginal gain. However, in case of higher variability, improvement in project delays are more drastic, but larger budget increases are needed to have a stabilized pattern and eliminate tardiness.

39.4 Discussion on Future Research Directions

Possible future research topics are summarized as follows:

1. Project management has mainly focused on the variations in activity durations. However, variations in resource requirements and resource availability could also have significant consequences. Studies on variability of resource are scarce; however, considering real instances there is a need for further analysis and analytical models. In this regard, robust versions of multi-mode resource-constrained scheduling models should be investigated.
2. For multi-mode scheduling, minmax and minmax regret models could also be studied. These models are usually difficult to solve, hence the development of efficient solution algorithms, especially approximate ones for large instances, is important and interesting for further research.
3. Other than well-known minmax algorithms, some recent approaches could be used to model robust project scheduling problems. Especially two of them are appealing:
   - The bw robustness proposed by Gabrel et al. (2013) aims to obtain objective values smaller than w in all scenarios and below the target value of b as much as possible $b < w$ (in a minimization problem).
   - *Lexicographic* α-robustness reduces the influence of worst-case scenario and incorporates some tolerance with the help of the threshold α (Kalai et al. 2012). Project scheduling problems would be a relevant application area for these approaches; but establishing efficient solution algorithms for large scale projects stays to be arduous and important.
4. The majority of project scheduling studies optimizes a single performance measure, such as minimization of the project completion time. However, schedules that are developed with traditional algorithms may result in poor performances in case of disruptions. Therefore, multi criteria optimization models that combine a time or cost based objective with a robustness criterion are promising research topics and are better adapted to the requirements of industry.
5. Modeling robust versions of multi project environments systems may constitute a rewarding research field.

6. If robustness with respect to project completion time is taken as the fundamental optimization criterion, there will be tendency to schedule activities at their earliest start times (see experimental study and results of Tian and Demeulemeester 2013 on road runner scheduling). On the other hand, late starts have the advantage of decreasing WIP inventory costs, which is one of the key elements of just-in-time (JIT) management philosophy. Moreover, in large capital-intensive projects, cash flow management is vital and delaying cash outlays as much as possible is usually preferred. Therefore, an optimal compromise between project costs and robustness has to be achieved. Modeling this relationship between robust scheduling and project costs is an interesting research topic.
7. Combining reactive and robust project scheduling improves project performance. This combined approach is new in scheduling literature and referred as "proactive-reactive scheduling". This approach protects against disruptions through the combination of a proactive scheduling and a reactive improvement procedure. The baseline schedule could be created by the maximization of a robustness measure so that it involves enough safety time to absorb anticipated disruptions. Even though this baseline schedule will be insensitive to some extent, all possible disruptions might not be anticipated. For this reason, it is better to include reactive scheduling as the second protection mechanism to prevent large performance deviations due to disruptions.
8. Specific project contract types should be further investigated and specific robust planning models approaches for these projects could be developed. One example is the BOT project types, which we have already mentioned. Recently, these types of contracts have been widely signed especially in developing countries.
9. Finally, developing robust project scheduling as a model base of a DSS to assist managers is crucial. Embedding these DSS tools in a commercial software package such as in the widely used Microsoft Project system would be of considerable help to project managers. At this point, combining these scheduling algorithms with efficient project control modules that define intervening strategies in case of disruptions will be a promising application area. We refer to the study of Hazır and Schmidt (2013) as an example to integrate scheduling and control functions in multi-mode projects.

39.5 Conclusions

Robustness of solutions has been increasingly attracting the attention of researchers and practitioners in operations research and related fields. In this chapter, we examine and discuss application of robust optimization on multi-mode project scheduling. We present mathematical models to generate schedules and measures to assess their robustness. Models and extensions have practical meaning in the sense that they could constitute the model basis of a decision support system on this subject. Furthermore, discussion on research directions can facilitate identifying research topics that have theoretical values or practical applications.

References

Aissi H, Bazgan C, Vanderpooten D (2009) Min-max and min-max regret versions of combinatorial optimization problems: a survey. Eur J Oper Res 197(2):427–438

Akkan C, Drexl A, Kimms A (2005) Network decomposition-based benchmark results for the discrete time-cost tradeoff problem. Eur J Oper Res 165(2):339–358

Al-Fawzan M, Haouari M (2005) A bi-objective model for robust resource-constrained project scheduling. Int J Prod Econ 96(2):175–187

Artigues C, Leus R, Talla Nobibon F (2013) Robust optimization for resource-constrained project scheduling with uncertain activity durations. Flex Serv Manuf J 25:175–205

Bertsimas D, Sim M (2003) Robust discrete optimization and network flows. Math Program 98:49–71

Bertsimas D, Brown D, Caramanis C (2011) Theory and applications of robust optimization. SIAM Rev 53(3):464–501

Chtourou H, Haouari M (2008) A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. Comput Ind Eng 55(1):183–194

Cohen I, Golany B, Shtub A (2007) The stochastic time-cost tradeoff problem: a robust optimization approach. Networks 49(2):175–188

De P, Dunne EJ, Ghosh JB, Wells CE (1995) The discrete time-cost tradeoff problem revisited. Eur J Oper Res 81(2):225–238

De P, Dunne EJ, Ghosh JB, Wells CE (1997) Complexity of the discrete time/cost trade-off problem for project networks. Oper Res 45:302–306

Demeulemeester EL, Herroelen WS, Elmaghraby SE (1996) Optimal procedures for the discrete time/cost trade-off problem in project networks. Eur J Oper Res 88(1):50–68

Erenguc SS, Tufekci S, Zappe CJ (1993) Solving time/cost trade-off problems with discounted cash flows using generalized benders decomposition. Nav Res Log 40(1):25–50

Gabrel V, Murat C (2010) Robustness and duality in linear programming. J Oper Res Soc 61:1288–1296

Gabrel V, Murat C, Wu L (2013) New models for the robust shortest path problem: complexity, resolution and generalization. Ann Oper Res 207(1):97–120

Garaix T, Artigues C, Briand C (2013) Fast minimum float computation in activity networks under interval uncertainty. J Sched 16(1):93–103

Goren S, Sabuncuoglu I (2008) Robustness and stability measures for scheduling: single-machine environment. IIE Trans 40(1):66–83

Hazır O, Schmidt K (2013) An integrated scheduling and control model for multi-mode projects. Flex Serv Manuf J 25(1–2):230–254

Hazır O, Haouari M, Erel E (2010a) Discrete time/cost trade-off problem: a decomposition-based solution algorithm for the budget version. Comput Oper Res 37(4):649–655

Hazır O, Haouari M, Erel E (2010b) Robust scheduling and robustness measures for the discrete time/cost trade-off problem. Eur J Oper Res 207(2):633–643

Hazır O, Erel E, Gunalay Y (2011) Robust optimization models for the discrete time/cost trade-off problem. Int J Prod Econ (1):87–95

Herroelen W (2005) Project scheduling: theory and practice. Prod Oper Manag 14(4):413–432

Herroelen W, Leus R (2005) Project scheduling under uncertainty: survey and research potentials. Eur J Oper Res 165(2):289–306

Kalai R, Lamboray C, Vanderpooten D (2012) Lexicographic α robustness: an alternative to min-max criteria. Eur J Oper Res 220(3):722–728

Klerides E, Hadjiconstantinou E (2010) A decomposition-based stochastic programming approach for the project scheduling problem under time/cost trade-off settings and uncertain durations. Comput Oper Res 37(12):2131–2140

Kolisch R, Padman R (2001) An integrated survey of deterministic project scheduling. Omega-Int J Manag S 29(3):249–272

Kouvelis P, Yu G (1997) Robust discrete optimization and its applications. Nonconvex optimization and its applications, vol 14. Kluwer Academic, Dordrecht

Li H, Womer K (2009) Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. J Sched 12:281–298

McCarthy SC, Tiong RLK (1991) Financial and contractual aspects of build-operate-transfer projects. Int J Proj Manag 9(4):222–227

Sherali HD, Bae KH, Haouari M (2010) Integrated airline schedule design and fleet assignment: polyhedral analysis and benders decomposition approach. INFORMS J Comput 22(4):500–513

Tian W, Demeulemeester E (2013) On the interaction between roadrunner or railway scheduling and priority lists or resource flow networks. Flex Serv Manuf J 25:145–174

Vanhoucke M, Debels D (2007) The discrete time/cost trade-off problem under various assumptions exact and heuristic procedures. J Sched 10:311–326

Węglarz J, Józefowska J, Mika M, Waligóra G (2011) Project scheduling with finite or infinite number of activity processing modes: a survey. Eur J Oper Res 208(3):177–205

Xu J, Zheng H, Zeng Z, Wu S, Shen M (2012) Discrete time-cost-environment trade-off problem for large-scale construction systems with multiple modes under fuzzy uncertainty and its application to Jinping II hydroelectric project. Int J Proj Manag 30(8):950–966

Yamashita D, Armentano VA, Laguna M (2007) Robust optimization models for project scheduling with resource availability cost. J Sched 10:67–76

Chapter 40
Robust Optimization for the Resource-Constrained Project Scheduling Problem with Duration Uncertainty

Christian Artigues, Roel Leus, and Fabrice Talla Nobibon

Abstract In this chapter, we examine the RCPSP for the case when there is considerable uncertainty in the activity durations, to the extent that the decision maker cannot with confidence associate probabilities with the possible outcomes of a decision. Our modeling techniques stem from robust discrete optimization, which is a theoretical framework that enables the decision maker to produce solutions that will have a reasonably good objective value under any likely input data scenario. We develop and implement a scenario-relaxation algorithm and a scenario-relaxation-based heuristic. The first algorithm produces optimal solutions but requires excessive running times even for medium-sized instances; the second algorithm produces high-quality solutions for medium-sized instances and outperforms two benchmark heuristics.

Keywords Project scheduling • Robust optimization • RCPSP • Scenario relaxation • Uncertain durations

40.1 Introduction

Project parameters such as activity durations and resource requirements are seldom precisely known and usually subject to estimation errors. Uncertainty is the prime cause of incomplete and unreliable data. This uncertainty can originate from a

C. Artigues (✉)
CNRS, LAAS, Univ de Toulouse, Toulouse, France
e-mail: artigues@laas.fr

R. Leus
ORSTAT, Faculty of Economics and Business, KU Leuven, Leuven, Belgium
e-mail: Roel.Leus@kuleuven.be

F. Talla Nobibon
FedEx Europe, Middle East, Indian Subcontinent & Africa
e-mail: tallanob@gmail.com

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_10

great number of potential sources. As some of the most frequently encountered causes, we can cite activities that take more or less time than originally estimated, machine breakdowns, materials that arrive behind schedule, worker absenteeism and delays due to bad weather. Although the sources of variability in the project environment are manifold, the main scheduling objectives are mostly related to the activities' starting (or ending) times, with the project makespan being the single most studied objective, next to other ones such as weighted earliness-tardiness and net present value of the project. This justifies a restriction to the study of uncertainty in processing times only, although many different sources may be at the basis of this variability. Ultimately, inability to compensate adequately for activity-time variation is a prominent cause of project failure (see Hughes 1986).

Uncertainty in key project parameters is usually modeled in terms of probability theory (Jørgensen 1999). Malcolm et al. seem to have been the first in 1959 to recognize that randomness in the duration of a project's individual activities can be modeled by a stochastic variable. Subsequently, a large number of stochastic models for evaluating project duration have been developed, see for instance Adlakha and Kulkarni (1989), Elmaghraby (1977), Kulkarni and Adlakha (1986) and Ludwig et al. (2001). All these studies neglect (renewable) resource constraints and assume that proper resource allocation decisions have already been made at a higher decision level. As coherently described by Stork (2001), an important new aspect comes into play when moving from the deterministic to the stochastic case: what is a solution? A solution should define for each possible 'event' that occurs during the execution of the project an appropriate 'action', typically the start of new activities. To make such decisions, one may want to exploit the information given by the current state of the project. One schedule does not contain enough information to make decisions in all possible execution scenarios of the project. Stork (in line with Igelmund and Radermacher 1983, among others) uses the term 'policy' to refer to a suitable set of decision rules that constitutes a solution. In the absence of resource constraints, the minimum-makespan objective requires no real scheduling effort: it is a dominant choice to start each activity as soon as its predecessors are completed. We formally define scheduling policies and related concepts in Sect. 40.3; most of the material on scheduling policies developed for stochastic scheduling can be transferred to robust optimization without major alterations.

Decision theory distinguishes between *risk*, *uncertainty* and *ignorance*. In a risk situation, the distribution of the outcomes under study is known with certainty. This is to be contrasted with 'unmeasurable' uncertainty, in a decision-theoretic context often simply termed 'uncertainty', in which it is not possible to attribute probabilities to the possible outcomes of a decision (French 1988; Knight 1921; Rosenhead et al. 1972). The case where even the possible outcomes are not known is usually referred to as 'unawareness,' 'ignorance' or 'incomplete state space'; Loch et al. (2006) speak of 'unk unks' (unknown unknowns). Rosenhead et al. (1972) note:

> It may be possible to convert an uncertainty problem into a risk problem, for example by the subjective estimation of probabilities, and used appropriately this can be a valuable simplification. However, some aspects of the future are genuinely unknowable, even in

the probability sense. To insert notional probabilities may make the decision maker more comfortable, but that is not necessarily the objective in tackling a decision problem.

It is to be noted that the definitions of the concepts of risk, uncertainty and ignorance are still not universally accepted and remain subject to ongoing discussion—see, for instance, Walley (1991).

Our purpose is to propose models for the RCPSP that are useful when there is considerable uncertainty in the activity durations, and when the decision maker does not have sufficient confidence in the subjective probabilities that can be attributed to the different duration scenarios. Our modeling techniques stem from *robust discrete optimization*, which is a theoretical framework that enables the decision maker to produce solutions that will have a reasonably good objective value under any likely input data scenario (Aissi et al. 2009; Kasperski 2008; Kouvelis and Yu 1997).

In this chapter, we will (1) describe how robust optimization can be applied to project scheduling under uncertainty; (2) develop a scenario-relaxation algorithm to solve the optimization problem at hand; and (3) based on the scenario-relaxation algorithm, develop a heuristic procedure that produces better results than two benchmark heuristics for medium-sized instances.

The remainder of this chapter is organized as follows. First, we survey the literature on decision making under uncertainty in Sect. 40.2, with a particular focus on the objectives that are examined in the remainder of this chapter. Subsequently, we give a number of definitions and a detailed problem statement in Sect. 40.3. The evaluation of the adopted objective function is discussed in Sect. 40.4, followed by a description of an optimization routine (Sect. 40.5) and of a heuristic procedure (Sect. 40.6). The results of our computational experiments on larger datasets are presented in Sect. 40.7. Finally, a summary and some conclusions are provided in Sect. 40.8.

40.2 Decision Making Under Uncertainty

A number of criteria can be distinguished for decision making under uncertainty (for an overview, see French 1988); some of the most important ones for a minimization problem are (1) *minimax*: minimize the worst makespan realization that can occur (Wald 1950), (2) *minimin*: minimize the best outcome that can occur, which is an optimistic approach, as opposed to the pessimistic minimax (see Hurwicz 1951), who also proposes to optimize a weighted average of minimin and minimax), (3) *minimax regret*: minimize the largest possible difference in makespan between the policy to be selected and the optimal makespan for a given realization (Savage 1951), and (4) minimize the objective *in expectation*. Within the context of this chapter, the objectives (1) and (2) can be solved via the classic RCPSP, since the duration realizations of the different activities will be assumed to be independent of each other. The most-studied objective for the so-called *stochastic RCPSP* (Stork 2001; Ballestín and Leus 2009; Ashtiani et al. 2011), where a probability

distribution is known for the duration scenarios, is to select a policy that minimizes the expected value (4) of the project makespan within a specific class of policies. In the context of this chapter, however, probability distributions are not available and so expected values cannot be computed.

Assavapokee et al. (2008a) state that, because of incomplete information about the joint probability distribution of the uncertain parameters in the problem, decision makers are often unable to search for decisions with the best long-run average performance. Instead, they look for *robust* decisions, which perform well across all possible input scenarios without attempting to assign a fixed probability distribution to any ambiguous parameter. Daniels and Kouvelis (1995) motivate the choice of regret-based objectives as follows:

> A decision maker may be rightfully concerned not only with how a schedule's performance varies with the actual realizations of the task parameters, but also with how actual performance compares with the optimal performance that could have been achieved if perfect information had been available prior to scheduling. Such comparisons against optimal performance focus the decision maker on opportunities to free short-term capacity by reducing uncertainty and efficiently utilizing resources through scheduling, ...

Comparable regret-based objectives have recently been examined for various combinatorial optimization problems, see Assavapokee et al. (2008b,a), Averbakh (2000), Averbakh and Lebedev (2004), Lebedev and Averbakh (2006), Montemanni (2007), Talla Nobibon and Leus (2014a,b). Scheduling with regret-based objectives is studied by Kouvelis et al. (2000), Kouvelis and Yu (1997), Daniels and Kouvelis (1995) in a machine environment, with two objective functions: the *absolute-deviation robust* scheduling problem and the *relative-deviation robust* scheduling problem. The underlying deterministic machine problems studied in these references are easy (solvable in polynomial time). Aissi et al. (2009) also point out that they prefer to study robust versions only of problems that are solvable in polynomial or pseudo-polynomial time, in the hope that they could preserve the complexity. The RCPSP, however, is strongly $\mathcal{NP}$-hard (Błażewicz et al. 1983).

Other approaches to robust optimization can be found in the literature; we briefly discuss some of these in the following lines. An extensive survey is given by Nikulin (2006). Ben-Tal and Nemirovski (1999, 2000, 2002) find robust solutions to convex optimization problems with data uncertainty, when the data are drawn from ellipsoids; they produce solutions such that the constraints are respected whatever the realization of the data. A practical drawback of this approach is that it leads to non-linear, although convex, models, which are computationally rather demanding. Bertsimas and Sim (2003, 2004) propose an approach to address data uncertainty for discrete optimization and network-flow problems that allows the degree of conservatism of the solution to be controlled: protection is provided for the case where only a pre-specified number of the input coefficients changes from its base value, which allows to reduce the 'price of robustness' when the protection required is not too high. Finally, Mulvey et al. (1995) present an approach that integrates goal-programming formulations with a scenario-based description of the problem data; they distinguish between solutions that remain close to optimal and those that

remain 'almost feasible' and use the terms 'solution robust' and 'model robust,' respectively.

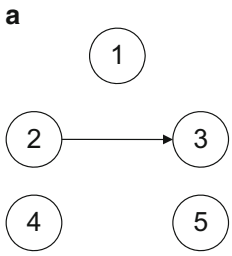
40.3 Definitions and Problem Statement

40.3.1 Project Scheduling

We examine the scheduling of a single project. The project consists of a set $V = \{0, 1, \ldots, n+1\}$ of activities that need to be performed. We associate with each activity $i \in V$ a set $\Sigma_i \subset \mathbb{R}_{\geq 0}$ containing the possible realizations of the duration of activity i. This set Σ_i can be a discrete set $\{p_{i1}, p_{i2}, p_{i3}, \ldots, p_{i|\Sigma_i|}\}$ or an interval $[p_i^{min}; p_i^{max}]$; in the first case, we also write $p_i^{min} \equiv \min_{\Sigma_i} p_{ik}$ and $p_i^{max} \equiv \max_{\Sigma_i} p_{ik}$. The (dummy) activities 0 and $n+1$ have zero duration (meaning that $P_0 = P_{n+1} = \{0\}$). We use vector $\sigma = (p_0, p_1, \ldots, p_{n+1})$ with $p_i \in \Sigma_i$ for all $i \in V$, to represent one particular *scenario* of the durations (also called *sample* or *realization*). The set containing all scenarios is denoted by $\Sigma = P_0 \times P_1 \times \ldots \times P_{n+1}$: the possible durations for one activity are not dependent on the values chosen for the other activities.

When each $|\Sigma_i| = 1$, we are in the case of the deterministic RCPSP. Each duration is a constant in this case, and so this corresponds with one scenario $\sigma = (p_0, \ldots, p_{n+1})$. A solution to the RCPSP is a schedule S, i.e., an $(n+2)$-vector of starting times $(S_0, S_1, \ldots, S_{n+1})$ with $S_i \geq 0$ for all $i \in V$. In most projects, some of the activities can only be started once other activities are completed. Such precedence relationships between the activities are represented by a binary relation $E \subset V \times V$. We assume that E is a (strict) partial order on V, i.e., an irreflexive and transitive relation. The activities 0 and $n+1$ represent the start and the end of the project, respectively, meaning that $\forall i \in V \backslash \{0\} : (0, i) \in E$, and $\forall i \in V \backslash \{(n+1)\} : (i, n+1) \in E$, or in other words, 0 and $n+1$ are predecessor, respectively successor, of all other activities. A so-called *precedence network* (V, E) is inferred, where the nodes correspond to activities and arcs represent precedence relations. For a binary relation E' on V, we let $TE = T(E')$ denote its *transitive closure*, defined as the minimal transitive relation on V that contains E'. Since E is transitive and irreflexive, (V, E) does not contain a cycle, and all precedence networks (V, E') with the same transitive closure (V, TE) represent the same scheduling instance. The schedule S is said to be *precedence feasible* if $S_i + p_i \leq s_j$ for all $(i, j) \in E$. Without loss of generality, we usually set $S_0 = 0$.

The project activities are to be scheduled on a set $\mathscr{R}$ of renewable resource types with availability R_k for each $k \in \mathscr{R}$ (e.g., groups of equivalent workers or machines). Each activity $i \in V$ occupies a fixed number $r_{ik} \in \mathbb{N}$ units of each resource type k during its execution. The activities 0 and $n+1$ do not use resources: $r_{0k} = r_{n+1,k} = 0$ for all $k \in \mathscr{R}$. A schedule S is said to be *resource feasible* if, at any time t and for each resource type $k \in \mathscr{R}$, it holds that $\sum_{i \in \mathscr{E}'(S,t)} r_{ik} \leq R_k$,



Project network

b

i	Σ_i	r_{i1}
0	$\{0\}$	0
1	$\{1,2,8\}$	1
2	$\{2,3\}$	2
3	$\{2\}$	1
4	$\{1,3\}$	2
5	$\{1,2\}$	1
6	$\{0\}$	0

Other activity data

Fig. 40.1 Example project network and activity data

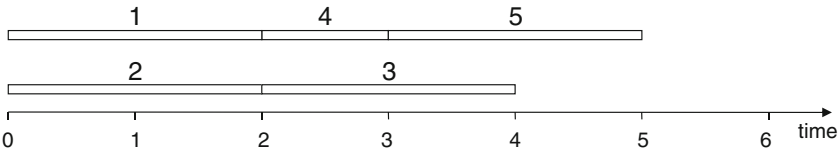


Fig. 40.2 A feasible schedule for the example project under scenario σ^1

where the *active set* $\mathscr{E}'(S,t) = \{i \in V | S_i \leq t < S_i + p_i\}$ contains the activities in $V \backslash \{0, n+1\}$ that are in progress at time t. The objective of the RCPSP is to find a precedence-feasible and resource-feasible schedule S that minimizes the project makespan S_{n+1}.

In this chapter, we examine the following problem: at the start of the project, the decision maker does not know which activity duration scenario will occur, and yet a number of sequencing decisions need to be made already (at least, he/she needs to decide which activities to release for execution at the start of the project horizon). We assume that an activity's duration realization is known only when the activity finishes (although this may implicitly be discovered earlier in the discrete case, namely as soon as the last-but-one scenario is exceeded). Sequencing decisions take the form of scheduling policies, which are the subject of the next subsection.

Figure 40.1a represents a precedence network for a small project with $n = 5$ non-dummy activities, so $V = \{0, 1, \ldots, 6\}$ (the dummy nodes 0 and 6 are omitted for brevity). In our example project, the resource availability of a single resource type ($K = |\mathscr{R}| = 1$) is $R_1 = 3$ units. All remaining data are provided in Fig. 40.1b. The schedule graphically represented in Fig. 40.2 is a feasible schedule for this project when the activity durations are the components of the vector $\sigma^1 = (0, 2, 2, 2, 1, 2, 0)$.

40.3.2 Scheduling Policies

The execution of a project with uncertain activity durations is a dynamic decision process. A solution is a *policy*, which defines *actions* at *decision times*. Decision times are typically the start of the project and the completion times of activities. An action can entail the start of a set of activities that is both precedence feasible and resource feasible. A schedule is thus constructed gradually through time. A decision at time t can only use information that has become available before or at time t; this requirement is often referred to as the *non-anticipativity constraint*. As soon as all activities are completed, the activity durations are known, yielding a realization σ.

A set of activities $F \subset V$ is a *forbidden set* of a precedence relation E' if it is an antichain of E' (a stable set in graph (V, E')) and if $\sum_{i \in F} r_{ik} > R_k$ for at least one $k \in \mathscr{R}$: these sets can give rise to resource conflicts during project execution. A subset-minimal forbidden set is called a *minimal forbidden set* or *mfs* (see for instance Stork and Uetz 2005). The set of mfss for precedence relation E' is denoted by $\mathscr{F}(E')$. For the example project presented in the previous subsection, we have $E = \{(2, 3)\}$ and $\mathscr{F}(E) = \{\{1, 2, 5\}, \{1, 3, 4\}, \{2, 4\}, \{3, 4, 5\}\}$. Several scheduling policies for projects with stochastic activity durations were presented by Igelmund and Radermacher (1983) based on the concept of forbidden sets. In this chapter, we study the set of earliest-start policies (ES-policies), which can be applied also when the probability distributions are not known. An ES-policy is characterized by a set of activity pairs $X \subset (V \times V) \setminus E$, such that for the extended set of activity pairs $E \cup X$ it holds that $\mathscr{F}(T(E \cup X)) = \emptyset$. The implication is that we can ignore the resource constraints if we respect the precedence constraints corresponding with $E \cup X$; in line with Balas (1971), we call X a *selection*. The policy is feasible if $(V, E \cup X)$ is still acyclic. A selection X of activity pairs that leads to a feasible ES-policy is called a *sufficient* set or *sufficient* selection.

An ES-policy parameterized by a sufficient selection X can be interpreted (Igelmund and Radermacher 1983; Stork 2001) as a function $\mathbb{R}^{n+2}_{\geq 0} \to \mathbb{R}^{n+2}_{\geq 0} : \sigma \mapsto S(X, \sigma)$ that maps given samples σ of activity durations to feasible schedules S. Let $(V, E \cup X, \sigma)$ denote the weighted graph where each arc $(j, k) \in E \cup X$ is valued by p_j. The starting time $S_i(X, \sigma)$ is the length of a longest path from 0 to i in $(V, E \cup X, \sigma)$, which can be determined recursively (via standard longest-path calculations in acyclic graphs). The optimal makespan $S^*_{n+1}(\sigma)$ for the RCPSP with durations σ equals

$$S^*_{n+1}(\sigma) = \min_{X \in \mathscr{X}} S_{n+1}(X, \sigma) \tag{40.1}$$

where $\mathscr{X}$ is the set containing all sufficient selections. For known durations, this model is an extension of the disjunctive-graph representation of the classical job-shop scheduling problem (Roy and Sussmann 1964), and has been known for quite some time already (see, for instance Balas 1971).

In what follows, we will use transshipment networks that represent the flow of resource units between activities; these networks are subsequently referred to as

(resource) flow networks. Such networks have recently been proposed by various sources (Artigues and Roubellat 2000; Bowers 1995; Leus and Herroelen 2004; Naegler and Schoenherr 1989; Neumann et al. 2003). In this chapter, the word *flow* usually refers to a resource flow, unless noted otherwise. A flow ϕ assigns to each triple $(i, j, k) \in V \times V \times \mathscr{R}$ a value $\phi(i, j, k) \equiv \phi_{ij}^k \in \mathbb{N}$, representing the number of resource units of type k that are transferred from the end of activity i to the start of activity j. These values must satisfy the following constraints, which are flow-conservation constraints as well as lower and upper bounds on the flow through intermediate nodes (not the start or end node):

$$\sum_{j \in V: j \neq i} \phi_{ji}^k = \sum_{j \in V: j \neq i} \phi_{ij}^k = r_{ik} \qquad (i \in V \setminus \{0, n+1\}; k \in \mathscr{R})$$

For each resource type $k \in \mathscr{R}$, R_k resource units are sent into the network from the start node and collected at the end node:

$$\sum_{j \in V: j \neq 0} \phi_{0j}^k = \sum_{j \in V: j \neq (n+1)} \phi_{j(n+1)}^k = R_k \qquad (k \in \mathscr{R})$$

We are most interested in the flow-carrying arcs that are not in E, which do not coincide with technological precedence constraints; these are gathered in the set $C(\phi) = \{(i, j) \in V \times V : \phi(i, j, k) > 0 \text{ for at least one } k \in \mathscr{R}\} \setminus E$. A flow ϕ entails a detailed resource allocation decision for the individual units of each resource type, and induces additional precedence constraints via the elements of $C(\phi)$ under the condition of invariant resource allocation (for a discussion, see Bowers 1995). We say that a flow ϕ is *feasible* when $(V, E \cup C(\phi))$ is acyclic, in which case the project can be implemented with the resource-allocation decisions inherent in ϕ.

It is obvious that for a feasible flow ϕ, $X = C(\phi)$ is a sufficient set; conversely, if the selection X defines a feasible ES-policy, then a feasible flow ϕ exists with $E \cup C(\phi) \subseteq T(E \cup X)$. A further discussion of the equivalence between ES-policies and resource flows can be found in Leus and Herroelen (2004) and Leus (2011a,b).

For the example project, the schedule depicted in Fig. 40.2 does not provide information on the detailed allocation of the activities to the individual resources. Two possible allocations corresponding with the same schedule are depicted in Fig. 40.3a, b, where each horizontal band corresponds with a resource unit (e.g., one machine); the resource units are denoted by u_i $(i = 1, 2, 3)$. The resource flow networks corresponding with Fig. 40.3a,b are depicted in Fig. 40.4a,b. The dummy activities 0 and 6 function as source and sink for the three resource units of the single resource type: the three units are dispatched into the network from activity 0 and gathered at node 6. Obviously, if more than one resource type is considered $(K > 1)$, there will be a separate flow network for each resource type.

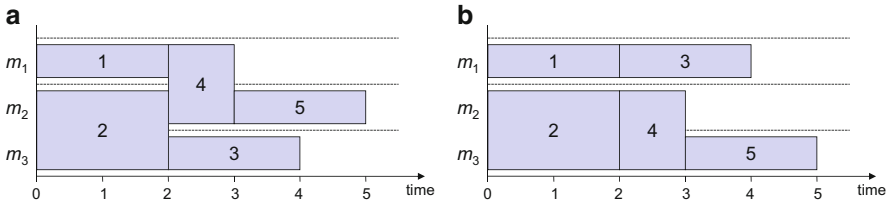


Fig. 40.3 Two possible resource allocations for the example project; the durations correspond with scenario σ^1. (**a**) Allocation 1. (**b**) Allocation 2

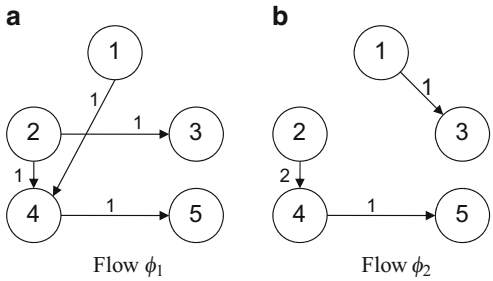


Fig. 40.4 Flow networks corresponding with the resource allocations in Fig. 40.3a,b. Flow quantities are indicated next to each arc. (**a**) Flow ϕ_1. (**b**) Flow ϕ_2

In the flow networks, some resource units are transported between activities that are not originally precedence-related (e.g., from activity 1 to 4 in case of ϕ_1). If we decide to maintain the same resource allocation throughout the execution of the project then arcs such as $(1, 4)$ in the flow network induce additional 'hard' precedence constraints. In fact, once a decision has been made regarding the allocation of resources and as long as all (original and extra) precedence constraints are respected, we can disregard resource constraints altogether and still produce a resource-feasible schedule. The schedule in Fig. 40.2, for instance, is the result of starting all activities as early as possible subject to the original precedence constraints augmented with the extra arcs from either Fig. 40.4a or b.

40.3.3 Problem Statement

In this chapter, we examine the minimax absolute-regret robust resource-constrained project scheduling problem or AR-RCPSP. The objective of the AR-RCPSP is to find an ES-policy that minimizes the maximum absolute regret over all scenarios. The absolute regret $\rho(X, \sigma)$ for a sufficient selection X and duration vector σ is the difference between the makespan $S_{n+1}(X, \sigma)$ obtained by selection X and the optimal makespan $S^*_{n+1}(\sigma)$ for σ, or

$$\begin{aligned}\rho(X,\sigma) &= S_{n+1}(X,\sigma) - S^*_{n+1}(\sigma)\\ &= S_{n+1}(X,\sigma) - \min_{Y\in\mathscr{X}} S_{n+1}(Y,\sigma)\\ &= \max_{Y\in\mathscr{X}} \{S_{n+1}(X,\sigma) - S_{n+1}(Y,\sigma)\}\end{aligned}$$

If we define the regret of a policy X relative to a policy Y as $\rho(X,Y,\sigma) = S_{n+1}(X,\sigma) - S_{n+1}(Y,\sigma)$ then $\rho(X,\sigma) = \max_{Y\in\mathscr{X}} \rho(X,Y,\sigma)$.

The maximum regret $\rho^{max}(X)$ for a given sufficient selection X is

$$\begin{aligned}\rho^{max}(X) &= \max_{\sigma\in\Sigma} \rho(X,\sigma)\\ &= \max_{\sigma\in\Sigma, Y\in\mathscr{X}} \{S_{n+1}(X,\sigma) - S_{n+1}(Y,\sigma)\} = \max_{\sigma\in\Sigma, Y\in\mathscr{X}} \rho(X,Y,\sigma)\end{aligned}$$

The optimization problem that we wish to solve can now be stated as follows:

$$(AR-RCPSP) \qquad \begin{aligned}\rho^* = \min_{X\in\mathscr{X}} \rho^{max}(X) &= \min_{X\in\mathscr{X}} \max_{\sigma\in\Sigma} \rho(X,\sigma)\\ &= \min_{X\in\mathscr{X}} \max_{\sigma\in\Sigma, Y\in\mathscr{X}} \rho(X,Y,\sigma)\end{aligned}$$

A problem closely related to AR-RCPSP is the minimax relative-regret robust resource-constrained project scheduling problem RR-RCPSP. For given X and σ, the relative regret $\tilde{\rho}(X,\sigma)$ is given by:

$$\tilde{\rho}(X,\sigma) = \frac{S_{n+1}(X,\sigma) - S^*_{n+1}(\sigma)}{S^*_{n+1}(\sigma)} = \max_{Y\in\mathscr{X}} \frac{S_{n+1}(X,\sigma)}{S_{n+1}(Y,\sigma)} - 1 = \max_{Y\in\mathscr{X}} \tilde{\rho}(X,Y,\sigma)$$

where the last equality serves as a definition for $\tilde{\rho}(\cdot,\cdot,\cdot)$. The maximum relative regret can be written as follows:

$$\tilde{\rho}^{max}(X) = \max_{\sigma\in\Sigma, Y\in\mathscr{X}} \tilde{\rho}(X,Y,\sigma)$$

The RR-RCPSP then amounts to the following problem:

$$(RR\text{-}RCPSP) \qquad \tilde{\rho}^* = \min_{X\in\mathscr{X}} \tilde{\rho}^{max}(X) = \min_{X\in\mathscr{X}} \max_{\sigma\in\Sigma} \tilde{\rho}(X,\sigma)$$

40.3.4 Objective-Function Evaluation

The RCPSP, which has known durations σ, is strongly $\mathscr{NP}$-hard, and RCPSP reduces to the evaluation of the regret for a known selection X and duration vector σ: $\rho(X,\sigma)$ is the difference between $S_{n+1}(X,\sigma)$ and $S^*_{n+1}(\sigma)$, where the first term can be obtained by a longest-path computation in $(V, E\cup X, \sigma)$ and the second term is

the optimal solution to the RCPSP instance. Consequently, once we know $\rho(X,\sigma)$, we also know $S^*_{n+1}(\sigma)$. Hence, since the RCPSP is $\mathcal{NP}$-hard, computing $\rho(X,\sigma)$ is also $\mathcal{NP}$-hard. A similar reasoning shows the $\mathcal{NP}$-hardness of evaluating the relative regret.

Since computing the regret for a fixed ES-policy and duration vector is itself $\mathcal{NP}$-hard, the computation of the maximum regret is not easy either. More precisely, evaluating the maximum absolute regret $\rho^{max}(X)$ as well as the maximum relative $\tilde{\rho}^{max}(X)$ is $\mathcal{NP}$-hard for a given ES-policy defined by X: if the set of possible durations Σ_i is a singleton for each activity $i \in V$ then $|\Sigma| = 1$ and the evaluation of $\rho^{max}(X)$, respectively $\tilde{\rho}^{max}(X)$, is equivalent to the evaluation of $\rho(X,\sigma^*)$, respectively $\tilde{\rho}(X,\sigma^*)$, where $\Sigma = \{\sigma^*\}$. In project scheduling, when activity durations are decision variables, one deals with a so-called *multi-mode* scheduling problem. The problem of evaluating the maximum regret of a given ES-policy amounts to a multi-mode resource-constrained project scheduling problem.

By similar arguments, we also see that the computation of ρ^* and $\tilde{\rho}^*$ is hard: when $|\Sigma| = 1$, minimizing the maximum regret amounts to finding a policy X with critical-path length of the extended network equal to the minimal RCPSP makespan, which is equivalent to solving the RCPSP.

40.3.5 Extreme Duration Scenarios

We will say that the following two duration scenarios are *extreme* scenarios: $\sigma^{min} = (p_1^{min},\ldots,p_n^{min})$ and $\sigma^{max} = (p_1^{max},\ldots,p_n^{max})$. According to Averbakh (2005), for the category of subset-type combinatorial optimization problems, the maximum regret can always be attained at an extreme scenario, and this for both the absolute and the relative maximum regret. In this section, we show that this is also the case for $\rho^{max}(X)$ but not for $\tilde{\rho}^{max}(X)$.

Theorem 40.1. *There is always an extreme duration scenario in which the maximum absolute regret of an ES-policy X is reached.*

Proof. For a duration vector $\sigma = (p_1,\ldots,p_n)$, let $V'(\sigma)$ denote the set of activities with a duration that is strictly between its extreme values, so $V'(\sigma) = \{i \in V \mid p_i^{min} < p_i < p_i^{max}\}$. We let $\sigma^* = (p_1^*,\ldots,p_n^*)$ and Y^* represent a duration scenario and an ES-policy that achieve the maximum regret, i.e.,

$$\rho(X,Y^*,\sigma^*) = S_{n+1}(X,\sigma^*) - S_{n+1}(Y^*,\sigma^*) = \rho^{max}(X)$$

Suppose that σ^* is not an extreme scenario (i.e., $|E(\sigma^*)| \geq 1$) and that $|E(\sigma^*)|$ is minimal. We choose an activity $j \in E(\sigma^*)$, so $p_j^{min} < p_j^* < p_j^{max}$. Recall that $S_{n+1}(X,\sigma^*)$ is equal to the length of ℓ_{X,σ^*}, which denotes the longest path in $(V, E \cup X, \sigma^*)$, while $S_{n+1}(Y^*,\sigma^*)$ is the length of ℓ_{Y^*,σ^*}, the longest path in $(V, E \cup Y^*, \sigma^*)$. Changing the duration of j from p_j^* to a different value in

P_j preserves the feasibility of ES-policies X and Y^*. Let Y' denote an ES-policy of minimal makespan for a modified duration vector $\sigma' = (p'_1, \ldots, p'_n)$, we have necessarily

$$S_{n+1}(Y', \sigma') \leq S_{n+1}(Y^*, \sigma') \tag{40.2}$$

The following four possibilities are mutually exclusive and jointly exhaustive:

1. If $j \in \ell_{X,\sigma^*}$ and $j \in \ell_{Y^*,\sigma^*}$, let σ' be such that $p'_i = p^*_i$ for $i \neq j$ and $p'_j = p^{max}_j$. ℓ_{X,σ^*} and ℓ_{Y^*,σ^*} remain the longest paths in the respective graphs with new lengths $S_{n+1}(X, \sigma') = S_{n+1}(X, \sigma^*) + p^{max}_j - p^*_j$ and $S_{n+1}(Y^*, \sigma') = S_{n+1}(Y^*, \sigma^*) + p^{max}_j - p^*_j$.
2. If $j \notin \ell_{X,\sigma^*}$ and $j \notin \ell_{Y^*,\sigma^*}$, let σ' be such that $p'_i = p^*_i$ for $i \neq j$ and $p'_j = p^{min}_j$. ℓ_{X,σ^*} and ℓ_{Y^*,σ^*} remain the longest paths with unchanged lengths $S_{n+1}(X, \sigma') = S_{n+1}(X, \sigma^*)$ and $S_{n+1}(Y^*, \sigma') = S_{n+1}(Y^*, \sigma^*)$.
3. If $j \in \ell_{X,\sigma^*}$ and $j \notin \ell_{Y^*,\sigma^*}$, let σ' be such that $p'_i = p^*_i$ for $i \neq j$ and $p'_j = p^{max}_j$. This keeps ℓ_{X,σ^*} as the longest path in $(V, E \cup X, \sigma')$ and its length is increased, so $S_{n+1}(X, \sigma') = S_{n+1}(X, \sigma^*) + p^{max}_j - p^*_j$. If a path in $(V, E \cup Y^*, \sigma')$ becomes longer than $S_{n+1}(Y^*, \sigma^*)$, its length increases by at most $p^{max}_j - p^*_j$. Hence we have $S_{n+1}(Y^*, \sigma') \leq S_{n+1}(Y^*, \sigma^*) + p^{max}_j - p^*_j$.
4. If $j \notin \ell_{X,\sigma^*}$ and $j \in \ell_{Y^*,\sigma^*}$, we let σ' be such that $p'_i = p^*_i$ for $i \neq j$ and $p'_j = p^{min}_j$. This keeps ℓ_{X,σ^*} as the longest path in $(V, E \cup X, \sigma')$ and its length is unchanged with $S_{n+1}(X, \sigma') = S_{n+1}(X, \sigma^*)$. The length of the longest path in $(V, E \cup Y^*, \sigma')$ decreases (by at most $p^*_j - p^{min}_j$ time units), so we have $S_{n+1}(Y^*, \sigma') \leq S_{n+1}(Y^*, \sigma*)$.

For each of the four listed cases together with expression (40.2), the new regret verifies $S_{n+1}(X, \sigma') - S_{n+1}(Y', \sigma') \geq S_{n+1}(X, \sigma^*) - S_{n+1}(Y^*, \sigma')$. Hence, the value of p_j can be changed to p^{min}_j or p^{max}_j without decreasing $\rho(X, Y^*, \sigma^*)$, which reduces $|E(\sigma^*)|$ and contradicts the hypothesis of minimality. In conclusion, σ^* and Y^* always exist such that $E(\sigma^*) = \emptyset$. □

Unfortunately, the same property does not hold for the relative regret. Consider a two-activity example with $P_1 = \{2, 3, 6\}$ and $P_2 = \{1, 3, 5\}$, without precedence constraints and with a resource usage such that the two activities can be scheduled in parallel. For any of the nine scenarios, the optimal makespan is $\max\{p_1, p_2\}$. Suppose we want to evaluate the maximum regret of ES-policy $X_1 = \{(1, 2)\}$. For any scenario $\sigma = (p_1, p_2)$ (we omit the dummy activities, for brevity), we have $S_{n+1}(X_1, \sigma) = p_1 + p_2$. The relative regret of ES-policy X_1 for duration scenario σ is equal to

$$\tilde{\rho}(X_1, \sigma) = \frac{p_1 + p_2}{\max\{p_1, p_2\}} - 1 = \frac{\min\{p_1, p_2\}}{\max\{p_1, p_2\}}$$

It is easily verified that the unique maximizer of this value is the duration scenario $(3, 3)$, which is not an extreme scenario. For this reason as well as due to the non-

linearity inherent in the relative regret, we will focus only on the absolute regret in the remainder of this chapter. Note that the absolute regret $\rho(X_1, \sigma)$ of policy X_1 is equal to $p_1 + p_2 - \max\{p_1, p_2\} = \min\{p_1, p_2\}$, which is maximized by the extreme scenario with $p_1 = 6$ and $p_2 = 5$.

40.3.6 Example Project

For the example project presented in Sect. 40.3.1, define $X_1 = C(\phi_1)$ and $X_2 = C(\phi_2)$, with ϕ_1 and ϕ_2 as described in Fig. 40.4. The regret of X_1 is maximized for duration vector $\sigma^2 = (p_{01}, p_{13}, p_{21}, p_{31}, p_{42}, p_{52}, p_{61}) = (0, 8, 2, 2, 3, 2, 0)$, with an optimal selection for this scenario being $Y = \{(2, 5), (4, 2)\}$, which isolates activity 1 on a separate resource unit because this activity may have a high duration (namely 8); the regret in this case is equal to the highest possible duration of activities 4 and 5, which are successors of activity 1 according to X_1. We have $\rho^{max}(X_1) = \rho(X_1, \sigma^2) = s_6(X_1, \sigma^2) - s_6(Y, \sigma^2) = 13 - 8 = 5$. Similarly, the maximum regret of X_2 equals 2, which is the duration of activity 3 (which has only one possible value).

The maximum regret $\rho^{max}(\cdot)$ is minimized by both the policies Y and X_2 and equals 2 (so $\rho^* = 2$). A maximum-regret scenario for Y is $\sigma^3 = (0, 1, 3, 2, 3, 1, 0)$, with $s_6(Y, \sigma^3) = 8$ while $s_6(Z, \sigma^3) = 6$, where $Z = \{(1, 3), (2, 1), (2, 4), (5, 4)\}$. The value of $\rho^{max}(Z)$, on the other hand, is 5.

40.4 Evaluation of the Maximum Regret of an ES-Policy

The absolute maximum regret of an ES-policy X is given by:

$$\begin{aligned}\rho^{max}(X) &= \max_{\sigma \in \Sigma, Y \in \mathscr{X}} \{S_{n+1}(X, \sigma) - S_{n+1}(Y, \sigma)\} \\ &= \max_{\sigma \in \Sigma} \left\{ \max_{c \in \mathscr{C}(X,\sigma)} l(c) - \min_{Y \in \mathscr{X}} \max_{c \in \mathscr{C}(Y,\sigma)} l(c) \right\}\end{aligned}$$

where $\mathscr{C}(Z, \sigma)$ denotes the set of all paths from 0 to $(n + 1)$ in $(V, E \cup Z, \sigma)$ for a selection Z and $l(c)$ is the length of the path c in the appropriate graph. The determination of each of these longest-path lengths can be cast into a linear formulation. Since the path lengths appear in the objective function with a positive sign for graph $(V, E \cup X, \sigma)$ and with a negative sign for $(V, E \cup Y, \sigma)$, we opt for the 'event-oriented' formulation for $(V, E \cup X, \sigma)$ and for the 'flow-oriented' formulation for $(V, E \cup Y, \sigma)$; see Wiest and Levy (1969) for more details. This leads to

$$\rho^{max}(X) = \max_{\sigma \in \Sigma} \left\{ \left(\sum_{(i,j) \in E \cup X} p_i \varphi_{ij} \right) - S^*(\sigma) \right\}$$

s.t.

$$\sum_{(i,j) \in E \cup X} \varphi_{ij} = \sum_{(j,i) \in E \cup X} \varphi_{ji} \qquad (i \in V \setminus \{0, n+1\})$$

$$\sum_{(0,j) \in E \cup X} \varphi_{0j} = \sum_{(j,n+1) \in E \cup X} \varphi_{j,n+1} = 1$$

$$\varphi_{ij} \geq 0 \qquad ((i,j) \in E \cup X)$$

$$S^*(\sigma) = \min_{Y \in \mathscr{X}} \quad S_{n+1}$$

$$\text{s.t.} \quad S_j \geq S_i + p_i \qquad ((i,j) \in E \cup Y)$$

$$S_0 = 0$$

By replacing the longest-path lengths by their linear-programming expressions, the maximum regret $\rho^{max}(X)$ of a given selection X is the optimal objective value of a bi-level mathematical program with, in our case, an RCPSP instance at the lower level. The variables φ_{ij} search for the longest path in $(V, E \cup X)$ by routing a unit flow through the network. An integration of the two levels of optimization is easily achieved:

$$\rho^{max}(X) = \max_{\sigma \in \Sigma} \left\{ \left(\sum_{(i,j) \in E \cup X} p_i \varphi_{ij} \right) - S_{n+1} \right\} \tag{40.3}$$

s.t.

$$\sum_{(i,j) \in E \cup X} \varphi_{ij} = \sum_{(j,i) \in E \cup X} \varphi_{ji} \qquad (i \in V \setminus \{0, n+1\}) \tag{40.4}$$

$$\sum_{(0,j) \in E \cup X} \varphi_{0j} = \sum_{(j,n+1) \in E \cup X} \varphi_{j,n+1} = 1 \tag{40.5}$$

$$\varphi_{ij} \geq 0 \qquad ((i,j) \in E \cup X) \tag{40.6}$$

$$S_j \geq S_i + p_i \qquad ((i,j) \in E \cup Y) \tag{40.7}$$

$$S_0 = 0 \tag{40.8}$$

$$Y \in \mathscr{X} \tag{40.9}$$

We neglect for a moment the variable duration vector σ and focus only on the RCPSP formulation with variable Y (cf. Eq. 40.1). The RCPSP formulation can be linearized using a resource-flow formulation, which has the benefit that it contains

only a polynomial number of constraints and that it does not require the explicit determination of all minimal forbidden sets (Demassey 2008). In particular, we add the following constraints:

$$\sum_{j \in V: j \neq i} \phi_{ji}^k = \sum_{j \in V: j \neq i} \phi_{ij}^k = r_{ik} \qquad (i \in V \setminus \{0, n+1\}; k \in \mathscr{R}) \tag{40.10}$$

$$\sum_{j \in V: j \neq 0} \phi_{0jk} = \sum_{j \in V: j \neq (n+1)} \phi_{j,(n+1),k} = R_k \qquad (k \in \mathscr{R}) \tag{40.11}$$

$$\phi_{ij}^k \geq 0 \qquad ((i, j) \in V \times V; \forall k \in \mathscr{R}) \tag{40.12}$$

The sufficient selection Y in the optimization problem is replaced by the set $C(\phi)$, with ϕ a flow satisfying the above constraints (40.10)–(40.12) as well as acyclicity of $(V, E \cup C(\phi))$. We replace the constraints (40.7) and (40.9) by the following:

$$0 \leq \phi_{ij}^k \leq M y_{ij} \qquad ((i, j) \in V \times V; k \in \mathscr{R}) \tag{40.13}$$

$$S_j \geq S_i + p_i - M(1 - y_{ij}) \qquad ((i, j) \in V \times V : i \neq j) \tag{40.14}$$

$$y_{ij} = 1 \qquad ((i, j) \in E) \tag{40.15}$$

$$y_{ij} \in \{0, 1\} \qquad ((i, j) \in V \times V) \tag{40.16}$$

The constraint sets (40.13) and (40.14) both contain 'big-M'-type constraints. The large number M can be chosen more specifically for each particular value of the indices i, j and k, a convenient choice is $\min\{r_{ik}, r_{jk}\}$ in (40.13) for $i, j \notin \{0, n+1\}$, with replacement of r_{ik} by R_k in this min-expression for $i = 0, n+1$. In (40.14), M can be an upper bound on the project makespan with durations σ^{max}. When combined with the constraint sets (40.13)–(40.16), acyclicity of $(V, E \cup C(\phi))$ is not an issue when the activity durations are non-zero.

Reverting to the optimization over Σ in (40.3), we should remove the non-linearity in the first term of (40.3) caused by the multiplication of p_i and φ_{ij} in order to obtain a linear model. According to Theorem 40.1, we need only consider two values p_i^{min} and p_i^{max} for p_i, with $p_i^{min} = p_i^{max} = 0$ for $i = 0, n+1$ (these zero values can be substituted immediately). We introduce n binary variables a_i $(i = 1, \ldots, n)$, where $a_i = 0$ means that duration p_i^{min} is selected for activity i, whereas $a_i = 1$ indicates that $p_i = p_i^{max}$. In Eq. (40.14), we replace p_i by $(1 - a_i)\, p_i^{min} + a_i\, p_i^{max}$. The non-linear terms $p_i \varphi_{ij}$ in (40.3) are replaced by $p_i^{min} \varphi_{ij}^{min} + p_i^{max} \varphi_{ij}^{max}$ and each occurrence of φ_{ij} in the constraints is replaced by $\varphi_{ij}^{min} + \varphi_{ij}^{max}$, in which φ_{ij}^{min} fulfills the role of φ_{ij} when $a_i = 0$ and φ_{ij}^{max} functions as φ_{ij} in the cases where $a_i = 1$. This is achieved by adding the following two equation sets to the formulation:

$$\sum_{(i,j)\in E\cup X} \varphi_{ij}^{max} \leq a_i \qquad (i \in V \setminus \{0, n+1\}) \tag{40.17}$$

$$\sum_{(i,j)\in E\cup X} \varphi_{ij}^{min} \leq 1 - a_i \qquad (i \in V \setminus \{0, n+1\}) \tag{40.18}$$

Summarizing the foregoing, we find that the determination of the maximum regret of an ES-policy X reduces to an instance of a multi-mode RCPSP with a composite objective function consisting in the maximization of the difference between the length of the longest path in $(V, E \cup X)$ and the optimal makespan S_{n+1}. We call this formulation *integrated*, because it simultaneously finds an optimal duration vector (an optimal scenario) and an optimal selection for the scenario (Y in the above). The full integrated formulation is included in the Appendix of this chapter.

As an alternative to this integrated formulation, we also propose a *scenario-based* formulation, in which some intermediate results are to be computed beforehand. We have

$$\begin{aligned} \rho^{max}(X) &= \max_{\sigma\in\Sigma} \left\{ S_{n+1}(X,\sigma) - S^*_{n+1}(\sigma) \right\} \\ &= \min \rho \qquad (40.19) \\ &\quad \text{s.t.} \\ &\quad \rho \geq S_{n+1}(X,\sigma) - S^*_{n+1}(\sigma) \qquad (\sigma \in \Sigma) \qquad (40.20) \end{aligned}$$

If the longest-path length $S_{n+1}(X,\sigma)$ and the RCPSP solution $S^*_{n+1}(\sigma)$ are known for each scenario σ then (40.19)–(40.20) is a linear formulation.

40.5 Absolute Minimax-Regret Optimization

In this section, we present a procedure for finding an optimal solution to the problem AR-RCPSP, which was defined in Sect. 40.3.3. The procedure is based on an extension of the scenario-based formulation for evaluation of a given policy (see Sect. 40.4). In principle, we could also extend the integrated formulation. This, however, would lead to a rather cumbersome model, and the main interest of the scenario-based solution procedure lies in its modifications that will lead to an effective heuristic for the AR-RCPSP, as will be set out in Sect. 40.6.

When we plug the scenario-based model (40.19)–(40.20) into the definition of the absolute-regret objective, we obtain the following bi-level formulation of AR-RCPSP:

$$\rho^* = \min_{X \in \mathscr{X}} \rho^{max}(X)$$

$$= \min_{X \in \mathscr{X}} \left\{ \begin{array}{ll} \min \rho & \\ \text{s.t.} & \\ \rho \geq S_{n+1}(X, \sigma) - S^*_{n+1}(\sigma) & (\sigma \in \Sigma) \end{array} \right\}$$

Again, we can assume that Σ contains only the extreme duration scenarios. The two levels of optimization are easily integrated and we resort to an event-based formulation for the longest-path computations in the graph $(V, E \cup X)$. Let $\Sigma = \{\sigma^1, \ldots, \sigma^{|\Sigma|}\}$. This leads to

$$\rho^* = \min \rho \tag{40.21}$$

s.t.

$$\rho \geq S^h_{n+1} - S^*_{n+1}(\sigma^h) \qquad (h = 1, \ldots, |\Sigma|) \tag{40.22}$$

$$S^h_j \geq S^h_i + p^h_i - M(1 - x_{ij}) \qquad ((i, j) \in V \times V : i \neq j; h = 1, \ldots, |\Sigma|) \tag{40.23}$$

$$S^h_i \geq 0 \qquad (i \in V; h = 1, \ldots, |\Sigma|) \tag{40.24}$$

together with the following scenario-independent constraints, which are similar to (40.10)–(40.13) and (40.15)–(40.16) (substituting x_{ij} for y_{ij}):

$$\sum_{j \in V: j \neq i} \phi^k_{ji} = \sum_{j \in V: j \neq i} \phi^k_{ij} = r_{ik} \qquad (i \in V \setminus \{0, n+1\}; k \in \mathscr{R}) \tag{40.25}$$

$$\sum_{j \in V: j \neq 0} \phi_{0jk} = \sum_{j \in V: j \neq (n+1)} \phi_{j,n+1,k} = R_k \qquad (k \in \mathscr{R}) \tag{40.26}$$

$$\phi^k_{ij} \geq 0 \qquad ((i, j) \in V \times V; k \in \mathscr{R}) \tag{40.27}$$

$$0 \leq \phi^k_{ij} \leq M x_{ij} \qquad ((i, j) \in V \times V; k \in R) \tag{40.28}$$

$$x_{ij} = 1 \qquad ((i, j) \in E) \tag{40.29}$$

$$x_{ij} \in \{0, 1\} \qquad ((i, j) \in V \times V) \tag{40.30}$$

In the worst case, we have $|\Sigma| = 2^n$. Hence, the MILP (mixed-integer linear program) includes an exponential number of variables S^h_i and constraints (40.22)–(40.24). Furthermore, for each duration vector σ^h the optimal RCPSP solution $S^*_{n+1}(\sigma^h)$ has to be computed. We therefore investigate the possibility of solving a relaxed version of the foregoing formulation by only incorporating the constraints

corresponding with a subset $\hat{\Sigma} \subset \Sigma$, iteratively adding scenarios until it can be guaranteed that the solution obtained for $\hat{\Sigma}$ has the same objective as the full model with Σ. Following Assavapokee et al. (2008a), we will refer to this approach as *scenario relaxation*. We call the resulting MILP the *master problem*, by analogy with Benders' decomposition, with objective function value $\rho^*(\hat{\Sigma})$ for set $\hat{\Sigma}$. Clearly, $\rho^*(\hat{\Sigma})$ is a lower bound on $\rho^* \equiv \rho^*(\Sigma)$. The variables $S_i^h = 0$ for scenarios $\sigma^h \in \Sigma \setminus \hat{\Sigma}$ can be removed from the model without any influence.

In Assavapokee et al. (2008a), a scenario-relaxation method is proposed to solve a general absolute min-max regret optimization problem with two-stage variables. The first stage-variables are binary "choice" variables, corresponding to our ϕ and x representing the ES-policy. The second-stage variables are continuous "recourse" variables, in our case the variables S^h. To overcome the implementation problems caused by an exponential number of constraints, Assavapokee et al. (2008a) propose a three-stage algorithm, based on the iterative solution of the model on a restricted scenario set. The first stage consists in solving the master problem with a restricted scenario set so as to obtain a lower bound and the corresponding values for the first-stage decision variables. For a general optimization problem, these values can be infeasible for some scenarios excluded from the scenario set. For this reason, the second stage consists in finding such 'infeasible' scenarios, which are added to the scenario set and the algorithm returns to the first stage. If no infeasibilities are found, the algorithm proceeds to the third stage, which aims to identify a scenario in $\Sigma \setminus \hat{\Sigma}$ achieving the largest regret for the candidate robust solution x and ϕ. In this chapter, each ES-policy produced by the master problem will be feasible for all scenarios: the feasibility of an ES-policy is independent of the activity durations. A comparable iterative solution approach for an inventory model has recently been examined by Bienstock and Özbay (2008).

We propose the framework described by Algorithm 40.1, in which *LB* and *UB* constitute a lower and upper bound on ρ^*, which are stepwise tightened over the course of the algorithm. Since an extreme duration vector is generated at each iteration, the solution framework converges within at most 2^n iterations. The restricted set of scenarios is updated at each iteration; we consider $\hat{\Sigma}_q$ at iteration q. At initialisation (Step 1), $|\hat{\Sigma}_1| = 1$ (although any number is possible). Bienstock and Özbay (2008) call the master problem (Step 2) the 'decision maker's problem,' where the decision maker makes a first-stage decision (the ES-selection) while accounting for only a subset of the scenarios, and Step 3 is the 'adversarial problem,' in which the worst scenario is generated for the candidate solution from Step 2, to verify its objective function against the full set of scenarios. Put differently, Step 3 looks for a scenario σ^h that is not currently in $\hat{\Sigma}$ and for which constraint (40.22) does not hold.

Computationally, the multi-mode-like instances in Step 3 of Algorithm 40.1 turn out to be especially hard. We can slightly modify the procedure by noting that at Step 3, it is not necessary to solve the maximum-regret evaluation to optimality. Let z represent the objective function of the subproblem; the correct functioning of the algorithm only requires that a duration vector σ^{q+1} be found such that $z \geq LB + 1$.

Algorithm 40.1 scenario relaxation for AR-RCPSP

1: (initialisation)
Consider duration-vector set $\hat{\Sigma}_1$ containing a single duration vector σ^1. Set $q := 1$, $LB := 0$ and $UB := +\infty$. Compute $S^*_{n+1}(\sigma^1)$.
2: (master problem)
Solve the restricted master problem (40.21)–(40.30) to obtain $LB = \rho^*(\hat{\Sigma}_q)$ and the corresponding ES-policy X_q. **If** $LB = UB$ then stop.
3: (maximum-regret computation)
Evaluate the maximum regret $\rho^{max}(X_q)$ of policy X_q using the integrated formulation of Sect. 40.4 and obtain the corresponding worst-case duration vector σ^{q+1} and the associated optimal RCPSP makespan $S^*_{n+1}(\sigma^{q+1})$. Set $UB := \min\{\rho^{max}(X_q); UB\}$.
4: (optimality check)
If $LB = UB$ then stop; **else** set $q := q + 1$, $\hat{\Sigma}_q := \hat{\Sigma}_{q-1} \cup \{\sigma^q\}$ and **goto** Step 2.

If one such duration vector exists then it can be included in the master problem, otherwise LB is optimal. To that purpose, we replace the objective function in the integrated formulation of Sect. 40.4 by

$$f^*(X) = \min f$$

and we add the constraints

$$\begin{aligned} z + f &\geq LB + 1 \\ z &\leq \left(\sum_{(i,j)\in E\cup X} p_i^{min}\varphi_{ij}^{min} + p_i^{max}\varphi_{ij}^{max} \right) - S_{n+1} \\ z, f &\geq 0 \end{aligned}$$

The resulting model has a solution $f^*(X) = 0$ if and only if there exists $z \geq LB+1$. The drawback of this approach is that for the value S^*_{n+1} corresponding to a duration vector σ^* output by this new subproblem, there is no guarantee that it equals the optimal makespan. Consequently, we additionally need to solve a standard RCPSP instance to obtain $S^*_{n+1}(\sigma^*)$ (in Step 3) before adding σ^* to $\hat{\Sigma}$ (in Step 4). We refer to the resulting subproblem as scenario generation with *bounded contribution*.

40.6 A Heuristic for AR-RCPSP

Our computational results (see Sect. 40.7) indicate that the execution of the standard scenario-relaxation procedure (Algorithm 40.1) until convergence may take an inordinate amount of time even for medium-sized instances. We will therefore proceed with the development of heuristic solution procedures in this section, with the framework provided by Algorithm 40.1 as a basis. A first obvious such heuristic

Algorithm 40.2 Heuristic framework for AR-RCPSP

1: (initialisation)
Consider duration-vector set $\hat{\Sigma}_1$ containing a single duration vector σ^1, set $q := 1$ and compute $\hat{s}^*_{n+1}(\sigma^1)$.
2: (generate new solution)
Use $\hat{s}^*_{n+1}(\sigma^q)$ and $\hat{\Sigma}_q$ to produce a new approximate solution (ES-policy) X_q.
3: (generate new duration vectors)
Generate one or more new duration vectors σ^{q+1} that represent scenarios under which policy X_q performs badly, together with an RCPSP upper bound $\hat{s}^*_{n+1}(\sigma^{q+1})$.
4: (iterate)
Set $q := q + 1$, $\hat{\Sigma}_q := \hat{\Sigma}_{q-1} \cup \{\sigma^q\}$ and **goto** Step 2.

is the variant of Algorithm 40.1 that is not run until the stopping criterion $LB = UB$ is met, but rather until LB and UB are 'reasonably close' to each other, for instance when $(UB - LB)/LB < \epsilon$ for a fixed (small) value of ϵ.

However, even this truncated run of Algorithm 40.1 will sometimes require very high running times, mainly due to the computational effort needed for performing Steps 2 and 3. We therefore propose a different approach, still following the same overall algorithmic structure but with significant efficiency gains also in each execution of both Steps 2 and 3; the main steps are presented as Algorithm 40.2. The essential drawback is that we again abandon the guarantee of finding an optimal solution: a number of approximations are inserted throughout the procedure. In the general variant of the algorithm, $\hat{S}^*_{n+1}(\sigma^h)$ is a heuristic solution (upper bound) to the RCPSP instance with optimal objective value $S^*_{n+1}(\sigma^h)$.

Step 2 produces a new solution, which is hopefully better than the current best solution. In our implementation, for the current scenario (duration vector σ^h), we solve the deterministic RCPSP to optimality (so $\hat{s}^*_{n+1}(\sigma^q)$ is actually $S^*_{n+1}(\sigma^{q+1})$ in our computations). With this schedule, we associate an activity list ℓ that orders the activities in non-decreasing starting times (subsequently referred to as 'associated list'). This list is then compared to the list ℓ^* associated with the current best solution. If the two are identical, the algorithm returns to Step 3 to obtain a new scenario, otherwise we consider all intermediary lists obtained by modifying the list ℓ^* step-by-step until ℓ is obtained. To each intermediary list, we apply a serial schedule generation scheme (Kolisch 1996) to find a new schedule, which then in turn is used to produce a new solution (ES-policy) via the algorithm of Artigues et al. (2003). This solution will be used as current best solution if it performs better (based on its regret) than the latter on the scenarios already generated. Let $\ell^* = (\ell^*_1, \ell^*_2, \ldots, \ell^*_n)$ and $\ell = (\ell_1, \ell_2, \ldots, \ell_n)$; notice that $\ell^*_0 = \ell_0$ and $\ell^*_{n+1} = \ell_{n+1}$. The intermediary lists between ℓ^* and ℓ are generated as follows. Let i be the lowest index for which $\ell^*_i \neq \ell_i$. We consider the list ℓ' obtained from ℓ^* by moving the activity ℓ_i from its current position in ℓ^* to position i in ℓ'. Hence, the activities between position i and the current position of ℓ_i in ℓ^* are shifted. Next, we set $\ell^* = \ell'$ and repeat the procedure until $\ell' = \ell$. This step constitutes a path-relinking procedure (Glover et al. 2000; Glover and Laguna

1997), generating feasible policies on a neighborhood path from an optimal policy (in terms of makespan for a given scenario) to another one.

In Step 3, a new scenario is generated by running the integrated formulation for evaluation of the current ES-policy X (as described in Appendix) but from which all the y_{ij}-variables are removed (apart from those corresponding to $(i, j) \in E$): we effectively solve the problem

$$\max_{\sigma \in \Sigma} \{S_{n+1}(X, \sigma) - S_{n+1}(\emptyset, \sigma)\}$$

which delivers an upper bound for the actual regret of policy X and so also an upper bound for the minimax regret. This upper-bound formulation is handed to a MIP solver, which yields a new scenario. If the addition of this scenario does not lead to a solution different from the current best solution, we have a *cycling phenomenon*. In that case, we identify a longest path in the graph $(V, E \cup X)$, where X is the current best solution, to generate a new scenario: the activities on the path are set at their maximum durations, all other activities receive the minimum duration. In case this scenario also leads to cycling, a new scenario is generated randomly.

When solving the example project described in Sect. 40.3.1 using the basic implementation of Algorithm 40.1, the optimal value of 2 is obtained after three iterations and a running time of 0.03 s. Using the implementation with bounded contribution, four iterations are used and the running time is only 0.02 s. Finally, Algorithm 40.2 finds an optimal solution after six iterations, but its running time is 0 s.

40.7 Computational Results

All algorithms have been coded in C using Visual Studio C++ 2005; all the experiments were run on a Dell Optiplex 760 personal computer with Pentium R processor with 3.16 GHz clock speed and 3.21 GB RAM, equipped with Windows XP. CPLEX 10.2 was used for solving the MIP instances. Below, we first provide some details on the generation of the datasets, then we discuss some implementation details, and subsequently we present the computational results. Throughout this section, computation time is referred to as $t_{cpu}^{\emptyset}$ and is expressed in seconds.

40.7.1 Data Generation

The algorithms are tested on randomly generated instances of AR-RCPSP with n non-dummy activities, for $n = 10$, 20 and 30. We use the software RanGen (Demeulemeester et al. 2003) to generate instances of the deterministic RCPSP. Using this software, we can choose different values for the number

of activities n, the order strength (OS), the resource factor (RF) and the resource constrainedness (RC) (for more information on these parameters, see Demeulemeester et al. 2003). For our experiments, we have chosen three different values for OS and two values for RF and RC, as follows. OS takes its value in the set $\{0.4, 0.6, 0.8\}$, RF is chosen from $\{0.45, 0.9\}$ and RC is a value in $\{0.3, 0.6\}$. For each combination of n, OS, RF and RC, we randomly generate ten instances of the deterministic RCPSP. From each RCPSP instance, we create an instance of AR-RCPSP by randomly choosing for each activity i with processing time p_i an integer δ_i between zero and $p_i - 1$. In the AR-RCPSP instance, the lower (upper) bound on the processing time of activity i is $p_i - \delta_i$ (respectively $p_i + \delta_i$). In total, there are $3 \times 2 \times 2 \times 10 = 120$ instances for each value of n.

40.7.2 Implementation Details

In this section, we describe some of the implementation details for the algorithms proposed in Sects. 40.5 and 40.6. We also present two simple heuristics that will serve as benchmarks in Sect. 40.7.3 for evaluation of the performance of the algorithms on the generated instances.

40.7.2.1 Algorithm 40.1

The implementation of Algorithm 40.1 follows the pseudocode of Sect. 40.5 with the following details and adaptations: we start with $\hat{\Sigma}_1$ containing only $\sigma^1 = \sigma^{min}$, the scenario in which the processing time of each activity is minimal. We use the branch-and-bound algorithm developed by Demeulemeester and Herroelen (1992) to solve the resulting deterministic RCPSP. Subsequently, the first master problem is set up but instead of solving it, we use the algorithm proposed by Artigues et al. (2003) to find a feasible resource flow and hence an ES-policy. The master problem is solved using CPLEX starting from the second iteration. We wish to underline that this implementation was chosen after preliminary experiments with other variants, among which an implementation where the initial scenario is chosen randomly, a variant where we initially add the two extreme scenarios (minimum and maximum durations) and one that initially adds three scenarios (minimum and maximum durations, and the third scenario is selected randomly).

Since Algorithm 40.1 is an exact procedure, we have investigated the bottleneck of its CPU time on the set of 10-activity instances. The CPU time of this algorithm is mainly made up of the time spent solving the master and the time needed to evaluate a given policy; we study the contribution of each of these two computations to the overall CPU time. In Table 40.1, we report the average CPU time for the master problem, for the evaluation procedure and the total average running time. We also report the average number of iterations (itr.) and the number of instances

Table 40.1 Distribution of average CPU time of Algorithm 40.1

Parameters			Algorithm 40.1 (I)					Algorithm 40.1 (II)				
			$t_{cpu}^{\emptyset}$					$t_{cpu}^{\emptyset}$				
OS	RF	RC	Master	Evaluation	Total	itr.	f_{opt}	Master	Evaluation	Total	itr.	f_{opt}
0.4	0.45	0.3	0.03	0.04	0.07	2.7	10	0.02	0.02	0.04	2.7	10
		0.6	0.12	1.15	1.27	2.1	10	0.12	0.71	0.83	2.1	10
	0.9	0.3	11.30	15.27	26.57	9.0	10	10.23	9.15	19.38	15.40	10
		0.6	16.48	882.59	899.07	1.6	8	19.06	734.84	753.90	4.25	9
0.6	0.45	0.3	0.01	0.02	0.03	2.1	10	0.01	0.02	0.03	2.8	10
		0.6	0.06	0.12	0.18	2.1	10	0.06	0.07	0.13	2.8	10
	0.9	0.3	0.15	0.40	0.55	3.4	10	0.17	0.24	0.41	4.3	10
		0.6	0.20	11.17	11.37	1.2	10	0.22	14.77	14.99	1.7	10
0.8	0.45	0.3	0.01	0.02	0.03	1.2	10	0.01	0.00	0.01	1.2	10
		0.6	0.01	0.02	0.03	1.2	10	0.01	0.02	0.03	1.2	10
	0.9	0.3	0.03	0.05	0.08	1.8	10	0.03	0.04	0.07	1.8	10
		0.6	0.05	0.14	0.19	1	10	0.05	0.14	0.19	1.0	10

solved to guaranteed optimality within a time limit of 30 min (f_{opt}). Each reported value in the table is the average of ten values, except in the last column (f_{opt}). "Algorithm 40.1 (I)" refers to the "basic" implementation while "Algorithm 40.1 (II)" is the implementation "with bounded contribution." In general, the second implementation outputs results that are slightly better than those produced by the first implementation. Moreover, the implementation with bounded contribution cannot optimally solve one instance while the implementation without bounded contribution fails to solve two instances to optimality. We observe, however, that for one group (OS $=$ 0.6, RF $=$ 0.9 and RC $=$ 0.6) the average CPU time of the implementation with bounded contribution is larger than the standard implementation. In fact, for that group of ten instances, there are two instances for which the implementation with bounded contribution takes substantially more time than the implementation without bounded contribution.

As mentioned, among the 120 instances, there is only one (for variant II) or two (for I) that are not solved within the time limit (for these two instances, the optimal solution was actually found but a certificate of optimality could not be produced within the time limit). These two instances belong to the same group with OS $=$ 0.4, RF $=$ 0.9 and RC $=$ 0.6, which also has the highest average CPU time. We also observe that very few iterations are usually needed to arrive at an optimal solution: the average is never higher than 16, and in most cases even below 4; the algorithm with bounded contribution generally uses more iterations than the basic variant. When we compare the running times for evaluation and for the master problem, the former come out considerably higher than the latter.

40.7.2.2 Algorithm 40.2

The implementation of Algorithm 40.2 also follows its pseudocode. We again start with $\hat{\Sigma}_1 = \{\sigma^{min}\}$. For this minimum scenario, the deterministic RCPSP is solved and an ES-policy is constructed following Artigues et al. (2003); this solution is set as the current best solution and is optimal for the scenario set $\hat{\Sigma}_1$. In the first iteration, we can therefore skip Step 2 and immediately go to Step 3.

The stopping criteria for the algorithm are a limit on the computing time (30 min), a maximum number of scenarios generated (100) and a maximum number of consecutive scenarios giving rise to cycling (10). This implementation was chosen after preliminary experiments with different values for the maximum number of scenarios generated and the maximum number of consecutive scenarios engendering cycling.

40.7.2.3 Two Simple Heuristics

We present two additional heuristics that will be used as benchmarks for our two main algorithms. The rationale behind the choice for these simple heuristics is the fact that it is extremely difficult to provide meaningful bounds for AR-RCPSP even for medium-sized instances. The first heuristic is referred to as Heuristic 1 and is described in pseudocode below. As before, the deterministic RCPSP is solved using the branch-and-bound algorithm of Demeulemeester and Herroelen (1992).

Algorithm 40.3 Heuristic 1

1: determine the average duration $p_i = \lfloor \frac{p_i^{min} + p_i^{max}}{2} \rfloor$ for each activity i
2: solve the corresponding deterministic RCPSP
3: impose a resource flow on this schedule with the algorithm of Artigues et al. (2003)
4: output the solution found

The second heuristic is named Heuristic 2 and is outlined below. The problem encountered in Step 2 is formulated as a MIP where the objective is to minimize the number of arcs in the transitive closure, which is solved using CPLEX. We refer to Leus (2011b) for a motivation for this choice of objective function.

Algorithm 40.4 Heuristic 2

1: ignore the activity durations
2: find a solution (ES-policy) with a transitive closure having the minimum number of arcs

40.7.3 Computational Experiments

Below, we present our computational results for instance sets with 10, 20, and 30 activities.

40.7.3.1 Comparison of the Algorithms for 10-Activity Instances

Table 40.2 displays for every algorithm the average CPU time ($t_{cpu}^{\emptyset}$), the number of instances for which the algorithm finds an optimal solution (opt.) and the average gap ($\Delta_{opt}^{\emptyset}$) for the set of instances with ten non-dummy activities. We opt for variant II of Algorithm 40.1. In the table, each value reported in the columns $t_{cpu}^{\emptyset}$ and $\Delta_{opt}^{\emptyset}$ is the average of ten values. Since the optimal objective value can be zero, the usual 'gap' is not well defined. Therefore, throughout this section, we use $\Delta_{opt}^{\emptyset}$ defined as follows. For an instance, let $f(H)$ be the value of the solution found by the considered algorithm and f_{opt} be the optimal objective value (for $n = 10$, this optimal value is found by Algorithm 40.1). Let δ be the average difference between minimum and maximum duration of activities, so $\delta = \frac{1}{n}\sum_{i=1}^{n}\left(p_i^{max} - p_i^{min}\right)$. We define $\Delta_{opt}^{\emptyset} = 100 \times \frac{f(H)-f_{opt}}{\delta}$.

Table 40.2 shows that Algorithm 40.1 optimally solves all the instances apart from one; this, however, is sometimes coupled with a high average CPU time. Among the remaining algorithms, Algorithm 40.2 solves 82 instances out of 120 to optimality, while Heuristic 1 finds optimal solutions for 47 instances and Heuristic 2 provides optimal solutions for only 40 instances. Algorithm 40.2 usually produces the smallest average $\Delta_{opt}^{\emptyset}$ for this dataset. However, the average CPU time of

Table 40.2 Comparison for $n = 10$

Parameters			Algorithm 40.1			Algorithm 40.2			Heuristic 1			Heuristic 2		
OS	RF	RC	$t_{cpu}^{\emptyset}$	$\Delta_{opt}^{\emptyset}$	f_{opt}	$t_{cpu}^{\emptyset}$	$\Delta_{opt}^{\emptyset}$	f_{opt}	$t_{cpu}^{\emptyset}$	$\Delta_{opt}^{\emptyset}$	f_{opt}	$t_{cpu}^{\emptyset}$	$\Delta_{opt}^{\emptyset}$	f_{opt}
0.4	0.45	0.3	0.04	0.00	10	2.03	17.78	5	0.00	26.09	2	0.02	75.98	2
		0.6	0.83	0.00	10	2.53	31.37	5	0.00	80.21	3	0.04	90.12	2
	0.9	0.3	19.38	0.00	10	3.88	44.87	4	0.00	59.71	3	0.02	62.35	3
		0.6	753.90	0.00	9	2.58	25.00	5	0.00	138.70	0	0.06	45.60	3
0.6	0.45	0.3	0.03	0.00	10	1.36	28.21	7	0.00	21.65	6	0.02	52.57	5
		0.6	0.13	0.00	10	2.44	25.00	7	0.00	30.93	4	0.04	38.56	3
	0.9	0.3	0.41	0.00	10	2.24	30.51	4	0.00	52.07	3	0.07	91.88	2
		0.6	14.99	0.00	10	1.86	11.90	8	0.00	20.41	6	0.44	33.15	4
0.8	0.45	0.3	0.01	0.00	10	0.21	0.00	10	0.00	7.32	7	0.01	94.59	4
		0.6	0.03	0.00	10	0.00	0.00	10	0.00	13.96	7	0.13	47.53	6
	0.9	0.3	0.07	0.00	10	1.22	17.50	7	0.00	75.02	0	0.46	199.15	0
		0.6	0.19	0.00	10	0.73	0.00	10	0.00	11.10	6	0.90	10.40	6

Algorithm 40.2 is higher than for Heuristic 2, which, in turn, is higher than for Heuristic 1.

40.7.3.2 Comparison of the Algorithms for 20-Activity Instances

In this section, we focus on the set of instances with 20 non-dummy activities. For these instances, attempts to provide optimal solutions by solving the mixed-integer formulation with CPLEX produced extremely poor results within the time limit of 30 min: when interrupting CPLEX after 30 min, the best lower bound is usually zero and the best upper bound is very large. Without an optimal value nor a good lower bound, $\Delta_{opt}^{\emptyset}$ defined in the previous section is meaningless. Therefore, to compare the different algorithms we first report the average CPU time and the number of optimal solutions found in Table 40.3. In fact, each algorithm may have found more optimal solutions than what is reported in Table 40.3 because we count only the number of optimal solutions with objective value zero (for Algorithm 40.1, other instances that were solved to guaranteed optimality within the time limit are obviously also counted).

Variant I of Algorithm 40.1 provides guaranteed optimal solutions to 52 instances out of 120 within the time limit of 30 min; this number goes up to 58 for the bounded-contribution implementation. The average CPU time of this algorithm is very large for both implementations. We observe that for most unsolved instances, the time limit is reached before two iterations are fully completed. This confirms that solving both the master problem and the evaluation problem in an exact fashion is simply overly time-consuming. Algorithm 40.2 produces optimal solutions for 18 instances, with a maximum average CPU time less than 6 s. Heuristic 1 is the fastest algorithm but obtains optimal solutions for only six instances, while Heuristic 2 provides such solutions for nine instances but needs more time.

In order to further compare the quality of the outputs of the algorithms, we have performed a pair-wise comparison, the results of which are reported in Table 40.4. For a given instance of AR-RCPSP, let X and Y be the solutions output by two different algorithms. Using CPLEX, we compute the quantity

$$LB(X,Y) = \max_{\sigma \in \Sigma} \left\{ S_{n+1}(X,\sigma) - S_{n+1}(Y,\sigma) \right\}$$

which entails the maximum regret of the output of the first algorithm with respect to the output of the second algorithm over all the scenarios; $LB(X,Y)$ is a lower bound for $\rho^{max}(X)$ for any feasible Y. The quantity $LB(Y,X)$ represents a similar comparison of the second algorithm with respect to the first one; $LB(X,Y)$ and $LB(Y,X)$ need not be the same. In Table 40.4, we denote Algorithm 40.1 (with bounded contribution) by A1, Algorithm 40.2 by A2, Heuristic 1 by H1 and Heuristic 2 by H2. Observe that any number in this table is the average of ten values. We conclude that Algorithm 40.1 achieves the best comparison for those instances where it regularly finds optimal solutions. Overall, Algorithm 40.2 tends to display

Table 40.3 Comparison for $n = 20$

Parameters			Algorithm 40.1 (I)		Algorithm 40.1 (II)		Algorithm 40.2		Heuristic 1		Heuristic 2	
OS	RF	RC	$t_{cpu}^{\emptyset}$	f_{opt}	$t_{cpu}^{\emptyset}$	f_{opt}	$t_{cpu}^{\emptyset}$	f_{opt}	$t_{cpu}^{\emptyset}$	f_{opt}	$t_{cpu}^{\emptyset}$	f_{opt}
0.4	0.45	0.3	629.22	8	542.31	8	2.00	1	0.00	0	14.72	1
		0.6	2201.50	0	1947.81	1	3.22	0	0.00	0	1207.69	1
	0.9	0.3	2340.17	0	2247.11	0	3.59	0	0.00	0	1800.00	0
		0.6	1985.90	0	1830.47	1	5.18	0	0.00	0	1800.00	0
0.6	0.45	0.3	56.42	10	61.00	10	1.63	2	0.00	2	0.21	1
		0.6	1990.98	0	1559.89	0	2.57	1	0.00	1	36.24	0
	0.9	0.3	2208.07	0	1466.95	0	2.58	3	0.00	1	1153.71	0
		0.6	1800.10	0	2081.57	1	1.87	0	0.00	0	1800.16	0
0.8	0.45	0.3	1.27	10	0.68	10	0.84	5	0.00	1	0.06	3
		0.6	190.95	10	133.15	10	1.10	4	0.00	1	0.37	2
	0.9	0.3	183.94	10	177.56	10	1.41	2	0.00	0	0.56	1
		0.6	1799.99	4	1808.54	7	0.29	0	0.00	0	113.78	0

Table 40.4 Pair-wise comparison for $n = 20$

Parameters			Algorithm 40.1 (A1)			Algorithm 40.2 (A2)			Heuristic 1 (H1)			Heuristic 2 (H2)		
OS	RF	RC	A2	H1	H2	A1	H1	H2	A1	A2	H2	A1	A2	H1
0.4	0.45	0.3	0.00	0.00	0.00	31.48	12.73	8.22	42.87	23.37	46.70	87.12	62.26	32.25
		0.6	49.29	46.31	30.66	33.12	22.70	12.10	42.38	28.49	38.12	49.75	121.16	73.56
	0.9	0.3	80.63	45.34	72.98	38.10	10.63	28.95	119.21	164.25	46.11	71.08	135.26	58.63
		0.6	89.45	22.39	66.87	4.56	0.20	0.00	92.08	178.24	47.32	21.70	166.96	65.45
0.6	0.45	0.3	0.00	0.00	0.00	36.21	26.12	47.87	76.36	46.85	0.00	55.18	65.14	23.90
		0.6	79.05	39.23	54.65	42.92	32.61	0.00	68.49	155.15	45.17	32.57	75.51	33.80
	0.9	0.3	77.81	64.52	39.06	42.65	78.93	96.35	69.84	131.16	37.54	49.01	124.32	62.64
		0.6	51.26	25.79	49.07	33.64	52.68	72.95	44.98	73.63	64.94	27.23	59.10	37.24
0.8	0.45	0.3	0.00	0.00	0.00	33.29	15.23	13.25	67.66	47.56	12.52	38.11	31.50	11.86
		0.6	0.00	0.00	0.00	75.42	23.56	36.45	92.35	45.68	25.23	74.00	40.27	39.60
	0.9	0.3	0.00	0.00	0.00	47.38	25.12	24.65	109.22	78.54	23.69	164.47	66.45	46.10
		0.6	18.04	8.59	13.37	68.00	39.24	27.45	99.65	64.52	45.63	139.80	64.00	31.50

the smallest difference with respect to the output of the other algorithms. Between Heuristic 1 and Heuristic 2, however, there is no clear domination of one over the other.

40.7.3.3 Comparison of the Algorithms for 30-Activity Instances

We have performed comparisons for the set with 30 non-dummy activities similar to the previous sections. We do not, however, include the exact algorithm (Algorithm 40.1) because within the imposed time limit, this algorithm is unable to perform even a single iteration. In Table 40.5, we compare the average CPU time and the number of solutions with zero objective value found by each of the three remaining algorithms. Algorithm 40.2 obtains solutions with zero objective value for nine instances while Heuristic 1 and Heuristic 2 do so for only one instance. Again, it is important to note that this does not mean that these algorithms did not solve more instances until optimality, since we do not know the optimal objective value. From Table 40.5, we also observe that (somewhat logically) the average CPU time of each algorithm has increased compared to Table 40.3. The pair-wise comparison of the three algorithms is described in Table 40.6. For this dataset, Algorithm 40.2 displays the smallest values when compared to the other two heuristics.

Table 40.5 Comparison for $n = 30$

Parameters			Algorithm 40.2		Heuristic 1		Heuristic 2	
OS	RF	RC	$t_{cpu}^{\emptyset}$	f_{opt}	$t_{cpu}^{\emptyset}$	f_{opt}	$t_{cpu}^{\emptyset}$	f_{opt}
0.4	0.45	0.3	4.99	2	0.00	0	0.27	0
		0.6	6.89	1	0.00	0	105.54	1
	0.9	0.3	25.79	0	0.00	0	703.85	0
		0.6	17.59	0	0.00	0	1800.17	0
0.6	0.45	0.3	21.15	1	0.00	1	976.51	0
		0.6	17.30	1	0.00	0	1800.06	0
	0.9	0.3	52.18	0	0.01	0	1800.03	0
		0.6	53.07	0	0.00	0	1800.04	0
0.8	0.45	0.3	14.74	2	0.01	0	1753.83	0
		0.6	11.62	1	0.01	0	1800.07	0
	0.9	0.3	31.81	1	1.19	0	1800.05	0
		0.6	32.46	0	0.00	0	1800.02	0

Table 40.6 Pair-wise comparison for $n = 30$

Parameters			A2		H1		H2	
OS	RF	RC	H1	H2	A2	H2	A2	H1
0.4	0.45	0.3	36.90	41.65	150.30	57.24	239.00	77.62
		0.6	39.20	43.70	403.50	108.31	344.00	94.87
	0.9	0.3	15.50	18.80	381.80	163.44	310.10	124.64
		0.6	55.20	58.80	420.10	122.18	390.70	210.23
0.6	0.45	0.3	34.20	39.40	244.40	132.52	197.00	97.56
		0.6	39.80	54.90	332.00	142.10	304.60	56.78
	0.9	0.3	34.60	56.50	354.80	180.30	365.10	169.23
		0.6	73.70	86.90	467.30	156.34	444.80	274.59
0.8	0.45	0.3	18.50	28.10	206.60	93.24	169.30	73.52
		0.6	57.00	86.40	376.00	196.03	358.80	216.76
	0.9	0.3	59.70	90.50	406.70	215.73	400.80	91.08
		0.6	80.20	100.80	445.20	106.07	490.30	134.27

40.8 Conclusions

In practical project management, a project's parameters such as activity durations and resource requirements are seldom precisely known and usually subject to estimation errors. In this chapter we have proposed a robust optimization approach to project scheduling with uncertain activity durations, assuming that the decision maker cannot with confidence associate probabilities with possible activity durations. The resulting robust project scheduling problem that we have studied, has turned out to be exceptionally difficult, in that even exact objective-function evaluation is intractable and computationally overly demanding, even for medium-sized instances. We have developed and implemented a scenario-relaxation algorithm and a scenario-relaxation-based heuristic. The first algorithm produces optimal solutions but requires excessive running times even for medium-sized instances; the second algorithm produces high-quality solutions for medium-sized instances, which are significantly better than those produced by two benchmark heuristics—although the latter consume less CPU time.

Further research should be oriented towards the formulation and solution of more practical variants of the AR-RCPSP, for instance by considering an objective function that can be evaluated in polynomial time for a given scheduling policy, so that the corresponding decision problem is at least in $\mathcal{NP}$. This could be the case for the minimization of the upper bound of the minimax regret of a policy X defined as $\max_{\sigma \in \Sigma} \{S_{n+1}(X, \sigma) - S_{n+1}(\emptyset, \sigma)\}$, the complexity status of which, to the best of our knowledge, is open. Incorporation of uncertainties different from activity durations is another avenue for future work; the most important factors that can play a role in practical project management are probably the resource availabilities, the structural properties of the project network (the precedence relations) and the addition of extra activities. The incomplete specification of probabilities also leads

to the question of what the value of information associated with a full or partial resolution of these uncertainties would be. In particular: future research might study the case where the decision maker can acquire more information about probabilities, possibly at a cost, and decide how these efforts of information acquisition should be targeted.

Acknowledgements This work was supported by research project G.0508.09 of the Belgian Research Foundation—Flanders (FWO) and by the French National Research Agency (ANR) project on Robustness and Cooperation in Scheduling "ROBOCOOP" ANR-08-BLAN-0331-01.

Appendix: Integrated Formulation for Max-Regret Evaluation

The full integrated formulation for evaluation of the maximum regret of a policy X, which was developed in Sect. 40.4, is given below.

$$\rho^{max}(X) = \quad \max \left\{ \left(\sum_{(i,j)\in E\cup X} p_i^{min}\varphi_{ij}^{min} + p_i^{max}\varphi_{ij}^{max} \right) - S_{n+1} \right\}$$

$$\text{s.t.} \quad \sum_{(i,j)\in E\cup X} \varphi_{ij}^{min} + \varphi_{ij}^{max} = \sum_{(j,i)\in E\cup X} \varphi_{ji}^{min} + \varphi_{ji}^{max} \quad (i \in V \setminus \{0, n+1\})$$

$$\sum_{(0,j)\in E\cup X} \varphi_{0j}^{min} + \varphi_{0j}^{max} = \sum_{(j,n+1)\in E\cup X} \varphi_{j,n+1}^{min} + \varphi_{j,n+1}^{max} = 1$$

$$0 \le \phi_{ij}^k \le M y_{ij} \quad ((i,j) \in V \times V; k \in \mathscr{R})$$

$$S_j \ge S_i + (1-a_i)p_i^{min} + a_i\, p_i^{max} - M(1-y_{ij}) \quad ((i,j) \in V \times V : i \ne j)$$

$$\sum_{j\in V: j\ne i} \phi_{ji}^k = \sum_{j\in V: j\ne i} \phi_{ij}^k = r_{ik} \quad (i \in V \setminus \{0, n+1\}; k \in \mathscr{R})$$

$$\sum_{j\in V} \phi_{0j}^k = \sum_{j\in V} \phi_{j(n+1)}^k = R_k \quad (k \in \mathscr{R})$$

$$\sum_{(i,j)\in E\cup X} \varphi_{ij}^{max} \le a_i \quad (i \in V \setminus \{0, n+1\})$$

$$\sum_{(i,j)\in E\cup X} \varphi_{ij}^{min} \le 1 - a_i \quad (i \in V \setminus \{0, n+1\})$$

$$\varphi_{ij}^{min}, \varphi_{ij}^{max} \ge 0 \quad ((i,j) \in E \cup X)$$

$$y_{ij} = 1 \quad ((i,j) \in E)$$

$$y_{ij} \in \{0,1\} \quad ((i,j) \in V \times V)$$

$$S_0 = 0$$

$$\phi_{ij}^k \geq 0 \quad ((i,j) \in V \times V; \forall k \in \mathscr{R})$$

$$a_i \in \{0,1\} \quad (i \in V)$$

$$a_0 = a_{n+1} = 0$$

References

Adlakha VG, Kulkarni VG (1989) A classified bibliography of research on stochastic PERT networks: 1966–1987. INFOR 27:272–296

Aissi H, Bazgan C, Vanderpooten D (2009) Min-max and min-max regret versions of combinatorial optimization problems: a survey. Eur J Oper Res 197:427–438

Artigues C, Roubellat F (2000) A polynomial activity insertion algorithm in a multiresource schedule with cumulative constraints and multiple modes. Eur J Oper Res 127:297–316

Artigues C, Michelon P, Reusser S (2003) Insertion techniques for static and dynamic resource-constrained project scheduling. Eur J Oper Res 149:249Ú–267

Ashtiani B, Leus R, Aryanezhad MB (2011) New competitive results for the stochastic resource-constrained project scheduling problem: exploring the benefits of pre-processing. J Sched 14(2):157–171

Assavapokee T, Realff MJ, Ammons JC (2008a) A new min-max regret robust optimization approach for interval data uncertainty. J Optim Theory Appl 137:297–316

Assavapokee T, Realff MJ, Ammons JC, Hong IH (2008b) Scenario relaxation algorithm for finite scenario-based min-max regret and min-max relative regret robust optimization. Comput Oper Res 35:2093–2102

Averbakh I (2000) Minmax regret solutions for minimax optimization problems with uncertainty. Oper Res Lett 27:57–65

Averbakh I (2005) Computing and minimizing the relative regret in combinatorial optimization with interval data. Discrete Optim 2:273–287

Averbakh I, Lebedev V (2004) Interval data minmax regret network optimization problems. Discrete Appl Math 138:289–301

Balas E (1971) Project scheduling with resource constraints. In: Beadle ELM (ed) Applications of mathematical programming techniques. The English Universities Press, London, pp 187–200

Balletsín F, Leus R (2009) Resource-constrained project scheduling for timely project completion with stochastic activity durations. Prod Oper Manage 18(4):459–474

Ben-Tal A, Nemirovski A (1999) Robust solutions of uncertain linear programs. Oper Res Lett 25:1–13

Ben-Tal A, Nemirovski A (2000) Robust solutions of linear programming problems contaminated with uncertain data. Math Program 88:411–424

Ben-Tal A, Nemirovski A (2002) Robust optimization: methodology and applications. Math Program 92:453–480

Bertsimas D, Sim M (2003) Robust discrete optimization and network flows. Math Program 98: 49–71

Bertsimas D, Sim M (2004) The price of robustness. Oper Res 52:35–53

Bienstock D, Özbay N (2008) Computing robust basestock levels. Discrete Optim 5:389–414

Błażewicz J, Lenstra JK, Rinnooy Kan AHG (1983) Scheduling subject to resource constraints: classification and complexity. Discrete Appl Math 5:11–24

Bowers JA (1995) Criticality in resource constrained networks. J Oper Res Soc 46:80–91

Daniels RL, Kouvelis P (1995) Robust scheduling to hedge against processing time uncertainty in single-stage production. Manag Sci 41:363–376

Demassey S (2008) Mathematical programming formulations and lower bounds. In: Artigues C, Demassey S, Néron E (eds) Resource-constrained project scheduling: models, algorithms, extensions and applications. Wiley, Hoboken, pp 49–62

Demeulemeester E, Herroelen W (1992) A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. Manag Sci 38:1803–Ű1818

Demeulemeester E, Vanhoucke M, Herroelen W (2003) A random network generator for activity-on-the-node networks. J Sched 6:13–34

Elmaghraby SE (1977) Activity networks: project planning and control by network models. Wiley, New York

French S (1988) Decision theory. An introduction to the mathematics of rationality. Ellis Horwood, New York

Glover F, Laguna M (1997) Tabu search. Kluwer Academic, Boston

Glover F, Laguna M, Martí R (2000) Fundamentals of scatter search and path relinking. Control Cybern 39:653–684

Hughes MW (1986) Why projects fail: the effects of ignoring the obvious. Ind Eng 18:14–18

Hurwicz L (1951) Optimality criteria for decision making under ignorance. Cowles Commission Discussion Paper no. 370

Igelmund G, Radermacher FJ (1983) Preselective strategies for the optimization of stochastic project networks under resource constraints. Networks 13:1–28

Jørgensen T (1999) Project scheduling as a stochastic dynamic decision problem. Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim

Kasperski A (2008) Discrete optimization with interval data. Minmax regret and fuzzy approach. Studies in fuzziness and soft computing, vol 228. Springer, Berlin

Knight FH (1921) Risk, uncertainty, and profit. Houghton Mifflin, Boston

Kolisch R (1996) Serial and parallel resource-constrained project scheduling methods revisited: theory and computation. Eur J Oper Res 90:320–333

Kouvelis P, Yu G (1997) Robust discrete optimization and its applications. Kluwer Academic, Dordrecht

Kouvelis P, Daniels RL, Vairaktarakis G (2000) Robust scheduling of a two-machine flow shop with uncertain processing times. IIE Trans 32:421–432

Kulkarni VG, Adlakha VG (1986) Markov and Markov-regenerative PERT networks. Oper Res 34:769–781

Lebedev V, Averbakh I (2006) Complexity of minimizing the total flow time with interval data and minmax regret criterion. Discrete Appl Math 154:2167–2177

Leus R (2001a) Resource allocation by means of project networks: dominance results. Networks 58(1):50–58

Leus R (2011b) Resource allocation by means of project networks: complexity results. Networks 58(1):59–67

Leus R, Herroelen W (2004) Stability and resource allocation in project planning. IIE Trans 36:667–682

Loch CH, De Meyer A, Pich MT (2006) A new approach to managing high uncertainty and risk in projects. Wiley, Hoboken

Ludwig A, Möhring RH, Stork F (2001) A computational study on bounding the makespan distribution in stochastic project networks. Ann Oper Res 102:49–64

Malcolm DG, Rosenbloom JM, Clark CE, Fazar W (1959) Application of a technique for research and development program evaluation. Oper Res 7:646–669

Montemanni R (2007) A mixed integer programming formulation for a single machine robust scheduling with interval data. J Math Model Algorithm 6:287–296

Mulvey JM, Vanderbei RJ, Zenios SA (1995) Robust optimization of large-scale systems. Oper Res 43:264–281

Naegler G, Schoenherr S (1989) Resource allocation in a network model: the Leinet system. In: Słowiński R, Węglarz J (eds) Advances in project scheduling, chap. II.8. Elsevier, Amsterdam

Neumann K, Schwindt C, Zimmermann J (2003) Project scheduling with time windows and scarce resources: temporal and resource-constrained project scheduling with regular and nonregular objective functions. Springer, Berlin

Nikulin Y (2006) Robustness in combinatorial optimization and scheduling theory: An extended annotated bibliography. Technical Report 606, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel, Kiel

Rosenhead J, Elton M, Gupta SK (1972) Robustness and optimality as criteria for strategic decisions. Oper Res Q 23:413–431

Roy B, Sussmann B (1964) Les problèmes d'ordonnancement avec contraintes disjonctives. Technical Report Note D.S. no. 9 bis, SEMA, Paris

Savage LJ (1951) The theory of statistical decision. J Am Stat Assoc 46:55–67

Stork F (2001) Stochastic resource-constrained project scheduling. Ph.D. dissertation, TU Berlin, Berlin

Stork F, Uetz M (2005) On the generation of circuits and minimal forbidden sets. Math Program 102:185–203

Talla Nobibon F, Leus R (2014a) Complexity results and exact algorithms for robust knapsack problems. J Optimiz Theory App 161(2):533–552

Talla Nobibon F, Leus R (2014b) Robust maximum weighted independent-set problems on interval graphs. Optim Lett 8(1):227–235

Wald A (1950) Statistical decision functions. Wiley, New York

Walley P (1991) Statistical reasoning with imprecise probabilities. Monographs on statistics and applied probability, vol 42. Chapman and Hall, London

Wiest JD, Levy FK (1969) A Management Guide to PERT/CPM with GERT/PDM/DCPM and other Networks. Prentice-Hall, Englewood Cliffs

Part XIV
Project Scheduling Under Interval Uncertainty and Fuzzy Project Scheduling

Chapter 41
Temporal Analysis of Projects Under Interval Uncertainty

Christian Artigues, Cyril Briand, and Thierry Garaix

Abstract Given an activity-on-node network where every activity has an uncertain duration represented by an interval, this chapter takes an interest in computing the minimum and maximum earliest start times, latest start times and floats of all activities over all duration scenarios. The basic results from the literature are recalled and efficient solving algorithms are detailed. A particular focus is put on the computation of minimum float, which remains an $\mathcal{NP}$-hard optimization problem. For this last case, a recent and efficient branch and bound algorithm is described that outperforms previously proposed methods.

Keywords Algorithms • Branch-and-bound • Complexity • Interval uncertainty • Project scheduling • Temporal analysis

41.1 Introduction

In standard deterministic project scheduling, temporal analysis aims at determining the temporal degree of freedom of activities under simple finish-start precedence constraints. More precisely, it aims at computing for every activity i its earliest start and completion times ES_i and EC_i, its latest start and completion times LS_i and LC_i and its total float TF_i. It is well known that these values can be computed via longest path computation in the project network where each arc (i, j) is evaluated by the duration of i. More precisely, if d_{ij} denotes the length of the longest path from i to j in this graph, ES_i is the longest path from dummy node 0 to node i: $ES_i = d_{0i}$. LS_i is the length of the longest path from dummy node 0 do dummy node $n + 1$ (schedule length or makespan) minus the length of the longest path from node i to

C. Artigues (✉) • C. Briand
CNRS LAAS, Université de Toulouse, Toulouse, France
e-mail: artigues@laas.fr; briand@laas.fr

T. Garaix
CIS ROGI-LIMOS UMR CNRS 6158, École Nationale Supérieure des Mines de Saint-Étienne, Saint-Étienne, France
e-mail: garaix@emse.fr

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_11

node $n+1$: $LS_i = d_{0(n+1)} - d_{i(n+1)}$. The total float can be defined as the difference between LS_i and ES_i or, equivalently, $TF_i = d_{0(n+1)} - d_{0i} - d_{i(n+1)} = LS_i - ES_i$. Standard longest path computations in acyclic graph allow to compute all these values in $\mathcal{O}(|E|)$ time, E being the set of precedence constraints.

When problem parameters are ill-known, a common way of modeling uncertainty is to define each such parameter as an interval, as discussed in Chap. 40 of this book. In this chapter, we consider that the duration of each activity $i \in V$ is defined as an interval $[p_i^{min}, p_i^{max}]$. Under such an assumption, temporal analysis now focuses on the computation of the minimum and maximum values of the earliest start times, latest start times and total floats.

Let us define the scenario set Σ as the set of possible duration vectors:

$$\Sigma = \{\sigma \in \mathbb{R}^n | p_i^{min} \leq p_i \leq p_i^{max}, \forall i \in V\}$$

The temporal analysis of a project network under interval uncertainty consists in computing the following values for each activity $i \in V$:

- Best-case (minimum) earliest start time $ES_i^{min} = \min_{\sigma \in \Sigma} ES_i(\sigma)$.
- Worst-case (maximum) earliest start time $ES_i^{max} = \max_{\sigma \in \Sigma} ES_i(\sigma)$.
- Best-case (maximum) latest start time $LS_i^{max} = \max_{\sigma \in \Sigma} LS_i(\sigma)$.
- Worst-case (minimum) latest start time $LS_i^{min} = \min_{\sigma \in \Sigma} LS_i(\sigma)$.
- Best-case (maximum) float $TF_i^{max} = \max_{\sigma \in \Sigma} TF_i(\sigma)$.
- Worst-case (minimum) float $TF_i^{min} = \min_{\sigma \in \Sigma} TF_i(\sigma)$.

These values are of interest for providing valuable information to the decision-maker about the level of criticality of the activities despite the uncertain nature of the processing times. Any activity i such that $TF_i^{max} = 0$ is necessarily critical and should consequently be carefully monitored independently of the scenario. Any activity i such that $TF_i^{min} = 0$ is possibly critical and a special attention should be paid to it, especially for risk-adverse decision policies. On the contrary, if $TF_i^{min} > 0$, the information that the activity has flexibility for all scenarios is obtained.

Table 41.1 provides the minimum and maximum earliest start times, latest start times and floats for the project network displayed in Fig. 41.1.

We see in Table 41.1 that some non-dummy activities are necessarily critical (activities 3 and 8) while others are possibly critical (activities 1, 5, and 7). We also remark that intuition does not necessarily work to obtain the minimum total float. In fact we have some cases where $TF_i^{max} \neq LS_i^{max} - ES_i^{min}$ and others where $TF_i^{min} \neq LS_i^{min} - ES_i^{max}$.

Table 41.1 Minimum and maximum earliest start times, latest start times and floats

i	1	2	3	4	5	6	7	8	9
$[ES_i^{min}, ES_i^{max}]$	[0,0]	[0,0]	[0,0]	[3,4]	[10,10]	[4,8]	[12,13]	[19,22]	[20,25]
$[LS_i^{min}, LS_i^{max}]$	[0,8]	[2,14]	[0,0]	[6,17]	[10,18]	[17,23]	[13,20]	[19,22]	[20,25]
$[TF_i^{min}, TF_i^{max}]$	[0,8]	[2,14]	[0,0]	[2,14]	[0,8]	[9,19]	[0,8]	[0,0]	[0,0]

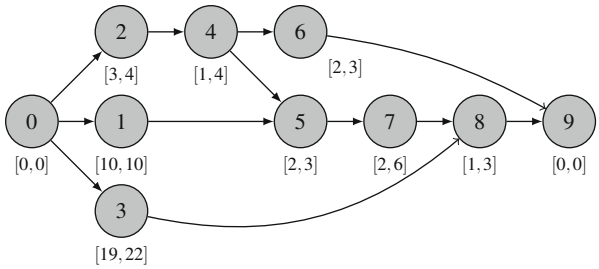


Fig. 41.1 Project network with interval uncertainty

Several authors have studied these temporal analysis problems. The chapter is based on the results obtained by Chanas et al. (2001, 2002), Chanas and Zieliński (2002, 2003), Dubois et al. (2003, 2005), Fargier et al. (2000), Fortin et al. (2010), Kasperski and Zieliński (2010), Zieliński (2003, 2005, 2006), Garaix et al. (2013). Among these works we refer in particular to the survey by Fortin et al. (2010) in which it is also remarked that the interval problems are particular cases of fuzzy project scheduling problems (see Chap. 42 of this book). Section 41.2 summarizes the main structural properties complexity results and proposed solution algorithms detailed by Fortin et al. (2010). Section 41.3 focuses on algorithms for the minimum and maximum latest start time problems. Section 41.4 presents the algorithms for solving the minimum and maximum float problems. Finally, Sect. 41.5 provides concluding remarks and direction for future works in connection with related problems, such as controllability of simple temporal networks with uncertainty.

Let us precise that in this chapter, in contrast with some of the above-cited papers, the formalism on Activity-on-Node graph is used instead of the one of Activity-on-Arc (AoA).

41.2 Basic Properties, General Algorithms and Complexity Results (Fortin et al. 2010)

41.2.1 Extreme Scenarios

Let us define the notion of extreme scenario induced by an activity subset. Let $\sigma^{max}(Q)$, with $Q \subseteq V$, the extreme scenario such that each activity $i \in Q$ is set to p_i^{max} while each activity of $i \in V \setminus Q$ is set to p_i^{min}. Remark that the minimum and maximum earliest start times of each activity $i \in V$ are attained on extreme scenarios induced by the empty set and V, respectively:

$$ES_i^{min} = ES_i\ (\sigma^{max}(\emptyset))$$
$$ES_i^{max} = ES_i\ (\sigma^{max}(V))$$

For determining the latest start times and floats, the solution is not so trivial. However Dubois et al. (2003) showed that the searched minimum and maximum values are attained for extreme scenarios.

Theorem 41.1 (Dubois et al. 2003). *For every activity $i \in V$:*

$$LS_i^{max} = \max_{Q \subseteq V} LS_i\left(\sigma^{max}(Q)\right)$$

$$LS_i^{min} = \min_{Q \subseteq V} LS_i\left(\sigma^{max}(Q)\right)$$

$$TF_i^{max} = \max_{Q \subseteq V} TF_i\left(\sigma^{max}(Q)\right)$$

$$TF_i^{min} = \min_{Q \subseteq V} TF_i\left(\sigma^{max}(Q)\right)$$

Proof. To understand these properties, let us illustrate them on the minimum float case. Suppose that the minimum float of an activity i is reached on a scenario σ that is not extreme. Let P denote the longest path passing through i on scenario σ. Suppose in addition that among all scenarios yielding the minimum float for i, σ is such that the position on P of the first activity j not set to its largest or smallest duration is minimum. First, let us change scenario σ in scenario σ' by modifying the duration of all activities—except those located on P—by switching them to their minimum duration. Obviously the longest path in the network cannot be increased by this operation and P remains one of the longest path passing through i. It follows that the float of i cannot be increased by this modification of σ. We further modify σ' by increasing j to its maximum duration so as to obtain scenario σ''. This obviously increases the longest path passing through i by $p'_j - p_j^{min}$. This also potentially increases the longest path in the network, but at most by $p'_j - p_j^{min}$. Hence the float of i cannot be decreased by this last change, which contradicts the minimality of the position of j in P. □

A purely enumerative algorithm would then enumerate all 2^n extreme scenarios and, for each of them, compute the required longest paths. In the eight non-dummy activities example of Fig. 41.1, these would yield to 256×2 longest path computations.

41.2.2 Path-Induced Extreme Scenarios

Dubois et al. (2005) further restricted the search space by establishing the following properties on extreme scenarios induced by paths. They first show that the minimum and maximum floats, as well as the maximum latest start time of any activity are always attained for an extreme scenario induced by the activities located on a $(0\text{–}n+1)$ path. Let $\mathscr{P}_{ij}$ denote the set of paths from activity i to activity j. By a slight abuse of notation, we identify the set of activities located on a path P by the path itself.

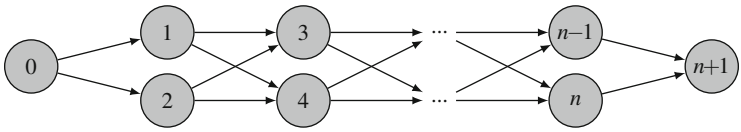


Fig. 41.2 Project network with an exponential number of paths

Theorem 41.2 (Dubois et al. 2005). *For every activity $i \in V$:*

$$LS_i^{max} = \max_{P \in \mathscr{P}_{0(n+1)}} LS_i\,(\sigma^{max}(P))$$

$$TF_i^{max} = \max_{P \in \mathscr{P}_{0(n+1)}} TF_i\,(\sigma^{max}(P))$$

$$TF_i^{min} = \min_{P \in \mathscr{P}_{0(n+1)}} TF_i\,(\sigma^{max}(P))$$

Proof. Observing the proof of Theorem 41.1, we can only change the definition of σ as an optimal extreme scenario not induced by a path and of j as the activity located in P with minimal position such that $p_j \neq p_j^{\max}$. Then the modifications change P into an optimal path where j is set to $p_j^{\max}$. □

Then, to compute the maximum latest start times, the minimum and the maximum float, one could enumerate all paths in $\mathscr{P}_{0(n+1)}$, and for each of them, obtain the induced extreme scenario and compute the longest path lengths.

In the example from Fig. 41.1, the number of longest path computations boils down to $|\mathscr{P}_{09}| \times 2 = 8$. Although the number of paths from 0 to $n+1$ is always smaller than 2^n and can be small for sparse graphs, it can also be unfortunately exponential in n, as illustrated in the network given in Fig. 41.2 where the number of paths from 0 to $n+1$ is equal to $2^{\frac{n}{2}}$.

The reader may have noticed that the minimum latest start time was excluded from Theorem 41.2. For this value, Dubois et al. (2005) show that one can restrict the search to scenarios induced by a path from i to $n+1$.

Theorem 41.3 (Dubois et al. 2005). *For every activity $i \in V$:*

$$LS_i^{min} = \min_{P \in \mathscr{P}_{i(n+1)}} LS_i\,(\sigma^{max}(P))$$

Proof. The proof proceeds the same way as for Theorems 41.1 and 41.2. Suppose σ is an optimal extreme scenario not induced by a $(i\text{–}n+1)$ path and let P denote the longest path from i to $n+1$ for scenario σ. Setting all activities in $V \setminus P$ to their minimum duration cannot increase the latest start time. Setting then all activities duration of P to their maximum duration can only decrease or leave unchanged the latest start time. □

There can still be an exponential number of paths in $\mathscr{P}_{i(n+1)}$. However Fortin et al. (2010) provided a dynamic programming recursion allowing to compute the optimal path of an activity i in a polynomial time (see Sect. 41.3.2), given the optimal paths of its direct successors.

Can we obtain the same positive complexity results for the maximum latest start time, the minimum and maximum floats?

41.2.3 Complexity Results

Actually, asserting the necessary criticality of an activity requires $\mathscr{O}(|E||V|)$ time using the algorithm proposed in Fortin et al. (2010), while the asserting the possibly criticality is strongly $\mathscr{NP}$-complete (Chanas and Zieliński 2002). However, both problems were shown to be polynomial by Zieliński (2005) in $\mathscr{O}(|E| + |V|)$ time, when durations of the predecessors of the activities are precisely known (Zieliński 2005).

Even if the minimum float problem is polynomial for series-parallel graphs (Fargier et al. 2000; Zieliński 2006), these positive complexity results cannot be extended to this problem for general graphs. The minimum float problem was indeed proven to be strongly $\mathscr{NP}$-hard and even has no polynomial approximation (Chanas et al. 2002; Chanas and Zieliński 2002; Zieliński 2005).

41.2.4 Link with Min Max Regret Longest Path Problems

Let us point out that the minimum float problem can be linked to the min max regret longest path problem in acyclic graphs. Let P be a path between 0 and $n + 1$. Given a scenario σ, the regret of P is the difference between the length of the longest path in the network for scenario σ and the length of P for scenario σ. The min max regret longest path problems amounts to find a path P^* from 0 to $n + 1$ that minimizes this maximum regret.

If there exists a scenario for which the min max regret is zero, we have found a scenario for which P is critical, which amounts to find a set of possibly critical activities, with a zero minimum float. Conversely if we have a path of necessarily critical activities (of zero maximum floats), we have found a critical path for all scenarios and consequently a path of zero min max regret.

In the remaining of this chapter, we describe the polynomial algorithms for computing the minimum, maximum latest start times and the maximum float. We also describe a branch and bound algorithm that performs remarkably well, compared to the path algorithm, for the minimum float problem.

41.3 Maximum and Minimum Latest Start Time

In Zieliński (2005), two similar polynomial algorithms are given to compute the minimum and maximum latest start times of an activity. They are respectively linked to two underlying problems; asserting on possibly and necessary criticality of an activity when durations of its predecessors are fixed. For the case of minimum latest start time, another recursive algorithm is proposed in Fortin et al. (2010). In this section, we propose a slightly different description of these algorithms, for the sake of clarity and concision.

41.3.1 Maximum Latest Start Time

Theorem 41.4 allows to restrict the number of scenarios to consider during the search of the maximum latest start time of an activity i. We refer to $\overline{Succ}(i)$ ($\overline{Pred}(i)$) as the set of all immediate and transitive successors (predecessors, respectively) of each activity i.

Theorem 41.4. *The set of scenarios* $\Sigma^{i,max} = \{\sigma \in \Sigma | p_j(\sigma) = p_j^{max}, \forall j \notin \overline{Succ}(i) \cup \{i\}\}$ *is dominant when the maximum latest start time of activity* $i \in V$ *is sought.*

Proof. The proof is straightforward as the latest start time of an activity LS_i, $i \in V$, is defined by the gap between $d_{0(n+1)}$ and $d_{i(n+1)}$. Transformations of a scenario by switches from p_j^{min} to p_j^{max} on activities $j \notin \overline{Succ}(i) \cup \{i\}$ can only increase this gap and, finally, reach a scenario of $\Sigma^{i,max}$. □

In order to describe the algorithm of Zieliński (2005), it is necessary to link the maximum latest start time of an activity to its necessary criticality. Let $\Sigma^{i,max}(p_i = v)$ the subset of scenarios of $\Sigma^{i,max}$ where in addition $p_i = v$, v being any value in $[p_i^{min}, p_i^{max}]$.

Theorem 41.5. $LS_i^{max} = ES_i^{max} + \Delta$, *where* $\Delta = \min\{\delta \in \mathbb{R} | i$ *is necessary critical in* $\{\sigma \in \Sigma^{i,max}(p_i = p_i^{min} + \delta)\}\}$, $\forall i \in V$.

Proof. It is enough to show that $ES_i^{max} + \Delta = ES_i(\sigma) + \Delta = LS_i(\sigma)$ for all scenarios of $\Sigma^{i,max}(p_i = p_i^{min} + \Delta)$. For any $\sigma \in \Sigma^{i,max}(p_i = p_i^{min} + \Delta)$, $LS_i(\sigma)$ can not be strictly greater than $ES_i(\sigma) + \Delta$, by definition of necessary criticality of i. A strictly lower value of $LS_i(\sigma)$ contradicts the minimality of Δ. □

The main idea of Algorithm 41.1 is to iteratively increment the duration of i until reaching the minimum value ($p_i^{min} + \Delta$), such that i becomes necessarily critical, the considered set of scenarios being restricted to $\Sigma^{i,max}$. Note that for large enough values of Δ, $\Sigma^{i,max}(p_i = p_i^{min} + \Delta)$ does not include feasible scenarios since p_i can become greater than p_i^{max}. The tricky point is to define the increment step at

Algorithm 41.1 Computing LS_i^{max}

1: $\Delta := 0$;
2: $\delta := \min_{j \in \overline{Succ}(i)} \{\delta_j(\sigma)\}$ on $\Sigma^{i,max}(p_i = p_i^{min} + \Delta)$;
3: **if** $\delta > 0$ **then**
4: $\quad \Delta := \Delta + \delta$ and **goto** Step 2;
5: **end if**
6: $LS_i^{max} := ES_i^{max} + \Delta$;

each iteration. That is done by considering activities $j \in \overline{Succ}(i)$ for which i is not j*-critical*, this concept being defined in Theorem 41.6.

Theorem 41.6. *Activity i is not necessary critical in $\sigma \in \Sigma^{i,max}(p_i(\sigma) = \bar{p})$, if and only if there exists at least one activity $j \in \overline{Succ}(i)$ on each longest path from i to $n+1$ such that $\delta_j(\sigma) = d_{0j}(\sigma) - (d_{0i}(\sigma) + d_{ij}(\sigma)) > 0$, i.e., the longest path from 0 to j traversing i is not a longest path from 0 to j in σ. Activity i is said j-critical when $\delta_j(\sigma) = 0$.*

Proof. The proof is directly derived from the definition of the criticality of an activity. □

According to Theorems 41.5 and 41.6, for any duration $\bar{p} < p_i^{min} + \Delta$, there exists at least one scenario $\sigma \in \Sigma^{i,max}(p_i(\sigma) = \bar{p})$ and one activity $j \in \overline{Succ}(i)$ such that i is not j-critical ($\delta_j(\sigma) > 0$). Then an increase of the duration of i by the minimum $\delta = \min_{j \in \overline{Succ}(i)} \delta_j(\sigma)$ will decrease by at least one (j itself) the number of activities for which i is not j-critical in each scenario. The algorithm ends when i becomes j-critical for all activities of $j \in \overline{Succ}(i)$, and so necessary critical as expected in Theorem 41.5.

At the initialization phase, the maximum earliest start time of i and the minimum earliest start times of activities $\overline{Succ}(i)$ in scenarios $\Sigma^{i,max}(p_i = p_i^{min})$ can be computed, as a preprocessing phase, through a PERT algorithm under extreme scenario $\sigma^{max}\left(V \setminus (\overline{Succ}(i) \cup \{i\})\right)$. The procedure called at Step 2 of Algorithm 41.1, which computes values $\delta_j(\sigma)$, is detailed in Algorithm 41.2. Since the topological order is followed, the value of $\delta_j(\sigma)$ only depends on (predecessor) activities with fixed durations; initialized at Step 1 and updated at Steps 4–8 of Algorithm 41.2. By construction (Steps 4–8), the gap between $d_{0j}(\sigma)$ and $d_{0i}(\sigma) + d_{ij}(\sigma)$, is decreased as much as possible for next activities $k \in \overline{Succ}(j)$. Thus, each computed value of $\delta_j(\sigma)$ is maximal on $\Sigma^{i,max}(p_i = p_i^{min} + \Delta)$. We highlight that $\delta_j(\sigma)$ can be viewed as the total float of activity i, under scenario $\Sigma^{i,max}(p_i = p_i^{min} + \Delta)$, in the subgraph involved by $\overline{Pred}(j)$ with $j \in \overline{Succ}(i)$. The evaluation of $\delta_j(\sigma)$ can be done in constant time as lengths of partial longest paths $d_{0j}(\sigma)$ and $d_{ij}(\sigma)$ can be dynamically updated. Therefore, Algorithm 41.2 runs in $\mathcal{O}(|E|)$ time and Algorithm 41.1 in $\mathcal{O}(|V||E|)$ time.

Note that Algorithm 41.2 allows to assert the necessary criticality of an activity when the durations of its predecessors are precisely known (Zieliński 2005).

Algorithm 41.2 Computing $\delta_j(\sigma), \forall j \in \overline{Succ}(i)$ on $\Sigma^{i,max}(p_i = p_i^{min} + \Delta)$

1: Set partial scenario $\sigma \in \Sigma^{i,max}(p_i = p_i^{min} + \Delta)$;
2: $j :=$ next activity of $\overline{Succ}(i)$ according to the topological order;
3: $\delta_j(\sigma) := d_{0j}(\sigma) - d_{0i}(\sigma) - d_{ij}(\sigma)$;
4: **if** $\delta_j(\sigma) > 0$ **then**
5: $\quad p_j(\sigma) := p_j^{max}$;
6: **else**
7: $\quad p_j(\sigma) := p_j^{min}$;
8: **end if**
9: **if** $j = n + 1$ **then**
10: $\quad$ **stop**;
11: **else**
12: $\quad$ **goto** Step 2;
13: **end if**

41.3.2 Minimum Latest Start Time

Theorem 41.3 gives graph-topological properties that reduce the search space to longest paths from i to $n + 1$. The recursion procedure of Theorem 41.7 allows to compute the optimal path from i to $n + 1$.

Theorem 41.7 (Fortin et al. 2010). *For each activity $i \in V$:*

$$LS_i^{min} = \min_{j \in Succ(i)} LS_i\left(\sigma^{max}(\{i\} \cup Q_j)\right) \text{ with } Q_j \in \mathscr{P}_{i(n+1)}, LS_j^{min} = LS_j\left(\sigma^{max}(Q_j)\right)$$

From this recursion, as a unique path can be computed for each successor, a polynomial algorithm can be obtained. Fortin et al. (2010) proposed an $\mathscr{O}((|E| + |V|)^2)$ algorithm. This shows that the minimum latest start time computation is polynomial despite the possibly exponential number of paths in $\mathscr{P}_{i(n+1)}$.

Another $\mathscr{O}((|E| + |V|)^2)$ time algorithm has been previously proposed in Zieliński (2005). Algorithm 41.3 is similar to the algorithm that computes the maximum latest start time of an activity but it is based on the relation between the minimum latest start time and the possible criticality of an activity.

The dominant set of scenarios is denoted $\Sigma^{i,min}$ and is defined by setting durations of activities of $V \setminus \{\overline{Succ}(i) \cup \{i\}\}$ to their respective lower bounds. This result is justified in the argument of Theorem 41.3.

Here Δ represents the minimum value to add to the duration of i to make it possibly critical, i.e., j-critical in at least one scenario. This procedure is given in Algorithms 41.3 and 41.4. The main difference with Algorithms 41.1 and 41.2, which gives LS_i^{max}, is the updating step of partial scenario σ. The gap between $d_{0k}(\sigma)$ and $d_{0i}(\sigma) + d_{ik}(\sigma)$ for next activities $k \in \overline{Succ}(i)$, can only be decreased by Steps 4–8 of Algorithm 41.4.

Again, one can derive an algorithm, presented in Zieliński (2005), which allows to assert the possibly criticality of an activity when its predecessors have fixed durations.

Algorithm 41.3 Computing LS_i^{min}

1: $\Delta := 0$;
2: $\delta := \min_{j \in \overline{Succ}(i)} \{\delta_j(\sigma)\}$ on $\Sigma^{i,min}(p_i = p_i^{max} + \Delta)$;
3: **if** $\delta > 0$ **then**
4: $\quad \Delta := \Delta + \delta$ and **goto** Step 2;
5: **end if**
6: $LS_i^{min} := ES_i^{min} + \Delta$;

Algorithm 41.4 Computing $\delta_j(\sigma), \forall j \in \overline{Succ}(i)$ on $\Sigma^{i,min}(p_i = p_i^{max} + \Delta)$

1: Set partial scenario $\sigma \in \Sigma^{i,min}(p_i = p_i^{max} + \Delta)$;
2: $j :=$ next activity of $\overline{Succ}(i)$ according to the topological order;
3: $\delta_j(\sigma) := d_{0j}(\sigma) - d_{0i}(\sigma) - d_{ij}(\sigma)$;
4: **if** $\delta_j(\sigma) > 0$ **then**
5: $\quad p_j(\sigma) := p_j^{min}$;
6: **else**
7: $\quad p_j(\sigma) := p_j^{max}$;
8: **end if**
9: **if** $j = n + 1$ **then**
10: $\quad$ **stop**;
11: **else**
12: $\quad$ **goto** Step 2;
13: **end if**

41.4 Minimum and Maximum Floats

41.4.1 Maximum Floats

In Zieliński (2005), the author proposed a polynomial time algorithm able to determine whether an activity i is necessarily critical. It is based on two properties. The first one states that if i is necessarily critical in a subgraph made of the activities of $\overline{Succ}(j)$, for some $j \in \overline{Pred}(i)$, then i is necessarily critical in the general graph if and only if there exists a scenario σ with $p_j(\sigma) = p_j^{min}$ such that $TF_i(\sigma) = 0$. The second one claims that if i is *not* necessarily critical in the subgraph made of the activities of $\overline{Succ}(j)$, for some $j \in \overline{Pred}(i)$, then i is necessarily critical in the general graph if and only if there exists a scenario σ with $p_j(\sigma) = p_j^{max}$ such that $TF_i(\sigma) = 0$. These properties allow to derive Algorithm 41.5 that asserts the necessary criticality of activity i. The algorithm is called with the initial scenario σ^{min}. At the first iteration (i.e., $j = i$), it first calls Algorithm 41.2 (see Step 2) to determine whether i is necessarily critical in the graph made of the activities of $\overline{Succ}(i)$. Thus it is possible to fix the durations of the activities immediately preceding i, with respect to both above properties, without modifying the necessary criticality of i (at Steps 3–7). We can reiterate this processus considering the activities of $\overline{Pred}(i)$ according to the reverse topological order (see Step 1). Once the durations of the activities belonging to $\overline{Pred}(i)$ are totally set, i is necessarily critical if and only if, at the last iteration, $TF_i(\sigma) = 0$. Algorithm 41.5 asserts the

Algorithm 41.5 Asserting the necessary criticality of activity i

1: $j :=$ next activity of $\{\overline{Pred}(i) \cup \{i\}\}$ according to the *reverse* topological order;
2: Compute $\delta = TF_i(\sigma)$ in the subgraph $\overline{Succ}(j)$ using Algorithm 41.2;
3: **if** $\delta > 0$ **then**
4: $\quad p_j(\sigma) := p_j^{max}$;
5: **else**
6: $\quad p_j(\sigma) := p_j^{min}$;
7: **end if**
8: **if** $j \neq 0$ **then**
9: $\quad$ **goto** Step 1
10: **end if**;

necessarily criticality of i in $\mathcal{O}((|E| + |V|)^2)$ time. Nevertheless, let us highlight that in the case where i is not necessarily critical, the value $TF_i(\sigma) > 0$ eventually found gives a lower bound of TF_i^{max}.

The algorithm that computes the maximum float of activities is based on the previous algorithm and uses the following property.

Theorem 41.8 (Zieliński 2005). *If Δ is the smallest positive real value such that i becomes necessarily critical in scenario σ such that $p_i(\sigma) = p_i^{min} + \Delta$, then $TF_i^{max} = \Delta$.*

Proof. The argument is based on the fact that, whatever the considered scenario $\sigma \in \Sigma$, any longest path traversing i remains a longest path if p_i is increased by a small value. If Δ is the minimum value to add to p_i such that i become necessarily critical (i.e., $TF_i(p_i^{min} + \Delta) = 0$) then, because $TF_i(p_i^{min} + \Delta) \leq TF_i^{max}$, it easy to deduce the claimed property. □

The problem is now to find Δ. The idea is to use a similar approach that the one used for computing LS_i^{max} using the property (proved in Zieliński 2003) that $TF_i^{max} = LS_i^{max} - ES_i^{max}$ in the case where the durations of the activities belonging to $\overline{Pred}(i)$ are known (that property being falsed in the general case). So, it becomes possible to determine the smallest δ such that i becomes necessarily critical in a subgraph made by the activities of $\overline{Succ}(j)$, with $j \in \overline{Pred}(i)$.

The above idea is implemented in Algorithm 41.6, which works as follows. A first assignment of the durations of the activities $\overline{Pred}(i)$ is made by Algorithm 41.5 in Step 3. If $TF_i(\sigma) = 0$ then i is necessarily critical. Otherwise, for every $j \in \overline{Pred}(i)$ and for the scenario σ determined by Algorithm 41.5 (i.e., the durations of $\overline{Pred}(i)$ are known), the lower bound of the maximum float $LS_i^{max}(\sigma) - d_{0i}(\sigma)$ is computed in the graph $\overline{Succ}(j)$ using Algorithm 41.1 (see Step 7). The value δ of the smallest positive lower bound is then memorized (see Steps 8–10). Finally, Δ and p_i are incremented by δ in Steps 14 and 15 and the value of $TF_i(\sigma)$ is recomputed. The algorithm ends when i becomes critical in σ and $TF_i^{max} = \Delta$. This algorithm works in $\mathcal{O}((|E| + |V|)^4)$.

Algorithm 41.6 Computing $TF_i^{max} = \Delta$

1: $p_j(\sigma) := p_j^{min}, \forall j \in V \setminus \{\overline{Pred}(i) \cup \overline{Succ}(i)\}$;
2: $\Delta := 0$;
3: Compute $TF_i(\sigma)$ and update σ with respect to decisions made in Algorithm 41.5;
4: **while** $TF_i(\sigma) > 0$ **do**
5: $\quad\delta := +\infty$;
6: $\quad j :=$ previous activity of $\overline{Pred}(i)$ according to the reverse topological order;
7: $\quad$Compute $LS_i^{max}(\sigma)$ in the subgraph $\overline{Succ}(j)$ using Algorithm 41.1;
8: $\quad$**if** $LS_i^{max}(\sigma) - d_{0,i}(\sigma) > 0$ **then**
9: $\quad\quad\delta_j(\sigma) := \min(\delta, LS_i^{max}(\sigma) - d_{0,i}(\sigma))$;
10: $\quad$**end if**
11: $\quad$**if** $j \neq 0$ **then**
12: $\quad\quad$**goto** Step 6
13: $\quad$**end if**;
14: $\quad\Delta := \Delta + \delta$;
15: $\quad p_i(\sigma) := p_i^{min} + \Delta$;
16: $\quad$Compute $TF_i(\sigma)$ and update σ with respect to decisions made in Algorithm 41.5;
17: **end while**

41.4.2 Minimum Floats

This subsection focuses on the computation of the minimum float TF_i^{min} of every activity i, which is an $\mathcal{NP}$-hard problem (Chanas et al. 2002; Chanas and Zieliński 2002; Zieliński 2005). From Theorem 41.2, we know that any optimum is obtained for a particular scenario $\sigma^{max}(P)$ with $P \in \mathcal{P}_{0(n+1)}$. Note that P is also the longest path from 0 to $n+1$ traversing i. Using this property, Dubois et al. (2005) and Fortin et al. (2010) proposed a first exact algorithm based on path enumeration, the float of activities being computed for every generated path using standard PERT method. Typically, this algorithm is able to compute in a few seconds the minimum float TF_i^{min} of an activity on medium-density graph with 100 activities. However, its performance gets worse on high density graph since the number of paths to explore grows exponentially.

A faster branch-and-bound procedure was recently proposed in Garaix et al. (2013). It takes benefits from other problem properties. This procedure is able to compute minimum floats within few milliseconds for 100-activity graph, this CPU-time remaining rather insensitive to graph density variation. We review in this section the basic ingredients of the Garaix et al. (2013) procedure: a dominance property and a bounding rule.

Let us first take an interest in the structure of the longest path obtained for a path-induced extreme scenario $\sigma^{max}(P)$.

The dominance Theorem 41.9 below states that the longest path $P' \in \mathcal{P}_{0(n+1)}$ in any optimal scenario $\sigma^{max}(P)$ differs from P only by a subpath $P'_{a\to b}$, a being a predecessor of i in P, and b a successor of i in P. Moreover, in the particular case where $TF_i^{min} = 0$ then, since P and P' are identical, $a = b = i$. This property is illustrated in Fig. 41.3 where the arcs in bold define the longest path P traversing i inducing the extreme scenario. The thin arc $P'_{a\to b}$ represents the deviation from

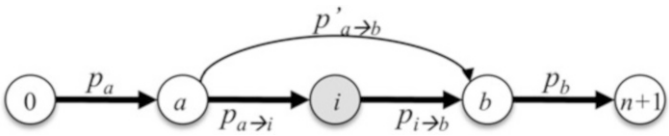


Fig. 41.3 Dominant path structures for float computation

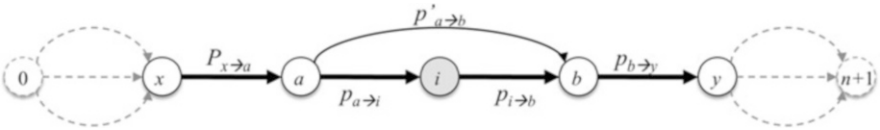


Fig. 41.4 Partial path structure for float computation

P of the longest path P' that results from the concatenation of three partial paths (i.e., $P' = (P_{0\to a}, P'_{a\to b}, P_{b\to n+1})$). Furthermore, the length of $P'_{a\to b}$ equals the length of the longest path from a to b in the scenario $\sigma^{min}(V)$, plus $p_a^{max} - p_a^{min}$ (the duration of activity a being set to p_a^{max} as it belongs to P).

Theorem 41.9 (Garaix et al. 2013). *For any activity $i \in V$, there exists an optimal path P with scenario $\sigma^{max}(P)$ and a pair of activities $a, b \in P$ such that the longest path P' in this scenario is $P' = (P_{0\to a}, P'_{a\to b}, P_{b\to n+1})$ verifying*

$$TF_i^{min} = d_{ab}(\sigma^{min}(V)) + p_a^{max} - p_a^{min} - \sum_{k \in P_{a\to b}\setminus\{b\}} p_k^{max}$$

Proof. The proof of this theorem goes by showing that if $P_{0\to a}$ is not a longest path from 0 to a then there exists an alternative path $P'_{0\to a}$ such that $TF_i(\sigma^{max}(P)) \geq TF_i(\sigma^{max}(P'_{0\to a}, P_{a\to b}, P_{b\to n+1}))$. Similarly, if $P_{b\to n+1}$ is not a longest path from b to $n+1$ then there exists another scenario path-induced scenario leading to a lower float for i. □

A major interest of Theorem 41.9 is to formalize the intuitive fact that any path-induced scenario $\sigma^{max}(P)$ such that the subpath from 0 to a is not also a longest path from 0 to a can be discarded (since it is dominated with respect to the minimization of TF_i). Symmetrically, any path $P \in \mathscr{P}$ whose subpath from b to $n+1$ is not also a longest path from b to $n+1$ under scenario $\sigma^{max}(P)$, is also dominated. In the sequel, any path which cannot be discarded in this way will be said *valid* with respect to the minimization of TF_i.

Let us take an interest now in the computation of lower bounds for TF_i^{min}. For that purpose, let us consider a valid *partial* path $P_{x\to y}$ with $i \in P_{x\to y}$ (see Fig. 41.4). We highlight that $P_{x\to y} = (P_{x\to a}, P_{a\to i}, P_{i\to b}, P_{b\to y})$ is said valid in the sense that $P' = (P_{x\to a}, P'_{a\to b}, P_{b\to y})$ is the longest path in the scenario $\sigma^{max}(P_{x\to y})$ between x and y.

From the optimality principle of Bellman it is easy to show that the following theorem holds:

Theorem 41.10 (Garaix et al. 2013). *Considering a valid* partial *path* $P_{x\to y}$ *with* $i \in P_{x\to y}$*, if there exists a valid path* $P_{0\to n+1}$ *extending* $P_{x\to y}$ *towards activities* 0 *and* $n+1$*, it satisfies:*

$$TF_i(\sigma^{max}(P)) \geq LB_i(P_{x\to y})$$

with

$$LB_i(P_{x\to y}) = d_{ab}(\sigma^{min}(V)) + p_a^{max} - p_a^{min} - \sum_{k\in P_{a\to b}\setminus\{b\}} p_k^{max}$$

Proof. Considering a valid path extension $P_{0\to n+1}$ of a partial path $P_{x\to y}$, we know from Theorem 41.9 that there exists a^* and $b^* \in V$ such that $TF_i(\sigma^{max}(P)) = d_{a^*b^*}(\sigma^{min}(V)) + p_{a^*}^{max} - p_{a^*}^{min} - \sum_{k\in P_{a^*\to b^*}\setminus\{b^*\}} p_k^{max}$. From Bellman's optimality principle, any path from 0 to $n+1$ diverging from P from an activity $a \neq a^*$ and converging back to P on an activity $b \neq b^*$ has a length lower or equal to the longest one passing through a^* and b^*. From this latter property, it is easy to deduce the inequality claimed in Theorem 41.10. □

In other words, whatever the considered possible valid extension P of $P_{x\to y}$, Theorem 41.10 ensures that $TF_i(\sigma^{max}(P))$ will never be lower than $LB_i(P_{x\to y})$. This value actually corresponds to the float of i under the scenario $\sigma^{max}(P_{x\to y})$ when only the activities belonging to $(P_{x\to y} \cup P'_{a\to b})$ are considered.

Theorems 41.9 and 41.10 allow to design an efficient branch-and-bound procedure for the computation of the minimum floats. In this procedure, the nodes of the search tree correspond to valid partial paths related to a given activity i, which are stored inside a stack $\mathscr{Q}$ for depth-first search. Given a valid partial path $P_{x\to y}$ with $i \in P_{x\to y}$, the branching scheme consists in alternatively extending the path to the left or to the right, by considering either all the immediate predecessors of x, or all the immediate successors of y, discarding the non-valid path extensions (with respect to the dominance rule). Thus, considering a given path-extension direction δ, a node has always as many children as immediate valid path-extensions. Each node of the search tree also memorizes the direction δ ($\delta \in$ {left, right}) to consider for the next path extension. A leaf of the search tree corresponds to a valid path P from 0 to $n+1$ and is evaluated by $TF_i(\sigma^{max}(P))$. Classically, a partial path $P_{x\to y}$ is deleted only if its current evaluation, $LB_i(P_{x\to y})$, is not smaller than the best float TF_i already found.

Let us comment on Algorithm 41.7. We remark first that the longest path values $d_{ij}(\sigma^{min}(V))$ are precomputed using the variant of Bellman–Ford's algorithm for DAGs. Computing the minimum float of all activities requires running the branch-and-bound procedure n times (see Step 1). At the beginning of each run (Steps 2 and 3), TF_i is set to ∞, stack $\mathscr{Q}$ only contains a single partial path $P_{x\to y} = i$, and the path-extension direction is set to left by default. The branch-

Algorithm 41.7 Branch-and-bound

1: **for all** $i \in V \setminus \{0, n+1\}$ **do**
2: $TF_i := \infty$;
3: push($\mathscr{P}$, (i, left));
4: **while** $\mathscr{P} \neq \emptyset$ **do**
5: $(p_{x\to y}, \delta)$:=pop($\mathscr{P}$);
6: $(a, b) := \underset{\{u \in p_{x\to i}, v \in p_{i\to y}\}}{\operatorname{argmax}} \; d_{u,v}(\sigma^{min}(V)) + p_u^{max} - p_u^{min} - \sum_{k \in P_{u\to v}\setminus\{v\}} p_k^{max}$;
7: **if** $d_{a,b}(\sigma^{min}(V)) + p_a^{max} - p_a^{min} - \sum_{k \in P_{a\to b}\setminus\{b\}} p_k^{max} < TF_i$ **then**
8: **if** $x = 0$ AND $y = n+1$ **then**
9: $TF_i := d_{a,b}(\sigma^{min}(V)) + p_a^{max} - p_a^{\min} - \sum_{k \in P_{a\to b}\setminus\{b\}} p_k^{max}$;
10: **else if** δ =left **then**
11: **for all** $x' \in \Gamma^{-1}(x)$ **do**
12: **if** $p_{x'}^{max} + \sum_{k \in P_{x\to u}\setminus\{u\}} p_k^{max} \geq d_{x',u}(\sigma^{min}(V)) + p_{x'}^{\max} - p_{x'}^{min} \quad (u \in P_{x\to i})$ **then**
13: **if** $y \neq n+1$ **then**
14: δ :=right;
15: **else**
16: δ :=left;
17: **end if**
18: push$\big(\mathscr{P}, \big((x', P_{x\to y}), \delta\big)\big)$;
19: **end if**
20: **end for**
21: **else if** δ =right **then**
22: **for all** $y' \in \Gamma(y)$ **do**
23: **if** $p_{y'}^{max} + \sum_{k \in P_{v\to y}\setminus\{v\}} p_k^{max} \geq d_{v,y'}(\sigma^{min}(V)) + p_v^{max} - p_v^{min} \quad (v \in p_{i\to y})$ **then**
24: **if** $x \neq 0$ **then**
25: δ :=left;
26: **else**
27: δ :=right;
28: **end if**
29: push($\mathscr{P}$, $\big((P_{x\to y}, y'), \delta\big)$);
30: **end if**
31: **end for**
32: **end if**
33: **end if**
34: **end while**
35: **end for**

and-bound procedure is implemented in Steps 4–34. While stack $\mathscr{Q}$ is not empty, a partial path $P_{x\to y}$ is taken from the stack, with its path-extension direction δ (Step 5). Preliminarily, the activities (a, b) for which the longest path from x to y differs from $p_{x\to y}$ are updated (see Step 6). Note that this can be done incrementally in linear time: if x (y) is the last activity toward which the path has been extended, only pairs (u, v) such that $u = x$ ($v = y$) and $v \in P_{i\to y}$ ($u \in P_{x\to i}$) are considered, respectively.

If $TF_i \leq LB_i(P_{x\to y})$, the path is deleted (see Step 7). Otherwise, if $P_{x\to y} \in \mathscr{P}_{0(n+1)}$, the new TF_i value is memorized (see Steps 8–9). If $P_{x\to y} \notin \mathscr{P}_{0(n+1)}$, it is extended with respect to direction δ. Below, only the case $\delta =$ left is commented on (see Steps 10–20), the case $\delta =$ right (see Steps 21–32) being symmetrical.

All the immediate predecessors x' of x are first considered for possible path extension $(x', P_{x \to y})$ (see Step 11). With respect to Theorem 41.9, Step 12 verifies that it does not exist any activity $u \in P_{x \to i}$ such that the path $(x', P_{x \to u})$ has a smaller length than the one of the longest path between x' and u, otherwise the path extension is not valid. We underline that the validity of a path extension can be checked in linear time since all longest path at minimum duration have been precomputed. Once a new left-path-extension is found, the next extension direction is set to right unless $y = n + 1$ (see Steps 13–18).

41.5 Conclusions

Computing the minimum and maximum values for the starting times and floats of project activities is a major concern of project managers. This assertion remains particularly valid when activity durations are modelled as intervals. Indeed, uncertain durations bring the concepts of possible and necessary activity criticality. This chapter showed how the necessary criticality can be checked in polynomial time, while the possible criticality remains $\mathscr{NP}$-hard to assert in general. An effective branch-and-bound procedure is proposed to cope with this last problem, which is able to compute the minimum float of the project activities.

It has already been pointed out by Fortin et al. (2010) the close connections of the criticality analysis presented in this chapter with fuzzy PERT scheduling problems on the one hand and min-max regret longest path problems on the other hand. We also mention here that a closely related model, the simple temporal networks under uncertainty (STNU) have also been studied in the artificial intelligence community, see, e.g., Morris et al. (2001). Actually STNUs can be seen as a generalization of the activity network with uncertain interval duration to generalized precedence constraints, i.e., where arcs can have negative value representing a maximum time lag between two time points. The research that is mainly done in STNU is to assert dynamic controllability, which is roughly the ability of defining a schedule for any uncertain scenario. The criticality analysis presented in this chapter could be of interest to provide additional information on STNUs. A step towards such a generalization has recently been made by Yakhchali and Ghodsipour (2010).

References

Chanas S, Dubois D, Zieliński P (2001) Criticality analysis in activity networks under incomplete information. In: Proceedings of the 2nd international conference of the European society for fuzzy logic and technology, pp 233–236

Chanas S, Dubois D, Zieliński P (2002) On the sure criticality of tasks in activity networks with imprecise durations. IEEE Trans Syst Man Cybern B 32(4):393–407

Chanas S, Zieliński P (2002) The computational complexity of the criticality problems in a network with interval activity times. Eur J Oper Res 136(3):541–550

Chanas S, Zieliński P (2003) On the hardness of evaluating criticality of activities in a planar network with duration intervals. Oper Res Lett 31(1):53–59

Dubois D, Fargier H, Fortin J (2005) Computational methods for determining the latest starting times and floats of tasks in interval-valued activity networks. J Intell Manuf 16(4–5):407–421

Dubois D, Fargier H, Galvagnon V (2003) On latest starting times and floats in activity networks with ill-known durations. Eur J Oper Res 147:266–280

Fargier H, Galvagnon V, Dubois D (2000) Fuzzy PERT in series-parallel graphs. In: Proceedings of the 9th IEEE international conference on fuzzy systems, San Antonio, TX, pp 717–722

Fortin J, Zieliński P, Dubois D, Fargier H (2010) Criticality analysis of activity networks under interval uncertainty. J Sched 13(6):609–627

Garaix T, Artigues C, Briand C (2013) Fast minimum float computation in activity networks under interval uncertainty. J Sched 16(1):93–103

Kasperski A, Zieliński P (2010) Minmax regret approach and optimality evaluation in combinatorial optimization problems with interval and fuzzy weights. Eur J Oper Res 200:680–687

Morris P, Muscettola N, Vidal T (2001) Dynamic control of plans with temporal uncertainty. In: Nebel B (ed) Proceedings of the 17th international joint conference on artificial intelligence, Seattle, WA, pp 494–499

Yakhchali S, Ghodsipour S (2010) Computing latest starting times of activities in interval-valued networks with minimal time lags. Eur J Oper Res 200(3):874–880

Zieliński P (2003) Latest starting times and floats of activities in network with uncertain durations. In: Proceedings of the 3rd international conference of the European society for fuzzy logic and technology, pp 586–591

Zieliński P (2005) On computing the latest starting times and floats of activities in a network with imprecise durations. Fuzzy Set Syst 150(1):53–76

Zieliński P (2006) Efficient computation of project characteristics in a series-parallel activity network with interval durations. In: Della Riccia G, Dubois D, Kruse R, Lenz HJ (eds) Decision theory and multi-agent planning. International centre for mechanical sciences: courses and lectures, vol 482. Springer, New York, pp 111–130

Chapter 42
The Fuzzy Time-Cost Tradeoff Problem

Hua Ke and Weimin Ma

Abstract The time-cost tradeoff problem is a specific type of the project scheduling problem, which studies how to modify project activities so as to achieve the tradeoff between the completion time and the project cost. In real projects, the tradeoff between the project cost and the completion time, and the uncertainty of the environment are both considerable aspects for managers. In this chapter, three new fuzzy time-cost tradeoff models are proposed, in which credibility theory is applied to describe the uncertainty of activity durations. A searching method by integrating fuzzy simulation and genetic algorithm is developed to search quasi-optimal schedules under some decision-making criteria. The purpose of this work is to reveal how to obtain the optimal balance of the completion time and the project cost in fuzzy environments.

Keywords Credibility theory • Fuzzy sets • Project scheduling • Time-cost tradeoff

42.1 Introduction

The time-cost tradeoff problem studies how to modify project activities so as to achieve the tradeoff between the completion time and the project cost, which is a specific type of the project scheduling problem. Kelley (1961) first studied this type of the project scheduling problem, which also initiated the research on project scheduling problems. In the following years, the research on the time-cost tradeoff problem mainly focused on the deterministic cases (Phillips and Dessouky 1977; Siemens 1971). For solving the deterministic time-cost tradeoff problem, the common analytical methods are linear programming and dynamic programming (Butcher 1967; Talbot 1982). Besides, some heuristic algorithms, such as genetic algorithm (Azaron et al. 2005; Chua et al. 1997; Feng et al. 1997), have also been introduced.

H. Ke (✉) • W. Ma
School of Economics and Management, Tongji University, Shanghai, China
e-mail: hke@tongji.edu.cn; mawm@tongji.edu.cn

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_12

Though most research work on the time-cost tradeoff problem assumes that the problem is always in some deterministic environment, the real world is full of nondeterministic factors. For instance, the project completion time may vary due to many external influence factors, such as the change of weather, the increase of productivity level, the use of additional manpower, etc. Hence, many recent studies introduced uncertain factors. Furthermore, Goldratt (1997) questioned the validity of deterministic environments in the project scheduling problem. The readers may refer to Charnes and Cooper (1962), Freeman (1960), Golenko-Ginzburg and Gonik (1997), and Ke and Liu (2005) to see the progress in stochastic project scheduling. In recent years, the stochastic time-cost tradeoff problem has also attracted many researchers' interest. Wollmer (1985) discussed a stochastic linear time-cost tradeoff problem, in which some discrete random variables were introduced. Gutjahr et al. (2000) designed a modified stochastic branch-and-bound approach and applied it into a specific stochastic discrete time-cost tradeoff problem. Laslo (2003) described a stochastic critical-path-method time-cost tradeoff model, including four fundamental formulations of the model and several new ideas for formulating the relationships between time-cost tradeoffs. Zheng and Ng (2005) presented a new approach for a time-cost optimization model by integrating fuzzy set theory and the nonreplaceable front concept with genetic algorithms, where fuzzy set theory was introduced to model the managers' prediction on activity durations and costs as well as the associated risk levels. Zahraie and Tavakolan (2009) embedded two concepts of time-cost tradeoff and resource leveling and allocation in a stochastic multiobjective optimization model, where fuzzy set theory was applied to represent different options for each activity. Ke et al. (2009) presented three stochastic time-cost tradeoff models to meet different practical optimization requirements.

Probability theory can be regarded as a tool for the description of objective uncertainty, while credibility theory, a new theory dealing with fuzziness, is a powerful instrument for treating with subjective uncertainty. In fact, the activities of some projects may have been processed many times before, and with historical data, the uncertainty of the activity durations can be described by probability distributions. While the activities of some other projects may be short of statistical data, the durations can be better described by fuzzy variables. Zadeh (1965) originally introduced the concept of fuzzy set to describe fuzzy phenomena via membership function. Zadeh (1978) proposed the concept of possibility measure for measuring a fuzzy event. Liu and Liu (2002) presented a self-dual credibility measure for measuring a fuzzy event, as possibility measure has no self-duality property, which is a very important property for most applications. Liu (2004) provided axiomatic foundation for credibility theory.

With the development of the research on fuzziness, fuzzy set theory was also applied to project scheduling problems, originally by Prade (1979). Furthermore, many other authors, such as Chanas and Kamburowski (1981), Kaufmann and Gupta (1988), and Ke and Liu (2010), discussed the fuzzy project scheduling problem.

In recent years, the study focuses on the resource-constrained project scheduling under fuzzy environments, which was initiated in Hapke et al. (1994) and Hapke and Slowinski (1993, 1996). Wang (1999, 2002) developed a fuzzy beam search approach for solving product development project scheduling. Hapke and Slowinski (2000) applied simulated annealing to the resource-constrained project scheduling problem for solving some multi-objective cases. Özdamar and Alanya (2001) established a nonlinear mixed-binary mathematical model for software development projects with fuzzy activity duration times, in which four priority-based heuristics were used on some case study. Long and Ohsato (2008) performed a fuzzy critical chain method for fuzzy resource-constrained project scheduling problem.

To the knowledge of the authors, the first work on the fuzzy time-cost tradeoff problem was done by Leu et al. (2001). In Leu et al. (2001), the activity durations were characterized by fuzzy numbers due to environmental variation, and the fuzzy relationship between the activity duration and the activity cost was taken into account by membership function. Furthermore, the philosophy of chance-constrained programming, which was initiated by Charnes and Cooper (1959), was introduced as a decision-making approach. Jin et al. (2005) gave a GA-based fully fuzzy optimal time-cost tradeoff model, in which all parameters and variables were characterized by fuzzy numbers and an example in ship building scheduling was demonstrated. Eshtehardian et al. (2008) established a multi-objective fuzzy time-cost model, in which fuzzy logic theory was introduced to represent accepted risk levels. Ghazanfari et al. (2008) and Ghazanfari et al. (2009) applied possibilistic goal programming to the time-cost tradeoff problem to determine the optimal duration for each activity in the form of triangular fuzzy numbers. However, as we mentioned above, possibility measure does not have self-duality property, which is an important property in many applications. Especially, self-duality property is necessary for well defining the concept of expected value of stochastic event or fuzzy event, which is the most widely used decision-making criterion in optimization problems.

In this chapter, with the credibility theory of Liu (2004), some decision-making criteria will be proposed, and some fuzzy time-cost tradeoff models will be established, which is the main contribution of this study. In addition, a hybrid intelligent algorithm integrating fuzzy simulation and genetic algorithm (GA) will be designed to deal with the proposed fuzzy time-cost tradeoff models.

This chapter is organized as follows: In Sect. 42.2, the fuzzy time-cost tradeoff problem is described, in which some assumptions and some parameters are given to deduce the project completion time and the project cost. In Sect. 42.3, some important concepts of credibility theory are introduced and based on these concepts, three fuzzy models are proposed. Section 42.4 introduces a hybrid intelligent algorithm integrating fuzzy simulation and GA. Then we give some numerical examples to illustrate the effectiveness of the proposed algorithm in Sect. 42.5. Section 42.6 draws some conclusions.

42.2 Problem Description

For the change of the environment influencing the project, the activity durations might vary, and meanwhile the corresponding activity costs also change. For example, hiring more workers might accelerate the project execution process and consequently decrease the project duration and simultaneously increase the total project cost. Actually, in most real projects, the managers always need to take account of the tradeoff between the total project cost and the project completion time. It is naturally desirable for managers to find the most effective way to complete a project within some predetermined completion time limit and with the "minimal" cost in some sense, which is just what the time-cost tradeoff problem is about.

Generally, a project can be described by a directed acyclic graph as illustrated in Fig. 42.1. Let $G = (V, E)$ be a directed acyclic graph with the activity-on-node (AoN) network structure representing a project, where $V = \{0, 1, 2, \ldots, n + 1\}$ is the set of nodes representing the activities of the project, and E is the set of arcs corresponding to the precedence relationships among the activities. Note that dummy activities 0 and $n + 1$ represent the beginning and completion of the project.

First we introduce the parameter $\hat{p}_i$ as a fuzzy variable representing the normal duration of activity i, whose uncertainty is attributed to the variation of the external environment, and c_i as the normal cost per time unit of activity i, which is a constant. That is, $\hat{p}_i$ represents the duration of activity i without the influence of the decision made by the manager. The fuzzy normal activity durations are concisely written as $\hat{p} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n)$. The decision variable x_i, which is assumed to be an integer, represents the change of the duration of activity i, which may be due to some decisions of the manager, such as hiring more or less workers, applying better or worse instruments, etc. Owing to some practical conditions, the variable x_i is bounded by some interval $[x_i^{min}, x_i^{max}]$, where x_i^{min} and x_i^{max} are assumed to be integers. Accordingly, for each activity i, there exists another associated cost d_i, which is the additional cost of per unit change of x_i and is also assumed to be a constant. Then our goal is to find the optimal vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ meeting some scheduling requirements.

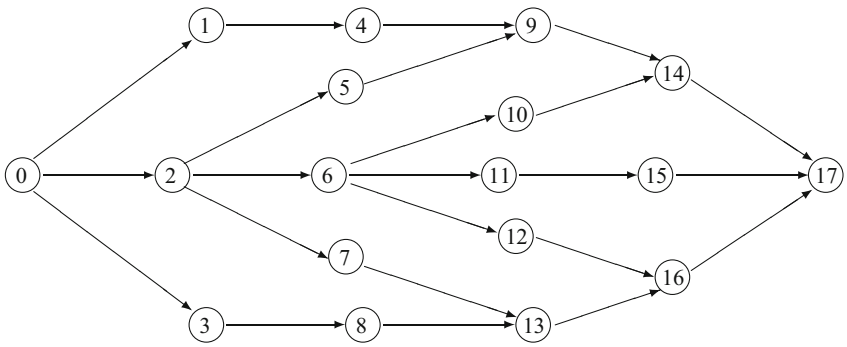


Fig. 42.1 Project network

We denote the starting time of activity i by $S_i(\boldsymbol{x}, \hat{p})$ and the starting time of the project is assumed to be 0. For simplicity, we assume that each activity can be processed only if all the foregoing activities are finished, and it should be processed without interruption. With these assumptions, the starting time of activity j, $j = 1, 2, \ldots, n$, can be determined by

$$S_j(\boldsymbol{x}, \hat{p}) = \max_{(i,j)\in E} \{S_i(\boldsymbol{x}, \hat{p}) + \hat{p}_i + x_i\}$$

and the completion time of the project can be calculated by

$$S_{n+1}(\boldsymbol{x}, \hat{p}) = \max_{(i,n+1)\in E} \{S_i(\boldsymbol{x}, \hat{p}) + \hat{p}_i + x_i\} \tag{42.1}$$

Consequently, the total cost of the project is

$$C(\boldsymbol{x}, \hat{p}) = \sum_{i=1}^{n} (c_i \hat{p}_i - d_i x_i) \tag{42.2}$$

42.3 Fuzzy Models of Time-Cost Tradeoff Problem

42.3.1 Credibility Theory

Credibility theory, founded by Liu (2004), is a branch of mathematics for studying the behavior of fuzzy phenomena. In this subsection, we will introduce some basic concepts, which will be helpful for establishing some fuzzy models for the time-cost tradeoff problem. Let Θ be a nonempty set and $\mathscr{P}(\Theta)$ be the power set of Θ.

Definition 42.1 (Liu and Liu 2002). The set function Cr is called a credibility measure if it satisfies:

(*i*) $\mathrm{Cr}\{\Theta\} = 1$.
(*ii*) $\mathrm{Cr}\{A\} \le \mathrm{Cr}\{B\}$ whenever $A \subset B$.
(*iii*) $\mathrm{Cr}\{A\} + \mathrm{Cr}\{A^c\} = 1$ for any $A \in \mathscr{P}(\Theta)$, where A^c represents the complement of set A.
(*iv*) $\mathrm{Cr}\{\cup_i A_i\} = \sup_i \mathrm{Cr}\{A_i\}$ for any A_i with $\sup_i \mathrm{Cr}\{A_i\} < 0.5$.

Next, we will introduce the concept of a credibility space, which will be used to define a fuzzy variable.

Definition 42.2 (Liu 2004). Let Θ be a nonempty set, $\mathscr{P}(\Theta)$ the power set of Θ, and Cr a credibility measure. Then the triplet $(\Theta, \mathscr{P}(\Theta), \mathrm{Cr})$ is called a credibility space.

Definition 42.3 (Liu 2004). A fuzzy variable is a function from a credibility space $(\Theta, \mathscr{P}(\Theta), \mathrm{Cr})$ to the set of real numbers.

With the concept of fuzzy variable, we can define the membership function of a fuzzy variable.

Definition 42.4 (Liu 2004). Let $\hat{z}$ be a fuzzy variable defined on the credibility space $(\Theta, \mathscr{P}(\Theta), \mathrm{Cr})$. Then its membership function is derived from the credibility measure by

$$\mu(z) = (2\mathrm{Cr}\{\hat{z} = z\}) \wedge 1 \quad (z \in \mathbb{R})$$

where $\wedge$ is the minimum operator, i.e., for $a, b \in \mathbb{R}$, $a \wedge b$ equals to the smaller one of a and b.

Actually, the credibility measure can also be derived from the membership function of a fuzzy variable, which is called the credibility inversion theorem.

Theorem 42.1 (Liu 2006a). *Let $\hat{z}$ be a fuzzy variable with membership function μ. Then for any set B of real numbers, we have*

$$\mathrm{Cr}\{\hat{z} \in B\} = \frac{1}{2}\left(\sup_{z \in B} \mu(z) + 1 - \sup_{z \in B^c} \mu(z)\right)$$

For giving out some decision-making criteria for managers, we will introduce the following definitions:

Definition 42.5 (Liu and Liu 2002). Let $\hat{z}$ be a fuzzy variable. The expected value of $\hat{z}$ is defined by

$$E[\hat{z}] = \int_0^{+\infty} \mathrm{Cr}\{\hat{z} \geq r\}\mathrm{d}r - \int_{-\infty}^0 \mathrm{Cr}\{\hat{z} \leq r\}\mathrm{d}r$$

provided that at least one of the above two integrals is finite.

Definition 42.6 (Liu 2002). Let $\hat{z}$ be a fuzzy variable, and $\alpha \in (0, 1]$. Then

$$\hat{z}_{inf}(\alpha) = \inf\{r | \mathrm{Cr}\{\hat{z} \leq r\} \geq \alpha\}$$

is called the α-pessimistic value of $\hat{z}$.

42.3.2 α-Cost Minimization Model

The philosophy of chance-constrained programming (CCP) initiated by Charnes and Cooper (1959) is a useful decision-making approach, with which managers prefer satisfying some chance constraints with at least some given confidence levels. Liu and Iwamura (1998a,b) have studied several types of fuzzy CCP models. Based on the philosophy of fuzzy CCP, we can present a model as follows:

$$\begin{cases} \text{Min. } \bar{C} \\ \text{s.t.} \quad \text{Cr}\{C(\boldsymbol{x}, \hat{p}) \le \bar{C}\} \ge \alpha \\ \qquad \text{Cr}\{S_{n+1}(\boldsymbol{x}, \hat{p}) \le d_{n+1}\} \ge \beta \\ \qquad x_i \in [x_i^{min}, x_i^{max}] \quad (i = 1, 2, \ldots, n) \\ \qquad x_i \in \mathbb{Z} \quad (i = 1, 2, \ldots, n) \end{cases}$$

where α and β are predetermined confidence levels, d_{n+1} is the due date of the project, x_i^{min} and x_i^{max} are integers given in advance, and $S_{n+1}(\boldsymbol{x}, \hat{p})$ and $C(\boldsymbol{x}, \hat{p})$ are defined by (42.1) and (42.2), respectively. The model is referred to as the α-cost minimization model, where the α-cost is defined by $\min \left\{\bar{C} \;\middle|\; \text{Cr}\{C(\boldsymbol{x}, \hat{p}) \le \bar{C}\} \ge \alpha\right\}$.

42.3.3 Expected Cost Minimization Model

Comparing expected values is the most widely used decision-making criterion in practice. Managers, who are risk-averse, usually want to find the optimal decision with minimum expected project cost subject to some expected project completion time constraint. With this criterion, we can present the following expected cost minimization model:

$$\begin{cases} \text{Min. } E[C(\boldsymbol{x}, \hat{p})] \\ \text{s.t.} \quad E[S_{n+1}(\boldsymbol{x}, \hat{p})] \le d_{n+1} \\ \qquad x_i \in [x_i^{min}, x_i^{max}] \quad (i = 1, 2, \ldots, n) \\ \qquad x_i \in \mathbb{Z} \quad (i = 1, 2, \ldots, n) \end{cases}$$

where d_{n+1} is the due date of the project, x_i^{min} and x_i^{max} are integers given in advance, and $S_{n+1}(\boldsymbol{x}, \hat{p})$ and $C(\boldsymbol{x}, \hat{p})$ are defined by (42.1) and (42.2), respectively.

42.3.4 Credibility Maximization Model

In practice, some project scheduling goals cannot be attained absolutely due to the uncertainty of the external environment. In that case, a realistic approach may be to maximize the chance of achieving the optimization goals, which corresponds to the philosophy of dependent-chance programming by Liu (1997, 1999). Following this approach, we can present the following credibility maximization model:

$$\begin{cases} \text{Max. Cr}\{C(\boldsymbol{x}, \hat{p}) \le b\} \\ \text{s.t.} \quad \text{Cr}\{S_{n+1}(\boldsymbol{x}, \hat{p}) \le d_{n+1}\} \ge \alpha \\ \qquad x_i \in [x_i^{min}, x_i^{max}] \quad (i = 1, 2, \ldots, n) \\ \qquad x_i \in \mathbb{Z} \quad (i = 1, 2, \ldots, n) \end{cases}$$

where α is a predetermined confidence level, d_{n+1} is the due date of the project, b is the budget, x_i^{min} and x_i^{max} are integers given in advance, and $S_{n+1}(\boldsymbol{x}, \hat{p})$ and $C(\boldsymbol{x}, \hat{p})$ are defined by (42.1) and (42.2), respectively.

42.4 Hybrid Intelligent Algorithm

In this section, we describe the design of a hybrid intelligent algorithm integrating fuzzy simulations and genetic algorithm for solving the above three models.

We have three types of fuzzy functions, i.e., $E[C(\boldsymbol{x}, \hat{p})]$, $\mathrm{Cr}\{S_{n+1}(\boldsymbol{x}, \hat{p}) \le d_{n+1}\}$, and $\min\left\{\bar{C} \;\middle|\; \mathrm{Cr}\{C(\boldsymbol{x}, \hat{p}) \le \bar{C}\} \ge \alpha\right\}$, which are all to be estimated by fuzzy simulations. With the relationship between credibility measure and membership function shown in the credibility inversion theorem, the above three fuzzy functions can be formulated or estimated by the form of membership function. The detailed procedure of fuzzy simulations will be explained in this section. The theory and the application of fuzzy simulations can be found in Liu (2002) and Liu (2006b).

The first type of fuzzy functions is $E[C(\boldsymbol{x}, \hat{p})]$. Let μ be the membership function of $\hat{p}$ and u_i the membership functions of $\hat{p}_i$, $i = 1, 2, \ldots, n$, respectively. According to the concept of expected value of a fuzzy variable, the first type of fuzzy simulations can be performed as follows:

Algorithm 42.1: (Fuzzy Simulation for Expected Value)

Step 1. Set $e = 0$.

Step 2. Randomly generate $u_{1h}, u_{2h}, \ldots, u_{nh}$ from the ε-level sets of fuzzy variables $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$, and put $\boldsymbol{u}^h := (u_{1h}, u_{2h}, \ldots, u_{nh})$, $h = 1, 2, \ldots, M$, where ε is a sufficiently small positive number and M is a sufficiently large number.

Step 3. Set $a = C(\boldsymbol{x}, \boldsymbol{u}^1) \wedge C(\boldsymbol{x}, \boldsymbol{u}^2) \wedge \cdots \wedge C(\boldsymbol{x}, \boldsymbol{u}^M)$, $b = C(\boldsymbol{x}, \boldsymbol{u}^1) \vee C(\boldsymbol{x}, \boldsymbol{u}^2) \vee \cdots \vee C(\boldsymbol{x}, \boldsymbol{u}^M)$.

Step 4. Randomly generate r from $[a, b]$, and set $e := e + \mathrm{Cr}\{C(\boldsymbol{x}, \hat{p}) \ge r\}$.

Step 5. Repeat the fourth step for N times, where N is a sufficiently large number.

Step 6. $E[C(\boldsymbol{x}, \hat{p})] = a + e \cdot (b - a)/N$.

The second type of fuzzy functions is credibility measure. According to the definition, the credibility can be obtained approximately by the following formula

$$L = \frac{1}{2}\left(\max_{1 \le k \le N} \left\{\mu(\boldsymbol{u}^k) \;\middle|\; S_{n+1}(\boldsymbol{x}, \boldsymbol{u}^k) \le d_{n+1}\right\} + \min_{1 \le k \le N} \left\{1 - \mu(\boldsymbol{u}^k) \;\middle|\; S_{n+1}(\boldsymbol{x}, \boldsymbol{u}^k) > d_{n+1}\right\} \right)$$

Hence, the fuzzy simulation for credibility measure can be performed as follows:

Algorithm 42.2: (Fuzzy Simulation for Credibility Measure)
Step 1. Let $k = 1$.
Step 2. Randomly generate u_i from the ε-level sets of fuzzy variables $\hat{p}_i$, $i = 1, 2, \ldots, n$, where ε is a sufficiently small positive number.
Step 3. Set $\boldsymbol{u}^k = (u_1, u_2, \ldots, u_n)$ and $\mu(\boldsymbol{u}^k) = \mu_1(u_1) \wedge \mu_2(u_2) \wedge \cdots \wedge \mu_n(u_n)$.
Step 4. $k := k + 1$. Turn back to Step 2 if $k \leq N$, where N is a sufficiently large number, and else, go to Step 5.
Step 5. Return L.

The third type of fuzzy functions is $\min\left\{\bar{C} \mid \mathrm{Cr}\{C(\boldsymbol{x}, \hat{p}) \leq \bar{C}\} \geq \alpha\right\}$. In order to find the minimal $\bar{C}$ such that $\mathrm{Cr}\{C(\boldsymbol{x}, \hat{p}) \leq \bar{C}\} \geq \alpha$, we define

$$L(r) = \frac{1}{2}\left(\max_{1\leq k\leq N}\left\{\mu(\boldsymbol{u}^k) \mid C(\boldsymbol{x}, \boldsymbol{u}^k) \leq r\right\} + \min_{1\leq k\leq N}\left\{1 - \mu(\boldsymbol{u}^k) \mid C(\boldsymbol{x}, \boldsymbol{u}^k) > r\right\}\right)$$

Then the process of fuzzy simulation can be performed as follows:

Algorithm 42.3: (Fuzzy Simulation for α-Cost)
Step 1. Let $k = 1$.
Step 2. Randomly generate u_i from the ε-level sets of fuzzy variables $\hat{p}_i$, $i = 1, 2, \ldots, n$, where ε is a sufficiently small positive number.
Step 3. Set $\boldsymbol{u}^k = (u_1, u_2, \ldots, u_n)$ and $\mu(\boldsymbol{u}^k) = \mu_1(u_1) \wedge \mu_2(u_2) \wedge \cdots \wedge \mu_n(u_n)$.
Step 4. $k := k + 1$. Turn back to Step 2 if $k \leq N$, where N is a sufficiently large number, and else, go to Step 5.
Step 5. Find the minimal r satisfying $L(r) \geq \alpha$.
Step 6. Return r.

Subsequently, we embed the fuzzy simulations, which can simulate the above three types of uncertain functions, into GA to design a hybrid intelligent algorithm for searching quasi-optimal solutions for the fuzzy time-cost tradeoff models.

The procedure of the hybrid intelligent algorithm can be sketched as follows.

Algorithm 42.4: (Hybrid Intelligent Algorithm)
Step 1. Initialize σ_{pop} chromosomes, where the three types of fuzzy functions can be calculated and the feasibility can be checked by the proposed fuzzy simulations.
Step 2. Update the chromosomes by crossover and mutation operations, in which the feasibility of offsprings may also be checked by the proposed fuzzy simulations.
Step 3. Compute the objective values for all chromosomes and accordingly calculate the fitness of each chromosome.
Step 4. Select the chromosomes by spinning the roulette wheel.
Step 5. Run the second to fourth steps for a given number of cycles and report the best chromosome as the quasi-optimal solution.

42.5 Computational Results

Now let us consider a project as shown in Fig. 42.1. The durations, which are assumed as triangular fuzzy variables, the normal costs, and the additional costs of the activities in the project are listed in Table 42.1.

First, let us consider the following 0.90-cost minimization model:

$$\begin{cases} \text{Min. } \bar{C} \\ \text{s.t.} \quad \mathrm{Cr}\{C(\boldsymbol{x}, \hat{p}) \le \bar{C}\} \ge 0.90 \\ \qquad \mathrm{Cr}\{S_{17}(\boldsymbol{x}, \hat{p}) \le 36\} \ge 0.90 \\ \qquad x_i \in [-3, 3] \quad (i = 1, 2, \ldots, 16) \\ \qquad x_i \in \mathbb{Z} \quad (i = 1, 2, \ldots, 16) \end{cases}$$

The parameters of the algorithm, including the population size of one generation σ_{pop}, the probability of mutation ρ_{mut}, and the probability of crossover ρ_{crs}, will be set to different values to compare the different results. It can be seen from Table 42.2 that the "Δ_{best}"s, calculated by the formula: (actual value−best value)/best value×100 %, do not exceed 0.88 %, which does not exceed the general project demand. Note that the "best value" here means the minimal value among the costs in Table 42.2.

The second numerical experiment is about the expected cost minimization model. The manager may want to minimize the expected project cost with the

Table 42.1 Fuzzy durations and costs of activities

Activity i	Normal duration $\hat{p}_i$	Normal cost c_i	Additional cost d_i
1	(7, 9, 12)	170	200
2	(4, 6, 8)	300	280
3	(7, 10, 12)	45	70
4	(4, 6, 9)	270	300
5	(8, 10, 13)	35	50
6	(7, 8, 10)	25	30
7	(6, 8, 11)	150	100
8	(5, 6, 8)	600	400
9	(6, 8, 11)	55	100
10	(7, 10, 12)	200	180
11	(5, 7, 9)	300	400
12	(9, 11, 14)	320	380
13	(7, 10, 13)	45	30
14	(6, 8, 10)	70	50
15	(9, 11, 13)	50	40
16	(5, 7, 9)	90	120

Table 42.2 Computational results for the α-cost minimization model ($\alpha = 0.90$)

σ_{pop}	ρ_{mut}	ρ_{crs}	Quasi-optimal solution	Cost	$\Delta_{best}(\%)$
50	0.4	0.3	$(0, 1, 0, 3, 0, -3, 2, 3, -2, 3, -3, 0, 0, 0, 1, 3)$	17,650	0.28
50	0.3	0.4	$(0, 0, 1, 2, 1, -1, 1, 3, 0, 3, -3, -1, 0, -1, 3, 3)$	17,755	0.88
60	0.3	0.2	$(1, 1, 0, 2, 0, -2, 3, 2, 0, 3, -1, -2, 0, -1, 3, 3)$	17,602	0.01
60	0.3	0.5	$(0, 1, -1, 3, 0, -2, 1, 3, -1, 3, -1, -2, 0, -3, 3, 3)$	17,734	0.76
70	0.2	0.3	$(2, 0, 0, 3, 0, -2, 0, 3, -1, 3, -3, -3, 0, 0, 1, 3)$	17,730	0.74
70	0.3	0.3	$(0, 1, 0, 3, 0, -3, 0, 3, -1, 3, -3, -1, 0, 0, 2, 3)$	17,600	0.00

Table 42.3 Computational results for the expected cost minimization model

σ_{pop}	ρ_{mut}	ρ_{crs}	Quasi-optimal solution	Cost	$\Delta_{best}(\%)$
30	0.3	0.4	$(1, -2, -1, 0, 0, -1, 0, 3, -1, 3, -3, -1, 0, -1, 0, 3)$	18,425	0.10
30	0.5	0.3	$(1, 0, 0, 0, 0, -2, 0, 3, -1, 1, -2, -1, 0, -2, 1, 3)$	18,529	0.67
40	0.2	0.3	$(2, 1, 0, 2, 1, -3, 0, 0, -2, 1, -1, -3, 0, -2, 3, 3)$	18,442	0.20
40	0.4	0.2	$(0, 0, 0, 2, 0, -1, 1, 2, -1, 2, -1, -2, -2, -2, 1, 2)$	18,406	0.00
50	0.2	0.4	$(0, 0, 0, 1, -1, -1, 0, 3, 0, 3, -3, -2, -2, -1, 0, 1)$	18,497	0.49
50	0.4	0.3	$(0, -1, 0, 2, 0, -2, 0, 3, 0, 2, -2, -3, -3, -2, 2, 3)$	18,489	0.45

expected project completion time limit as it is shown in the following expected cost minimization model:

$$\begin{cases} \text{Min. } E[C(\boldsymbol{x}, \hat{p})] \\ \text{s.t.} \quad E[S_{17}(\boldsymbol{x}, \hat{p})] \leq 34 \\ \qquad x_i \in [-3, 3] \quad (i = 1, 2, \ldots, 16) \\ \qquad x_i \in \mathbb{Z} \quad (i = 1, 2, \ldots, 16) \end{cases}$$

The result comparison is shown in Table 42.3. As the maximal error is only 0.67 %, we can say that the proposed hybrid intelligent algorithm performs well on the expected cost minimization model.

The last model is the credibility maximization model. Suppose that the project budget is 17,800 and the project completion time limit is 36. With the philosophy of dependent-chance programming, the credibility maximization model can be established as follows:

$$\begin{cases} \text{Max. Cr}\{C(\boldsymbol{x}, \hat{p}) \leq 17800\} \\ \text{s.t.} \quad \text{Cr}\{S_{17}(\boldsymbol{x}, \hat{p}) \leq 36\} \geq 0.9 \\ \qquad x_i \in [-3, 3] \quad (i = 1, 2, \ldots, 16) \\ \qquad x_i \in \mathbb{Z} \quad (i = 1, 2, \ldots, 16) \end{cases}$$

The results of the credibility maximization model are shown in Table 42.4. The designed hybrid intelligent algorithm is stable for solving the model as all the errors do not exceed 1.18 %.

Table 42.4 Computational results for the credibility maximization model

σ_{pop}	ρ_{mut}	ρ_{crs}	Quasi-optimal solution	Credibility	$\Delta_{best}(\%)$
40	0.2	0.4	$(3, 0, 0, 1, 0, 0, 0, 3, 0, 3, -1, -3, -1, -2, 3, 3)$	0.9201	1.05
40	0.3	0.2	$(2, 0, 0, 2, 0, -2, 1, 3, -1, 3, -1, -3, -1, -2, 3, 3)$	0.9195	1.18
50	0.4	0.2	$(0, 0, 0, 2, 0, -1, 3, 3, 0, 3, -2, -1, 0, -1, 3, 3)$	0.9283	0.17
50	0.2	0.3	$(0, 0, -2, 2, 0, -2, 2, 3, 0, 3, -1, -1, 0, 0, 3, 3)$	0.9204	1.02
60	0.3	0.4	$(0, 2, 0, 3, 0, -3, 0, 3, -2, 3, -1, 0, 0, -2, 1, 3)$	0.9270	0.31
60	0.5	0.2	$(1, 3, 0, 3, 0, -3, 1, 1, -2, 3, -1, -2, -2, -2, 3, 3)$	0.9299	0.00

42.6 Conclusions

The tradeoff between the project cost and the project completion time is an important issue for managers in real projects. In this chapter, we proposed three new fuzzy models: the α-cost minimization model, the expected cost minimization model, and the credibility maximization model of the time-cost tradeoff problem, in which the uncertainty of the activity durations was described by credibility theory. To solve the models, a hybrid intelligent algorithm integrating the fuzzy simulation and genetic algorithm was devised. The main contribution of this study is that we adopted credibility theory to establish a framework for the time-cost tradeoff problem with fuzzy factors, which can be studied more deeply in the future research.

Acknowledgements The work was partly supported by the National Natural Science Foundation of China (71001080, 71371141), and the Fundamental Research Funds for the Central Universities.

References

Azaron A, Perkgoz C, Sakawa M (2005) A genetic algorithm approach for the time-cost trade-off in PERT networks. Appl Math Comput 168:1317–1339

Butcher WS (1967) Dynamic programming for project cost-time curve. J Constr Div 93:59–73

Chanas S, Kamburowski J (1981) The use of fuzzy variables in PERT. Fuzzy Set Syst 5:11–19

Charnes A, Cooper WW (1959) Chance-constrained programming. Manag Sci 6:73–79

Charnes A, Cooper WW (1962) A network interpretation and a direct sub-dual algorithm for critical path scheduling. J Ind Eng 13:213–219

Chua DKH, Chan WT, Govindan K (1997) A time-cost trade-off model with resource consideration using genetic algorithm. Civil Eng Syst 14:291–311

Eshtehardian E, Afshar A, Abbasnia R (2008) Time-cost optimization: using GA and fuzzy sets theory for uncertainties in cost. Constr Manag Econ 26:679–691

Feng CW, Liu L, Burns SA (1997) Using genetic algorithms to solve construction time-cost trade-off problems. J Constr Eng M 11:184–189

Freeman RJ (1960) A generalized network approach to project activity sequencing. IRE Trans Eng Manag 7:103–107

Ghazanfari M, Shahanaghi K, Yousefli A (2008) An application of possibility goal programming to the time-cost trade off problem. J Uncertain Syst 2:22–28

Ghazanfari M, Yousefli A, Ameli MSJ, Bozorgi-Amiri A (2009) A new approach to solve time-cost trade-off problem with fuzzy decision variables. Int J Adv Manuf Tech 42:408–414
Goldratt E (1997) Critical chain. The North River Press, Great Barrington
Golenko-Ginzburg D, Gonik A (1997) Stochastic network project scheduling with non-consumable limited resources. Int J Prod Econ 48:29–37
Gutjahr WJ, Strauss C, Wagner E (2000) A stochastic branch-and-bound approach to activity crashing in project management. INFORMS J Comput 12:125–135
Hapke M, Słowiński R (1993) A DSS for resource-constrained project scheduling under uncertainty. J Decis Syst 2:111–128
Hapke M, Słowiński R (1996) Fuzzy priority heuristics for project scheduling. Fuzzy Set Syst 83:291–299
Hapke M, Słowiński R (2000) Fuzzy set approach to multi-objective and multi-mode project scheduling under uncertainty. In: Hapke M, Słowiński R (eds) Scheduling under fuzziness. Physica, Heidelberg, pp 197–221
Hapke M, Jaszkiewicz A, Słowiński R (1994) Fuzzy project scheduling system for software development. Fuzzy Set Syst 67:101–117
Jin C, Ji Z, Lin Y, Zhao Y, Huang Z (2005) Research on the fully fuzzy time-cost trade-off based on genetic algorithms. J Mar Sci Appl 4:18–23
Kaufmann A, Gupta MM (1988) Fuzzy mathematical models in engineering and management science. North-Holland, Amsterdam
Ke H, Liu B (2005) Project scheduling problem with stochastic activity duration times. Appl Math Comput 168:342–353
Ke H, Liu B (2010) Fuzzy project scheduling problem and its hybrid intelligent algorithm. Appl Math Mod 34:301–308
Ke H, Ma W, Ni Y (2009) Optimization models and a GA-based algorithm for stochastic time-cost trade-off problem. Appl Math Comput 215:308–313
Kelley JE Jr (1961) Critical path planning and scheduling mathematical basis. Oper Res 9:296–320
Laslo Z (2003) Activity time-cost tradeoffs under time and cost chance constraints. Comput Ind Eng 44:365–384
Leu SS, Chen AT, Yang CH (2001) A GA-based fuzzy optimal model for construction time-cost trade-off. Int J Proj Manag 19:47–58
Liu B (1997) Dependent-chance programming: a class of stochastic programming. Comput Math Appl 34:89–104
Liu B (1999) Dependent-chance programming with fuzzy decisions. IEEE T Fuzzy Syst 7:354–360
Liu B (2002) Theory and practice of uncertain programming. Physica, Heidelberg
Liu B (2004) Uncertainty theory: an introduction to its axiomatic foundations. Springer, Berlin
Liu B (2006a) A survey of credibility theory. Fuzzy Optim Decis Mak 5:387–408
Liu YK (2006b) Convergent results about the use of fuzzy simulation in fuzzy optimization problems. IEEE T Fuzzy Syst 14:295–304
Liu B, Iwamura K (1998a) Chance constrained programming with fuzzy parameters. Fuzzy Set Syst 94:227–237
Liu B, Iwamura K (1998b) A note on chance constrained programming with fuzzy coefficients. Fuzzy Set Syst 100:229–233
Liu B, Liu YK (2002) Expected value of fuzzy variable and fuzzy expected value models. IEEE T Fuzzy Syst 10:445–450
Long LD, Ohsato A (2008) Fuzzy critical chain method for project scheduling under resource constraints and uncertainty. Int J Proj Manag 26:688–698
Özdamar L, Alanya E (2001) Uncertainty modelling in software development projects (with case study). Ann Oper Res 102:157–178
Phillips S Jr, Dessouky MI (1977) Solving the project time/cost tradeoff problem using minimal cut concept. Manag Sci 24:393–400
Prade H (1979) Using fuzzy set theory in a scheduling problem: a case study. Fuzzy Set Syst 2:153–165
Siemens N (1971) A simple CPM time-cost trade-off algorithm. Manag Sci 17:354–363

Talbot FB (1982) Resource-constrained project scheduling with time-resource tradeoffs: the non-preemptive case. Manag Sci 28:1197–1210

Wang JR (1999) A fuzzy set approach to activity scheduling for product development. J Oper Res Soc 50:1217–1228

Wang JR (2002) A fuzzy project scheduling approach to minimize schedule risk for product development. Fuzzy Set Syst 127:99–116

Wollmer RD (1985) Critical path planning under uncertainty. Math Program Stud 25:164–171

Zadeh LA (1965) Fuzzy sets. Inform Control 8:338–353

Zadeh LA (1978) Fuzzy sets as a basis for a theory of possibility. Fuzzy Set Syst 1:3–28

Zahraie B, Tavakolan M (2009) Stochastic time-cost-resource utilization optimization using nondominated sorting genetic algorithm and discrete fuzzy sets. J Constr Eng M 135:1162–1171

Zheng DXM, Ng ST (2005) Stochastic time-cost optimization model incorporating fuzzy sets theory and nonreplaceable front. J Constr Eng M 131:176–186

Part XV
General Project Management

Chapter 43
Further Research Opportunities in Project Management

Nicholas G. Hall

Abstract The practice of project management has expanded exponentially over the last 15 years. Currently, one-fifth of the world's economic activity, or U.S. $12 trillion annually, is organized as projects. Meanwhile, the range of business applications that are conducted as projects has also expanded greatly into areas with quite different characteristics. However, these developments have not been matched by a corresponding increase either in research activity, or in the training of academic researchers in project management. This mismatch is creating significant opportunities for academic research in project management to be conducted over the next 10 years. The present work is a successor to a previous article (Hall, J Syst Sci Syst Eng 21(2):129–143, 2012) on recent developments and research opportunities in project management. It updates the information given in the previous article, and identifies an additional research agenda for project management. The 11 new topics presented support a wide range of practical aspects of project management, and require the use of widely varying research methodologies. The conclusions suggest that significant research opportunities remain open within project management.

Keywords Emerging research directions • Opportunities for research • Overview • Project management

43.1 Introduction

There are several reasons for the increasing importance of project management as a business process. Principal among these reasons are the following.

1. Project management controls change, allowing organizations to introduce new products, processes and programs effectively.
2. Projects are becoming more complex, making them more difficult to control without a formal management structure.

N.G. Hall (✉)
Fisher College of Business, The Ohio State University, Columbus, OH, USA
e-mail: hall.33@osu.edu

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_13

3. The life cycles of products and services are becoming shorter, thereby motivating the use of project management to reduce time to market.
4. Projects with substantially different characteristics, especially in IT, are emerging.
5. Companies are using project management to develop and test their future leaders.

The use of project management as a business process goes back a long time. Indeed, the building of the Egyptian pyramids is believed by many to have been assisted by the use of simple project management principles. For much of the history of project management, the predominant application type was engineering and construction projects—for example, roads, bridges and skyscrapers. This was still the case when project management became formalized in the 1960s with the help of new computing power. A particularly impressive project management achievement at that time was the Apollo moon landing project (1961–1969), which required the coordination of about 410,000 workers at a cost of $25 billion in 1961 dollars, or $154 billion in 2011. Another impressive achievement, repeated every 2 years, is the organization of the summer or winter Olympic Games using project management. This is an example of an event project, where the project deadline is fixed and cannot be violated.

There are several features of projects that make them particularly challenging to perform, compared to other business processes. One such feature is their uniqueness (Kerzner 2009), which inhibits learning from previous experience and introduces greater variance into project performance. Another feature is precedence constraints between the tasks (Klastorin 2004), which complicate resource planning and result in the proliferation of delays at the task level. By contrast, early completion of tasks tends to become lost within the project, owing to Parkinson's Law (Wikipedia 2013a). Also, in typical projects, the resource requirements vary widely over time, which stresses the available resources of the organization. A further complication is that the problem of scheduling the tasks of a project, subject to resource constraints, is a highly intractable one. As a result, for most projects of practical size, optimal solution of the project scheduling problem is substantially beyond the capabilities of available software (Demeulemeester and Herroelen 1992). Due to these difficulties, the frequency with which projects are "successful", i.e. on schedule, on budget and meeting scope, remains problematically low, especially in modern applications like IT (The Standish Group 2009).

However, these difficulties have failed to discourage the use of project management as a business process. Currently, one-fifth of the world's economic activity, or U.S. $12 trillion annually, is organized as projects (Project Management Institute 2013). The Project Management Institute operates branches in 180 countries, and has grown from a membership of less than 10,000 in 1980, to about 50,000 by 1996 and about 500,000 members today. Few, if any, other business processes have demonstrated such dramatic, indeed exponential, growth in professional interest. This expansion of interest has surpassed the availability of trained labor, resulting in, for example, 10,000 unfilled jobs in IT project management in Asia during the

summer of 2010. Moreover, academic research has failed to keep pace with the development of new practice-based methodologies for project management.

This is particularly problematic because the characteristics of modern projects are often very different from those of traditional ones. Traditional project management applications in engineering and construction, for example skyscrapers and bridges, are usually highly *deterministic*, meaning that the eventual configuration of the project is known before the start of the execution phase. However, modern applications of project management, such as research and development, software, pharmaceuticals and organizational change management, are *nondeterministic*, i.e. the eventual project configuration needs to be found as part of the execution process. Further, new product and service development is a midrange application, with some characteristics of both deterministic and nondeterministic projects.

Modern applications of project management also suffer from a difficulty in estimating progress. Whereas progress in skyscraper construction is easily observable, in software development it is much less so, and indeed much of the value is only delivered on completion of the software program. This lack of information about project progress makes it difficult to estimate project time and cost variance during the execution phase, which in turn makes decisions about resource reallocations to the project difficult. Another defining difference relates to the shortness of product and service life cycles in industries for which modern projects are performed. Examples include new product development in consumer electronics and technology products. Where competitive time to market is a major performance factor, the balance within the familiar "triple constraint" (Wikipedia 2013b) of project management changes in favor of time, and puts pressure on cost and on scope fulfilment. For these reasons, many project managers and academic researchers believe that traditional project management methodology does not match modern applications well, in that it is too inflexible and too focused on time consuming planning and documentation. A variety of research opportunities emerge from this mismatch, as we discuss below with specific examples.

This work is organized as follows. In Sect. 43.2, we provide an update regarding research progress on the topics discussed in Hall (2012). In Sect. 43.3, we identify many specific research opportunities in project management. These are classified by subsection into three broad categories, based on the type of research issue that is being considered. Section 43.4 provides a conclusion.

43.2 Update on Previous Research Agenda

There are some recently published works that relate to the research topics discussed in Hall (2012). In this section, we provide a brief summary of several of those works. We refer to the research topics by their section number in Hall (2012). The literature discussed here is representative, rather than complete.

43.2.1 Robust Optimization for Project Scheduling (Hall 2012, Sect. 4.1.1)

Wiesemann et al. (2012) consider a resource constrained project management problem where task times are uncertain, but can be influenced by the amount of resources allocated to the task. The decision maker chooses an allocation of resources to all the tasks, so as to minimize the worst case project makespan. The authors describe upper and lower bounds on the optimal objective value that converge monotonically as the computation proceeds, along with a procedure for generating a feasible resource allocation to the tasks that is lower than the upper bound. A computational study for projects with up to 300 tasks demonstrates substantial improvement over decision rule approximation benchmarks. Further, for projects with 200 tasks, the gaps between upper and lower bounds are reasonable. The proposed procedure works best when the resource budget is large and the uncertainty budget is small, in which case the bounds converge faster.

Artigues et al. (2013) consider a resource constrained project scheduling problem, where task times are chosen from a known discrete set of scenarios. The decision maker needs to make sequencing decisions, i.e. to decide for each resource what is the sequence of tasks that it will process, without knowledge of future task time realizations. The objective considered is the minimization of the maximum relative regret. The authors describe an optimal scenario-relaxation algorithm that requires excessive computation time for instances with 20 or more tasks. They therefore describe a heuristic procedure which has reasonable computation time, and show computationally that it provides good quality solutions and outperforms two benchmark approaches.

Goh and Hall (2013) consider projects with activity times from a partially specified distribution within a family of distributions. This family is described by one or more of the following details about the uncertainties: support, mean, and covariance. The objective considered is total completion time penalty plus crashing and overhead costs, using a robust optimization model with a conditional value-at-risk satisficing measure. Decision rules are developed for activity start time and crashing decisions. Computational studies show that, compared to PERT and Monte Carlo approaches, the robust crashing policies provide both a higher level of performance, i.e. higher success rates and lower budget overruns, and substantial robustness to activity time distributions.

43.2.2 Robust Optimization for Project Selection (Hall 2012, Sect. 4.1.2)

Hall et al. (2013) consider a project selection problem where each project has an uncertain return with a partially characterized probability distribution. The decision maker selects a feasible subset of projects to minimize the underperformance risk

of the project portfolio and the uncertain portfolio return. The model captures correlation and interaction effects such as synergies, and is solved using binary search, with solution of the subproblems from Benders decomposition techniques. A computational study shows that project portfolios generated by minimizing the underperformance risk are at least competitive with those found by all the standard benchmark approaches.

Hassanzadeh et al. (2013) develop a multiobjective binary integer programming model for R&D project portfolio selection with competing objectives, when the data in both the objective functions and constraints are uncertain. They apply robust optimization to deal with the uncertainty, and an interactive procedure to evaluate tradeoffs between the different objectives. Robust nondominated solutions are found by solving the linearized version of a robust augmented Tchebycheff program. The final project portfolio chosen by the decision maker is robust in terms of all possible realizations of the uncertain problem coefficients.

43.2.3 Earned Value Analysis (Hall 2012, Sect. 4.1.3)

Kwak and Anbari (2012) note that earned value management (EVM) is now mandated for many U.S. government projects and programs, and provide an extensive historical perspective on implementations of earned value management in government. Current practices in the adoption and implementation of EVM at NASA are studied in detail. The study concludes that implementation of EVM delivers substantial value. Recommendations for broadening and improving the use of EVM within government programs are provided. For example, the authors propose that EVM can be used to encourage innovation in projects, sharpen management estimates of project resources and scope as they change during the execution stage, and advocate for additional rigor in project planning and implementation.

Mortaji et al. (2013) formulate earned value management in a vagueness environment, using fuzzy numbers. This makes earned value management more useful in uncertain conditions, leading to better management decisions. They develop an efficient procedure for calculating estimates at completion in this environment, and illustrate the use of their methodology with a case study.

43.2.4 Policies for Task Notification (Hall 2012, Sect. 4.1.4)

Hou et al. (2013) consider the flow of information between the project manager and the task operators. It is inefficient and costly for a task to become available either earlier or later than the completion time of the last of its predecessor tasks. However, uncertainty in the task times makes it difficult to match that time exactly. The authors study the issue of what notification should be given to the task operator about the

time when their task should be made available, and when should that notification be given as the project execution proceeds and the task time uncertainties are realized. Various models of the problem, which allow for a baseline schedule or for multiple notifications, are studied and solved, either optimally or heuristically. The authors also discuss the implications of their work for agile project management.

43.2.5 Cooperation in Project Management (Hall 2012, Sect. 4.2.1)

Estévez-Fernández (2012) studies how to divide the total reward or penalty in a project that is not executed according to plan. The reward and penalty are arbitrary nondecreasing functions of the earliness or lateness of the project, respectively. She establishes a link between this game and bankruptcy and taxation games, and uses this result to establish nonemptiness of the core. She also includes an interesting discussion of what should be considered as desirable properties for allocations, indicating that it is not the properties of the game as a whole that matter, but rather the properties of the derived solutions.

Cai et al. (2013) consider a project management problem where the project manager outsources expeditable tasks to independent subcontractors. An optimal project schedule requires coordination among the subcontractors, in sharing their resources and completing their tasks. This problem falls within the principal-agent framework. The subcontractors' cooperative game is balanced, and a profit sharing scheme is developed using linear programming. Algorithms are described to compute the optimal contract parameters for profit sharing. The pooling effect of the subcontractors' cooperation mitigates the cost of poor parameter estimation by the project manager. Interesting results include: (1) the subcontractors' profits may decrease if they strategically provide false information, and (2) it is safer for the project manager to overestimate subcontractors' crashing costs than to underestimate them.

43.2.6 Real Options Analysis in Project Evaluation (Hall 2012, Sect. 4.2.2)

Zhu (2012) applies a real options approach to the investment evaluation of nuclear power projects. These projects are difficult to evaluate because of substantial uncertainties, especially with respect to technology, nuclear energy generating cost, radioactive leakage, and energy prices. A real options analysis model that uses Monte Carlo simulation to represent these uncertainties is developed, and applied to two case studies. An interesting conclusion is that, in view of the low electricity price, third-generation nuclear power is currently not a worthwhile investment in China.

Chang (2013) discusses the problem of evaluating renewable energy projects. Three important recent advances are critically reviewed: the systematic use of financial risk management instruments, the integration of real options analysis techniques, and incorporation of the logic of real options analysis into decision analysis and system dynamics frameworks. However, some potential deficiencies of these approaches are also identified, specifically behavioral uncertainty and the danger of contract breakup. The paper addresses these two problems by incorporating the concept of risk-bearing capacity into a net present value framework.

43.2.7 Design of Early Completion Incentives (Hall 2012, Sect. 4.2.3)

Wu et al. (2013) identify an important behavioral factor in project performance, which they term cost salience. This factor causes project team members to view the cost of immediate effort as greater than the cost of future effort. This leads to procrastination during the early stages of project execution, and overwork during the later stages, which in turn result in delays in project delivery and also loss of quality. Traditional analysis of incentives in project management, without consideration of behavioral issues, focuses on the later stages of the project. However, the authors conclude that, as a result of cost salience, incentives should be focused on the early stages of a project, where they are needed more and can also be more effective. A variety of practical issues that affect the design of incentives in project management are discussed in Chap. 47 of this handbook.

In project management, the widely observed behavioral phenomenon known as Parkinson's Law results in the benefit towards project completion time from potential early completion of tasks being wasted. In many projects, this leads to poor project performance. Chen et al. (2013) describe an incentive compatible mechanism to resolve Parkinson's Law for projects planned under the critical path method (CPM). This scheme can be applied to any project where, among the tasks that are allocated to a single task owner, none is a predecessor of another. They also describe an incentive compatible mechanism to resolve Parkinson's Law for projects planned under critical chain project management (CCPM). The incentive payments received by all task owners under CCPM weakly dominate those under CPM. Finally, the authors develop an incentive compatible mechanism for repeated projects, where commitments to early completion continue for subsequent projects.

43.2.8 Learning Between Projects (Hall 2012, Sect. 4.3.1)

Eggers (2012) explores the relationship between organizational experience and product development capability. An empirical study of new products in the U.S. mutual fund industry is conducted. Organizations face initial challenges as they

adapt their processes to meet new market opportunities. Quality increases with the number of similar projects in particular product niches, but decreases as the portfolio of products in the organization's portfolio broadens.

Bartsch et al. (2013) study the difficulties that organizations have in learning across boundaries and in making project-level knowledge available to the whole organization. They conduct an empirical study of a large number of engineering firms in Germany. Their conclusion is that the social ties between project team members and other employees of the organization are highly important in disseminating lessons learned from the project. Indeed, they compensate for a lack of organizational incentives and formal structure for knowledge transfer. Other benefits of such social ties include organizational learning about market conditions and technologies.

43.2.9 Scalability of Agile Project Management Methodologies (Hall 2012, Sect. 4.3.2)

Paasivaara et al. (2012) address one of the key issues in scaling agile project management methodology. Customer expectations are managed by the product owner, who based on communication with the customer prioritizes backlogged items and communicates them to the project team. When an application is sufficiently large to require tens of project teams, several area product owners may be needed in place of a single overall one. The authors conduct 58 interviews to identify the key activities that make this process successful. These include having local product owner representatives, forming a product owner team, frequent communication between the local product owner and his/her team, and clear communication of the backlog to all stakeholders.

43.2.10 Sustainable Project Management (Hall 2012, Sect. 4.3.3)

Maltzman and Shirley (2012) provide a comprehensive overview of many practical aspects of sustainable project management. This book provides an informative starting point for relevant research on this topic. Hwang and Tan (2012) use survey and interview techniques with 31 construction industry experts in Singapore, to identify problems in green construction projects. Foremost among these is cost, which can be addressed by a broadening of government incentives. They further conclude that a higher level of communication is needed among the project team members in environmentally sensitive projects, due to the need for in-depth understanding of green principles. Also recommended is the development of a project management framework for green building construction.

Esenduran et al. (2013) study the environmental regulation of projects. They consider a regulator who attempts to maximize social welfare, and project companies who respond to regulation by controlling their costs. In this bilevel nonlinear program, the upper level regulator specifies waste reduction targets, and the lower level project companies respond using waste stream reduction or remediation. High waste diversion targets lead to outcomes with little pollution, but excessive project costs and only modest waste stream reduction. Projects that have lower task precedence density, or pollutants with different environmental impacts, show larger increases in project cost and time resulting from regulation. The authors design a subsidy that coordinates the system and a bonus that encourages truthful reporting by the project companies.

43.3 Research Opportunities

In this section, we present 11 research topics that will represent important advances in support of the rapidly growing and changing project management environment. These topics are divided into three categories. Section 43.3.1 discusses three topics that inform a choice between fundamentally different project management methodologies. Section 43.3.2 discusses four opportunities for enhancing existing project management planning techniques, either to improve efficiency, or to respond to new practical challenges. Finally, Sect. 43.3.3 discusses four opportunities for using new or more advanced modeling techniques to address various specific issues that arise in project management.

43.3.1 Strategic Choice of Methodology

This section consider tradeoffs and choices between different project management methodologies.

43.3.1.1 Choice Between CPM and CCPM Methodology

Critical chain project management (CCPM) was developed by Goldratt (1997) as a response to concerns about the performance of projects that are planned using traditional methodology based on the critical path method (CPM). These concerns begin with the inclusion of safety slack for individual tasks during project planning. It is well documented that, in many projects, this slack proliferates due to Parkinson's Law (Parkinson 1955, 1958), and project performance is compromised. Hence, projects routinely take longer than expected to perform, and fail to meet due dates that have been previously agreed with clients.

CCPM builds the project plan using median task times, and thus without including safety slack time for individual tasks. In addition, the traditional notion of due dates for tasks, based on late finish times from CPM calculations, is suppressed. Then, the management of the project is based on the monitoring of three types of buffers: a project buffer for the project makespan, a resource buffer for the startup time of both internal and external resources, and a feeding buffer for protection of the critical chain which is the longest sequence of tasks with respect to the given resources. There is substantial anecdotal evidence that the use of CCPM can improve project performance (Patrick 1998; Leach 1999). Nonetheless, CCPM can be criticized for a lack of solid statistical foundation and a lack of clarity about how specific issues should be handled during the execution phase (Raz et al. 2003).

In managing a given project, an important issue is the choice between traditional and CCPM methodology. There are many projects in which Parkinson's Law is not a major issue, but where the due dates for the tasks provide an important source of control. Such projects are better managed using traditional CPM methodology. However, where the use of safety slack is extensive but tasks are routinely not delivered early due to Parkinson's Law, the CCPM methodology is likely to be more effective. For each project, a choice of methodology is needed. Apparently, no standard protocol exists for making this choice. A factor-based approach seems ideal for this purpose. We now discuss how such an approach might be developed.

Relevant factors include:

- The extent to which proliferation of slack and Parkinson's Law are likely to be problematic in a given project.
- The extent to which project team members feel comfortable and motivated working with median time estimates, and without due dates, for the tasks.
- The availability of a reliable formula for calculating project buffer size.
- The development of a reliable procedure for allocating a feeding buffer among different successor tasks.
- The availability of a good algorithm or heuristic for solving the resource leveling problem as part of buffer calculations.
- The development of a reliable procedure for handling changes to the critical chain during the execution phase.
- The availability of reliable information about lead times from subcontractors.

A multiple regression or other statistical analysis (Black 2001), can be developed from these factors to predict the relative success of CCPM methodology compared to that of CPM. This analysis can be used to inform an organization's choice between the CPM and CCPM methodologies for project management, either across all projects or on an individual project basis.

43.3.1.2 Choice Between Traditional and Agile Project Management Methodology

Many traditional applications of project management methodology were largely deterministic, i.e. the final configuration of the end of project deliverables was mostly specified before the start of project execution. Examples of deterministic applications include construction and engineering, where very detailed blueprints are typically developed during project planning. By contrast, many modern applications include uncertainty about the final project configuration, and new information which resolves that uncertainty is revealed as the project execution proceeds. Depending on the application, this information is revealed as a result of testing, clinical trials, regulatory decisions, or technological and business developments. Examples of such nondeterministic applications include research and development, and the development of software and pharmaceuticals. A widely used response to the different characteristics of modern projects is the development of agile project management methodology (agilemanifesto.org 2001). The principles of agile project management include reduced planning and documentation, a focus on customer requirements, and the submission of prototype deliverables in small increments, followed by rapid user feedback and rework. Real world success stories for agile methodology are reported at objectmentor.com (2012).

Paykina and Zhou (2011) compare the performance of traditional plan-driven and agile project management methodologies for software development projects. The purpose of their study is to identify the main organizational and project characteristics that appear to be significant factors in this choice. The methodology used is semi-structured interviews and questionnaires with experienced employees of an IT company. The four major characteristics that are identified are: project complexity (e.g., project size and the number of interdependent parts), communication capability between the customers and the project team, competencies (e.g., knowledge, abilities, skills, motivations, and attitudes) and product requirements (e.g., accuracy and urgency).

Estler et al. (2012) conduct a study of 66 globally distributed software development projects run by 31 companies. They consider the effect of the choice between traditional project management methodologies such as waterfall, and agile project management ones such as Scrum and extreme programming. The project performance criteria studied include (a) overall success, (b) economic savings, (c) the importance customers attribute to projects, (d) the motivation of the project team, and (e) the amount of real-time communication needed during project development. Perhaps surprisingly, the results show no significant differences between the project performance that results from the two methodologies. It should be noted, however, that the range of applications used in this study is rather narrow, and hence the same conclusions may not apply more generally.

In many projects, the final project configuration is mostly deterministic and the planning requirements are not excessive, often due to similar projects having been performed before. Such projects should be managed using traditional methodology (Nerur et al. 2005). In research and development projects, however, the

flexibility provided by agile project management methodology is an essential asset (Boehm 2002). There are many intermediate projects, however, where the choice of methodology is less obvious. Examples arise frequently in new product and service development projects.

Apparently, no standard protocol exists for making this choice. However, anecdotal evidence suggests that some organizations, especially those which manage a variety of projects, have developed their own protocol. A factor-based approach seems ideal for this purpose. We now discuss how such an approach might be developed.

Relevant factors include whether:

- The project deliverables are well specified.
- Rapid completion of the project is a higher priority than cost control.
- There is availability of a cross-trained project team with substantial project management, ideally agile project management project, work experience.
- The project team is enthusiastic about working in a dynamic, interactive, synergistic, agile project management environment.
- The existence of a corporate culture, including senior management, that is supportive of an agile project.
- Colocation of the project team is possible.
- The size of the project team is not more than 15.
- The project can be divided into modules that can be developed separately.
- The project has few dependencies within the organization.
- A system is in place for continuous, incremental, implementation of the project.
- A fast test and response mechanism for evaluating prototypes is available.
- Archiving of lessons learned from the project is not a critical issue.

A few of the above factors require some explanation. First, agile project management tends to deliver projects to completion quickly, but with less formal control of costs than traditional project management. An experienced and cross-trained project team is important because agile project management requires sharing of management responsibilities. Also, most successful agile project management implementations have been for small project teams. It is apparently difficult to sustain the synergy of the agile project management environment beyond this, and hence scalability of agile project management is problematic (see Hall 2012, Sect. 4.3.2). Finally, agile project management emphasizes limited documentation, which may inhibit the systematic archiving of lessons learned from the project. It would be valuable to consider the above factors in a multivariate regression or other statistical analysis (Black 2001), to predict project performance using agile methodology.

43.3.1.3 Whether to Use a Project Management Office

A project management office (PMO) is an organizational structure used by companies that run many projects. A PMO performs several functions. First, it makes

recommendations about which of the projects that are available to the company will be undertaken. Second, it makes decisions about resource allocations, including the appointment of a project manager, to projects. Third, it monitors the performance of current projects, relative to benchmarks and to other similar projects, which possibly results in resource reallocations among projects. Fourth, on completion of a project, the PMO archives information about what worked well and what did not, in the format of lessons learned, and acts as a repository for this information for the benefit of similar future projects. See Chap. 44 of this handbook for a discussion of different forms of PMOs.

The advantages that are observed for PMOs include the following (Santosus 2003). First, the group of subject matter experts and experienced project managers who form the PMO become highly skilled, through experience and training, at dealing with project management issues. Second, the PMO encourages and supports the use of the best project management practices and tools. Third, the PMO establishes a formal mechanism to reallocate resources between projects, as needed, which resolves political issues often associated with such decisions. Fourth, the PMO provides and frequently updates detailed assessments of project risk and impact.

However, some concerns about PMOs are also documented (D'Amico 2010). First, the formation and maintenance of a PMO introduces a substantial overhead cost. This cost is only likely to be justifiable if the organization routinely runs a sufficient number of projects. There is often a concern among project managers about the PMO creating more bureaucracy, more paperwork, and more meetings. Since the PMO is typically viewed as an extension of senior management, project managers may protect themselves by hiding information that reflects badly on them. Some poorly functioning PMOs have a tendency to collect information from their projects, but provide little in return. Another problem is that PMOs may attempt to standardize project performance standards, even where doing so is unrealistic due to the unique circumstances of each project. This leads to unfair comparisons and resentment among project managers.

Consequently, the choice of whether or not to use a PMO is not always an easy one. Factors in this choice include:

- How many projects the organization typically operates simultaneously.
- How many projects the organization typically completes during the year.
- The availability of experienced subject matter experts and project managers to run an effective PMO.
- A senior management that is supportive of projects.
- The availability of experienced project managers.
- The need to manage external resources, for example through subcontracting, on many projects.
- The turnover rate among the organization's project managers.
- The similarity of the organization's projects.

The last two points require some explanation. An organization with a high turnover rate for its project managers faces more situations where a project manager

has not worked on a similar project before. In such situations, the support and lessons learned provided by a PMO are highly valuable. Also, where an organization's projects are more similar, the value of lessons learned becomes greater.

43.3.2 Enhancement of Traditional Methodology

This section identifies several ways in which traditional project planning techniques can be extended to improve project performance.

43.3.2.1 Effect of Work Breakdown Structure on Project Success

A work breakdown structure (WBS) is a decomposition of the deliverables involved in a project, during the early stages of its planning process (Devi and Reddy 2012). The WBS defines the work to be completed within the project, establishes a hierarchical network showing the decomposition of that work by similarity and ownership, and enables the creation of a detailed schedule and budget for the project. The decomposition is exhaustive and mutually exclusive, i.e. all parts of the project are assigned to exactly one branch. At the lowest level of a WBS is a "work package" that is assigned to a specific organizational unit. A work package typically contains very similar subtasks. Globerson (1994) summarizes the importance of a WBS, as follows, "The correct use of a WBS contributes significantly to the probability of successful project completion", and provides several supporting examples.

However, the connection between WBS design and project success has not been extensively studied. Moreover, some fundamentally different WBS designs can emerge from the same project. Globerson (1994) provides an interesting example involving the opening of five new restaurants on five different campuses at the same university. The WBS can be organized according to (1) geographical location, (2) restaurant function, e.g. process design, (3) subsystem, e.g. the dining room, (4) logistics activity, e.g. operations, or (5) project phase, e.g. restaurant startup. Two issues arise from this variety of choices. The first issue is that, depending on the WBS chosen, the work packages from which the detailed planning and scheduling of the project is performed may be different. Differences in work package definition also produce differences in precedence relations between them, which in turn affects opportunities for concurrent processing (Hoedemaker et al. 1999). The second issue is that, even if the work packages resulting from different WBSs are the same, the level of compatibility between the WBS hierarchy and the organizational structure of the company may vary considerably. Both these issues may affect the time and cost performance of the project. These issues are further complicated by the following tradeoff. Decomposition of work packages creates opportunities for overlapping of tasks; however, this is not always beneficial, since the planning complexity of the project is increased.

There is some related literature. Jung and Woo (2004) propose a flexible work breakdown structure that integrates schedule and cost control. This methodology is illustrated using a case study. Golpayegani and Emamizadeh (2007) use modular neural networks to plan the WBS of projects, and also provide a case study.

It would be valuable to study the specific influence of WBS design on project success in a scientific way. Doing so would require detailed modeling of the linkages between WBS design and project performance. A successful model would provide guidance to a company about the appropriate hierarchy to use in designing the WBS, and also about the appropriate level of decomposition in specifying its work packages. A related topic of interest is how to design a WBS that provides project performance which is robust against changes, for example in the scope or resource availability of the project, that occur during the execution stage.

43.3.2.2 New Methodologies for Data Estimation

Among the most widespread and financially significant modern applications of project management is software development. Data estimation in software development projects is performed differently from that in traditional projects, for two reasons. First, the tasks in software development are less likely to have well documented data, although the task programmer sometimes has a reasonably accurate estimate based on related experience. Second, planning in software development projects is typically lightweight in style, such as agile project management methodology, and so less time and other resources are invested in data estimation. For these two reasons, specialized data estimation techniques have been developed for such projects.

Such techniques fall into two categories. First, using individual task estimation, task data is estimated individually by the task operator. This procedure is subject to a natural bias for the task operator to inflate estimates. Second, an alternative that tends to reduce this bias is the use of group consensus methods, where a procedure is developed for combining individual expert estimates. Moløkken-Østvold et al. (2008) identify six group consensus methods in popular use: Delphi, wideband Delphi, unstructured groups, statistical groups, decision markets and planning poker. The last of these is of particular interest here. Planning poker is a lightweight planning technique in which each participant writes an estimate of task cost on a card, without discussion. After all the estimates are revealed simultaneously, a discussion moves the process towards consensus. An advantage of planning poker, relative to other group consensus techniques, is that everyone is required to justify their estimate, therefore everyone participates. Based on an empirical study, the authors conclude that planning poker reduces a bias towards optimism that occurs in statistical combination of individual estimates.

However, the features of the planning poker methodology do not restrict its applicability uniquely to agile projects. Therefore, of great interest is the applicability of planning poker beyond the software application domain. Some specific questions related to this issue include the following. How would planning poker

work in different size, especially larger, project teams? How can planning poker be combined along with project re-estimation during the execution phase? Also, how does planning poker influence the motivation level of the project team and the quality of its work? Answering these questions could reveal the potential of this recent planning technique from software development projects to influence the practice of data estimation across a broader range of project management applications.

43.3.2.3 Estimation of PERT Adjustment Factors

The project planning methodology known as Program Evaluation and Review Technique (PERT) was developed by the consulting firm Booz Allen Hamilton in 1958, in connection with the Polaris military defense program (Fazar 1959). The purpose of PERT is to estimate a probability distribution for the occurrence time of various milestones within the project, including the overall project makespan. Knowledge of these probability distributions enables the project company to quote milestone completion times to the project client, based on its minimum probability of on-time delivery, or "service level".

However, PERT is based on three statistical assumptions that are, in practice, hard to justify for many projects. The first assumption is that the uncertain task times of the tasks are probabilistically independent. This assumption is difficult to justify when the same resources are used for different tasks. The second assumption is that, based on the Central Limit Theorem (Black 2001), the sum of several uncertain task times in series is closely approximated by the normal distribution. This assumption is difficult to justify when the number of tasks in series is less than 30. Moreover, depending on the configuration of the project network, this may require a much larger number of tasks in the project. While both these assumptions are questionable, they introduce inaccuracy but not systematic bias into PERT estimates.

However, the third assumption does introduce bias. This assumption is that the path within the network that is critical, i.e. longest, in expectation remains critical when all the task times are realized. The reason for making this assumption is that it greatly simplifies the calculations required to estimate the probability distribution of the project makespan, since only the longest path in expectation needs to be considered. However, by ignoring the possibility that other paths may become longer than, or *overtake*, the path that is critical in expectation, many possible scenarios that could increase the project makespan are removed from consideration. As a consequence, PERT estimates are systematically too optimistic. Thus, the estimate of the expected project makespan is systematically too low, whereas the probability that the project completes by a given time is systematically too high. Examples are provided by Schonberger (1981).

Although the Project Management Body of Knowledge (Project Management Institute 2008) now recommends Monte Carlo simulation as an alternative to PERT, many companies continue to use PERT (White and Fortune 2002). In many cases, they do so with the help of an ad hoc adjustment procedure that attempts to

compensate for the bias described above in the estimate of the expected project makespan. Anecdotal evidence suggests that typical adjustment factors range from 10 to 25 %. However, these adjustment factors lack any scientific basis. Hence, they often lack robustness in performance. This is especially problematic for organizations that manage projects with widely varying sizes or characteristics.

What is needed here is a statistically robust procedure for calculating PERT adjustment factors, based on reliable estimation of the amount of overtaking that is likely, using easily observable characteristics of the project. The main factors that affect the amount of overtaking, and hence the amount of PERT adjustment necessary, are (a) the number of paths lengths that are near critical in expectation, (b) how close to critical those path lengths are, (c) the variance of those path lengths, and (d) the correlation between those path lengths, for example due to having tasks in common. It would be valuable to incorporate these factors into a robust procedure for estimating adjustments to the expected makespan found by PERT. Even more challenging, but potentially even more valuable, would be developing a methodology to find robust estimates of the probability of completing the project by any given time. This information could be used to implement service level preferences in quoting project durations to potential clients.

43.3.2.4 Buffer Sizing in CCPM

The influential work of Goldratt (1997) has popularized the project planning approach of critical chain project management (CCPM). The critical chain is the set of tasks, considering both precedence and resource availability, that determines the overall project duration (Patrick 1998). The critical chain is protected by three types of buffers. A project buffer at the end of the critical chain protects the project delivery date. Resource buffers serve as warning devices to ensure that resources are ready when needed. Feeding buffers prevent delays on non-critical chains from affecting the performance of the critical chain. The size of these buffers is important. Buffers that are too large add costs and make bids less competitive. Whereas, buffers that are too small result in financial penalties for late project delivery, or expensive crashing or outsourcing measures to avoid those penalties (Tenera 2008). Problematically, however, the CCPM literature provides little scientifically validated guidance about how buffer sizes should be determined (Ashtiani et al. 2007).

Newbold (1998) proposes two simple buffer sizing methods. First, the cut and paste method aggregates the savings achieved by allocating to the tasks only 50 % of their possibly inflated safe estimates. Second, the root square error method computes the difference between the safe time estimate and the average time estimate of each task, and then finds the square root of the sum of squared differences along the critical chain. Both these methods are widely used, although computational studies indicate a lack of robustness in their performance. Herroelen and Leus (2001) conduct a computational study, from which they report that the cut and paste method substantially overstates buffer sizes, whereas the root square error method performs better, especially for larger projects. Tukel et al. (2006) introduce

two additional variables that intuitively should influence the appropriate buffer size, project complexity and resource tightness. These variables are incorporated into two more complex buffer size calculation algorithms. The results of a computational study suggest that the new algorithms outperform the two previous ones.

Because of the widespread use of CCPM, further refinement of buffer sizing methods would be highly valuable. This refinement could include incorporating other relevant features of projects into more complex, but potentially more accurate, buffer sizing algorithms. Also, as mentioned by Tukel et al. (2006), cost asymmetries are an issue that is relevant; for example, the costs of early completion are typically much less than those of late completion. Tenera (2008) incorporates Monte Carlo simulation into a buffer sizing algorithm, and this approach can be developed further. A reason for optimism about the potential of simulation approaches is that they consider the effect of the characteristics of different parts of the project, instead of relying on overall average measures. This intuition can also be useful for the development of other algorithms that do not use simulation. Finally, the adjustment of buffer sizes based on real time information during the execution stage is also worth studying.

43.3.3 Modeling Extensions

This section explores possibilities for the power of modeling to improve detailed decision making in project management.

43.3.3.1 Combining Resources for Performing Tasks

Consider a generic project with several tasks. Further, consider two resources that can be applied to the tasks. Each task in the project can be performed by either of the resources, individually, with a known task time distribution. Alternatively, the same task can be performed by the use of both resources together, with a different task time distribution, and on average more quickly. This example illustrates the problem of combining, or "teaming", resources. The fundamental question, which can be applied to every task, is when is it efficient for the project to combine resources in completing a task, and when not (Klastorin 2004). This problem can be viewed as equivalent to the general resource-constrained project scheduling problem, and therefore it is highly intractable both in theory and in practice (Brucker et al. 1999). However, local improvements may be achieved by developing intuition about the problem.

A factor in favor of combining resources is that there may be a synergistic effect on productivity. For example, if two identical resources are combined, the task time may be reduced by more than 50 %. This is particularly evident in project management applications where detailed checking of work is important. A common example is the pairing of programmers in software development (see Sect. 43.3.3.2).

In many other applications, however, the reduction in task time from the use of two identical resources is likely to be less than 50 %, especially where the task cannot be divided equally between the resources, or the subtasks cannot be accelerated by applying additional resources. An example of the latter situation occurs with various physical tasks that require a fixed amount of time, such as concrete setting, paint drying or various chemical processes.

When task times are uncertain, there is an additional factor in favor of combining resources. Recall that the makespan of a project is defined by the length of its longest chain of tasks with respect to the given resources. When resources are combined, the number of chains is reduced. A smaller number of chains provides fewer opportunities for the longest chain to be very long. An equivalent way of viewing this is that allocating resources separately increases the parallelization of the project.

The tradeoff issues involved in combining resources for certain tasks in a project have received little scientific investigation. It would be of great practical value to develop some robust rules of thumb for this decision. As discussed above, the main factors to consider are the improvement in task time distribution from combining resources, and the effect of reduced parallelization of the project when they are combined. The extent of the latter effect depends on the distribution of the lengths of the chains that are close to critical in the project network. Such rules of thumb could be used to make decisions about combining resources either for individual tasks, or for subsets of tasks, in a project. In view of the intractability of the problem, what is being sought here is only a local, rather than globally, optimal solution. Kolisch and Hartmann (2006) provide an extensive computational comparison of the performance of several well-known heuristics for the general project scheduling problem. It is possible that some ideas from these heuristics can be used to develop good rules of thumb for the problem of combining resources.

43.3.3.2 Models of Paired Programming

A common practice in software development projects is paired programming (Williams and Kessler 2003). Using paired programming, two programmers work side by side at a single computer, keyboard and mouse, to produce a software product. The programmers assume the roles of a driver who writes code, and a navigator who looks for errors and tries to identify opportunities for improvement.

An important issue is how should pairs of programmers be formed. Characteristics such as skill level, motivation, and personality type should naturally be included in pairing choices. Pöyhönen (2001) finds that gender and age issues may also be important. Kangas (2004) conducts a literature survey, and provides a "suggestive summary" of the literature. Example suggestions include two extroverts being a great pairing but one that requires close supervision, introvert/extrovert pairs requiring balanced communication, and the need to avoid pairing of programmers with excessive egos.

An opportunity that has not so far been explored is the use of formal modeling techniques for pairing programmers. This problem can be addressed using matching techniques (Ahuja et al. 1993), which can solve the pairing problem for many pairs of programmers. The advantage of this modeling technique is that it efficiently compares all the numerous combinations of pairings, and selects one that is, subject to the accuracy of the available data, the best.

A more complex, but also potentially more powerful, modeling opportunity arises from simultaneous consideration of not only the pair of programmers, but also the software development task that is assigned to them. This problem is equivalent to the highly intractable optimization problem Three-Dimensional Matching (Garey and Johnson 1979), but it should still be solvable optimally for small instances, or heuristically for larger ones.

43.3.3.3 Managing a Secret Project

Pinker et al. (2013) discuss the management of a secret project. This problem involves project scheduling in a competitive environment with an adversary, who might represent a competitor company. As the tasks of the project are completed, the adversary obtains a clearer understanding of the project, until at some point the project is fully "exposed". At that point, the adversary initiates a competitive reaction, such as the development of a competing product. The objective is to minimize the difference between the project completion time and the time of exposure. The project manager uses a combination of task scheduling and crashing to achieve this. The authors establish intractability of a general form of their problem, and provide efficient solution procedures for special cases.

The secret project problem is closely related to the problem of project interdiction. Brown et al. (2005) model the interaction between a project manager and an adversary as a Stackelberg game. In this game, the leader wants to delay a project managed by the follower, to the greatest extent possible. The authors provide algorithmic and intractability results. Brown et al. (2009) model the project manager's problem as minimizing project duration, given some nonrenewable resources, crashing of task times by the expenditure of additional resources, and decision nodes that allow different ways of achieving milestones. The interaction between the project manager and the adversary is modeled as a two-stage Stackelberg game. Algorithms for solving this problem are presented.

The secret project environment is one that suggests numerous interesting research questions. Below, we provide a partial list of such questions.

- Various possible objectives can be considered for the problem, for example based not only on the project makespan but also on the completion times of several milestones within the project.
- A useful and practical generalization would be to consider stochastic task times, with known probability distributions, where crashing can be used to modify those distributions favorably.

- In environments where a sequence of related projects will be run, it would be valuable to minimize the amount of information exposed, subject to a given makespan, or in the case of stochastic task times the expected makespan, of the project.
- The information available to the different parties can be modeled in various ways, for example symmetrically or asymmetrically.
- Models where the adversary makes multiple decisions over time, with information and modification of his strategy, should be studied.
- The mechanism by which the adversary learns about the project can be modeled in a variety of practically relevant ways, and can influence the decisions and outcomes available to the project manager.
- The existing models of secret projects can be extended to include multiple adversaries, i.e. multiple potential competitors of the project firm, who act independently or in competition with each other.
- Models of strategic behavior can be used to increase the range of options available to the adversary; for example, the adversary can attempt to mislead the project company as to its intentions.
- The concurrent development of a suite of several secret projects, such as a new product range to be introduced simultaneously, including resource allocation issues between the projects, should be studied.

From the above examples, it appears that work on secret projects is only beginning, and can enrich the methodology and practical applicability of project management over the next several years.

43.3.3.4 Maximization of Risk-Adjusted NPV

One of the most widely used measures for evaluating a project is the net present value (NPV) of its cash flows. Given any schedule, including the timing of all the tasks, the NPV of the project can be calculated. Then, using NPV, a company can reasonably compare the value of investing in a project with the value of alternative investments. However, every practical project involves both cash inflows and outflows, with discounted values that depend on the decisions about the timing of the tasks. Moreover, as a result of precedence relationships between the tasks, the timings of the tasks are connected. This complicates the problem of determining the timing of the tasks that maximizes the NPV.

This problem has apparently been studied only under the assumption of a constant discount rate over time. For example, if c_i^F denotes a cash flow for tasks $i = 1, \ldots, n$, C_i is the time when it occurs, and α is the constant project discount rate, then the present value of a schedule is given by $\sum_{i=1}^{n} c_i^F e^{-\alpha C_i}$. Russell (1970) models this problem as a nonlinear program with linear constraints and a nonconcave objective. However, Grinold (1972) shows that this problem can be modeled as an equivalent linear program that allows an efficient specialized algorithm based on tree networks. He further investigates the tradeoff between

project duration and net present value. Russell (1986) provides a computational comparison of the performance of six heuristic scheduling rules for the more general problem that considers resource constraints. Schwindt and Zimmermann (2001) consider the maximization of project NPV subject to general temporal constraints, and describe a steepest ascent procedure that runs efficiently.

The above works can all be criticized, in that they ignore the fact that the risk component in the discount rate may change over time. Moreover, the change over time is a function of project scheduling decisions. While there is typically a risk free component in a discount rate, the level of risk in the project is also a large component in the discount rates used by most companies. As most projects are executed, the meeting of various challenges embodied in the tasks, and the passing of various reviews and tests, lowers the amount of risk in the project. An example is the development of a new pharmaceutical, a project that becomes less risky with the passage of each stage of clinical trial success or regulatory approval. The monotonically decreasing risk level in the project can be modeled, for example, by reducing the project discount rate, whenever a task is completed, by a prespecified task-dependent amount.

However, modeling the project discount rate in this way introduces additional mathematical complexity into the problem, because the discount rate becomes a function of the decisions about the timing of the tasks. For this reason, the linear program proposed by Grinold (1972) is no longer equivalent to NPV maximization. Indeed, the objective function of maximizing the NPV now contains additional nonlinearities. While this problem is computationally challenging, solving it would represent a major contribution to the project management literature, since the assumption of a constant discount rate in the existing literature is unrealistic for most projects. Even heuristic solution of this problem could provide more realistic results than optimal solution of the problem under the constant discount rate assumption.

43.4 Conclusions

This work provides a detailed list of research opportunities within project management. These research opportunities arise from several sources. The first source is industry's increasing focus on new project management applications with characteristics that differ greatly from those of traditional applications. The second source is the development of new methodologies that have evolved from project management practice, but are not yet well supported by academic research. The third source is the considerable underestimation of the value of project management as a planning methodology over the last 20 years. This has in turn caused research to lag behind recent business innovation and the growing range of applications. Perhaps because of this underestimation, there has been little Ph.D. education in project management during the last 15 years, and hence few new researchers have entered the project management field. Finally, leading academic journals in OR/OM have published few articles on project management in the last 10 years, compared to many other topics

of comparable or less practical importance or potential for interesting academic research.

The open research problems described here and in Hall (2012) require a wide variety of analytical techniques, including network analysis of WBS structures, multiple regression and other statistical analyses, robust and nonlinear optimization, cooperative and noncooperative game theory, competitive and strategic behavior, and stochastic analysis of project networks. This variety itself presents an interesting challenge to researchers.

Over the next 10 years, emerging trends within the practice of project management will identify further important research questions, besides those discussed here. One of the most important is understanding to what extent agile project management methodology can be effective for applications outside the range for which they were originally developed. It is unlikely that agile project management methodology will come to be used routinely for highly deterministic applications such as construction and engineering, since it apparently offers few advantages in such environments. However, there are applications with a mixture of deterministic and nondeterministic characteristics for which agile project management methodology may be useful. One natural application area to consider in this range is new product and service development. The special difficulties that arise for project management in this application area are discussed in Chap. 45 of this handbook. Some interesting practical discussions about the potential for agile project management methodology to be applied in this environment appear in Smith (2007). The research issues that surround this potential will crystallize over time.

In conclusion, we expect that the next 10 years will see fundamental advances in academic research on project management, and that such advances will greatly improve the practice of the challenging business process of project management.

Acknowledgements This research is supported by the Summer Fellowship Program, Fisher College of Business, The Ohio State University. Helpful comments on an earlier draft were provided by Prof. Xiaoqin Wen, Shanghai University, China. A version of this work was presented as part of a keynote address at PMS 2014 Conference, Munich, Germany, March 2014.

References

agilemanifesto.org (2001) Manifesto for Agile software development. Available at http://agilemanifesto.org/. Cited 18 Oct 2013

Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows: theory, algorithms and applications. Prentice Hall, Englewood Cliffs

Artigues C, Leus R, Nobibon FT (2013) Robust optimization for resource-constrained project scheduling with uncertain activity durations. Flex Serv Manuf J 25:175–205

Ashtiani B, Jalali G-R, Aryanezhad M-B, Makui A (2007) A new approach for buffer sizing in critical chain scheduling. In: Proceedings of IEEE international conference on industrial engineering and engineering management, Singapore, pp 1037–1041

Bartsch V, Ebers M, Maurer I (2013) Learning in project-based organizations: the role of project teams' social capital for overcoming barriers to learning. Int J Proj Manag 31:239–251

Black K (2001) Business statistics: contemporary decision making. South-Western Thomson Learning, Independence

Boehm B (2002) Get ready for agile methods, with care. Computer 35:64–69

Brown G, Carlyle WM, Harney R, Skroch E, Wood RK (2009) Interdicting a nuclear-weapons project. Oper Res 57(4):866–877

Brown G, Carlyle WM, Royset J, Wood RK (2005) On the complexity of delaying an adversary's project. In: Golden B, Raghavan S, Wasil E (eds) The next wave in computing optimisation and decision technologies. Springer, New York, pp 3–17

Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999) Resource-constrained project scheduling: notation, classification, models, and methods. Eur J Oper Res 112:3–41

Cai X, Hall NG, Zhang F (2013) Cooperation and contract design in project management with outsourcing. Working paper, Fisher College of Business, The Ohio State University, Columbus, Oper Res

Chang C-Y (2013) A critical analysis of recent advances in the techniques for the evaluation of renewable energy projects. Int J Proj Manag 31:1057–1067

Chen B, Deng X, Hall NG (2013) Incentive schemes to resolve Parkinson's Law in project management. Working paper, Fisher College of Business, The Ohio State University, Columbus

D'Amico V (2010) Project management office can cause problems too. IndUS Bus J. http://www.indusbusinessjournal.com/ME2/Audiences/dirmod.asp?sid=&nm=&type=Publishing&mod=Publications%3A%3AArticle&mid=8F3A7027421841978F18BE895F87F791&tier=4&id=895504D9A6B246BAAA04AF0CEF85B2AB&AudID=EEB7C7075C2E462F969310BCC0CAA619

Demeulemeester E, Herroelen W (1992) A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. Manag Sci 38(12):1803–1818

Devi TR, Reddy VS (2012) Work breakdown structure of the project. Int J Eng Res Appl 2(2):683–686

Eggers JP (2012) All experience is not created equal: learning, adapting, and focusing in product portfolio management. Strateg Manag J 33:315–335

Esenduran G, Hall NG, Liu Z (2013) Environmental regulation in project-based industries. Working paper, Fisher College of Business, The Ohio State University, Columbus

Estévez-Fernández A (2012) A game theoretical approach to sharing penalties and rewards in projects. Eur J Oper Res 216:647–657

Estler H-C, Nordio M, Furia CA, Meyer B, Schneider J (2012) Agile vs. structured distributed software development: a case study. In: IEEE seventh international conference on global software engineering, Porto Alegre, pp 11–20

Fazar W (1959) Program evaluation and review technique. Am Stat 13(2):10

Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, New York

Globerson S (1994) Impact of various work-breakdown structures on project conceptualization. Int J Proj Manag 12(3):165–171

Goh J, Hall NG (2013) Total cost control in project management via satisficing. Manag Sci 59(6):1354–1372

Goldratt EM (1997) The critical chain. North River Press, Great Barrington

Golpayegani SAH, Emamizadeh B (2007) Designing work breakdown structures using modular neural networks. Decis Support Syst 44:202–222

Grinold RC (1972) The payment scheduling problem. Nav Res Logist Q 19(1):123–136

Hall NG (2012) Project management: recent developments and research opportunities. J Syst Sci Syst Eng 21(2):129–143

Hall NG, Long Z, Qi J, Sim M (2013) Managing underperformance risk in project portfolio selection. Working paper, Fisher College of Business, The Ohio State University, Columbus, Oper Res

Hassanzadeh F, Nemati H, Sun M (2013) Robust optimization for interactive multiobjective programming with imprecise information applied to R&D project portfolio selection. Working paper, College of Business, University of Texas at San Antonio, San Antonio

Herroelen W, Leus R (2001) On the merits and pitfalls of critical chain scheduling. J Oper Manag 19:559–577

Hoedemaker GM, Blackburn JD, Van Wassenhove LN (1999) Limits to concurrency. Decis Sci 30(1):1–18

Hou C, Allen TT, Hall NG (2013) Models of information flow in project management. Working paper, Department of Integrated Systems Engineering, The Ohio State University, Columbus

Hwang B-G, Tan JS (2012) Green building project management: obstacles and solutions for sustainable development. Sustain Dev 20:335–349

Jung Y, Woo S (2004) Flexible work breakdown structure for integrated cost and schedule control. J Constr Eng Manag 130(5):616–625

Kangas M (2004) The impact of individual differences on pair programming. In: Seminar in software engineering, Spring

Kerzner H (2009) Project management: a systems approach to planning, scheduling, and controlling. Wiley, Hoboken

Klastorin TD (2004) Project management: tools and trade-offs, 1st edn. Wiley, Hoboken

Kolisch R, Hartmann S (2006) Experimental investigation of heuristics for resource-constrained project scheduling: an update. Eur J Oper Res 174:23–37

Kwak YH, Anbari FT (2012) History, practices, and future earned value management in government: perspectives from NASA. Proj Manag J 43:77–90

Leach LP (1999) Critical chain project management improves project performance. Proj Manag J 30(2):39–51

Maltzman R, Shirley D (2012) Green project management. CRC Press, Boca Raton

Moløkken-Østvold K, Haugen NC, Benestad HC (2008) Using planning poker for combining expert estimates in software projects. J Syst Softw 81:2106–2117

Mortaji STH, Bagherpour M, Noori S (2013) Fuzzy earned value management using L-R fuzzy numbers. J Intell Fuzzy Syst 24:323–332

Nerur S, Mahapatra R, Mangalaraj G (2005) Challenges of migrating to agile methodologies. Commun ACM 48(5):73–78

Newbold RC (1998) Project management in the fast lane: applying the theory of constraints. The St. Lucie Press, Boca Raton

objectmentor.com (2012) Agile/XP Object Mentor success stories. Available at http://www.objectmentor.com/omSolutions/agile_customers.html. Cited 18 Oct 2013

Paasivaara M, Heikkilä VT, Lannesius C (2012) Experiences in scaling the product owner role in large-scale globally distributed Scrum. In: Proceedings of IEEE seventh international conference on global software engineering, Porto Alegre, pp 174–178

Parkinson CN (1955) Parkinson's Law. Economist, November 19

Parkinson CN (1958) Parkinson's Law: the pursuit of progress. John Murray, London

Patrick FS (1998) Critical chain scheduling and buffer management: getting out from between Parkinson's rock and Murphy's hard place. Available at http://www.focusedperformance.com. Cited 18 Oct 2013

Paykina E, Zhou L (2011) What characteristics are suited to help choosing traditional or agile project management methods for software development methods? Master thesis, Umeå School of Business, Umeå

Pinker E, Szmerekovsky J, Tillson V (2013) Managing a secret project. Oper Res 61(1):65–72

Pöyhönen A (2001) Correlates of innovation in knowledge worker teams. Master thesis, Department of Social Psychology, University of Helsinki, Helsinki

Project Management Institute (2008) A guide to the project management body of knowledge. Project Management Institute, Newtown Square

Project Management Institute (2013) Should you be teaching project management? Project Management Institute, Newtown Square

Raz T, Barnes R, Dvir D (2003) A critical look at critical chain project management. Proj Manag J 36:24–32

Russell AH (1970) Cash flows in networks. Manag Sci 16(1):357–373

Russell RA (1986) A comparison of heuristics for scheduling projects with cash flows and resource restrictions. Manag Sci 32(10):1291–1300

Santosus M (2003) Why you need a project management office (PMO). CIO, July 1. Available at http://www.cio.com/article/29887/Why_You_Need_a_Project_Management_Office_PMO. Cited 18 Oct 2013

Schonberger RJ (1981) Why projects are "always" late: a rationale based on manual simulation of a PERT/CPM network. Interfaces 11:66–70

Schwindt C, Zimmermann J (2001) A steepest ascent approach to maximizing the net present value of projects. Math Methods Oper Res 53:435–450

Smith PG (2007) Flexible product development: building agility for changing markets. Wiley, Hoboken

Tenera AB (2008) Critical chain buffer sizing: a comparative study. In: Proceedings of the PMI research conference. Project Management Institute, Newtown Square

The Standish Group (2009) The CHAOS report, April. The Standish Group, Boston

Tukel OI, Rom WO, Eksioglu SD (2006) An investigation of buffer sizing techniques in critical chain scheduling. Eur J Oper Res 17:401–416

White D, Fortune J (2002) Current practice in project management - an empirical study. Int J Proj Manag 20:1–11

Wiesemann W, Kuhn D, Rustem B (2012) Robust resource allocations in temporal networks. Math Prog A 135:437–471

Wikipedia (2013a) Parkinson's Law. Available at http://en.wikipedia.org/wiki/ Parkinson's_Law. Cited 18 Oct 2013

Wikipedia (2013b) Project management triangle. Available at http://en.wiki pedia.org/wiki/ Project_management_triangle. Cited 18 Oct 2013

Williams L, Kessler R (2003) Pair programming illuminated. Addison-Wesley Professional, Boston

Wu Y, Ramachandran K, Krishnan V (2013) Managing cost salience and procrastination in projects: compensation and team composition. Prod Oper Manag. doi:10.1111/poms.12095

Zhu L (2012) A simulation based real options approach for the investment evaluation of nuclear power. Comput Ind Eng 63:585–593

Chapter 44
Project Management in Multi-Project Environments

Peerasit Patanakul

Abstract This chapter discusses project management in multi-project environments, which can be seen from the organizational level and project manager level. From the organizational level, such a practice can be seen in the contexts of multiple project management, project portfolio management, and project management office. From the project manager level, it can be seen in the context of the management of a group of multiple projects. To illustrate such a practice, first, the chapter introduces the concept of multiple project management (MPM) and contrasts it with that of project portfolio management (PPM). Second, the concept of the management of a group of multiple projects (MGMP) is introduced together with key factors contributing to MGMP effectiveness. Those factors include project manager assignment, resource allocation, organizational culture, project management processes, and competencies of multiple-project managers. Third, the chapter discusses the concepts of project management office (PMO) and different forms of PMO representing PMO as a project office, a functional project office, a customer group project office, a project management center of excellence, a coordinating center, and a corporate project office.

Keywords Multi-project environment • Process competency • Project management • Project management office

44.1 Introduction

It is typical in an organization that there are several projects going on at the same time. These projects can be selected as parts of the project portfolio and managed in a multi-project environment or multiple project environment. The term "multiple project management" is used to represent such a management condition. In an organization, these multiple projects can be managed as parts of a project management office (PMO). While some project managers are responsible

P. Patanakul (✉)
Sam and Irene Black School of Business, Pennsylvania State University, Erie, PA, USA
e-mail: pxp25@psu.edu

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_14

for leading one project at a time, many project managers lead more than one project at a time. Those project managers can be referred to as multiple-project managers. This chapter will first discuss the difference between project portfolio management and multiple-project management. Next, the chapter focuses on multiple-project management and discusses the issues related to multiple-project managers, especially the management of a group of multiple projects (MGMP). Furthermore, the chapter presents the concepts of project management office. Conclusions are presented at the end of the chapter.

44.2 Multiple Project Management

The term *multiple project management (MPM)* needs to be clearly defined. Broadly speaking, MPM can be referred to as an organizational-level environment in which multiple projects are managed concurrently. These projects are diverse in size and importance, may be at any point in their life cycle, and may not necessarily be interdependent or directly related (PMI 2008a). In the case that the projects are mutually dependent, share a common goal, and lead to a single deliverable product or service, these projects are part of a program. In an essence, *program management* is a centralized and coordinated approach that is used to manage a group of goal-related projects to achieve the program's strategic objectives and benefits (PMI 2008a). In some cases, projects in an MPM setting can be parts of a project portfolio. Typically, *project portfolio management (PPM)* involves the selection and management of the collection of projects and programs in which a company invests to implement its strategy so as to maximize the contribution of those projects and programs to the overall welfare and success of the enterprise (Levine 2005; Rajegopal et al. 2007). A typical goal of PPM is to ensure that an organization is doing the right work, rather than doing the work right (PMI 2008b). From industry practices, Cooper et al. (2001) suggest that the maximization of portfolio value, portfolio balance, and strategic alignment are the typical goals of PPM for new product development.

Contrasting PPM and MPM, PPM has a strategic focus. MPM, on the other hand, focuses on tactical issues. Typically, MPM emphasizes, e.g., the allocation of resources to multiple projects (Pennypacker and Dye 2002; Patanakul 2013), the assignment of project managers (Patanakul et al. 2007), and the use of project management processes, tools, and techniques. Despite the differences in their intent and focus, PPM and MPM are interrelated. With a constant change in most multiple project environments, a well-defined project selection and prioritization process can give guidance to project and resource managers for properly planning and allocating resources to multiple projects (Pennypacker and Dye 2002). Also, PPM has to be complemented by MPM as information from MPM can help a PPM committee making appropriate portfolio decisions.

In an MPM setting, some of the project managers have only one project under their responsibility. Typically, those projects are sufficiently large and strategic

in nature. Other projects in an MPM environment that are of smaller size and more tactical nature tend to be grouped such that one project manager (called a multiple-project manager) handles several concurrent projects at a time. This approach can be referred to as *management of a group of multiple projects (MGMP).* Typically, projects in the group are not mutually dependent in terms of objectives and goals but rather grouped for the sake of efficiency and better management at a project manager level, leading to interdependencies among these projects as they are managed by the same project manager (Archibald 1975; Ireland 1997). An example is the assignment of a product improvement project, an internal process improvement project, and a small IT upgrading project to one project manager. This form of project management is of the interest of the industries (some researchers and practitioners refer to it as multiple project management).

44.3 Management of a Group of Multiple Projects

At a project manager level, MPM can be perceived as the *management of groups of several concurrent projects.* Each group of projects managed by a project manager typically termed *a multiple-project manager*. These managers are tasked with making decisions lower in an organization hierarchy and have interrelationships with multiple functional units from which they draw resources (Galbraith 1994). Essentially, this form of MPM is designed as an overlay to an existing functional organization (Galbraith 1994) and is of strong interest to many organizations in various industries (Levy and Globerson 1997; Fricke and Shenhar 2000; Pennypacker and Dye 2002).

Multiple-project managers share many characteristics with single-project managers (project managers who manage one project at a time), but also differ in many ways. First, a major piece of multiple-project managers' role, linking multiple concurrent projects, doesn't exist in single-project management. Second, in dealing with multiple projects at a time, a multiple-project manager leads multiple teams for the projects of different objectives, while a single-project manager leads only one team. Third, multiple-project managers face the challenge of switchover from project to project, at times several times a day (Fricke and Shenhar 2000) while the switchover among projects does not exist in a single project.

As for factors impacting the effectiveness of MGMP, Patanakul and Milosevic (2009) suggest that such factors can be categorized in to the organizational factors and the operational level factors. The organizational level factors include project manager assignment, resource allocation, and organizational culture. The operational level factors include project management processes and competencies of multiple-project managers.

44.3.1 Project Manager Assignment

A realistic project manager assignment is one of the most important factors leading to the effectiveness in MGMP (Patanakul and Milosevic 2009; Patanakul 2013). With the realistic assignment, a multiple-project manager would have skills and time availability to effectively lead each project. With sufficient attention from a multiple-project manager, a project tends to be more successful.

What is a realistic project manager assignment? A realistic project manager assignment is the one that management attempts to find a good match between skill sets of multiple-project managers and the project requirements, including an additional consideration on the project priority and some other limitations (Patanakul and Milosevic 2006). Those limitations include, e.g., having an appropriate number of projects per multiple-project manager and having a balanced mix of projects in terms of project types and phases. Kuprenas et al. (2000) proposed that the effectiveness of MGMP depends on the number of projects a multiple-project manager leads at a time. Being assigned too many projects, a project manager would lose a tremendous amount of time catching up with all the issues in the projects instead of focusing on leading projects. There is no universal rule of thumb on how many projects should be assigned to a project manager. The appropriate number of projects being assigned to a project manager varies from company to company and from industries to industries. It also depends on the complexity of projects and their phase (Patanakul and Milosevic 2006). In manufacturing support environments, a study found that assigning two to three "major" projects to an engineering project manager is an effective maximization of his/her productivity (Fricke and Shenhar 2000). In an Information Technology environment where projects tend to be smaller in size, assigning five to six projects to a multiple-project manager seems to lead to the overall MGMP effectiveness (Patanakul 2011).

44.3.2 Resource Allocation

Having resources when needed is another critical factor and is a prerequisite for effective MGMP (Patanakul 2013). However, having sufficient resources is rare for most multiple-project managers. With the nature of MGMP, e.g. the smaller project size and tactical nature of projects, a multiple-project manager always faces a challenge with insufficient resources. In addition, smaller projects are always on the back burner resource. Usually, they have to deal with resource sharing and live with the risk of unsustainable resources. Even with such a systematic resource allocation process, there is no guarantee that multiple-project managers will end up with resources they need.

Resource sufficiency and sustainability are uncommon in many organizations. It is not only the nature of project management in MGMP but it is also the nature of the competitive environments these companies are in that leads to the insufficient

resource allocation. There are a lot of new opportunities for a company to pursue. Without a good resource management system, the company will always be resource-strapped (Adler et al. 1996). This discussion leads to the concept of project portfolio management. By having a balanced portfolio, first, a company would not implement too many projects such that the resource bottleneck is created (Adler et al. 1996). Second, the project priority would be set such that resources can be allocated to the projects accordingly (Seider 2006). The use of project management office can also help balance the resource usage (Crawford 2001; Kerzner 2003). The implementation of Enterprise Resource Planning (ERP) system should help alleviate the resource allocation issue. However, research found that ERP implementation is a complex and expensive process (Muscatello and Parente 2006; Seider 2006). Even though ERP promises to computerize an entire business, its focus is on promoting the integration between all functional areas within a firm's supply chain, especially related to manufacturing. Even though ERP was used in other parts of the organization that implement MGMP, it is not commonly used when it comes to project management, especially for managing project resources. Besides ERP, several studies proposed tools and techniques for scarce resource allocation, which include integer programming, heuristic methods, queuing theory, etc. (Dean et al. 1992; Morse et al. 1996; Levy and Globerson 1997; Hendriks et al. 1999). However, these techniques were proposed for a use in the functional level to allocate the functional resources across multiple projects. Since they were not proposed for the MGMP settings, these techniques may not be applicable to an operational level for a multiple-project manager to allocate resources across projects in his/her group.

44.3.3 Organizational Culture

To be effective in MGMP, an organization should establish the culture that supports such a management form. It is typical to witness in an MGMP setting that (1) the functional managers attempts to solve resource challenges across multiple projects, (2) the multiple-project managers juggles issues among projects, and (3) the project teams work on tasks of multiple projects. In such a setting, commitment, communication, strong working relationship, and reward for performance are needed for the effective MGMP (Patanakul and Aaronson 2012).

In terms of commitment, the project commitment has to come from top management and has to be supported by every level of the organization. At the project level, commitment of the project team is also important. With multiple projects to work on simultaneously, having the culture that supports communication is also significant. With clear communication channels, the project teams can share knowledge and experience across projects. Multiple-project managers can use these channels to communicate project objectives to the teams in order to engage them in project activities. In addition, the organization should have the culture that helps build a strong working relationship and supports reward for performance.

44.3.4 Project Management Processes

Having shared project management processes helps multiple-project managers to be effective in MGMP (Patanakul and Milosevic 2009). The shared processes include, first, the typical project management process for individual project, and second, interproject processes, which include sequence of steps to concurrently lead and complete multiple projects, while delivering results.

To be effective in MGMP, it is necessary that an organization should have standard project management processes and multiple-project managers should have a solid foundation of those processes. For each individual project, the process would lead a way for multiple-project manager to plan, schedule, monitor, and control project activities, allocate resources, manage risks, etc. In addition to a standard process to lead each project individually, to be effective in MGMP, multiple-project managers should be proficient in interproject processes and the management of interdependencies among projects (Patanakul 2013). Research has shown that even though knowing and executing interproject processes is a must because it is the hard core of MGMP, many companies still do not have a formal or a shared interproject process (Patanakul and Milosevic 2009). Research has shown that, to be effective, multiple-project managers use various methods to manage interproject processes e.g. consolidating projects' deliverables or milestones of projects and managing them together (Patanakul and Milosevic 2009). This will help multiple-project managers optimize their own resource capacity and also reduce the magnitude of multitasking. Having interproject processes also help multiple-project managers manage interdependencies and interactions among projects related to shared milestones, resources and technology. In other words, these processes help multiple-project managers manage the impact of one project on the others.

44.3.5 Competencies of Multiple-Project Managers

Literature in project management has always suggested that project manager is a key success factor of a project (Brown and Eisenhardt 1995). Multiple-project managers should possess a combination of competencies that help them lead each individual project and coordinate among projects to be effective (Patanakul and Milosevic 2008).

Multiple-project managers should have *experience* managing multiple-project for the organization for some time. Multiple-project managers should have *administrative competencies* that include planning, scheduling, monitoring and control, and management of cost, resource, and risk. They should also have a solid foundation of project management processes both individual project and interproject processes, discussed earlier. In addition, multiple-project managers should possess the ability in *interdependency management.* They should also have the *business competencies*, which include having business sense, understanding customers, having integra-

tive capability, having strategic thinking, and being profit/cost conscious. Having business competencies helps project managers solve interdependency problems to benefit all projects they lead as much as possible.

The *ability to multitask* is very important for effectiveness in MGMP (Patanakul 2013). Multiple-project managers must be able to estimate their own resource capacity in order to set priorities and switch contexts to multitask among different projects. Multitasking poses a significant challenge when managing more than one project because often, each project has its unique characteristics. During switchover from one task to another, it is not only multiple-project managers have to recognize the difference between tasks, but they also have to realize different project objectives associate with those tasks. As a result, they often lose some time while refocusing. Rubinstein et al. (2001) refer to it as "switchover-time cost".To be considered as being competent in multitasking, multiple-project managers must possess the ability to minimize the switchover-time cost. This includes those who usually are intensely organized, methodical, and focused. Sometimes, it is more effective to trust the project team and delegate some project activities.

Another group of competencies is *internal traits*, which includes being organized and disciplined, being proactive, being mature and self-controlled, being self-motivated, and being flexible. Last but not least, *leadership/simultaneous team management* should be mentioned. Multiple-project managers must be competent in simultaneously leading several project teams, literally at the same time. To do so, multiple-project managers should have knowledge, skills, and experience in interacting with numerous project stakeholders, a.k.a. interpersonal competencies. In a speedy manner and in a limited time, multiple-project managers must be able to (1) putting together a team that is committed and mutually accountable, (2) setting direction, (3) delegating authority, and (4) influencing a project team with fairness. Importantly, they must have ability to select and use different leadership styles specifically for each team. This is especially important in multidisciplinary and distributed teams, a frequent organizational design in a current business environment. In addition, to be effective, multiple-project managers should be a good communicator. They should be capable of listening, asking questions, communicating (verbally and in writing), and articulating and handling the information whether it is technical, legal, administrative, or interpersonal in nature. Problem solving competence is also significant in MGMP.

44.4 Project Management Office

To facilitate efficiency and effectiveness of project management in a multi-project environment, many organizations establish project/program management offices (PMOs). Contingently to the parent organization, PMOs take on different forms (Milosevic et al. 2007). A PMO of one organization may function as a project/program office, serving one specific project/program (Bernstein 2000). In other organizations, the PMO can be seen as a corporate project office,

focusing on corporate and strategic issues related to projects, programs, and portfolio management (Crawford 2001; Kerzner 2003). Many organizations utilize a PMO as a functional project office, managing a critical resource pool (Kerzner 2003). The PMO can also be seen as a center of project excellence, assisting project stakeholders on strategic matters and functional entities throughout the organization in the implementation of project management principles, best practices, methodologies, tools and techniques, including maintaining PM standards and organizing PM training (Block and Frame 1998; Dai and Wells 2004).

Within a company, depending on its needs and what PMO provides, multiple PMOs can be found (Crawford 2001; Crawford and Cabanis-Brewin 2006; Kerzner 2006; Milosevic et al. 2007). An organization can establish a project control office for direct management of a project. At a division level, e.g. IT department, a functional program office can be established to manage critical resource pool of that particular function. At the enterprise level, the establishment of a center of excellence or a corporate PMO can help set up a common platform for project management. Literature has suggested that having an established enterprise PMO is one measure of an organization maturity since project management is recognized as a true function of the organization (Archibald 2003). The following sections discuss various forms of PMO.

44.4.1 PMO as a Project Office

As a project office, also called, project control office (Crawford and Cabanis-Brewin 2006; Milosevic et al. 2007), this PMO is an organizational entity established to manage a specific project, a related series of projects, or a program, usually headed by a project or program manager. To serve the needs of a single, large, complex project, the PMO provides administrative and tracking support for the project teams. Work is focused on maintaining project procedures, schedule maintenance, earned-value tracking, tool usage and support, and project metrics and report generation (Milosevic et al. 2007).

44.4.2 PMO as a Functional Project Office

This type of PMO is utilized in one functional area or division of an organization and is set up to support the project managers within a division (Kerzner 2003). With its more administrative nature, the functional project office may not operate as a true function within the company (Milosevic et al. 2007). Even though this type of PMO still provides support for individual projects such as the maintenance of project schedules, project-data tracking, and development of project indicators and reports (Milosevic et al. 2007), its primary task is managing the integration of

multiple projects of varying sizes within the division (Crawford and Cabanis-Brewin 2006).

44.4.3 PMO as a Customer Group Project Office

Kerzner (2003) suggests that this type of PMO is set up for better customer management and customer communications. For better management, customer focus, and customer relations, projects with common customers are clustered together. As a result, within one organization, multiple customer group PMOs can exist at the same time. Within each PMO, a permanent project manager is assigned to manage projects.

44.4.4 PMO as a Project Management Center of Excellence

The responsibility of this organizational-level PMO is to supply project management support to the project team, provide the organization with project consulting and mentoring, develop and maintain commonality of project management methods, tools, and metrics, identify best practices, provide project management training, develop a corporate resource capability/utilization plan, be a knowledge center etc. (Block and Frame 1998; Crawford 2001; Kerzner 2003; Crawford and Cabanis-Brewin 2006; Milosevic et al. 2007).

44.4.5 PMO as a Coordinating Center

In many companies, multiple PMOs are established to serve the needs of different functions. To achieve better organization and control, a coordinating PMO is also established to network together those multiple PMOs (Kerzner 2006). This type of PMO may be considered a regional PMO in other companies serving as a coordinating center for groups of project managers or team members who perform project management duties within specific regions or industry-specific areas. The primary responsibility of this PMO may include promoting enterprise project management methodology, including the use of standard project management process, tools, and techniques; being a source of project management experts; and coordinating multinational project management knowledge (Kerzner 2006). This type of PMO can be seen as a variation of the center of excellence.

44.4.6 PMO as a Corporate Project Office

Another variation of PMO is a corporate project office. This type of PMO serves the entire company with the major responsibility on project management-related corporate and strategic issues rather than functional issues (Kerzner 2003, 2006). Some authors refer to this type of PMO as a strategic project office or an enterprise project office (Milosevic et al. 2007). Besides taking a role of a project management center of excellence or a project management coordinating center, an important responsibility of a corporate PMO is to ensure the alignment between the company's strategic direction with the projects or programs implemented within the company (Anonymous 2006). Its responsibilities may also include project selection, project portfolio management, and project manager assignments.

44.5 Conclusions

This chapter discusses project management in multi-project environment. First, the chapter presents the concept of multiple-project management (MPM), which can be broadly referred to as an organizational-level environment in which multiple projects are managed concurrently. Those projects can be parts of programs or portfolio. At the project manager level, MPM can be seen as the management of a group of multiple projects (MGMP). The majority of the chapter dedicates to the discussion of the factors impacting the effectiveness of MGMP. Those factors are project manager assignment, resource allocation, organizational culture, project management processes, and competencies of multiple-project managers. The chapter ends with the discussion of project management office (PMO) that many organizations establish to promote efficiency and effectiveness in management of multiple projects.

References

Adler PS, Mandelbaum A, Nguyen V, Schwerer E (1996) Getting the most out of your product development process. Harvard Bus Rev 74(2):134–146

Anonymous (2006) Effective PMOs cross the enterprise. PM Netw 20:12

Archibald RD (1975) Managing high-technology programs and projects. Wiley, New York

Archibald RD (2003) Managing high-technology programs and projects. Wiley, Hoboken

Bernstein S (2000) Project offices in practice. Project Manag J 31(4):4–6

Block TR, Frame JD (1998) The project office. Crisp Publications, Menlo Park

Brown SL, Eisenhardt KM (1995) Product development: past research, present findings, and future directions. Acad Manag J 20:343–378

Cooper RG, Edgett SJ, Kleinschmidt EJ (2001) Portfolio management for new products. Basic Books, New York

Crawford JK (2001) The strategic project office: a guide to improving organizational performance. Marcel Dekker, New York

Crawford JK, Cabanis-Brewin J (2006) Optimizing human capital with a strategic project office. Auerbach Publications, Boca Raton

Dai CX, Wells WG (2004) An exploration of project management office features and their relationship to project performance. Int J Proj Manag 22:523–532

Dean BV, Denzler DR, Watkins JJ (1992) Multiproject staff scheduling with variable resource constraints. IEEE Trans Eng Manag 39(1):59–72

Fricke SE, Shenhar AJ (2000) Managing multiple engineering projects in a manufacturing support environment. IEEE Trans Eng Manag 47(2):258–268

Galbraith JR (1994) Competing with flexible lateral organizations. Addison-Wesley, Reading

Hendriks M, Voeten B, Kroep L (1999) Human resource allocation in a multi-project R&D environment: resource capacity allocation and project portfolio planning in practice. Int J Proj Manag 17(3):181–188

Ireland LR (1997) Managing multiple project in the twenty-first century. The annual project management institute seminars & symposium. Project Management Institute, Upper Darby

Kerzner H (2003) Strategic planning for a project office. Proj Manag J 34(2):13–25

Kerzner H (2006) Project management. Wiley, Hoboken

Kuprenas JA, Jung CL, Fakhouri AS, Jreij WG (2000) Project manager workload-assessment of values and influences. Proj Manag J 31(4):44–51

Levine HA (2005) Project portfolio management. Jossey-Bass, San Francisco

Levy N, Globerson S (1997) Improving multiproject management by using a queuing theory approach. Proj Manag J 28(4):40–46

Milosevic DZ, Martinelli RJ, Waddell JM (2007) Program management for improved business results. Wiley, Hoboken

Morse LC, McIntosh JO, Whitehouse GE (1996) Using combinations of heuristics to schedule activities of constrained multiple resource projects. Proj Manag J 27(1):34–40

Muscatello JR and Parente DH (2006) Enterprise resource planning (ERP): a postimplementation cross-case analysis. Inform Resour Manag J 19(3):61–80

Patanakul P (2011) Project manager assignment and its impact on multiple project management effectiveness: an empirical study of an IT organization. Eng Manag J 23(4):14–23

Patanakul P (2013) Key drivers of effectiveness in managing a group of multiple projects. IEEE T Eng Manag 60(1):4–17

Patanakul P, Aaronson ZH (2012) Managing a group of multiple projects: do culture and leader's competencies matter? J Knowl Econ 3:217–232

Patanakul P, Milosevic DZ (2006) Assigning new product projects to multiple-project managers: what market leaders do. J High Tech Manag Res 17(1):53–69

Patanakul P, Milosevic DZ (2008) A competency model for effectiveness in managing multiple projects. J High Tech Manag Res 18(2):118–131

Patanakul P, Milosevic DZ (2009) The effectiveness in managing a group of multiple projects: factors of influence and measurement criteria. Int J Proj Manag 27(3):216–233

Patanakul P, Milosevic DZ, Anderson TR (2007) A decision support model for project manager assignments. IEEE Trans Eng Manag 54(4):548–564

Pennypacker JS, Dye LD (2002) Project portfolio management and managing multiple projects: two sides of the same coin? In: Pennypacker JS, Dye LD (eds) Managing multiple projects. Marcel Dekker, New York, pp 1–10

PMI (2008a) A guide to the project management body of knowledge. Project Management Institute, Newtown Square

PMI (2008b) The standard for portfolio management. Project Management Institute, Newtown Square

Rajegopal S, McGuin P, Waller L (2007) Project portfolio management. Palgrave Macmillan, New York

Rubinstein JS, Meyer DE, Evans JE (2001) Executive control of cognitive process in task switching. J Exp Psychol Hum 27(4):763–797

Seider R (2006) Optimizing project portfolios. Res Technol Manag 49(5):43–48

Chapter 45
Project Management for the Development of New Products

Dirk Pons

Abstract New product development (NPD) projects are difficult to manage because of the subjectivity of the customer requirements, the inter-dependencies between different parts of the solution, and the subjective nature of personal and collective innovation. This chapter describes the NPD life cycle and how it may be managed. A systems perspective is applied. This describes the processes of identifying the desired product functionality, assessing feasibility (and terminating the project if necessary), and how to manage the NPD team.

Keywords Human resource management • New product development • Product development life cycle • Project management • Systems engineering

45.1 Introduction

New product development? (NPD) is a difficult process with risky outcomes. There are many theories of how to do it, and many complexities to consider. This makes it difficult for practitioners to manage. What actions should they be taking? What are the downstream implications of decisions they are about to make right now? How should organisations manage themselves to optimise innovation

Existing theories of design theories primarily address the *engineering* aspects of the NPD process, and therefore are relatively weak at providing guidance for the underlying *project management* activities (cf. Chap. 48 of this handbook). Engineering design involves complex problem-solving (Söderlund 2002). The objectives (e.g. customer's needs) are often partly implicit, specific solutions may not be readily available, multiple solution paths may exist with outcomes that cannot be predicted. Different parts of a solution are entangled, so that solving one sub-problem affects other parts of the solution. The variables are often qualitative and so much of the body of standard engineering sciences is inoperable. Furthermore, the criteria for determining "success" usually involve qualitative factors, and the

D. Pons (✉)
Department of Mechanical Engineering, University of Canterbury, Christchurch, New Zealand
e-mail: dirk.pons@canterbury.ac.nz

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_15

decision-making processes have an element of subjectivity. Also, design involves the activity of personal and collective creativity, and this too adds a subjective complexity to the management of NPD. For all these reasons, NPD is a complex process, and hence difficult to manage (Smulders et al. 2003; Leus and Herroelen 2004; Olin and Wickenberg 2001). This chapter applies a systems engineering method to represent this complex situation. It demonstrates the value of approaching NPD as a net of interacting activities. Existing approaches to this problem are reviewed in Sect. 45.2, followed in Sect. 45.3 by considerations of the design life cycle. A systems perspective is applied in Sect. 45.4, and progressively developed to encompass the management of NPD projects (Sect. 45.5), the customer needs and product attributes (Sect. 45.6), design activities (Sect. 45.7), the assessment of project feasibility (Sect. 45.8), and the integration of NPD process with the production activities (Sect. 45.9). These sections focus on the task-oriented aspects of the NPD process, but there is also the people-oriented aspects to evaluate. These are considered in Sect. 45.10, which takes in a number of topics including creativity, management of innovative people, teams, conflict, and organisational culture. Implications for practitioners are summarised in Sect. 45.11, and opportunities for further research are also identified there.

45.2 Existing Approaches

There are several existing methods that are commonly applied to managing NPD projects. These include systematic design methods, project management, and general management. These are briefly reviewed below.

45.2.1 Systematic Design Methods

The systematic engineering design theories (Hubka 1987; Hubka and Eder 1988, 1996; Pahl and Beitz 1988; Hales 1994) take the position that design is primarily a creative activity, one of finding solutions to pre-defined problems. Consequently these theories require that the design problem needs to be fully specified before creative solutions can be found, and spend considerable attention on that specification process. The validity of the solution is then apparent by determining the extent to which it satisfies the initial specification. The acceptance decision itself is thus easy to make. The requirement for definition of specification is also important for artificial intelligence design methodologies, e.g. genetic algorithms and expert systems (Andert and Peters 1994; Biondo 1990; Haykin 1994; Kuipers 1994; Silverman 1994).

These formal design methodologies also rely heavily on another characteristic: that the problem is decomposable into sub-problems that can be solved independently and those solutions then reassembled to make a whole. Real design problems

may instead present with complex functional interdependencies between parts or production processes. This is also known as the *distributed design* problem (Finger and Dixon 1989a,b). This makes it difficult to plan design projects, because it can be difficult to anticipate when in time the different workstreams are likely to intersect, until closer to the time.

Another complexity arises because the NPD process invariably involves iterations, i.e. the *recursive* nature of design. This is particularly so for radical design, less so for routine incremental design. Concepts are selected and design proceeds, only to find an obstacle, and the designer goes back and takes a different conceptual approach. The locus of effort for any one design workstream within a NPD project is therefore convoluted over time: it intersects with other workstreams, and it folds back in rework loops. This makes it difficult to develop detailed project plans up front.

Consequently the formalised project management methods are often only partially applied in NPD. Their adoption is low during the conceptual stages but higher at detailed development (Lewis et al. 2002; Panico 2004).

45.2.2 Project Management Methods

Perhaps the best-known formalised project management (PM) method is the Project Management Body of Knowledge (PMBOK) (PMI 2004). It identifies nine knowledge areas: Project Integration, Scope, Time, Cost, Quality, Human Resource, Communications, Risk, and Procurement. Superficially, many of these elements would seem highly relevant to NPD. Scope corresponds to customer requirements and design specification, time and schedule are evident in the Gantt charts frequently used in NPD projects, quality is well-understood in design, human resources are important, as is the communication around the design process, e.g. design history files. In fact the project management method is often applied to NPD: it is just that the results are not always ideal, and the process cannot be followed completely (Olin and Wickenberg 2001).

The perceived irrelevance of the PM framework to NPD is undoubtedly partly due to lack of suitable contextualisation of the PMBOK, which otherwise is much more suited to infrastructure projects and activities of a more routine nature than NPD. Specific criticisms of the PMBOK are: that it takes a contractual approach to scope, and hence results in deterministic and pre-determined objectives that do not fit well with the changing landscape of design; that its understanding of quality is severely limited, being primarily manufacturing variance, and excludes the *voice-of-the customer* concept; that it treats people as mere homogenous units of labour that can be discarded at the end of the project; that its perspective of communication is overly directed at the formal contracting situation, and it has little real understanding of the difference between client and customer (end-user) so necessary in design. The *client* is the person who owns the intellectual or commercial rights to put the product on the market, as opposed to the *customer*

who buys the product and is the end user. The client may be internal or external to the organisation that does the design, and has specific needs that must be met, e.g. design for manufacture and production economics. In contrast the customer's needs are for product functionality. While these needs are less proximal, and may be merely implicit, they are vital since the product can only be commercially successful if customers desire to purchase it.

The NPD practitioners have their own more effective methods for all these things, and therefore do not find the PM method in its entirety particularly useful. The PM method is not well-suited to the type of thinking required to manage NPD. The complexity of the NPD tasks is more than can be managed by a simple piecemeal application of the nine knowledge areas. Methods have been developed to handle task dependencies in NPD, such as the design structure matrix (DSM) method (Yassine and Falkenburg 1999; Denker et al. 2001).

The need in NPD is therefore for a system of management that accommodates the complexities (flexibility), and can include all the workstreams (integration). While the PMBOK *does* have a particular section devoted to project integration, it advocates a rather rigid system of delegated authority, more suited to contractual situations than NPD.

One of the strengths of the PM method is its strong focus on the temporal dimension. This is evident in how PM defines itself (a time terminated endeavour), a focus on schedule, and its representation in the Gantt chart. Also, the PM method includes well-developed tools for identifying the critical path (in time), handling uncertain time estimates (PERT), and optimising schedule, as other chapters in this book illustrate. However, that also results in a limitation, in that PM is preoccupied with time as the critical success factor. This limitation becomes apparent in NPD, where uncertainties in the technology solution path are often a major disruptor of the planning. In an ideal world we would have a methodology that could identify the critical path for time, the critical route for technology completion (including interdependencies between design decisions), the critical cost drivers, etc.

45.2.3 Management Perspective

By comparison the management perspective tends to view "innovation" in a decontextualised manner. There is not much recognition that different types of innovation may require different types of support from management (exceptions exist, de Leede and Looise 2005). Instead there tends to be a one-size-fits-all approach based on an implicit premise that some universal key success factors exist for innovation. Thus the management literature is less explicit in identifying which type of innovation is being considered, though a distinction is typically made between process (also termed administration) and product innovation (Mavondo et al. 2005). The management view also encompasses the idea that "product innovation … [includes] diffusion of the product to new sets of customers" (Mavondo et al. 2005, p. 1246), whereas the engineering perspective would see that as effective marketing rather than innovation per se.

In this chapter we take the premise that NPD refers specifically to physical *product* development, and necessarily involves the application of design or engineering activities. Thus we do not specifically address financial "products" (e.g. saving accounts), or service "products" (e.g. mobile data plans), nor innovation generally.

45.3 Design Life Cycle: A Long-Term Strategic Project

The NPD process plays out in the time dimension. Thus a helpful perspective on design is the temporal life cycle. All products have an origin in time, and eventually become obsolete, hence life cycle.

The process starts with the customer's needs, and proceeds to deliver a product in satisfaction of those needs. Many activities are undertaken on the way, as shown in Fig. 45.1. Many organisations have multiple such NPD projects running concurrently at different stages of completion, and this diagram represents a single project. It can be helpful to perceive of the NPD process as catching a customer's need and (after some intense effort) replying back with a product.

The customer-orientation is therefore a core attribute of successful NPD. The complexity of the NPD process arises from the diversity of activities that need to be undertaken, and the interactions between them (including rework loops). The challenge in managing NPD projects is providing the integration necessary. Without

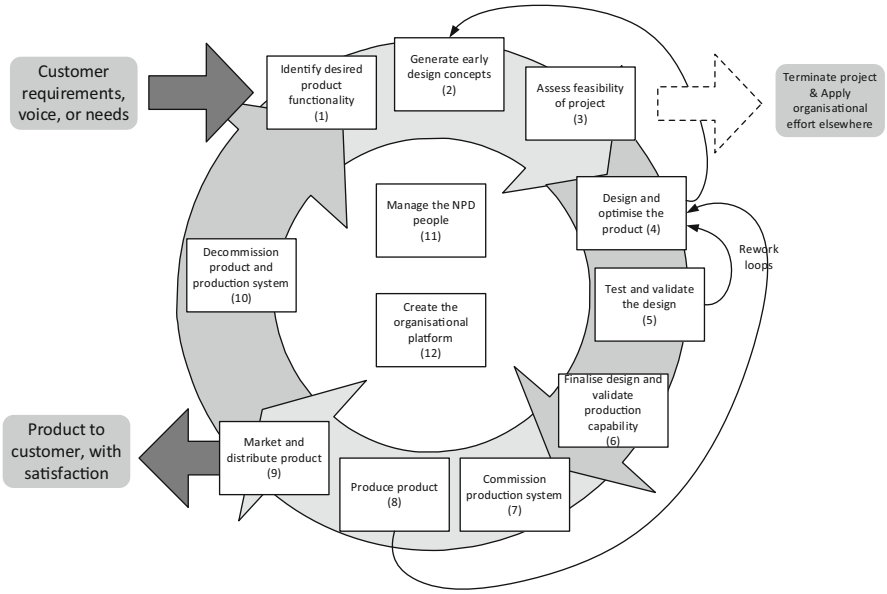


Fig. 45.1 The new product development (NPD) life cycle. The process starts in response to a customer need, and proceeds to reflect a product back to the customer and thereby satisfy the need

this integration it is difficult to develop a product that satisfies all the stakeholders needs (customer, producer, and potentially others) in a coherent manner. The temporal life cycle is an important perspective at the organisational level, for two reasons. The first is that NPD projects can have a long life cycle. It may be many years from concept development, through detailed design, into production, then several years in the market, and finally withdrawal of the project. Even then, there is a need to decommission production plant and retain sufficient spares for the rest-of-life usage of the product. Accommodating recycling and disposal of old product is also an increasingly important consideration, especially for the environmentally hazardous components of the product. The life cycle has to be managed over this entire time, and even though the individual designers only see a part of the product life cycle, the organisation still has to maintain a coherent approach to how it introduces its new products, supports the customer's use thereof, and eventually withdraws them. Organisations built up brand awareness, wherein customers know about their products and have specific expectations regarding the quality thereof. Since customers apply that knowledge to future purchase decisions, it benefits organisations to actively manage their market portfolio of product offerings. These decisions are strategically important and therefore form a significant part of the activities of executives and the board.

The second reason why the temporal NPD life cycle is important to the organisation is that projects consume resources, and any organisation only has finite resources. Organisations must therefore make strategic decisions about which product to develop, and which to abandon or suspend. This is the *capital rationing* consideration. Given that applying resources to product development *now* only results in sales income in the *future,* there are important cash-flow implications.

For start-up organisations with a capital intensive NPD project, the temporal considerations are paramount, and a matter of life-or-death for the organisation. This time in the life cycle of the start-up organisation is termed the *valley of death*. The issue is the prolonged negative cash-flow during the NPD process, and the frequent organisational failure that results. This phase is not attractive to professional venture capitalists, because of the risks, and the entrepreneurial founders of the organisation often have to rely on their own wealth to get through this phase. This puts considerable pressure on the management of such projects. These organisations value lean design, reliable manufactured products, and minimised time-to-market.

Fortunately it is relatively easy to apply project management methods to the life cycle part of the NPD process. When constructing a project plan with projections for perhaps 5–20 years, the complexities of the design process itself, including the inevitable rework loops, are less significant, and may be considered a lumped block of time. In addition, it is possible to anticipate the major milestones in the product life cycle and build them into the plan, as follows (adapted from Pons 2008):

New product development life cycle

1 Identify desired product functionality

    Identify customer perceptions and decision-making mechanisms
    Determine customer needs for the product

Evaluate quality of current system (product, service)
Identify strategic risks to the organisation
Define criteria for new system (product, service)

2 Generate early design concepts

Idea generation
Concept design

3 Assess feasibility of project

Consider production implications
Check strategic feasibility (e.g. SWOT—an analysis of strengths, weaknesses, opportunities, and threats)
Check market
Check technology capability
Check financial feasibility
Check schedule feasibility
Check for resources available
Create project proposal
Make decision to proceed/not
Close project (if required)

4 Design and optimise the product

Set the specifications
Design key characteristics
Evaluate production implications
Produce computer aided design (CAD) models
Analyse and optimise design
Produce drawings
Produce prototype

5 Test and validate the design

Test product for user satisfaction
Test key characteristics (e.g. engineering)
Finalise design

6 Review the design

Validate production capability
Board approval
Revise design
Freeze the design

7 Commission production system

Procure manufacturing capability
Design the tooling
Build the tools

- Modify building
- Obtain equipment
- Obtain manufacturing staff capability
- Commission plant

8 Produce product

- Get first parts from production
- Test parts
- Verify quality tolerances
- Produce in volume

9 Market and distribute product

- Market product
  - Identify key benefits of product
  - Identify potential users
  - Plan marketing strategy
  - Produce brochures, adverts
  - Produce campaign
- Arrange distribution
  - Establish sales chain
  - Find local representatives
  - Establish business procedures for ordering, shipping, accounting, repair
- Set up technical support capability
  - Write user manual
  - Write service manual
  - Decide on warranty conditions
  - Obtain staff capability
- Market economics
  - Market growth
  - Market maturation
  - Market decline and rejuvenation
    - Declining sales
    - Refresh product
    - Launch derivative product
    - Differentiate service
    - Launch new product

10 Decommission product and production system

- Decision to withdraw
- Produce lifetime spares requirement
- Decommission production
- Archive documentation

11 Project closure

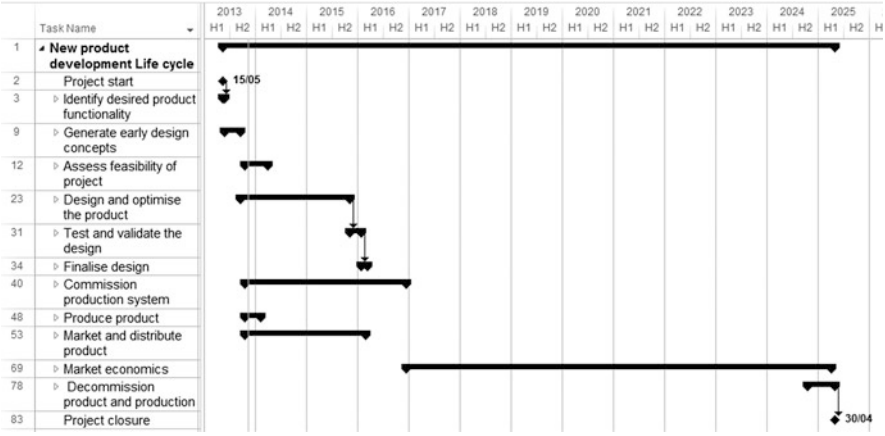


Fig. 45.2 Gantt chart showing the activities of the NPD life cycle placed into a work breakdown structure and scheduled in the temporal dimension. The detail here is merely representative, the more important point to note being the long duration of NPD projects, and the consequences for the organisation. Image adapted from Pons (2008)

These activities may readily be included into a Gantt chart, with its strong representation of the time dimension. The activities can then be scheduled, to determine likely time to market and other key milestones, see Fig. 45.2. It is then possible to estimate the future positive cash-flows, based on assumptions about production cost, price and sales volumes (Pons 2008). Naturally these estimates are of unknown validity, but that is not usually a problem with this type of project planning, for which the purpose is to understand the big-picture and the long-term potential of the NPD project. The time to *break-even* (as soon as possible is preferred) and the *return-on-investment* (ROI) can then be determined. Potential investors are particularly interested in the information that this type of longitudinal analysis provides. Furthermore the project management method readily provides a means to introduce some variability into these projections, through the PERT (project evaluation and review technique) method. PERT addresses stochastic variability in estimates of *duration,* which is a good start since time-based costs are important in this situation. However there are many other project variabilities that are not captured in PERT, such as unexpected rework loops, difficulties getting the product technology to work, cost variability, and discrete events (e.g. responses by competitors).

The temporal dimension is readily evident in Gantt charts. The cashflow implications can also be anticipated once the Gantt chart is created and the project scheduled and costed. However the Gantt chart does not capture the strong customer orientation that NPD organisations show: this is better represented in Fig. 45.1, though it will be seen that the two figures represent the same activities.

It is also possible to include the risk-management process with this type of long-term planning, by combining strategic methods (including SWOT and

PESTLE) with qualitative risk management (Pons 2010). (SWOT: strengths, weaknesses, opportunities, and threats, PESTLE: political, economic, social, technological, legal, environmental). Hence a form of *strategic risk management* may be achieved, one closely coupled to project management.

In this way, the conventional project management methods have a lot to offer in strategic scheduling at the organisational level. The PM methods, particularly the Gantt chart, also are very commonly used to manage the work breakdown structure and schedule of specific workstreams with the NPD project. Yet as the life cycle demonstrates, there is a lot more to managing NPD projects than schedule considerations alone, and it is to these other considerations that attention now turns.

45.4 Systems Perspective

Given the complexity, it is useful to take a systems engineering (SE) perspective of NPD projects. This lens is very similar to PM, and they share many commonalities and tools, but the key difference is how SE approaches the problem. Complex projects don't respond well to piecemeal solutions, because of the interaction of effects. These problems need to be treated holistically, and solutions implemented in an integrated manner. This is what SE aims to achieve. Examples of systems engineering applications are shown in Figs. 45.3 and 45.4.

Fig. 45.3 The Curiosity Mars rover was landed by a rocket-powered hovering skycrane. Getting all this hardware to this situation is a system problem, not merely a project. Image credit: http://mars.jpl.nasa.gov/m2020/images/PIA14839-fi.jpg

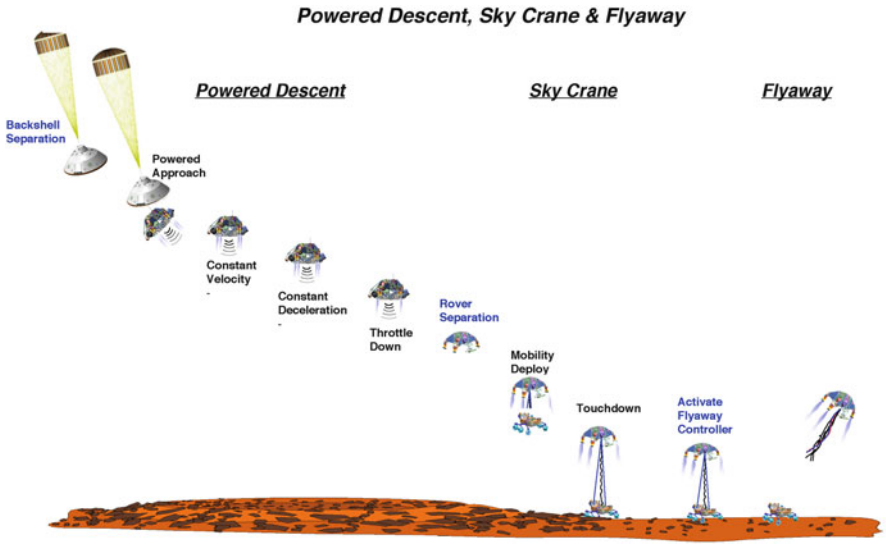


Fig. 45.4 Here are more details about the Mars rover deployment. More of the systems become apparent at this level. For example we see previous separation and control activities, and the subsequent decommissioning of the skycrane. Managing these system interactions for maximum dependability is an essential part of managing such NPD projects. Image credit: http://mars.jpl.nasa.gov/msl/images/msl/MSL_TL_EDL_1.jpg

Systems engineering is the application of engineering analysis tools in a systematic and integrated manner, for the solution of complex problems, typically for the development of new products and equipment. It is particularly focussed on:

1. Taking the big-picture perspective: e.g. using Integration definition zero (IDEF0) flowcharts (FIPS 1993)
2. Making sure that all the design is consistent with the customer requirements: e.g. using quality function deployment (QFD)
3. Ensuring that the analysis is realistically representative of real issues the product will have to face in its life, e.g. finite element analysis (FEA)
4. Effectively managing the design & product development process: typically using project management (PM) methods, and Risk management (RM)
5. Planning for appropriate testing regimes to validate the analyses and product performance: often using Design of Experiments (DoE), and Reliability testing.
6. Integrating the manufacturing into the design considerations: e.g. Design for manufacturing and assembly (DFMA) and concurrent engineering
7. Considering the full life of the project (from concept through to commissioning of plant), and of the product in the customer's hands (supply chain, manufacture, distribution & sales, user instructions, maintenance/support/repair, product disposal, recycling). Hence life cycle analysis (LCA).

A more formal definition of systems engineering from its professional body the International council on systems engineering (INCOSE) is: "Systems engineering integrates participating disciplines and specialty groups into a team effort by coordinating contributions throughout the system life cycle stages from concept to disposal. Systems engineering balances the social, business, and technical needs of all stakeholders to achieve a quality product that meets these needs" (INCOSE 2013). There is nothing in here that the project management method would contend with, and rather than seeing these as competing frameworks it is more useful to see them as different methodological approaches that can complement each other. As the descriptions show, systems engineering is particularly focussed on product development, whereas project management has a more general field of application. The PM method has the advantage of being more specific about process—it has a standardised methodology—whereas the systems engineering method is somewhat vague in places. On the other hand, it is that flexibility that makes it attractive to NPD. Systems engineering is about avoiding unintended consequences that could have been anticipated, whereas project management is more focussed on anticipating work packages and scheduling them temporally.

One of the ways that systems engineering helps the management of NPD projects is by providing a high-level integrated perspective. In the next section we show how this may be applied.

45.5 Managing NPD Projects

As we have seen, the development of a new product involves projects that typically go through phases, and have wide-ranging interactions with other activities in the organisation, such as marketing and manufacturing. These activities have already been shown in the simple flowchart of Fig. 45.1, and the Gantt chart of Fig. 45.2. Now we reconfigure them in the system-representation of Fig. 45.5. This shows the same information, but with additional detail. The value in doing this is that it permits a deeper exploration of some of the issues with managing NPD projects.

This diagram is represented in integration definition zero (IDEF0) system modelling notation (FIPS 1993). The IDEF0 model represents the proposed relationships of causality for a complex situation. As is usual with this method, we focus on the *activities* which in this case are the actions that are conducted as part of managing the NPD process. With IDEF0 the object types are inputs, controls, outputs, and mechanisms (ICOM) and are distinguished by placement relative to the block, with inputs always entering on the left, controls above, outputs on the right, and mechanisms below. A block represents an action. The inputs (if any) are transformed or even consumed by the function in the block, to produce one or more outputs. Input arrows are always on the left of the block, and outputs on the right. Controls (or constraints) enter above the block and initiate or ensure the output is correct. The mechanisms (if any) that support the function enter under the block. The notation therefore permits inputs and outputs to be clearly distinguished from other factors

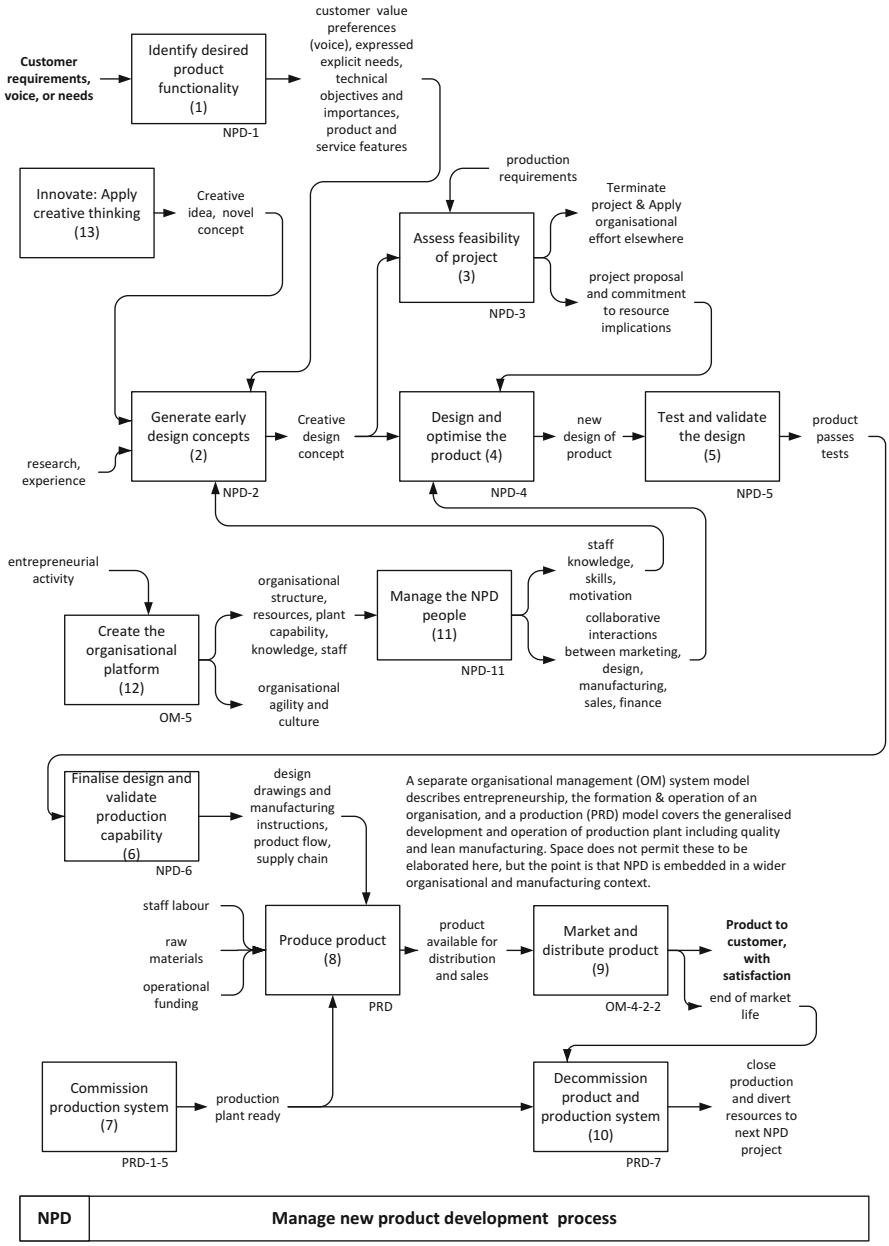


Fig. 45.5 System model of the new product development process. This diagram shows the same activities as the previous representations of the NPD life cycle, but with additional detail and using IDEF0 notation. Notice that the inputs to the process are the "Customer requirements, voice, or needs", and the output is a "Product to customer, with satisfaction", as before

that influence the activities. With other flowchart notations the meanings of the arrows is seldom explicit, and they generally represent sequence of activities or influence. However, with IDEF0 it is essential to note that arrows convey objects to activities. Therefore, an activity may begin autonomously when its required inputs are available, mechanisms are operable, initiators exist and constraints permit. Consequently, the IDEF0 notation readily provides that multiple activities can be simultaneously active, i.e. concurrent. It retains support for sequenced (serial) activities. Also, the locus of effort may make multiple iterations through some activities.

We briefly describe the activities apparent at this level. First is the customer orientation of identifying the desired product functionality (1). This is an important activity in understanding the NPD process, and we describe it in further detail below. One of the important outcomes is the identification of the test and acceptance criteria: these are necessary to know whether or not the design is sufficient for the intended purpose. Next are a bracket of design-centric activities. There is the development of creative thinking (13), and generation of early design concepts (2), which are where the creativity occurs for which design is renowned. This is usually followed by an activity of assessing the feasibility of the project (3), though this activity could alternatively be positioned earlier or later in the process.

The next activity is the design and optimisation of the product (4). For most products of a consumer nature, there is a strong reliance on computer aided design tools. The design process is primarily focussed on physical geometry, e.g. solid models and printed circuit board design. It is augmented by a variety of analysis tools, too many to list here, but including finite element stress/strain analysis (FEA), computational fluid dynamics (CFD) for fluid flow, electromagnetic analysis, mold-flow analysis (for plastic injection molded parts). These tools allow the optimisation of the design before parts are made.

Test and validation of the design (5) is next. The test and acceptance criteria are generally already available, having been set at (1), but a considerable effort may be involved in producing prototypes and testing them in all the load cases expected of the product in service. Rapid prototyping and 3D printing technologies are useful at this stage. The manufacturing and production requirements are considered throughout these design activities, i.e. design for manufacture and assembly (DFMA) methods. This finds its fulfilment in the next activity, which is to finalise the design (e.g. production of manufacturing drawings and instructions) and validation of the production capability (6). Sometime concurrent with this, it is necessary to commission the production system (7). This can have a long lead time, and benefits from the release of preliminary design information much earlier in the process, e.g. the sizes of the parts, even if not the details. When all that is ready, then its time to produce the product (8). Production is a large topic in its own right, and we do not go into it here. However it is relevant to note that the quality, continuous improvement, and lean management methods operate here. Consequently the production process will eventually start identifying areas for improvement in the design, and thus generate a re-design process (not shown in the figure). Naturally the marketing and distribution of the product (9) is a key activity

in actually getting the product to the customer. The diagram also shows some other activities. One is the withdrawal of the product from the market at the end of its economic life, and the decommissioning of the production system (10). Generally this activity is poorly anticipated, if at all, in project planning for NPD. In many cases the cost implications are low, but there are situations, such as nuclear power, where the decommissioning costs exceed the original fabrication costs, so it is worth keeping this is mind. Finally we wish to mention two other organisational activities. One is to create the organisational platform in the first place (12). The other activity worth considering is the management of this whole NPD process and the people involved (11). There are some particular challenges in here, because the creative temperament that drives the conceptual design at (2) also needs to be channelled constructively.

The IDEF0 model is progressively decomposed to show finer detail where necessary. As will be shown, this approach permits us to take a holistic perspective of NPD, and gives us the means to integrate the many activities in a coherent manner. It also permits detailed models of specific workstreams, and these can readily be converted to a work breakdown structure (WBS) and scheduled into a Gantt chart with the usual PM methods. Generally the detailed design activities, (4) onwards, enjoy more substantial project management effort than the conceptual stages (1) and (2) (Lewis et al. 2002). These earlier stages are the more challenging, and therefore we will concentrate our analysis here.

45.6 Identify Desired Product Functionality

The ultimate objective of this part of the NPD process is to define the design criteria for the new product. However there is much to be done beforehand, and the anticipated actions are represented in Fig. 45.6. Customers have perceptions about the product, that affect their purchasing decision (1) and these need to be understood. The organisation also wants to determine the customer needs for the product (2), because these will be directly useful in the engineering design activities. Another activity is for the organisation to evaluate the quality of its current product offering (3). Design and manufacturing engineers understand this well, since this type of thinking is embedded into quality and the continuous improvement process. This is a particularly valuable activity once the product (or a previous variant) has already been on the market, such that customer feedback is available. Much of the incremental product improvement activities are focussed in this area. It has been an enormously successful approach, as evident in the low cost, feature-rich and high reliability of modern automobiles for example. Another consideration is that NPD is invariably a strategic activity for an organisation, as the life cycle schedule showed. It is therefore important to identify strategic risks to the organisation (4) and the implications for new product developments. In this context risk refers to both threat and opportunity. Organisations always have the choice to do nothing different, merely maintain the status-quo. However that has risks that their product

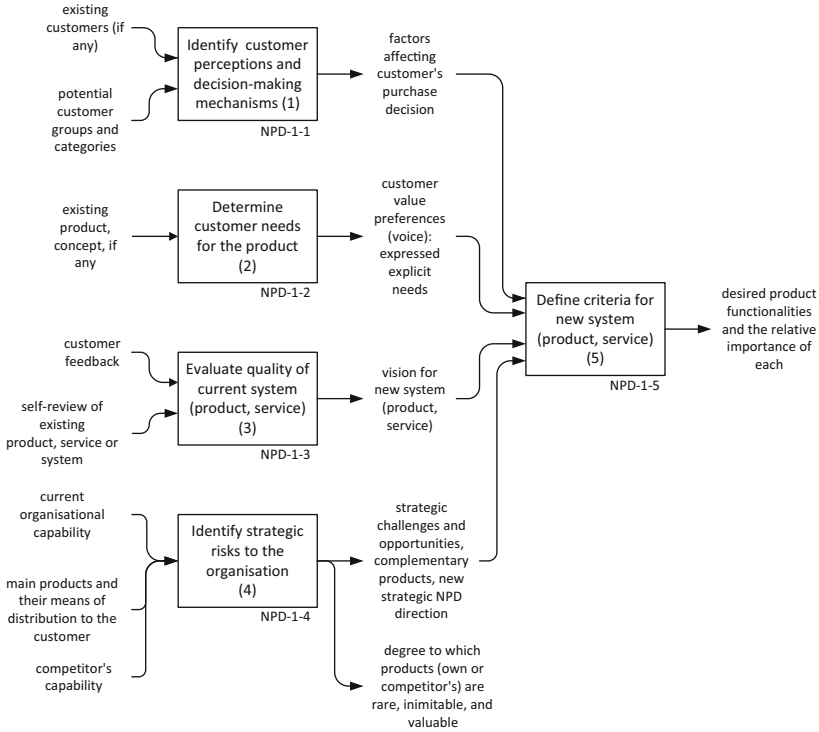


Fig. 45.6 System representation of the process for identifying the customer needs and product functionality. This is a key process in NPD, and drives the entire venture

becomes obsolete in the market, and then causes the organisation to fail. NPD is the route to organisational invigoration, but it can also fail spectacularly. Hence strategic choices have to be made.

Thus at this level of the model we see several very different activities taking place, before the NPD project is even initiated. There is marketing/advertising, voice of the customer, continuous improvement, and strategic decision-making occurring. Next we briefly look at several of these in turn. Space does not permit a full elaboration, rather our purpose here is to help the practitioner understand the issues involved, suggest some tools that may be used to manage these work streams, and identify some of the relevant literature. These IDEF0 flowcharts represent actions and outputs, and can therefore readily be repurposed into WBS work packages, scheduling applied, and Gantt charts created.

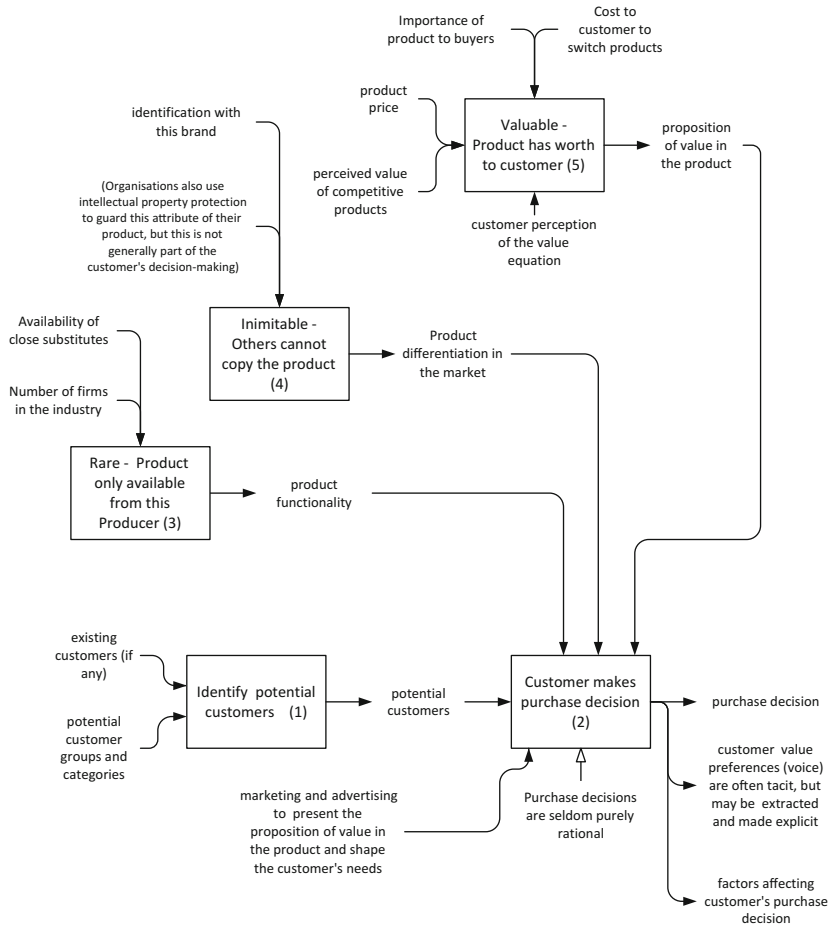
45.6.1 Identify Customer Perceptions and Decision-Making Mechanisms

In NPD it is desirable to know what the customer is likely to think and feel about the product. In the later stages of product development it is possible to present prototype products to focus groups and solicit this information. However that cannot always be done at the early NPD stages. In which case a useful way of approaching the problem is in terms of the competitive position: products are successful to the extent that they are *rare* (only available from this producer), *inimitabl*e (others cannot or may not copy the product), and *valuable* (has worth to customer) (Barney 1991). With the addition of *organisation* this is also called the VRIO framework (Barney and Hesterly 2012). The implications for products are modelled in Fig. 45.7. Thus products have a proposition of value to the customer, and the customer has her own perception of the value equation, and uses this to make the purchase decision. It should not be expected that the purchase decision will be purely rational, especially for consumer products. Nor is the organisation undertaking the NPD endeavour passive in the process: it uses marketing and advertising to present its proposition of value in the product, and it seeks to shape the customer's needs and priorities.

45.6.2 Determine Customer Needs for the Product

If seeking to affect the customer's decision-making process is one of the organisational activities associated with NPD, another is seeking to determine what the customer specific needs are in the product, see Fig. 45.8. The customer value preferences (voice) are often tacit, but this is not helpful to the NPD process: they need to be extracted and made explicit. Ultimately we need to obtain the customer value preferences (voice), which are expressed explicit functional needs that the customer wants from the product. We also want to know the relative importance of each, because this helps prioritise the downstream design activities (as will be shown). Customers will have general needs in the area under consideration (1), as well as preferences regarding specific products (own and competitor's). There are a number of market-research mechanisms available to determine customer needs, including quantitative methods (analytical hierarchy process, conjoint analysis), and qualitative (focus groups, analysis of user experience). However research is of no use if the organisation does not have the culture to act on it, and in this regard NPD organisations need staff with the necessary customer focus and market orientation.

Market research and focus groups are typical mechanisms for determining needs. The problem here is that customer needs are often implicit. Hence the situation is similar to knowledge management (KM), where the need is to convert implicit (tacit) knowledge into explicit (Polanyi 1958; Nonaka 1994; Nonaka et al. 2000). The KM capture process may use interviews, observations, simulations, or other mechanisms sometimes termed *cognitive task analysis* (Meso et al. 2002).



NPD-1-1	**Identify customer perceptions and decision-making mechanisms**

Fig. 45.7 Customer perceptions affect purchase decisions, thereby affecting NPD

The systems engineering method readily lends itself to analysis in this area. One such example is shown in Fig. 45.9, which analyses how a user (car driver) interacts with a car mirror. The analysis is broken into discrete activities, strung together in the time dimension. It is then a simple task of determining what additional functionality the user might appreciate at each of these stages. (These are shown in italics in the figure). Taken together, these provide a set of functional requirements.

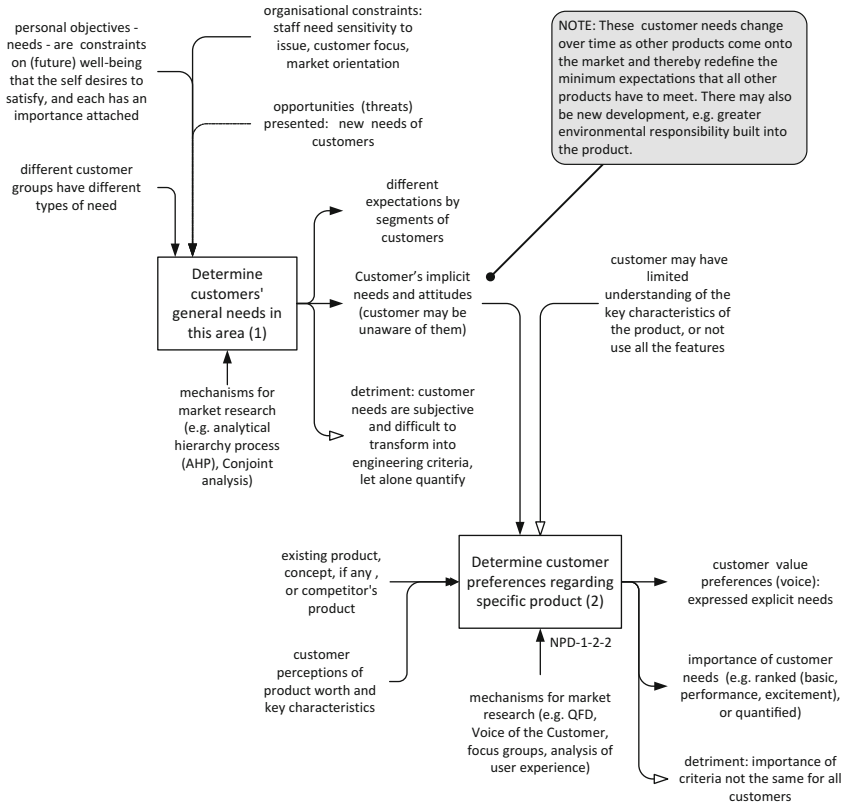


Fig. 45.8 NPD seeks to satisfy customer needs, and therefore seeks to identify those needs in the first place. This is termed customer voice. Note that customer needs may not be explicit to themselves

45.6.3 Evaluate Quality of Current Product, System or Service

Organisations that already have a product in the market have a potentially large and valuable source of ideas for new product features, if they can tap into the information. There are several ways of achieving this, shown in Fig. 45.10. The first is to analyse failed products (1), including warranty claim and maintenance reports. This information is typically used as part of the internal continuous improvement process (2), and can result in significant incremental improvements in the product by inexpensive improvements. The quality literature has much to contribute in this area. Another source of NPD ideas is customer feedback (3), including the complaints. In addition many NPD firms compare their product against that of a competitor (4), or compare their processes (e.g. design and analysis tools) against those of a friendly

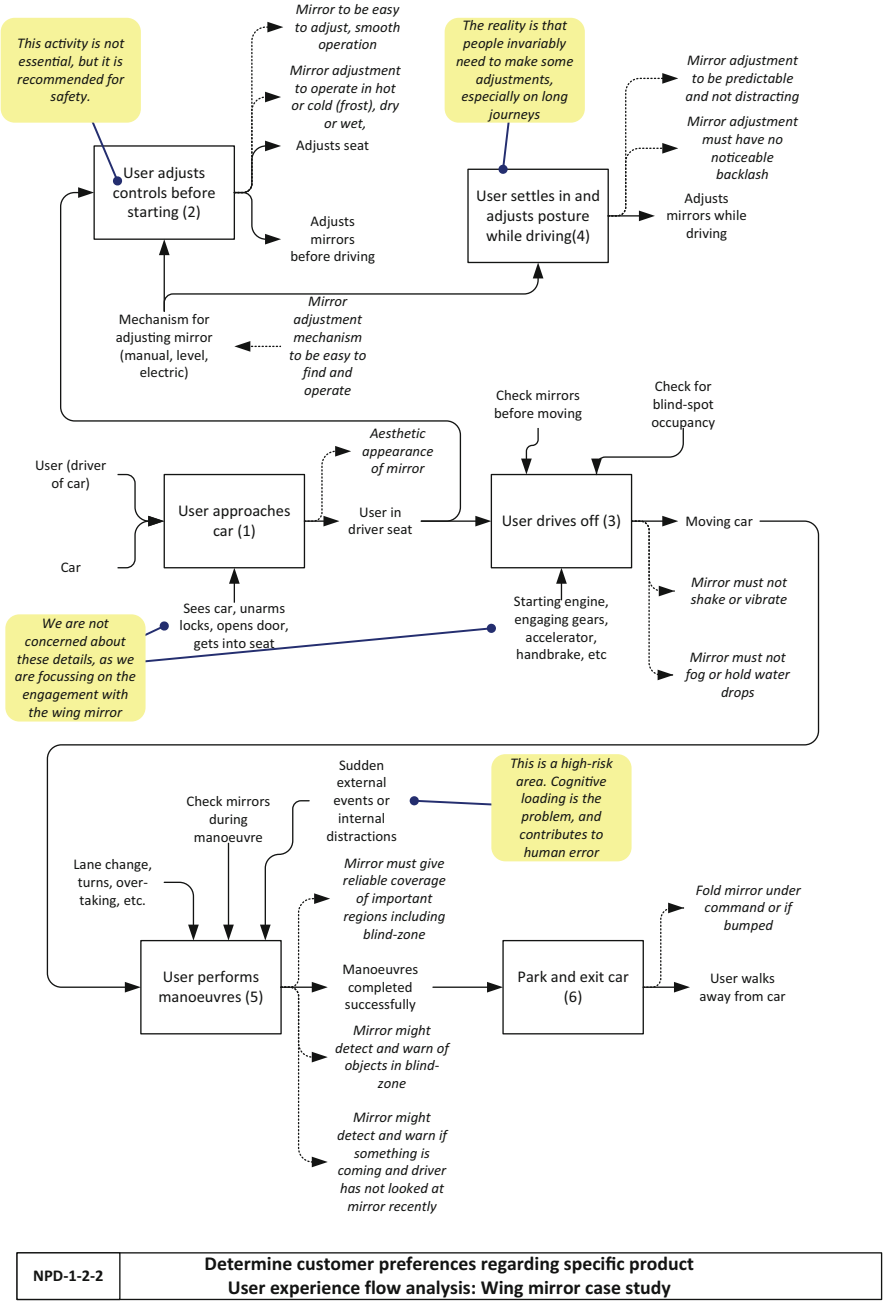


Fig. 45.9 Example of how user needs may be determined by analysing the interactions of the user with the product, in this case a car wing mirror

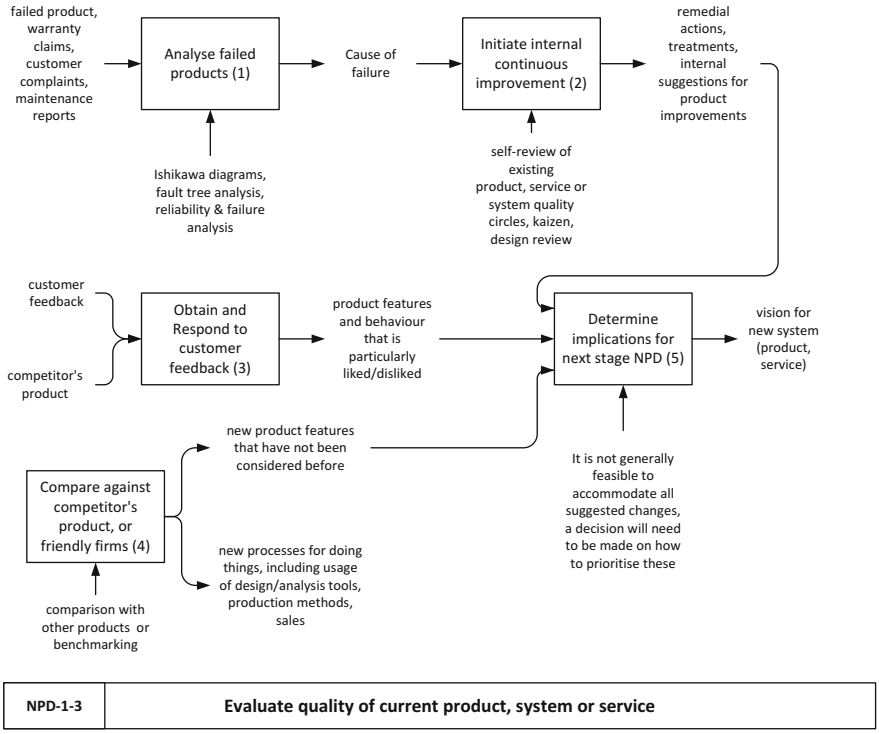


Fig. 45.10 Failed product and the evaluation of competitors' products is valuable information to NPD. Also, NPD is usually closely coupled to the manufacturing activities, and the continuous quality improvement processes within the latter also suggest new improvements to the design

firm which is not a competitor (benchmarking). From these ideas the NPD manager determines the vision for the functionality in the next generation of product.

45.6.4 Identify Strategic Risks to the Organisation

The strategic management of an NPD organisation is important to its success, because of the nature of the risks, threats and opportunities, that come with NPD. However the strategic issues have had limited attention in the NPD literature. In this part of the model, see Fig. 45.11, we show how the strategic considerations operate. This model is consistent with the Baldrige framework (NIST 2013), and the representation given here is necessarily a simplification. Typical activities here are the governance actions of setting organisational purpose (1), the evaluation of NPD opportunities (2) and the development of a vision for the future state of the organisation (3). A strategic plan (4) is developed for achieving the vision, taking into account the organisation's capability (5) and anticipating the organisational

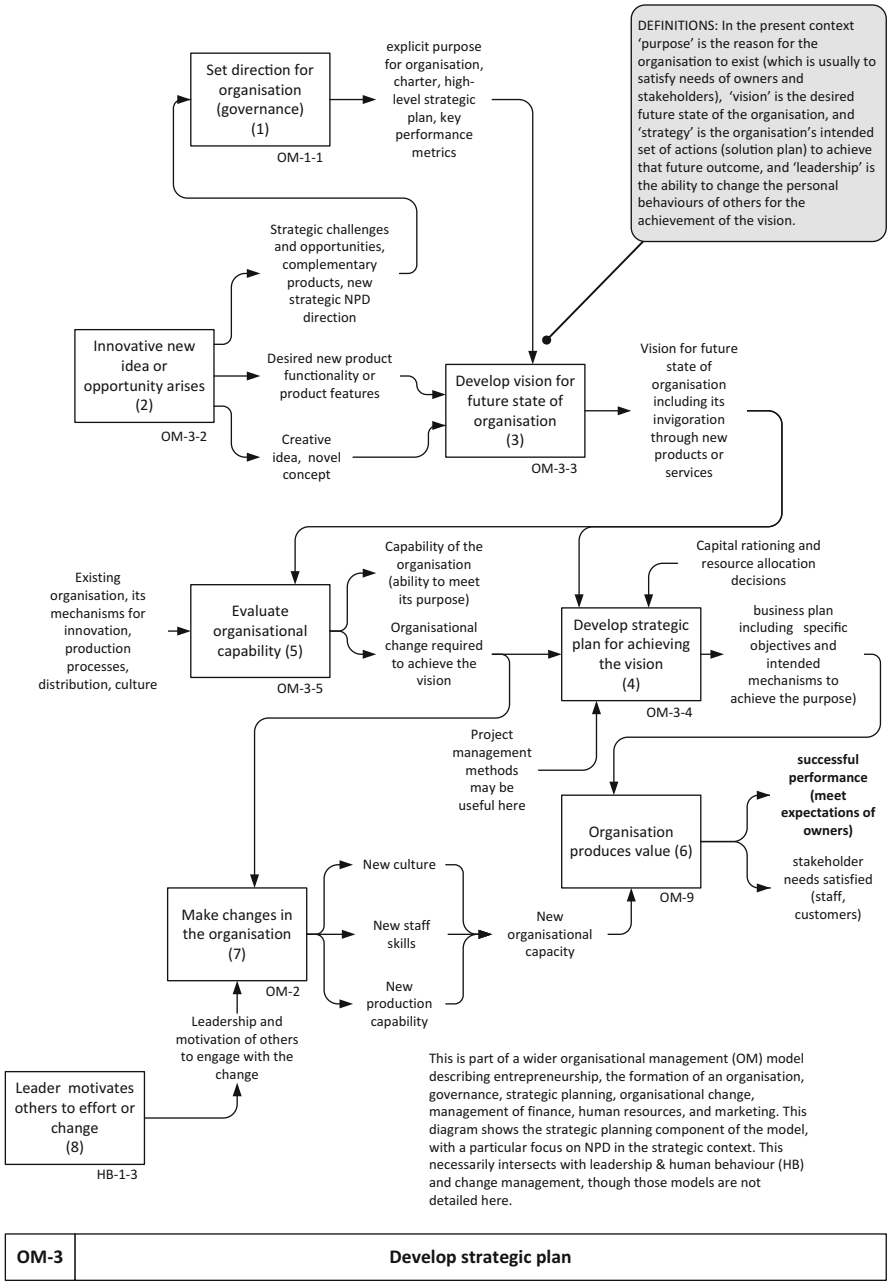


Fig. 45.11 NPD activities are key to the rejuvenation of the organisation, and therefore intimately connected with the strategy and vision activities. These activities tend to occur at more senior levels than the design engineers and project managers, but are nonetheless key drivers in the NPD process

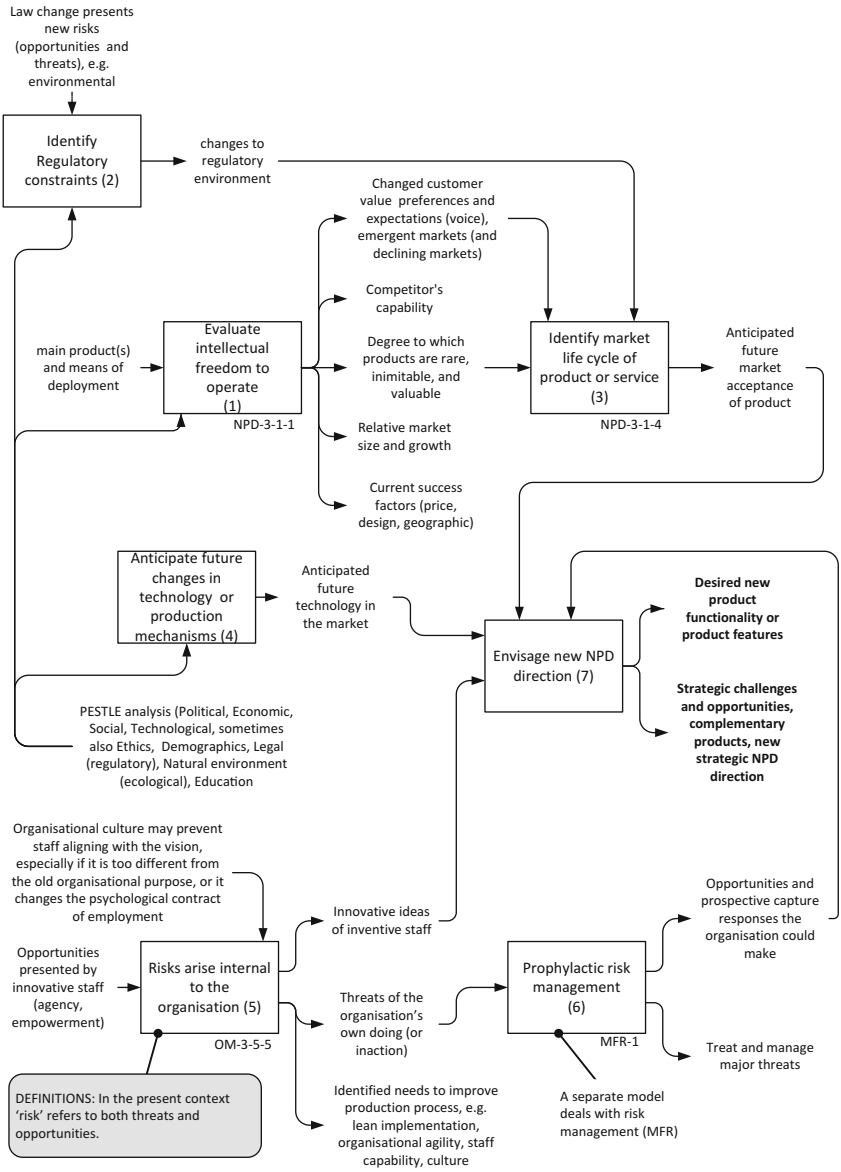
changes required (7) before value can be produced (6). The role of leaders is to motivate others to effort or change (8) within these processes. As this shows, there is an intersection of strategy, vision, leadership, and change-management. These set the context in which innovation and NPD occur.

The activities most directly relevant to NPD are the need to evaluate the NPD opportunities, and these processes are shown in Fig. 45.12. These high-level activities are conducted by the design manager, as opposed to the designers themselves. Chief of these activities is to evaluate the competitive position (1) of the existing and proposed products. This may include market research into relative market size, current success factors, changing customer value preferences and expectations (voice), emergent markets (and declining markets). An evaluation of the degree to which the product is rare, inimitable, and valuable can also be done here if not already completed (see Fig. 45.7 above). It is also necessary to scan for changed regulatory constraints (2). For an NPD organisation that is distributing a product into multiple countries, the regulatory constraints can be significant. Some examples in the NPD area are new national standards for product safety, and changes to product liability legislation. There is also a large class of constraints emerging in the environmental area, in the form of supply-chain and disposal considerations. These are discussed in further detail later (see Life Cycle Analysis), but the main point here is that regulatory constrains can be threats to existing products, and opportunities for new products.

All these considerations help form an opinion of the commercial prospects for the product, and identify where the product is in its commercial life cycle (3). Technology changes make certain products obsolete, and open opportunities for new products (4). For example digital cameras replaced film, and firms like Kodak failed to anticipate how profoundly these changes would affect their future product portfolio. Environmental scanning is important in building awareness of the external situation. Methods like PESTLE analysis are often used here, since they prompt to evaluate these different dimensions.

The above strategic risks originate external to the organisation. There are also internal risks to consider (5). Of particular relevance to NPD are the innovative ideas of inventive staff. Organisations need their staff to be active agents in the innovation process, as opposed to mere spectators. This requires an organisational culture that rewards rather than stifles such behaviours. This aspect of culture is also important in sustaining production quality improvements and implementing lean manufacturing. The lack of such a culture is thus a major threat, of internal origin, to NPD. There can also be threats from the organisation's own inaction, e.g. Kodak's complacency towards digital imaging, which may require a deliberate risk management treatment (6).

Taken together, these activities form the basis for envisaging the new NPD direction for the organisation (7). This might include the recognition of new product functionality, complementary products (or services), or whole new NPD directions. However it is seldom possible to only continue with the status-quo. NPD projects can have long lead times and it is therefore necessary to anticipate the remaining



NPD-1-4	Evaluate NPD risks to organisational purpose

Fig. 45.12 New product development involves risks, both threats and opportunities, and these may arise in several areas

market life of the existing product, and commence the new NPD process in a timely manner, rather than wait until sales are already dropping. Apart from Kodak, other examples of misjudging the strategic direction for new product development are Nokia and Microsoft dismissing smart phones until Apple, Google Android, and Samsung had already dominated the market, and General Motors being fixated on a future of large heavy motor cars and failing to develop a hybrid electric motor car until Toyota's Prius was already successful.

Much of the strategy and vision literature is relevant to NPD, in that the development of new products (and services) is a key part of competitive advantage, and provides a way to seek differentiation from other organisations, i.e. "performing *different* activities from rivals' or performing similar activities in *different* ways" (Porter 1996, p. 62). Porter went on to suggest that new entrant organisations were best placed to find these differences either because they found niches that others had already overlooked, or because they were more able to respond to new opportunities (their lack of established business activities making them more flexible). He felt that organisational forces in large firms tended to defeat the strategic efforts of managers. Nor is vision for new products solely or even mainly done by senior managers. NPD often requires an organisational transformation, and thus needs the collaborative foresight of many within the organisation, rather than the vision/dream of an individual leader, i.e. the "synthesis of many people's visions" (Hamel and Prahalad 2002, p. 30). This construct of emergent strategy also features in the literature on organisational innovation (Nonaka 1994; Lau and Ngo 2004).

45.6.5 Define Criteria for New Product

From the engineering perspective the objective of this set of activities, see Fig. 45.13, is first to convert the customer value preferences (voice) and the organisation's strategic vision for new products, into the engineering attributes of the product. It is important to know the relative importance of the various engineering features, so that the NPD project can be managed appropriately. The second activity is to define the test and acceptance criteria. These are important as they give confidence that the customer's needs will have been met. Another way to consider these activities is that they describe what a satisfactory solution will look like (Ullman 2001). This can also be considered the technical part of the project scope (PMI 2004).

Identifying the engineering attributes in a customer need is a special type of transformation process, and there are particular mechanisms used to achieve this, notably quality function deployment (Gustafsson 1996; Martin et al. 1998; Mill 1994). See Fig. 45.14 for an example applied to the car wing mirror case study.

From the systems engineering perspective these activities are part of the *requirements analysis* process, which seeks to take the client needs, in all the dimensions in which the client seeks value, and identify the implications for the engineering design and development process.

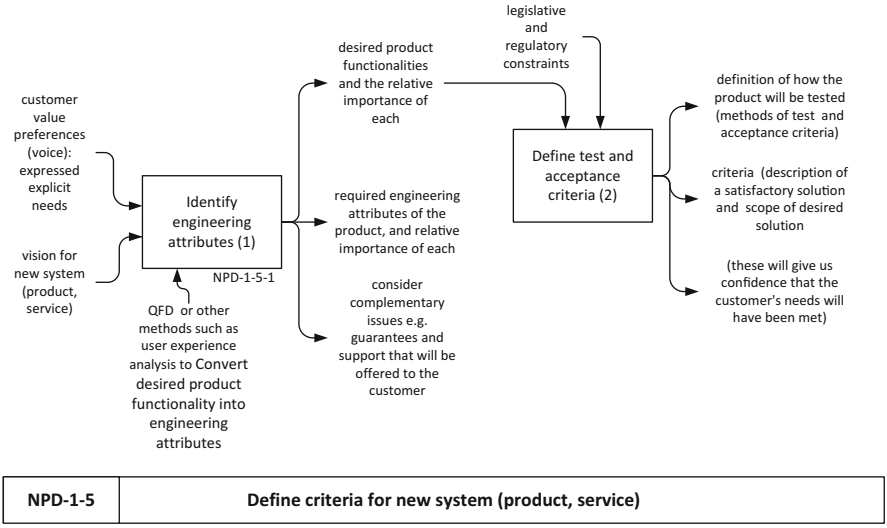


Fig. 45.13 Customer requirements flow into engineering attributes for NPD activities, and onwards to determine the testing implications

45.7 Generate Early Design Concepts

One approach to design is to simply progress as quickly as possible to designing the product details. This is especially relevant to simple problems for which the problem can be fully defined and the solution is obvious. However this is not necessarily a viable approach to more complex design problems. For these it is usually necessary *not* to converge too quickly onto a solution, but rather to explore the solution space beforehand. This exploration of the solution space is termed *conceptual design*. The objective is to generate multiple candidate solutions. The process is usually divergent, i.e. broad ranging and not closing on the first suitable solution found, since the intent is to see if there are any other solutions hidden in the space. The engineering community has long recognised that different cognitive processes are required for "early conceptual" design vs. "detailed" design (Andersson 1994; Calantone et al. 1999; Fairlie-Clarke and Muller 2003; Finger and Dixon 1989a,b).

The main design activities, see Fig. 45.15, are to generate concepts (1), as many and as creative as possible. The figure suggests several of the mechanisms that are used for this, such as but not limited to brainstorming. At this point the concepts are generally vague and even short in practical detail. A subsequent activity is to refine the concepts (2) and get them all up to a similar level of uncertainty, so that they can be compared.

In those cases where a precursor design existed, it is important to retrieve the past design intent and record the current one (3). The risk is that designers move on, and the new team, while "improving" the design, unintentionally breaks some functionality in the old design, or repeats a similar error.

(1) Functional requirements that the customer has of the product, ex the previous functional analysis.

(2) Functional requirements are scored with an importance.

(3) Engineering features or attributes of the product. May be physical sub-systems, as here, or design requirements more generally.

(4) Matrix evaluation is the extent to which the engineering feature is important in providing the functional needs. Requires appropriate knowledge to anticipate.

(5) Weighted total identifies which of the engineering features are the more important, in this case the mirror drive mechanism.

	Importance to user	Mirror controls inside car	Mirror mounting to car	Mirror pivot	Mirror drive mechanism	Mirror shroud	Structural rigidity of mirror pod	Shape details of shroud near mirror	Additional electronics housed in mirror	Shape of mirror including curvature
Aesthetic appearance of mirror	*3*					5		1		
Mirror adjustment mechanism to be easy to find and operate	*5*	5								
Mirror to be easy to adjust, smooth operation	*5*				5					
Mirror adjustment must have no noticeable backlash	*3*		4	4	4		4			
Mirror adjustment to be predictable and not distracting	*3*	4			4					
Mirror adjustment to operate in hot or cold (frost), dry or wet,	*5*				5					
Mirror must not shake or vibrate	*5*		5	5	5		5			
Mirror must not fog or hold water drops	*5*							3	4	
Mirror must give reliable coverage of important regions including blind-zone	*5*									5
Fold mirror under command or if bumped	*3*		2	5	4		2			
Mirror might detect and warn of objects in blind-zone	*2*								5	
Mirror might detect and warn if something is coming and driver has not looked at mirror recently	*4*								5	
Technology importance		**37**	**43**	**52**	**111**	**10**	**43**	**14**	**36**	**20**

NPD-1-5-1	**Identify engineering attributes: Example application of Quality function deployment**

Fig. 45.14 QFD applied to the car wing mirror case. User requirements are converted to prioritised engineering features

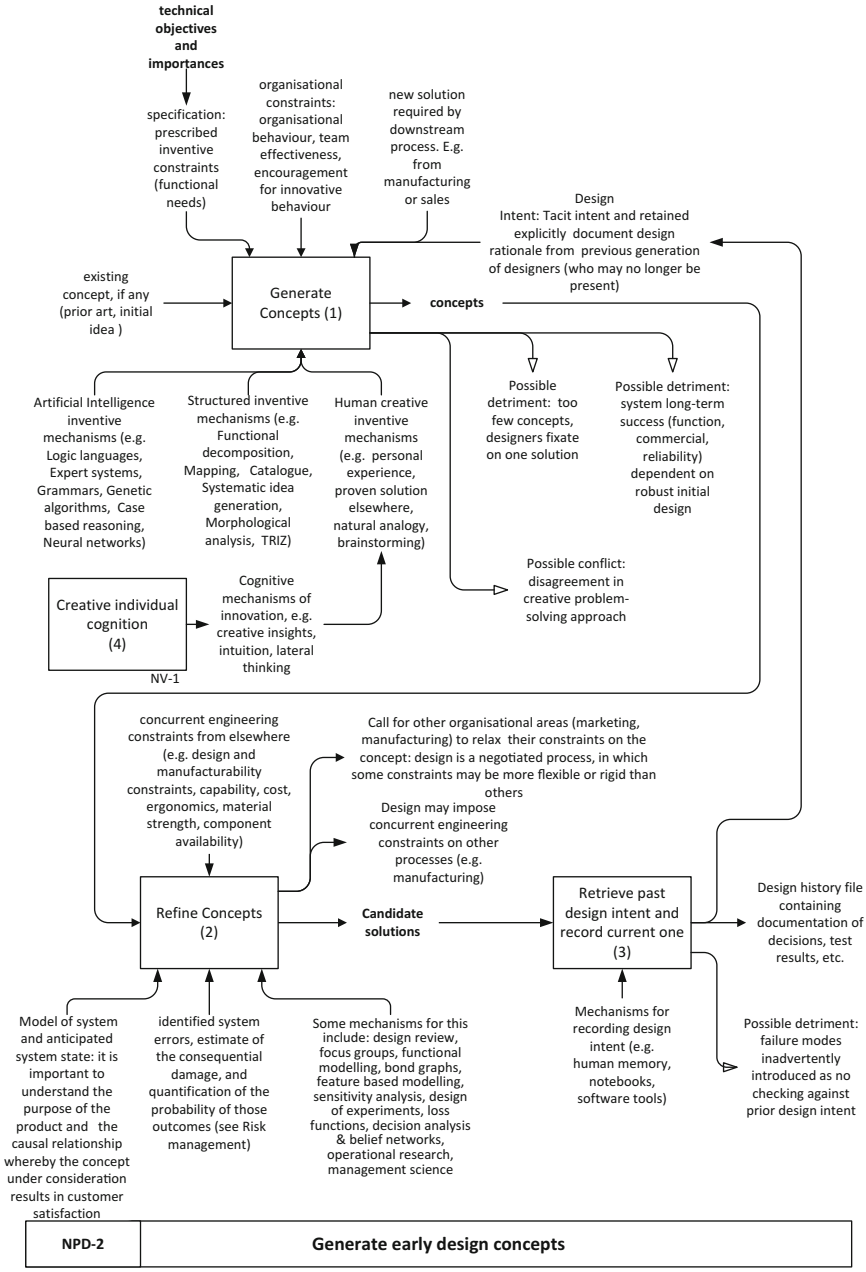


Fig. 45.15 Concept design is where the creative process is applied to give innovative ideas

In some cases research may be involved, either in the search for new ideas, new solutions, or to better understand the issues. Research and development (R&D) is a set of activities closely associated with NPD. R&D often precedes or overlaps the NPD processes. The main point of difference is that NPD, unlike research, necessarily seeks to produce an output artefact in the form of a product that the organisation can sell. By contrast, R&D is more focussed on finding new or improved processes, e.g. production mechanisms, usually by seeking to better understand the relationships of causality, i.e. what variables affect the outcome and how. R&D results in intellectual property (IP), and may also lead to NPD.

45.8 Assess Feasibility of Project

Not all new product ventures are likely to be successful, and since organisational resources are scarce it is necessary to evaluate the innovative new product idea, as shown in Fig. 45.16. Here we have nominally placed the assessment after concept design, but this is merely for descriptive purposes and the activity actually occurs at the outset, and periodically throughout the venture. The obvious main activity, where most people perhaps start, is determining the requirements specification for the product and developing the project plan (3). Other actions are to evaluate the idea for its intellectual property (IP) and economic prospects (1). There are many opportunities for stochastic and fuzzy decision analysis methods to be used here. Also necessary is to identify the technical ambiguities in the product and its production (2), since this has important implications for the amount of research required, and hence also consequences for risk and resources. The point being made here is that there are several activities, other than determining the product specification, that should be considered.

Next the temporal stages in the project may be identified, and the main work streams too (3). The project management methods are useful, particularly the work breakdown structure and scheduling approaches. Work streams have resource implications, and these need to be anticipated. In parallel and particularly important for products with an environmental attribute, is the activity of life cycle analysis (4), which is discussed in further detail later.

In this way the feasibility of the project is evaluated (6), both at the outset of the venture, and periodically later. In the latter case the project management monitoring methods, e.g. earned value, are useful tools for evaluating the temporal and financial adherence to the intended work streams. However in NPD there can be many changes in direction, especially if there are many ambiguities in the technical solution path requiring a large component of research. The PM monitoring methods (5) are good at comparing actual versus planned effort, but do not directly assess technological progress. Consequently NPD projects necessitate periodic reappraisal of the residual technical ambiguities, and the adaptation of the work plan. In some cases it is necessary to terminate the project and apply the organisational effort elsewhere (7).

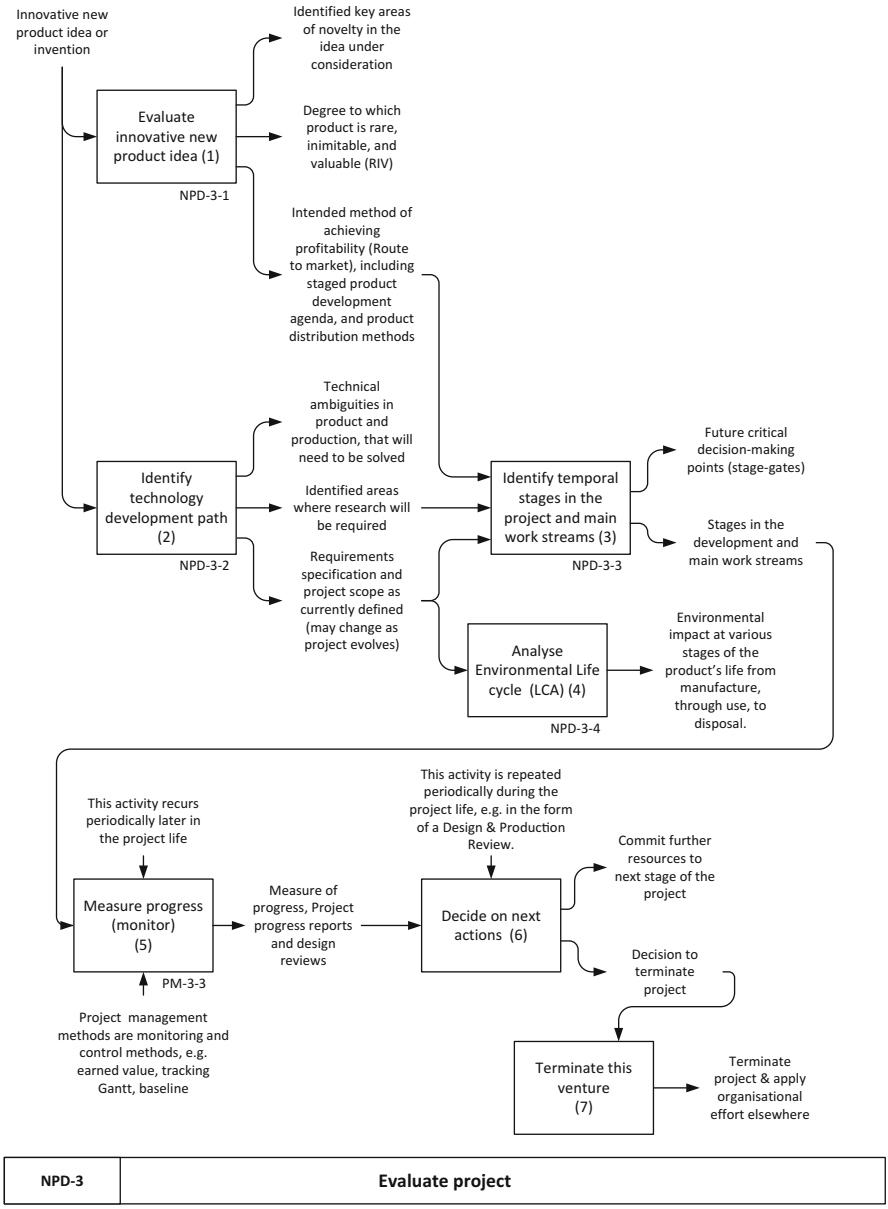


Fig. 45.16 NPD projects are evaluated for feasibility at the outset, and periodically throughout the life cycle

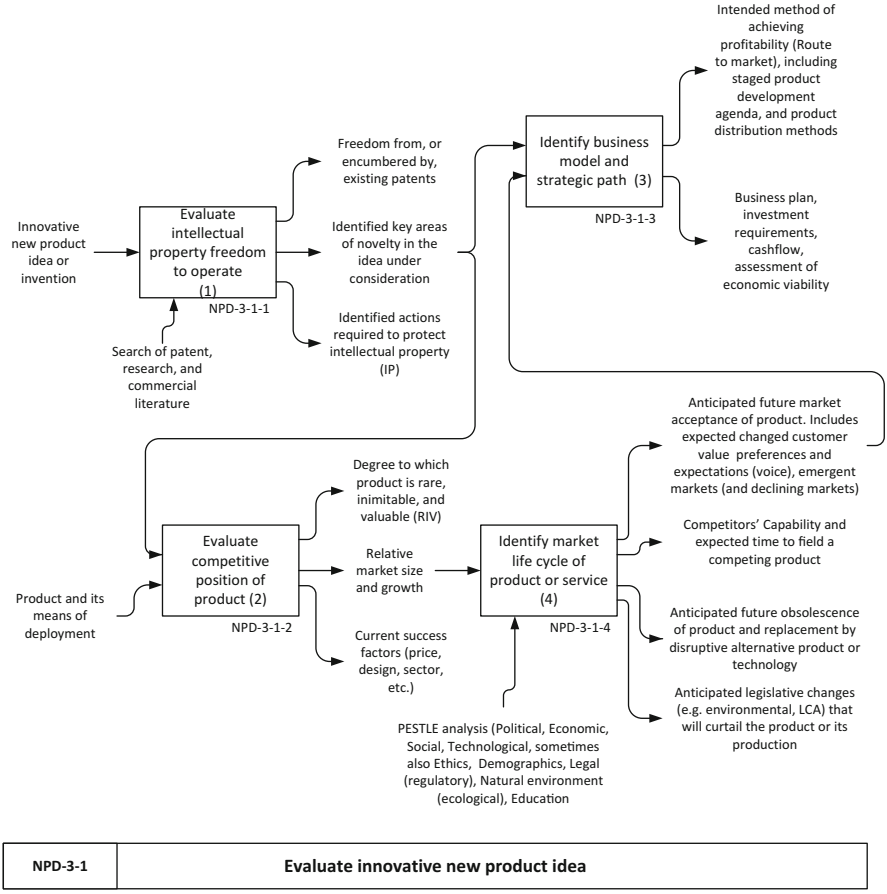


Fig. 45.17 New product ideas need to be evaluated against a number of criteria, to determine the economic attractiveness of taking them to market

There are particular challenges in the evaluation of an innovative new product idea, especially with regard to its economic prospects, and these are explored in Fig. 45.17. The first need is to evaluate the freedom to operate from the perspective of intellectual property (1). This involves searching patent literature and for evidence of prior disclosure. This is a complex activity because it is seldom that a new product is entirely novel in every attribute. Instead there is inevitably a degree to which it relies on existing principles, product features, or production mechanisms, all of which may be protected. Furthermore IP protections, such as patents, vary in their strength and defensibility. This means that it can be difficult to ascertain how strong the encumbrance may be, until it is tested in court. For entrepreneurial NPD, there is a need to identify the business model and the strategic path by which economic success may be achieved (3). This includes the intended method of developing the product for readiness for market, and the distribution method

(route to market). A staged product development agenda is typically applied, especially where the research and development ambiguities are large or the required capital investment is high. Included in here is the business plan, setting out the market opportunity, proposition of economic value, investment requirements, and the cash-flow. There is also a need to evaluate the competitive position of the product (2). This may be achieved by evaluating it according to the rare, inimitable and valuable framework (see above). For existing products there is valuable implicit information in the relative market size and growth. It is useful to consider what the current success factors might be. All products have a life in the market, and it is therefore necessary to consider the market life cycle of the product (4). Customer's needs change, competitors offer alternative products, disruptive new products or technologies emerge, and legislative changes may prevent production or sale of the product. For these and other reasons the product may have a shorter life in the market than its owner might hope, and this affects the economic case for its introduction. Agencies that fund innovative start-ups are particularly interested in an evaluation of the IP freedom to operate (1), for obvious reasons. They also invariably require a clear statement of the intended route to market (3).

A number of conventional project management methods have been mentioned above, and Fig. 45.18 puts these into context. It is not our intention to elaborate on these, but instead we simply point out that these methods may be adapted to serve parts of the NPD situation. The activities include definition of the scope and purpose (1), determination of work breakdown structure and tasks (2), estimation of durations (3) and determination of task interaction (4), allocation of resources (5), and the assembly of the work plan (6). Optimisation of this plan may be done (7) by analysing critical temporal (path), cost and technology attributes. Software is typically used to assist these processes (8), and frames what can and cannot be achieved for most practitioners.

These are the core activities of the general project management method, and this handbook provides much material elaborating on these methods. Specifically, the optimisation of project duration is addressed in Part I to II, resource allocation in Part III, V, and VIII, optimisation of cost in Part V and IX, and non-deterministic (e.g. stochastic and fuzzy methods) in Part XII, XIII, and XIV.

Attention now focusses on three of the activities that are particularly important for NPD: requirements analysis, environmental life cycle analysis, and project termination.

45.8.1 Requirements Analysis and Determination of Scope

An important early activity is to determine the project scope. The objective is to be able to list the project deliverables (objects and characteristics thereof), and the timelines. The scope is not the same as the work breakdown structure (WBS), though sometimes a WBS may be included. The scope defines the *ends* and the WBS defines the *means* (or at least the work that will be done to achieve those

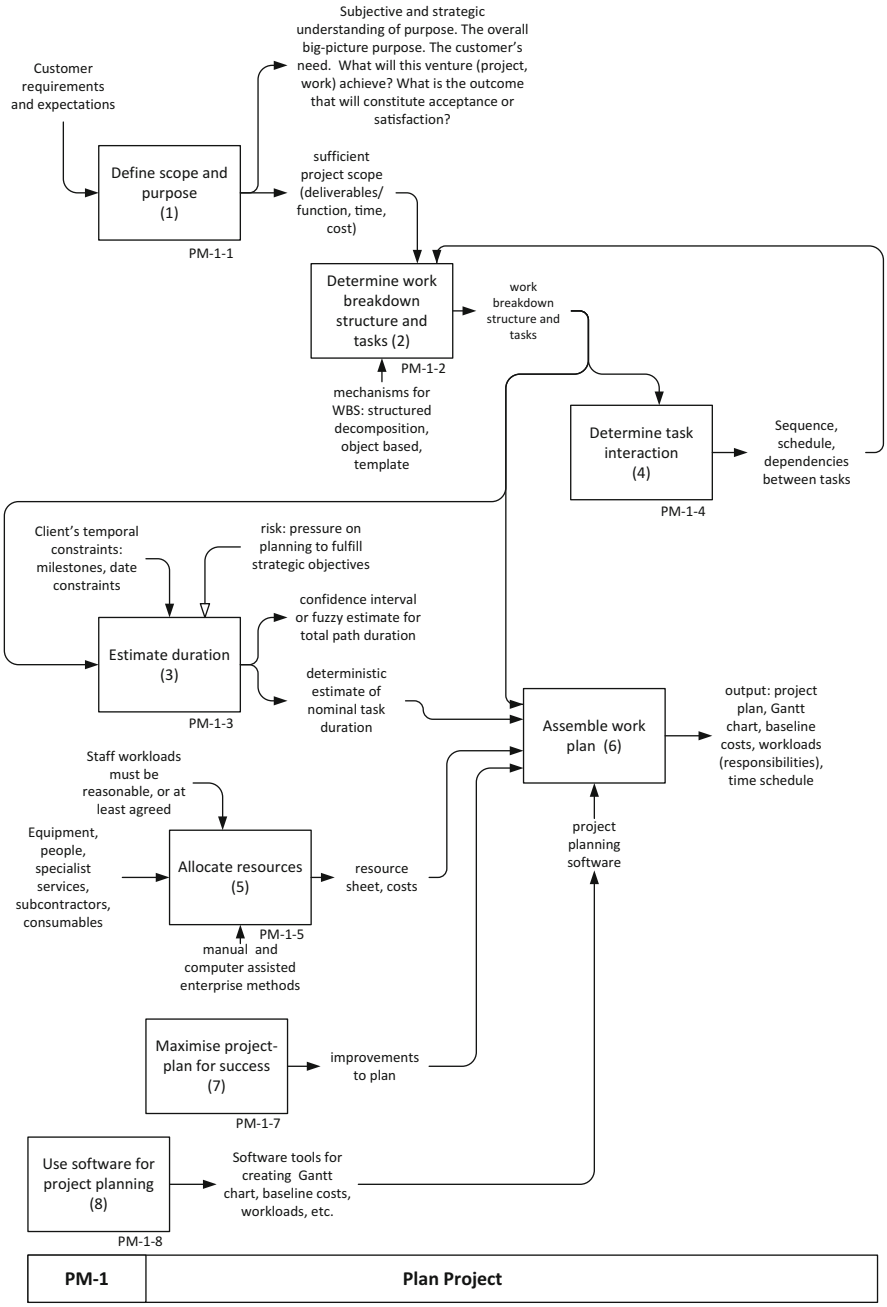


Fig. 45.18 Project management methods are applicable to NPD projects

ends). In engineering design the term *specification* is more often used, though it tends to be more limited to the functionality and engineering considerations. The dominant premise in the existing design theories is that the design problem needs to be fully specified before creative solutions can be found (Hubka 1987; Hubka and Eder 1988, 1996; Pahl and Beitz 1988). However that premise is questionable. A full and complete specification (scope) may work in many cases, but it does tend to under-emphasise the role of decision making. In particular, it is critically dependent on complete prior identification of the specification, something that is not always possible to achieve in practice (Frost 1999; Eder 1998). This is especially the case in complex design projects where the outcomes are qualitative and cover a range of domains, e.g. styling, mechanical function, electronics hardware, software. Specifications are sometimes difficult to set even in traditionally quantitative engineering areas, e.g. reliability, because of stochastic uncertainty (randomness). We take the perspective that it is only necessary to have a *sufficient* scope, and that a complete one is generally only obtainable for simple NPD problems. Inability to completely define a specification is usually not a problem for the creative design processes, providing that at least a *sufficient* specification is available. This is seldom a problem to define. The creative design processes, e.g. brainstorming, are not initially too focussed on the constraints anyway.

From the systems engineering perspective these activities are called *requirements analysis*. The focus is on defining the customer needs and the requirements (we covered the test requirements earlier). Obviously there are commonalities with the project management scope-setting process as evident in the PMBOK (PMI 2004), but SE has a wider field of view than merely the explicit activities and work-packages necessary to get to the deliverables. It is more explicitly aware of the whole product life, all the way through to field support and eventual disposal. SE takes a wide and temporally open-ended view, whereas PM is more task-oriented and time-terminated. By definition PM sees the project as a temporary endeavour that only extends to delivery of the outcomes.

The project evaluation phase is about making a decision on whether to proceed with the venture. The decision may be outright acceptance or rejection. Alternatively it could simply be a decision that not enough is known yet, and more work is required. The decision is made by considering the candidate solutions and the expected project effort required to get each one to market. That has to be weighted up in comparison to the organisation's constraints regarding technology, commercial (capital, production economics, market), and any time constraints (desired time to market, product life cycle). A common way of informing the decision is using feasibility analyses, or more generally risk assessment (ISO 31000 2009; ISO/IEC 31010 2009; AS/NZS 3931 1998; AS/NZS 4360 2004). The latter can look at not only the commercial dimension, but also the technical and the strategic. Project management methods are particularly useful at this point, as they permit the development cost to be estimated. Decisions at this point may involve senior executive or the board, depending on the resource implications. It is at this activity that the organisation makes a commitment to the project. Consequently this phase corresponds most closely to the charter stage in the PMBOK, though that term is

seldom used in NPD. See also Chaps. 49–51 of this handbook for risk management methods.

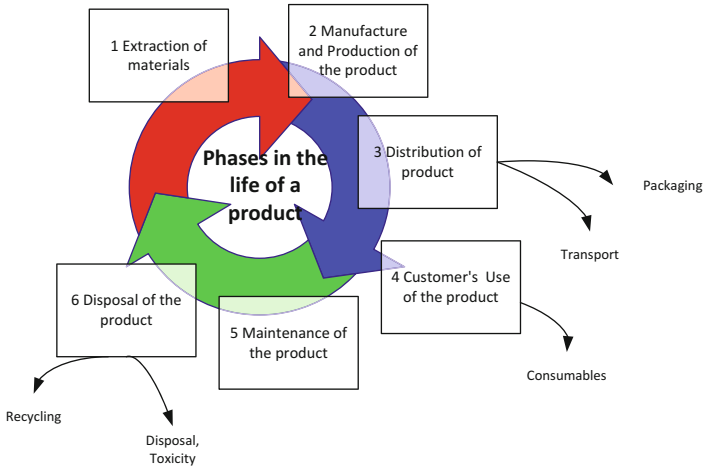
There are several other methodologies that are available to assist the design and decision-making processes. These include the theories of design science (Eder 1998; Hubka 1987; Hubka and Eder 1988, 1996; Pahl and Beitz 1988; Ullman 2001), decision theory (Clemen 1996; Herling et al. 1995), fuzzy theory, expert systems (Silverman 1994), neural networks, and management science. They all tend to construct the problem as rational selection among candidate solutions, based on quantitative determinants of value or likelihood or rule. However, people do not always structure their solution approaches this way. Instead, in complex situations they apparently instead strive for an adequate rather than a perfect solution (Simon 1981). Neither are managers somehow different: they too make adequate rather than purely rational decisions (Wagner 1991) This problem has been approached by others from the philosophical perspective, and as Simon observed, designers use "decision methods that look for good or satisfactory solutions, instead of optimal ones" (Simon 1981, p. 138). He termed this a process of *satisficing*.

Furthermore, this decision is almost never made by one individual acting alone. Instead, teams of people solve the technical problems and make decisions. These decisions cover a wide scope, including the selection of promising but uncertain solutions (and the abandonment of other candidates), and the specification of design parameters that have a significant downstream effect on production, organisational profitability, and customer satisfaction. With teams comes team behaviour, and with organisations comes politics. So the decision-making process is complex.

At the end of this phase the organisation has made a decision on which concept to take forward to detailed development. In rare cases more than one concept may be advanced, but this costs more. Assessment of feasibility occurs throughout the NPD life cycle, not only in the beginning. This is part of the project monitoring activities, at which the PM methods excel. In NPD these are typically augmented by design reviews.

45.8.2 Life Cycle Analysis

Historically product-design has focussed on functionality, delighting the customer, and production quality. To that list is now added the need to minimise the environmental impact of the use of the product. Life cycle analysis (LCA) is a systems engineering approach that involves evaluating everything that goes into and out of a product and its production and usage system, throughout the life of the product. When applied to environmental cases (ISO 14040), it is the environmental burden of the product that is under consideration. The life cycle for a product consists of several phases, see Fig. 45.19: Extraction of materials → Manufacture and Production of the part → Distribution of product (e.g. Packaging, Transport) → Customer's Use of the part (Consumables need to be considered too) → Maintenance of the product → Disposal of the product (including Recycling, Disposal, Toxicity in disposal).



NPD-3-4	Life cycle analysis

Fig. 45.19 Life cycle analysis examines the full life of the product, particularly the materials and their processing

The process of analysis involves the following steps. Each of these considers the full life of the product, i.e. all the phases identified above. This is sometimes called cradle-to-grave analysis, or other similar terms. The following is the ISO14040 process.

1. **Identify the goal and scope of the study**. This includes the product, the system boundaries, and the *impact categories* selected. The system boundary is the entire life of the product. The impact categories are typically within the broad areas of human health (human toxicity), ecological health effects (freshwater, marine and terrestrial ecotoxicity, pollution, acidification, eutrophication), climate effects (global warming, stratospheric ozone depletion, smog/photo-oxidants), and resource depletion. For example, the manufacture of a new product may have effects *(impacts*) in carbon value of energy usage (a type of resource depletion) and release wastes that are toxic in the aquatic environment (thus affecting ecological health).
2. **Inventory analysis:** This typically involves producing a flow chart showing the flow of energy and input raw materials into the system, and the discharges out of the system. These flows occur at different phases in the life of the product. The whole provenance chain should be included in the life cycle analysis, i.e. suppliers of suppliers are included. Data need to be collected of the actual or expected effects, which can be difficult.
3. **Impact assessment:** The different impacts are categorised and quantified. This is where the data are needed. Usually the environmental effects occur in multiple

dimensions, e.g. different types of waste with different effects on different parts of the ecological system. Thus there is often a need to normalise these and sum them into common equivalence units. This can be difficult to achieve. One option is to use subjective weights: this is very simple and easy, but also difficult to justify, and may be ill-advised if the LCA needs to be presented to external stakeholders, though it can be useful for an organisation's internal decision-making. A more robust option is to retain the different dimensions, and keep them all independent, as per the risk assessment method (ISO/IEC 31010 2009).

4. **Interpretation:** Here the results of the analysis are interpreted, and any significant issues are identified. These may be large environmental impacts that have simply never previously been noticed or connected to the product. In other cases it may be many small effects, which are individually insignificant but cumulatively important. Once these are identified, it may be possible to recommend treatments. Data are seldom perfect in this area, so it is important to determine the confidence in the results. How accurate really is the study under examination? How robust are the raw data in supporting the interpretation and recommendations? How sensitive would the outcomes be to small changes in the input assumptions? Where outcomes cannot be accurately foreseen, choices must be based on risk reduction and the precautionary principle (IPENZ 2005): that new risk be avoided in the absence of data.
5. **Implementation**: Organisations usually take a Pareto approach, by concentrating improvement efforts on the largest deficiency. They seek to at least reduce its consequences, since it will not always be possible to eliminate an effect altogether however preferable that may be. Once one effect is suitably reduced, the organisation can then put the resources into addressing the next biggest effect. Often this is accomplished in a project based manner.

45.8.3 Project Termination

Assessment may also result in a decision to close the project at any stage (4), and sometimes such checks are built into the project plan, hence a stage-gate approach (Hart et al. 2003; Howe et al. 2000). However the project closure/termination activities are generally done badly (Ceran and Dorman 1995). It is worth considering why, since this is an important consideration given the riskiness of all NPD projects especially those with a high content of research and development.

The preferred outcome is a project that provides the deliverables and is shut down in a controlled and progressive manner PMI (2008, 2004), leaving everyone content (Meredith and Mantel 1995). In other words, the project should satisfy all the stakeholders: the client, project organisation, sub-contractors, and the team members. If that is not possible then it is generally held to be desirable that a failing project be terminated cleanly. The problem is that a variety of partial failure modes are possible. For example a project can grind on interminably consuming a steady trickle of resources, or terminate abruptly by providing only the basic

deliverables and leaving loose ends. There are two aspects to decision-making regarding termination. One is to predict whether the project is likely to be successful, and this activity happens at the outset and at various decision-stages within the project life. The second is to respond to initiating hazard events.

As regards the first, the practitioner and research literatures assert that certain factors are critical (De 2001; Meredith and Mantel 1995; PMI 2006), though the evidence base for these assumptions is not always clear. For example a study (Balachandra and Brockhoff 1995) of research and development projects identified various factors as being related to project success: technical route (smoothness and probability of success); project champion. Others were associated with failure: deviations in cost schedule; chance events. However many other factors had an ambiguous contribution. Other research findings, also for R&D, are somewhat contradictory and suggest that technical feasibility and economic analysis are not strong components of decision-making, thus: "basic research projects are much less likely to be subjected to a formal economic analysis and are generally thought of as being 'strategic' investments" (Cook and Rizzuto 1989, p. 291). Hazard events that force the organisation to a decision whether to terminate or continue a project include (adapted from Meredith and Mantel 1995; Melymuka 2004):

- impending failure of project scope (e.g. either schedule, cost, function, or quality),
- lack of support from senior management (they have changed objectives, disinterest, or the external environment has changed),
- loss of motivation of project staff (they are stressed, pathological team relationships, despondent),
- loss of key capability (key technical skills lost, technology has been usurped by another project, intended technology solution is non-feasible, funding is drying up),
- loss of political support for project, e.g. loss of sponsor, project champion.

Much of the literature states that termination *should* be made on rational economic criteria (Cook and Rizzuto 1989; Dobson and Dorsey 1993; Melymuka 2004; Messica and Mehrez 2002; Rad and Levin 2005; Statman and Caldwell 1987). However the research is not entirely clear on what the termination decision is *actually* made on. Economic criteria do not feature in reality as much as might be expected. It seems that time, especially calendar time, is important at least in some cases (Dilts and Pence 2006). The *sunk cost bias* suggests that peoples' decisions about ongoing investment in a project are influenced by how much they have invested in it. The theory says that the more money and time they have invested, the more they are likely to want to persist with the project to completion, i.e. an *escalation of commitment bias* (Dilts and Pence 2006). However, the research evidence for this bias is ambiguous. Some research has found no support for the effect (Boehne and Paese 2000), while other studies have confirmed the existence of the bias (Garland 1990; Dilts and Pence 2006). Other research suggests the opposite effect, namely *de-escalation* of commitment as sunk cost increased (Garland et al. 1990). In some ways sunk *cost* might not be the best term, because it suggests

financial cost, whereas there is evidence that it can be *time* that really matters. Thus some research suggests it may be more accurate to say that decisions are based on how *close* the project is to completion (Boehne and Paese 2000; Conlon and Garland 1993; Paese 2000). This is consistent with the strong strategic motivation behind NPD projects: they are seen as vital for the competitive advantage of the organisation.

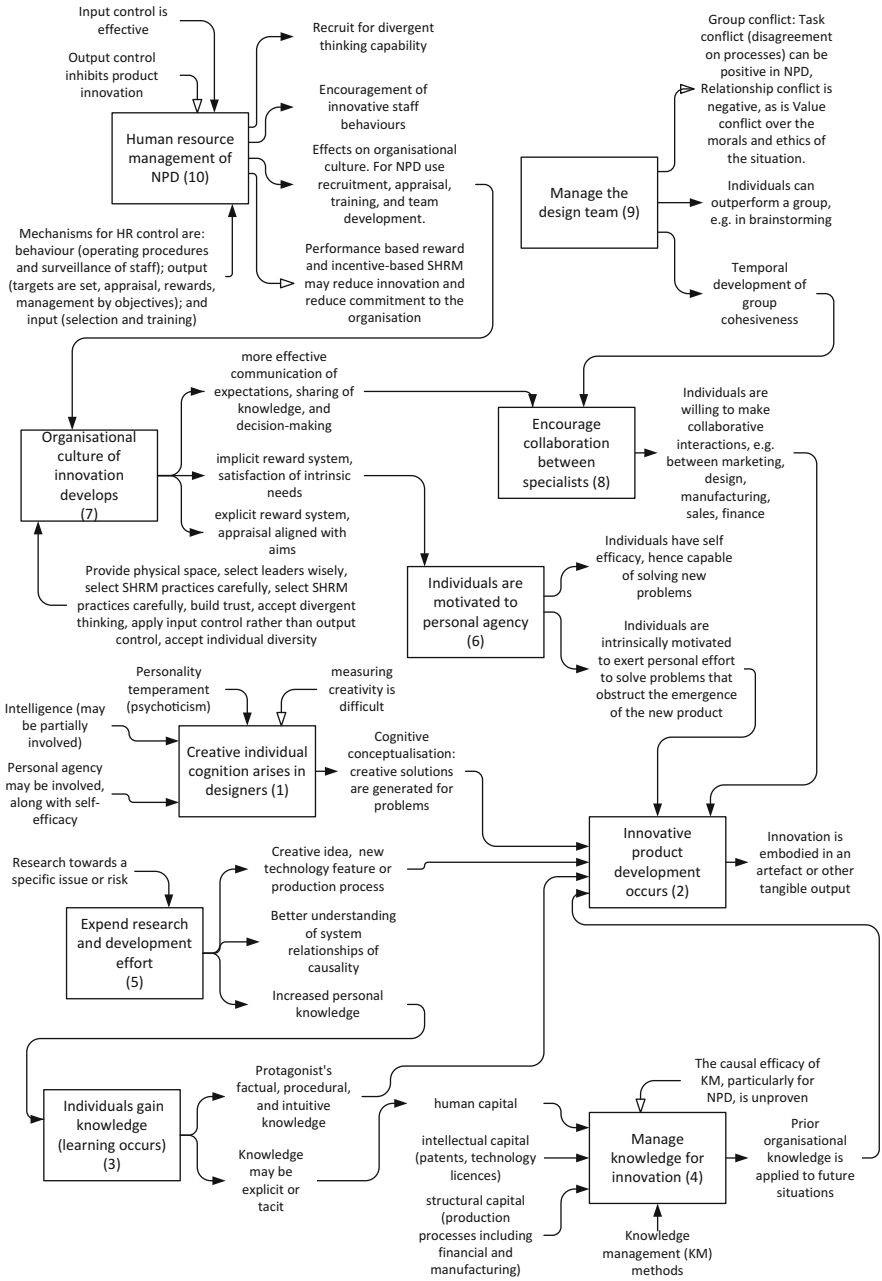
45.9 Other Design, Production and Economic Activities

There are several design and production activities identified in the top level system model (Fig. 45.4), that are beyond the present scope to detail. These include design and optimise the product, test and validate the design, finalise design and validate production capability, commission production system, production of the product and its distribution and marketing. Finally there is decommissioning of the product and production system. Most of these are relatively self-explanatory, and some have large literatures of their own. There is neither the space nor the need to elaborate them here. This system model has been extended to detailed design (Pons and Raine 2005), and commissioning (Lawry and Pons 2013).

That leaves two activities unaccounted for on the top level model. One is the need to create the organisational platform. This activity precedes and runs concurrently with the other activities. This is where the entrepreneurship occurs, and this is particularly prevalent in NPD start-ups. As discussed earlier, this type of organisation has acute problems with cashflow and therefore often builds organisational capability as and when needed. These tend to be small firms, hence small-to-medium enterprises (SMEs), and are often the focus of government encouragement of the innovation sector. Therefore it is not always safe to assume, as is common in most models of NPD, that the process is undertaken by a stable, mature, well-resourced firm. However we do not at this point need to open an entire discussion on NPD entrepreneurship, but instead we simply point out that the start-up nature of many NPD organisations causes constraints on the process. The last activity in this particular model is the management of the people. The NPD process is critically dependent on people. Consequently we close with a consideration of some of the issues involved.

45.10 Managing Innovation and the NPD Staff

NPD involves innovative thinking. Therefore part of the design manager's responsibility is managing the creativity of individuals, and the interaction of people. There are interesting human resource (HR) implications: Can managers selectively recruit for creativity? The model for this is shown in Fig. 45.20. The desired output is that human effort results in an innovative product being developed (2). This requires that individual designers show creativity (1) in the



NPD-11	Manage the NPD people and their creative capabilities

Fig. 45.20 Optimising the creative outputs of staff on NPD projects involves a complex set of organisational factors

application of their knowledge (3). Many organisations seek to deliberately record and manage their collective knowledge (4) to reuse it in future situations. Research and development effort (5) may need to be expended to address the residual technological ambiguities. Individuals need to be motivated to personal agency (6), which in the design situation often involves intrinsic motivation. This requires the development of an organisational culture (7). The design manager will also encourage collaboration between specialists (8), since significant NPD projects are complex and need a variety of skills. This means that the design team must be managed (9), and sufficient group cohesiveness created for effective collaboration. Human resource management (10) is therefore also involved, and in a more intimate way than in general project management where people are often merely considered resources.

45.10.1 Is There a Relationship Between Creativity and Intelligence?

The relationship between creativity and intelligence (IQ), despite being of interest to people for a long time, is unclear and probably depends on the measures used (Runco and Albert 1986; Singh 2006). Some authors are of the opinion that intelligence is not involved: that innovation is the combination of creativity and risk-taking (Byrd and Brown 2003, p. 7), e.g. "creativity is measured by originality" not intelligence (Byrd and Brown 2003, p. 13). However, this is not the majority view, and in this area where people can be so passionate it is prudent to instead examine the research evidence. Most studies have found either none or some (but not particularly strong) correlation between creativity and intelligence (Gardner 1988; Hauck and Thomas 1972; Kim 2005; Lindemann and Fullagar 1975; Nijsse 1975; Popescu-Neveanu and Facaoaru 1972; Richards 1976; Rushton 1990; Starr and Nicholl 1975). However others have found moderate correlation (Lopez et al. 1993; Meer and Stein 1955), or correlation only at low intelligence (Jaswal and Jerath 1991), or correlation consistent with IQ (Preckel et al. 2006; Sligh et al. 2005). Yet others reported significant correlations (Kazelskis et al. 1972; Singh et al. 1977; Virgolim 2006; Cheung 2008). A different perspective is that of neuroanatomy, where *creative innovation* (CI) is defined as "the ability to understand and express novel orderly relationships" (Heilman et al. 2003, p. 369). Those authors asserted that "a high level of general intelligence, domain-specific knowledge and special skills are necessary components of creativity, … [but] are not sufficient for CI." (p. 369). They suggested that the ability for divergent thinking was important, and associated these activities with particular anatomical regions.

Clearly the relationship, whatever it may be, is complex and not as simple as saying that intelligence results in creativity, or that intelligence is necessary for creativity. Based on current evidence all that can be concluded is that creativity and intelligence are related but independent characteristics, and that the relationship depends on how the variables are characterised and measured.

A second question is the relationship between creativity and innovation. It is widely held as an article of faith that creativity is necessary for innovation. Or as has been elegantly stated, "Innovation is the practical application of imagination" (Lin 2004, p. 13). The empirical evidence in this area is somewhat lacking, and few studies are available. One study showed that "knowledge innovation was significantly related to new product creativity" (Mingjie et al. 2008, p. 96). Further research is required before confident recommendations can be made to practitioners. For the present it is tentatively assumed that the concepts are closely linked: that creativity refers to cognitive conceptualisation processes, and innovation to embodiment in an artefact or other tangible output.

45.10.2 Is Creativity a Personality Variable?

It is commonly held that different temperaments have different creative abilities, whether for arts or engineering or other areas of creative endeavour. Creativity is at least partly a personality factor, as research has confirmed (Popescu-Neveanu and Facaoaru 1972). To what extent it is learnable is uncertain, but for the present this is presumed to be the case to some extent. In Eysenck's three factor model (extroversion-introversion, neuroticism-stability, and psychoticism) he has suggested a "connection between creativity and personality psychopathology (psychoticism)" (Eysenck 1995, p. 217). Indeed, one study found a moderately strong correlation between enjoyment of research and psychoticism (Rushton 1990). Again, stereotypes suggest that highly creative people can be temperamental and emotionally unstable (hence somewhat psychotic). In the five factor personality trait model (e.g. OCEAN, NEO-PI) (McAdams 1992; McCrae and Costa 1990, 1999; Saucier and Goldberg 2001), creativity is lumped together with intelligence in one factor. That does not necessarily mean that they are the same measure, but simply that there is a statistical similarity in the language used to describe creativity and intelligence. Thus part of the problem in finding relationships between creativity and intelligence is that we are not totally sure what we mean by each construct. On top of that, measuring creativity is difficult (Forteza 1974), and diverse tests exist (Singh 2006).

The management literature offers a bewildering variety of personality categorisations for innovation/creativity. Lacking, as they generally do, any supporting evidence, or underlying theory, it is prudent to consider them interesting ideas but of uncertain validity. Thus innovators have been categorised into *innovation-scientists*, ~*engineers*, ~*explorers*, and ~*astronomers* (Lin 2004). Or, another author has offered *isolators*, *stagnators*, and *navigators* (McAllum 2004). Somewhat related, though representing the personality of the organisation rather than individuals, the traditional Miles and Snow topology (e.g. Daft 2004) posits that organisations have four strategies for innovation: *innovator* (or "prospector" seeks opportunities and accepts risks), *defender* (steady efficiency, stability), *analyzer* (seeks opportunities and stability), and *reactor* (ad hoc reaction to external events).

There is no particular methodological difficulty in conceptualising different types of innovators, and creating personality tests that people can undertake to categorise themselves. Thus, "Tests have been developed that allow us to map individual innovation profiles" (Lin 2004, p. 23). However the ability to implement a categorisation scheme does not necessarily mean that the underlying theory for the typology is sound, or that the categorisation is helpful to people who undertake it, unless there is evidence to support it. Overall, there is an obvious lack of any evidence-based approach to typologies of personality-innovation, and much of the work in this area must be viewed sceptically.

Apart from the obvious issue of validating these various categorisations and weeding out those that are unreliable, there is much work to be done to develop an underpinning conceptual theory of why the different styles exist. Furthermore, there are many unasked questions in this area. For example, what about non-innovators? Are there different types of them too? What about precursor types of innovators, and how may they be developed into full innovators?

Returning to the five factor personality model, openness to experience has been associated with creativity, but conscientiousness not (Furnham et al. 2006). This has interesting implications because it suggests that creative people are open to experience, but not necessarily conscientious. Indeed, we have popular stereotypes ready to categorise these people, e.g. absent-minded professor or alternative-lifestyle artist, so perhaps this research is simply confirming something we already know intuitively.

On the current research evidence it would seem that creativity is partly associated with intelligence, but is distinguishable from it. Thus creativity does not necessarily require intelligence (depending on the construct for creativity, i.e. how one elects to define or measure it). Thus, one construct of creativity could be that it is a form of problem-solving (Gardner 1988). However, other constructs are possible. For example, some have constructed creativity as an internal drive for meaning, thus "our creations are the gifts of our selves to the world" (Byrd and Brown 2003, p. 15). Thus a personal agency may be involved, along with self-efficacy (Bandura 1989, 1997). From this perspective managers would foster creativity by providing intrinsic rewards that express appreciation and value for the ideas of staff. Some of those intrinsic rewards would presumably be provided by the organisational culture, therefore the manager would perhaps need to work at creating an appropriate culture. All this goes to show that the implications for managers will depend on how creativity is constructed in the first place.

45.10.3 Managing Knowledge for Innovation

That knowledge is essential for innovation is undisputed (Nonaka 1990, 1994). What is not so certain is how to best manage that knowledge for effective innovation. The field of knowledge management (KM) has arisen over the years, but there is still some distance to be covered before it can be considered to effectively and reliably

add value to the process of new product development (NPD). The issue is that the causal efficacy of KM, particularly for NPD, is unproven.

There are many concepts behind KM, and even implementation in information and computer technology, yet our understanding of implementation for NPD is still tenuous. The core concept is one of capturing and reusing the knowledge of individuals, thereby equipping the organisation for innovation. The diversity arises in the implementation of this. The types of knowledge that concern KM have been categorised into intellectual capital (patents, technology licences) (Brooking 1998), structural capital (production processes including financial and manufacturing), and human capital (people's professional skills) (Goh 2005). There has been substantial work done by Nonaka and others on the amplification of knowledge within organisations. However, there is an implicit premise within KM that increased knowledge somewhat automatically results in innovation, or at least that information enables innovation, thus: "In the information economy we have unlimited amounts of information, and so our potential and opportunity for innovation is unlimited as well" (Lin 2004, p. 11). How then can knowledge be reused to aid innovation? One of the processes, proposed by Nonaka, is that people who are given surplus ("redundant") information may be able to form other associations between that information and their own internal ("tacit") knowledge, thereby resulting in innovative ideas (Nonaka 1990, 1991, 1994). On the whole KM has several theories for increasing knowledge (see below), but does not have a lot to say about how knowledge is converted into innovation (Lee and Choi 2003). The process is obscure, and its efficacy more assumed than demonstrated. It is an area that has received little research attention. The underlying premise of causality, that information begets innovation, has not been tested.

Accepting a partition into tacit and explicit knowledge, Nonaka further postulated that knowledge was created by converting between tacit and explicit forms (four combinations of "entanglement"), in what he termed a spiral (Nonaka 1994). He held that shared experiences ("socialization") were essential for creating tacit knowledge, "externalization" used metaphor to express perspectives as tacit knowledge, "combination" was the assembly of explicit facts, and that conventional learning ("internalization") simply converted explicit knowledge to tacit, hence "SECI". From his perspective the individual's experiences, especially the variety thereof, were important because they "crystallized into a unique perspective" (p. 22) that would grow knowledge in the next part of the cycle. Nonaka, following Plato, asserted that knowledge was "justified true belief" (Nonaka 1994), i.e. was a personal belief that had been personally validated through experience. Nonaka's four SECI types of knowledge and the amplification spiral have become the dominant paradigm on knowledge creation, accepted as a given and widely adopted by many others as the basis for further theoretical work or practical deployment (Dyck et al. 2005; Hasan and Al-hawari 2003; Herschel et al. 2001; Johnson 2002; Meso and Smith 2000).

45.10.4 How Can We Assess Innovation?

Those who see value in classifying and storing organisational knowledge (Kakabadse et al. 2001) naturally also see value in measuring that knowledge. The knowledge is perceived to be *intellectual capital* (IC), and quite as important, if not more so, than physical assets and financial capital. However, there is no consensus on the measures to use (Marr et al. 2003). Although methods exist, such as balanced scorecard (Kaplan and Norton 1992), and human capital (Elias and Scarbrough 2004) they are difficult to calibrate. Furthermore, they tend to be preoccupied with quantifying artefacts, e.g. adding up quantities of patents as if they were of equal value, which misses the qualitative nature of innovation. Creating more holistic measures of an organisation's innovation is relatively straightforward. For example one such instrument (Tidd et al. 2001) consisted of a set of questions, grouped into categories. Those categories included strategic planning (external opportunities, threats), external linkages (partnerships, supply chain, customers), design (stage gate, cross-functional teams, new product development process), organisational culture (rewards, climate), and learning (knowledge management, training, knowledge sharing). At a superficial level, this type of approach is presumably useful in that it gives organisations and their leaders a tangible measure of innovation. Thus it can be for them a key performance indicator for which they can set target, exhort staff effort, and monitor progress. So these instruments may be useful as tools for leadership action. However they have some weaknesses, in that their external validity is unknown. The fundamental question is how we know whether the right questions have been included in the first place. The second weakness is that few if any of these tools have been validated by empirical evidence, so their reliability is unknown. Will a project that builds up the organisation's score on these measures *really* result in greater commercial innovation? The risk is that practitioners could apply the tools enthusiastically, but in an illusion of purposeful activity and end up being disappointed by the actual eventual outcomes. There is also the risk of too great an emphasis on managing the commodity of knowledge rather than managing the people (Smoliar 2003).

45.10.5 Implications for Human Resource Management of NPD

What then are the implications from an organisational perspective, particularly that of human resource management? The management literature makes much of the need for innovative staff behaviour, and seeks ways to make staff more innovative (Cheung 2008). Three forms of HR control are: *behaviour* (operating procedures and surveillance of staff); *output* (targets are set, appraisal, rewards, management by objectives); and *input* (selection and training) (Snell 1992). Some research has showed that *input* control can facilitate product innovation, and *output*

control inhibit it (Liao 2006). It was found that product innovation was associated by an emphasis on input control. The corollary was that innovation characterised by high task analysability (determinism of outcomes) was better managed by output or behaviour control. However one would have to question whether any process involving high task analysability was really innovative: certainly in the engineering design tradition this would be considered routine incremental design at best. Training (a type of input control) is commonly held to be effective for innovation and organisational performance in general, and indeed there is some support for this (Garcia 2005), however the relationship is not as certain as might appear.

In another study, a developmental culture was found to be associated with new product development (Lau and Ngo 2004). They further concluded that "an HR system which emphasizes training, performancebased reward, and team development is critical for creating a developmental culture" (p. 697). However, they also pointed out that "a simple relationship between HR systems or organizational culture and innovation outcomes should not be assumed" (p. 699), and that an "HR system alone may not be able to elicit innovation performance" (p. 699). HR practices such as selection, training and incentives have been found to be important for knowledge exchange (Collins and Smith 2006). In a longitudinal study it was found that "recruitment and selection, induction, appraisal and training—predict organizational innovation in products and production technology" (Shipton et al. 2005, p. 118). In fact there is some evidence that HR practices can have a larger effect on innovation than on organisational performance (Panayotopoulou and Papalexandris 2004), which is surprising as they are generally intended for the latter.

At this point strategic human resource management (SHRM) practices are commonly invoked. These seek to align the staff with the organisational purpose. Within those SHRM practices appraisal and performance-based pay feature strongly. However organisations generally need to be successful in multiple dimensions, e.g. productivity, short-term profitability, innovation, etc., and the SHRM practices are too blunt to be able to differentiate between the different outcomes. Furthermore, it is possible that SHRM is quite the wrong way to go about developing innovation. That contrary finding emerged from a longitudinal study which suggested that a focus on the strategic role of HR does not give sustainable competitive advantage (Hailey et al. 2005). SHRM was indeed found to result in financial performance, but at the cost of staff dissatisfaction and low commitment to the organisation. They concluded that SHRM "does not necessarily enhance the value of the firm's human capital" (p. 63). Thus it is possible that SHRM can be effective in the short term, but may be too cold, harsh, and incentivised, to create the type of organisational culture necessary to sustain tasks like innovation that would seem to require high motivation and commitment. Another longitudinal study found that appraisal linked to remuneration is a significant impediment to innovation (Shipton et al. 2005). Thus core SHRM practices (appraisal, remuneration) which may be useful in optimising some organisational outcomes, appear to be counter-productive to other outcomes like innovation. This paradox has been identified by others regarding management of diversity (Bassett-Jones 2005).

45.10.6 Managing the NPD Managers

The discussion up to here has been on managing the creativity of staff. How does one get managers who are good at managing NPD projects? This is an under-explored area of NPD, and though there is some research (Clift and Vandenbosch 1999; Lewis et al. 2002; Swink 2005; Barczak and Wilemon 2003; Jeffrey et al. 2003), it is still difficult to be definitive about the implications for practitioners. Since it has been identified that unintended consequences of creativity can include low conscientiousness and high psychoticism, one has to question whether a strategy of appointing many creative managers is necessarily the best approach for every organisation. It can be anticipated that a cadre of creative managers and leaders would result in a dynamic and quirky organisational culture. Indeed, the management literature is full of case studies about highly innovative companies and their organisational cultures.

Thus it is possible that a quirky culture is a necessary unintended consequence of highly creative managers and staff. However, this is not a possibility that the literature has considered in any depth. The common wisdom is that all companies should strive for maximum creativity, on the premise that it will result in innovative products and thus organisational success. But if this should have unintended consequences on the culture, then is this still the best strategy? The literature does not address this in detail. Perhaps it might be better to nurture different types of creativity in different parts of the organisation?

It is also prudent to consider also how the creativity of the leader, e.g. chief executive officer (CEO) may affect the subordinates. In this regards there is not a great amount of research, but the little that exists is not encouraging in that it suggests that leaders with intelligence and creativity tended, under stress, to be associated with reduced group performance: they inhibited other members (Gibson et al. 1993).

Putting those caveats aside, and assuming that an organisation seeks to increase the creativity of its staff and managers, whether all or some, then what are the means to achieve this? If the general consensus is true in stating that creativity is a personality characteristic, then it may respond to nurturing and development, but ultimately could be limited. It may be necessary to also recruit staff who already have the required high levels of creativity. How can this be done? Research has found that measures of divergent thinking adequately predict manager creativity (Scratchley and Hakstian 2001). This suggests a selection method for those organisations who wish to optimise manager creativity.

A manager who wishes to encourage innovation would therefore do well to create workplace relationships of trust by modelling trustworthy behaviour towards subordinates, providing intrinsic motivation opportunities for staff, and ensuring trusting relationships within teams (may require active management). In addition, the manager would want to develop the skills and experiences of staff, by training, project teams, temporary assignments, and hiring people who already had those skills. Similar recommendations for specific environments, e.g. research, have also been identified by other authors (Graversen et al. 2005).

A discussion of NPD would be incomplete without mention of reputation and ethical behaviour of the organisation. Products are very strongly part of the brand of the organisation. The greater the competitive advantage conferred by the product (through being rare, valuable and inimitable) the more important organisational reputation becomes. Consequently the ethical behaviour of the organisation becomes important. In this context ethics refers to the decisions made during the NPD process. Many are the product-reliability issues that have been caused by an expedient decision in design or manufacturing. Not only do such failures cause reputational damage, but they can also expose the organisation to product liability claims. The financial penalties in these cases can far outweigh the savings intended by the original short-cut. Managers need wisdom in how they make decisions when there are conflicting priorities. Wisdom in this context has been described as the values underpinning the choices people make in the application of their intelligence and creativity toward a common good (Sternberg 2004). This is a job for leaders, and it is difficult since the NPD process is full of conflicting priorities, and organisations have both altruism and selfishness in their mission statements.

45.10.7 Teams: Temporal Development

One of the perspectives that is almost totally lacking in the conventional project management paradigm is the concept of teams, specifically the temporal development of group cohesiveness. The general PM approach is to consider people as resources, hence homogenous units of labour that can be applied against tasks in the WBS as the project manager determines. Hence we have the typical Gantt-chart approach to managing the people. This may work adequately for construction projects where the staffing comprises contractors who come and go. When such people come onto the project they can safely be assumed to immediately bring to bear a high level of competence and productivity. These are not safe assumptions in the NPD arena.

NPD is characterised by being done in teams of closely-interacting people. They develop their skills co-dependently over time, and the resulting attitudes towards personal interaction are cemented in the organisational culture that they create. This culture and the nature of the personal interactions is an important component in the success of NPD teams. Consequently such teams cannot be considered merely a summation of independent agents, as conventional project management would view the situation.

Perhaps the most popular theory of temporal (time) development is Bruce Tuckman's model of group development (Tuckman 1965): forming, storming, norming, performing (FSNP). He proposed a classification system with three independent dimensions: *setting* (therapy, training, natural, laboratory), *realm* (task or interpersonal/group structure), and four temporal *stages*, and then showed how the literature of the time slotted into those categories. His original model was a complex one, with a separate strand of activities for each *realm.* This model

has subsequently been simplified over the years. This has become the dominant paradigm in social-work and management. It has been influential with practitioners and researchers, and continues to attract attention (e.g. Sweet and Michaelsen 2007). The model has formed the underlying premise for other research (Vroman and Kovacich 2002; Farrell et al. 2001; Aubert and Kelsey 2003) and inspired replication of the categorisation method (e.g. Cassidy 2007). The current models of group development are based on Tuckman's model, e.g. the integrated model proposes that the stages are (1) dependency and inclusion, (2) counter-dependency and fight, (3) trust and structure, (4) work, and (5) termination (Wheelan et al. 2003). Others offer: (1) Identify acceptable behaviours (member anxiety), (2) conflict and eventual stability (power), (3) development of trust, (4) task orientation, and (5) termination (Arrow et al. 2004). Models of group development are widely accepted (Wheelan et al. 2003, p. 225), but nonetheless the processes are imperfectly understood.

Its popularity notwithstanding, there are significant theoretical gaps in the Tuckman model. Implicit in the categorisation was the premise that social and task elements of group-interaction are separable (independent), and while he acknowledged that it was a fuzzy distinction he kept it nonetheless to maintain the continuity. In other words the *realm* dimension was a somewhat artificial construct. How, if at all, those two strands were related was not clarified, and indeed the model has subsequently been better-known by the colloquial names and the *realm* dichotomy has not survived into practitioner usage and is rarely mentioned in research papers.

As a conceptual-model the lack-of-connectedness between the elements is apparent: one expects to a see variables being passed from one stage to the other, mechanisms identified for the execution of the stages, and a description of those factors that impeded or enhanced the completion of the stage. By mechanisms we refer to underlying processes for achieving the purported outcomes. The model fails to identify the mechanisms that bring about the various stages, e.g. how does stage 2 conflict progress to stage 3 cohesion given the obvious differences in those states? The model is a summation of observable outcomes rather than a theory of how those outcomes arise. This also makes it difficult to test the model, because of a lack of reliable measures (Farrell et al. 2001). The model is not a conceptual model, since it does not propose any causal theory, but rather a categorisation—which was all that Tuckman claimed for it. Tuckman anticipated that the main value of the model would be the "derivation of many specific hypotheses relating independent variables to the sequence of temporal change" (p. 398), but that has not occurred as expected, nor have there been the expected major modifications to the model (bar the subsequent addition of the minor phase of "adjourning").

The setting considered by Tuckman, and invariably adopted by all subsequent theorists, is one of a team of peers. However the reality is that most teams, at least in commercial and industrial practice, have some internal authority structures. There is delegation of power and designation of leadership from still higher authorities in the organisation. There are names, which are in common use, for these people: "Team Leader", "Design Manager". The existing Tuckman-derivative theories totally ignore this important dynamic.

The bigger question, of which Tuckman investigated but a fraction, is the development of groups over time: the *temporal* perspective (Arrow et al. 2004). Temporal is one of many ways to examine groups (Chang et al. 2006). Even within the temporal perspective there are multiple ways of conceiving time, including social (e.g. events, seasons, epochs) (Ballard et al. 2008), resource (scarcity of time, pressure, time allocation), level (individual, group, organisation), life cycle (purposeful, cyclical, sequential) (Arrow et al. 2004). Thus Tuckman provided what can be identified as a *sequential stage model*. That has been a good base on which many subsequent works have built, but the model is not a theory and cannot further inform practitioners or researchers. It will eventually be necessary to develop more sophisticated theories of how groups develop and how practitioners can make them more effective. Therein lies an opportunity for further research: to determine how the temporal team development process works in NPD teams, and if possible enhance the process.

45.10.8 Do Teams Actually Work?

The dominant way of thinking is that teams are better than individual effort. There seems little doubt that in situations where the tasks are complex and beyond the ability of a single person, e.g. NPD, then teams are necessary at least for the division of labour. Whether teams, even in these situations, deliver higher reliability or greater cost-effectiveness than individuals, is less certain. However in the more general business processes that exist within organisations, evidence suggests teams are less effective than commonly perceived. Individuals are often better at certain tasks e.g. creativity and decision-making (Allen and Hecht 2004). In fact, overall the research literature does not confirm that teams have any great advantage over individuals, and that sometimes a single individual can outperform a group. Evidence shows that for certain tasks such as brainstorming, individuals are better than teams (Mullen et al. 1991; Oxley et al. 1996), for example there is "strong evidence that interacting groups actually generate far fewer or, at best, the same number of ideas, as compared with the combined efforts of several individuals working alone" (Allen and Hecht 2004, p. 440). Training can help compensate to some degree, since research has shown that "groups with a highly trained facilitator may achieve the productivity of nominal groups [individuals]" (Oxley et al. 1996, p. 644). However this result is unsurprising given that the treatment (specialist facilitator) was available to the group and not the control (individuals). To turn the case around, it appears that individuals perform as well as, if not better, than groups, even groups with trained facilitators.

The reasons theorised for the inferior performance of teams at brainstorming include evaluation-apprehension ("fears of being negatively evaluated by the other group members" (Oxley et al. 1996, p. 634), free-riding ("group members may feel that their efforts are dispensable and not work to their full potential" p. 635), production blocking ("inability of group members to state ideas freely and without

interruption" p. 635), and social influence ("decrease in performance is subsequently maintained because individuals match their performance to that of the other group members" p. 635). These theories have not been validated. While groups potentially have more resources, and more diversity of resources, than an individual or set of individuals, they are not always more efficient in their usage of those resources. In particular groups loaf, become polarised, seek consensus rather than novel ideas, and sustain an "illusion of group productivity" (Paulus and Van der Zee 2004, p. 476). The implications for practitioners are that if quantity and quality of ideas is required (which is usually the overt purpose of brainstorming), then individuals are generally better, especially compared to groups led by untrained facilitators. If groups are used then facilitators seem advisable, and the attributes of such people might be as follows, though it is worth noting that those authors could not identify precisely which attributes were the cause: "more experienced, … training in recognizing ideas and in keeping members focused, … reintroduce[d] topics that were not fully discussed" (Oxley et al. 1996, p. 644).

It is possible that there might be other secondary benefits to brainstorming, perhaps socialisation effects, but if so these are not evident in the existing research. This suggests several areas where research is required. Why do teams not work reliably? Why do people typically believe they succeed when the evidence suggests otherwise (for consideration of this question, see Allen and Hecht 2004)? In what situations do teams succeed? Are there other structures for work that are more effective and in what situations? How do within-team behaviours affect performance (Cordery 2004)? The need from a research and practitioner perspective is therefore to better understand the way teams succeed (fail), so as to more effectively deploy them (or alternative work structures) (Allen and Hecht 2004).

45.10.9 Conflict

The phase models suggest that conflict arises not at the beginning but some way into the group life, the storming stage (Tuckman 1965). Those models perceive the conflict as a process of clarifying the group's objectives, a positive force for making sure the purpose is well designed, well understood by everyone, and that the processes for achieving it are accepted by the whole group. They state that conflict is necessary for trust to develop, and is a natural part of developing a common set of goals (Wheelan et al. 2003). Likewise some have suggested that groups operate, or should operate, with well-defined expectations of decision-making: "unambiguous, consensually approved expectations about decision making, about each member's rights and responsibilities in his or her professional role, and about the procedures for working together" (Farrell et al. 2001). The difficulty is the naivety of such positions regarding the known negative consequences of conflict. They fail to identify the mechanisms whereby the negativity of conflict is transformed into positive outcomes.

A more insightful perspective is provided by the communication and psychology perspective, whereby conflict is the expressed struggle of people over interference from each other in the pursuit of incompatible individual goals (Rothwell 2007). The conflict is expressed in behaviour. Conflict can be positive: encouraging creative problem solving; initiating good changes; balance power; enhance group cohesiveness (Rothwell 2007). In this mode the conflict is based on seeking mutual satisfaction, and this requires communication that is cooperative. Conflict can also be destructive: aggressive and domineering behaviour; deliberate hurting of adversary; selfish personal gain. This type of conflict reduces group productivity, reduces member satisfaction, increases stress, and results in further behavioural consequences. Another perspective is that conflict may be over the task, relationship, or values:

- **Task conflict** is over processes. When the processes are routine (little variability possible) then the conflict tends to have negative effects, whereas conflict over non-routine (high uncertainty) tasks can have positive effects (Rothwell 2007).
- **Relationship conflict** is over personality styles. In these situations avoidance is often used as a solution.
- **Value conflict** is over the morals and ethics of the situation. These are expressed in strongly-held beliefs. Examples are homeland, racial, and origin-of-life beliefs. People do not readily compromise their beliefs, so that solution is unavailable. In fact there are no easy solutions other than on-going competition or plain avoidance. This type of conflict can cause break-away organisations.

Various strategies have been proposed for solving conflict. A popular model is the win-win approach, which seeks to satisfy both parties, as opposed to win-lose where one is satisfied but not the other, or lose-lose where neither are satisfied. The win-lose model might be suitable for negotiation type conflicts where the outcomes can be represented in binary states. However it does not capture the complexity of general conflict situations where there may be more outcomes and states other than fully satisfied-fully dissatisfied. Consequently the more general approach has several solutions:

- Collaboration: working together to find a solution that generally satisfies everyone. This corresponds to the win-win solution.
- Accommodation: giving up own needs and agreeing with the antagonist. This is the surrender solution, corresponding to win-lose.
- Compromise: mutual sacrifice, both parties give up some of their needs to find a solution that partially satisfies both.
- Avoidance: ignoring the conflict and not seeking any solution.
- Competing: seeking to dominate the antagonist, and to satisfy own needs even if at the cost of denying the needs of the antagonist. i.e. may be destructive. Both parties may compete: it is not necessary that the other moves to accommodation.

Late-onset conflict arises because opposition slowly builds. People resent the fact that they have had to take a conciliatory approach and suppress their own needs for the benefit of the aggressor and his faction. When the time is right and the antagonist

is weak, they launch a counter-attack. Possible solutions for managers: Change group roles; Re-plan the tasks and work breakdown structure for the remaining tasks in the project; Recalibrate expectations within the team, by voicing personal concerns and expressing own hopes and fears for the project.

45.10.10 Creation of a Culture of Innovation

The opinions of many authors strongly suggest that organisational culture encourages collaboration and learning (Mårtensson 2000). Innovation is a complex activity and managers should not expect to control it with any determinism. This is because many of the activities cannot simply be mandated to occur. Indeed, it is likely that an authoritarian management style will do more harm than good, at least to innovation. Nonetheless, there are many ways in which managers can affect the innovation process (Jiménez-Jiménez and Sanz-Valle 2005), even if indirectly. The following is not intended to be an exhaustive list, but simply identifies and elaborates on several features that appear to be more important.

45.10.10.1 Provide Physical Space

Nonaka asserted that self-organising teams needed to have a place (called "ba" in Japanese) in which to discuss their perspectives and resolve their conflicts, if knowledge was to be successfully amplified (Nonaka 1994). He implied that the process of knowledge creation in Japanese culture involved strong socialisation driven by the "sharing of mental and physical rhythm" (Nonaka 1994). He felt that dialogue was essential for the social process of knowledge creation, and asserted the value of face-to-face communication, free and candid expression, and redundancy of information (people are given more information than they need because some of the excess may generate new associations).

45.10.10.2 Select Leaders Wisely

It is likely that certain personalities are better than others at managing innovation. The leader's personality sets the relationships at the top management team level and propagates into the rest of the organisation to profoundly affect the overall culture. Likewise the leader's priorities, which can very easily be for short-term gain (George 2003) at the expense of future developments, propagate as goals into the organisation and can hinder or advance innovation. Desirable leader attributes for innovation might be openness, lack of ego-defences, participative style (non-authoritarian) (Jeffrey et al. 2003), along with intelligence, creativity, and divergent thinking (Scratchley and Hakstian 2001). Task-relevant knowledge is probably valuable in providing the right resources to subordinates. A board should

expect that appointment of an egotistical, narcissist chief executive officer (CEO) could put at risk the development of a trusting positive learning culture within the organisation. Likewise, a CEO who provides strong motivational rewards that are contingent on successful performance of strategic business units, should expect that the same could provide incentives for internecine intraorganisational competition that damages staff collaboration.

45.10.10.3 Select SHRM Practices Carefully

Unfortunately many strategic human resource management incentives that are intended to motivate staff and align them with the organisational purposes can have unintended consequences. These incentives may include internal competition between units, performance based pay, appraisals, etc., and are typical of SHRM high-performance practices. Thus workgroup competition suppresses knowledge sharing (Burgess 2005), and appraisal linked to remuneration impedes innovation (Shipton et al. 2005). While SHRM may indeed motivate staff to work harder for the organisational success, some of its practices do so by appealing to selfish needs at the individual and work-group level. Consequently SHRM output control (Liao 2006; Garcia 2005) may increase some aspects of organisational success, at least in the short-term, but risks destroying the contribution that knowledge sharing can make to long-term success. This should not surprise us, as there is no reason to believe that one perfect management strategy exists that will simultaneously optimise organisational performance for every dimension in which that can be measured. Instead managers must seek to obtain a balanced basket of organisational outcomes, and dynamically use components of various strategies to achieve this. Sometimes the causality reverses: innovation shapes the HR practices (Jiménez-Jiménez and Sanz-Valle 2005). Organisations that have been innovative are likely to adjust their HR practices to sustain those outcomes.

45.10.10.4 Encourage Knowledge Sharing

How can an organisation create a climate where people are motivated to share knowledge? Solutions might include: provide extrinsic rewards and recognition for sharing (Carmen and Cano 2006), reduce internal competition, build identification with whole organisation not just work group, promote on competence rather than influence, and build trust with the organisation. The softer HR practices are better (Collins and Smith 2006): “firms that employ commitmentbased HR practices [selection, training and development, and pay incentives] are associated with organizational climates containing higher levels of trust, cooperation [and knowledge sharing]” (p. 555).

45.10.10.5 Build Trust

If there is one thing that a leader can do to enhance knowledge within an organisation, it is creating a culture of trust (Lee and Choi 2003; Mårtensson 2000; Follon 1998). The trust in this case is peer-to-peer trust, not necessarily trust of subordinates in the leader though that is also important. To achieve this managers may model trustworthy behaviour towards subordinates, provide intrinsic motivation opportunities for staff, provide visible support, and ensure trusting relationships within teams. If trust is lacking, then collaboration cannot thrive (Nonaka 1994), because trust provides the willingness to lower ego defences and risk damage by the other. The organisational culture provides a common set of values (Meso et al. 2002) that support trust (Kakabadse et al. 2001). Mutual trust has been held to be essential for self-organising teams (also called "communities of practice") to be effective at innovation (Nonaka 1994). Collaboration results in greater individual learning (Lee and Choi 2003) as well as immediate organisational benefits of greater work effectiveness and efficiency (potentially better *and* faster projects).

45.10.10.6 Accept Divergent Thinking

It is within the manager's control to provide an environment where creative ideas are not scorned, risks are genuinely embraced along with opportunities, and the pressure on staff is not so great that creativity is suppressed.

45.10.10.7 Apply Input Control Rather Than Output Control

Input control (selection and training) can facilitate product innovation, and output control (targets are set, appraisal, rewards, management by objectives) inhibit it (Liao 2006; Garcia 2005; Shipton et al. 2005). Managers may also develop the skills and experiences of staff, by training, cross functional project teams, diversity of work experiences, temporary assignments, etc.

45.10.10.8 Accept Individual Diversity

In the five factor personality model, openness to experience has been associated with creativity, but conscientiousness not (Furnham et al. 2006). This has interesting implications because it suggests that creative people are open to experience, but not necessarily conscientious. Likewise Eysenck's three factor model links creativity to psychopathology (psychoticism) (Rushton 1990; Eysenck 1995). How will these people be managed if conscientiousness is not their strong point and they may tend towards psychoticism? Some simple solutions can be anticipated: provide a mix of personalities in a team, and manage the team to prevent excess conflict arising over

different expectations of conscientiousness. Factors such as conscientiousness are practically universally appreciated across engineering in whatever culture, but not so agreeableness (Norman 1963; McCrae and Costa 1985). So it may be necessary to accept that diversity and a level of disagreement is a necessary part of creativity.

45.11 Discussion

As this chapter has shown, the management of innovative NPD projects involves solving a complex problem in scheduling, with a long timeframe over the product life cycle. There is also the challenge of indeterminism in the work breakdown structure, caused by decisions that cannot be made until future information becomes available. Adding further complexity is the need to manage the people and the culture within which they work.

45.11.1 Outcomes

A system engineering model has been presented for the NPD process. There are other models of design, and separately of the project management process, but the present one is novel in that it explicitly includes the broader organisational features involved in the *management* of NPD projects. At a superficial level it is straight forward to apply PM methods, such as the Gantt chart, to the NPD process. Indeed it is useful to do so. However the issues with managing NPD run much deeper, being a consequence of the complexity of the process. In this chapter we have shown how the systems engineering approach is a useful complementary tool. SE provides a mechanism to identify the *context* in which the NPD process occurs. These correspond approximately to the *integration* activities of the PM method. SE lacks the detailed process control methods inherent in the PM method. However it does bring the bigger picture into focus, and this is valuable because the issues for NPD are not so much in the scheduling but in the other complex interactions that occur between people in activities that in all likelihood will not even be on the WBS.

The model captures many of the existing concepts in the literature, and using the systems perspective it also introduces new ideas and a different way of looking at NPD projects. The model we have presented here identifies the main NPD activities. Naturally there are many more activities that could be included, either in deeper models or at the top level. Examples are distributed NPD projects (conducted across multiple workplaces) (Pratim Ghosh and Chandy Varghese 2004), procurement/outsourcing, product families (Tatikonda 1999).

45.11.2 Contrary Perspective

The general premise throughout the literatures in engineering and management is that innovation and design are crucial for organisational success. Lack of innovation is thus widely perceived pejoratively. That innovation could occur in the form of product, service, or process. This is the perspective that we have taken here. However to be fair we need to present the contrary perspective, which is that innovation (at least product innovation) is not necessarily critical for success in mature industries (Mavondo et al. 2005). Innovation carries risks and costs, which may be too great for these industries compared to the anticipated benefits. Those same authors found that process and administrative innovation were nonetheless related to performance, even for mature industries.

45.11.3 Implications for Practitioners

Having sketched a system model of how NPD and innovation operates in an organisation, and having reconciled it with some of the literature, the next question is the implications for practitioners, namely managers. We suggest that the usual PM lens, of a project being a time-terminated endeavour, is unhelpful in the NPD situation. This is not to diminish the importance of time, especially time-to-market, but simply to point out that time is not the sole, nor even the most important, determinant of quality in NPD. The successful development of new products is strongly dependent on the context, which needs to be managed just as much as the schedule. What this means, to frame it in the language of the PMBOK, is that the *integration activities* are core. Project practitioners in the NPD area might consider using systems engineering approaches, like that shown here, in parallel to their project planning, as a method to ensure the integration activities are happening. Important tasks are managing the trial and error processes, fostering cooperation between the people, and channelling the collective effort into sub-problems where the ambiguity really needs resolving.

45.11.4 Implications for Further Research

As we have shown throughout this chapter, there are many interesting research questions that are still open, particularly in the area of managing people for innovation. There is also the related area of knowledge management with its open questions.

45.12 Conclusions

The process of new product development design is a complex process, and the time schedule represents only one dimension of the endeavour. Whereas a conventional project is typically viewed as a time-terminated endeavour to achieve specific deliverables, NPD is somewhat different. It has very long time frames (the product life cycle), is a strategic activity that changes focus but is never really terminated, and relies on the tacit knowledge of people (and therefore does not view them as merely resources). It is an ambiguous process too, in that the deliverable is not so much a product, but a *new* product, i.e. one that has the potential to be rare, inimitable, and valuable in the eyes of customers. The ambiguity also means that significant parts of the problem have no pre-existing proven solution path. Instead the team have methods that they will try, and they may have to change them part way through if they are not getting the results they need. This complex problem-solving situation requires careful management. There is a very obvious place in NPD for careful project planning, in terms of work breakdown structure and schedule, but this needs to be balanced by the other considerations that the systems engineering perspective provides. Ultimately a successful NPD project is not so much about how close the deployed project follows the project plan, but what the point of difference is of the finished product.

References

Allen NJ, Hecht TD (2004) The 'romance of teams': toward an understanding of its psychological underpinnings and implications. J Occup Organ Psychol 77:439–461

Andersson K (ed) (1994) A vocabulary for concept design. formal design methods for CAD. Elsevier, Amsterdam

Andert E, Peters L (1994) Computer-aided system design assessment tool. In: Proceedings of the IEEE 13th annual international Phoenix conference on computers and communications, Phoenix. IEEE Service Center, Piscataway, pp 193–199

Arrow H, Poole MS, Henry KB, Wheelan S, Moreland R (2004) Time, change, and development: the temporal perspective on groups. Small Gr Res 35:73–105

AS/NZS 3931 (1998) Risk analysis of technological systems: application guide. In: Joint Technical Committee MB/2, Risk Management. Standards Australia Ltd, Homebush

AS/NZS 4360 (2004) Risk management. In: Australian/New Zealand Standard, Standards Australia Ltd, Sydney

Aubert BA, Kelsey BL (2003) Further understanding of trust and performance in virtual teams. Small Gr Res 34:575–618

Balachandra R, Brockhoff K (1995) Are R&D project termination factors universal? Res Technol Manag 38:31–36

Ballard DI, Tschan F, Wailer MJ (2008) All in the timing: considering time at multiple stages of group research. Small Gr Res 39:328–351

Bandura A (1989) Human agency in social cognitive theory. Am Psychol 44:1175–1184

Bandura A (1997) Self-efficacy. Harvard Men Health L 13:4–6

Barczak G, Wilemon D (2003) Team member experiences in new product development: views from the trenches. R&D Manag 33:463–479

Barney J (1991) Firm resources and sustained competitive advantage. J Manag 17:99–120
Barney J, Hesterly WS (2012) Strategic management and competitive advantage. Prentice Hall, Englewood Cliffs
Bassett-Jones N (2005) The paradox of diversity management, creativity and innovation. Creativity Innov Manag 14:169–175
Biondo SJ (1990) Fundamentals of expert systems technology. Ablex Publishing, Norwood
Boehne DM, Paese PW (2000) Deciding whether to complete or terminate an unfinished project: a strong test of the project completion hypothesis. Organ Behav Hum Dec 81:178–194
Brooking A (1998) Intellectual capital. ITP, London
Burgess D (2005) What motivates employees to transfer knowledge outside their work unit? J Bus Comm 42:324–348
Byrd J, Brown PL (2003) The innovation equation: building creativity and risk-taking in your organization. Jossey-Bass, San Francisco
Calantone RJ, Di Benedetto CA, Schmidt JB (1999) Using the analytic hierarchy process in new product screening. J Prod Innovat Manag 16:65–76
Carmen PC, Cano PQ (2006) Human resources management and its impact on innovation performance in companies. Int J Technol Manag 35:11–27
Cassidy K (2007) Tuckman revisited: proposing a new model of group development for practitioners. J Exp Educ 29:413–417
Ceran T, Dorman AA (1995) The complete project manager. J Archit Eng 1:67–72
Chang A, Duck J, Bordia P (2006) Understanding the multidimensionality of group development. Small Gr Res 37:327–350
Cheung PK (2008) Can creativity be learned: a knowledge management approach to creativity support. Ph.D. dissertation, University of Hong Kong, Hong Kong
Clemen RT (1996) Making hard decisions: an introduction to decision analysis. Duxbury Press, Pacific Grove
Clift TB, Vandenbosch MB (1999) Project complexity and efforts to reduce product development cycle time. J Bus Res 45:187–198
Collins CJ, Smith KG (2006) Knowledge exchange and combination: the role of human resource practices in the performance of high-technology firms. Acad Manag J 49:544–560
Conlon DE, Garland H (1993) The role of project completetion information in resource allocation decisions. Acad Manag J 36:402–413
Cook T, Rizzuto R (1989) Capital budgeting practices for R&D: a survey and analysis of business week's R&D scoreboard. Eng Econ 34:291–304
Cordery J (2004) Another case of the Emperor's new clothes? J Occup Organ Psychol 77:481–484
Daft RL (2004) Organizational theory and design. Thomson, Mason
de Leede J, Looise JK (2005) Innovation and HRM: towards an integrated framework. Creativity Innovat Manag 14:108–117
De PK (2001) Project termination practices in Indian industry: a statistical review. Int J Proj Manag 19:119–126
Denker S, Steward DV, Browning TR (2001) Planning concurrency and managing iteration in projects. Proj Manag J 32:31–38
Dilts DM, Pence KR (2006) Impact of role in the decision to fail: an exploratory study of terminated projects. J Oper Manag 24:378–396
Dobson J, Dorsey R (1993) Reputation, information and project termination in capital budgeting. Eng Econ 38:143–152
Dyck B, Starke FA, Mischke GA, Mauws M (2005) Learning to build a car: an empirical investigation of organizational learning. J Manag Stud 42:387–416
Eder WE (1998) Design modeling: a design science approach (and why does industry not use it?). J Eng Design 9:355–372
Elias J, Scarbrough H (2004) Evaluating human capital: an exploratory study of management practice. Hum Resour Manag J 14:21–40
Eysenck HJ (1995) Can we study intelligence using the experimental method? Intelligence 20:217–228

Fairlie-Clarke T, Muller M (2003) An activity model of the product development process. J Eng Design 14:247–272
Farrell MP, Schmitt MH, Heinemann GD (2001) Informal roles and the stages of interdisciplinary team development. J Interprof Care 15:281–295
Finger S, Dixon J (1989a) A review of research in mechanical engineering design. Part 1: descriptive, prescriptive, and computer-based models of design process. Res Eng Des 1:51–67
Finger S, Dixon J (1989b) A review of research in mechanical engineering design. Part 2: representations, analysis, and design for the life cycle. Res Eng Des 1:121–137
FIPS (1993) Integration definition for function modeling (IDEF0): FIPS Publication 183. In: Federal information processing standards publications, National Institute of Standards and Technology. http://www.itl.nist.gov/fipspubs/idef02.doc. Cited 15 Nov 2013
Follon M (1998) Removing limits to growth: implementing knowledge sharing loops between operators and engineers. Eur J Work Organ Psychol 7:567–576
Forteza JA (1974) Problems related to the measurement of creativity. Res Psicol Gen Apl 29:1033–1055
Frost RB (1999) Why does industry ignore design science? J Eng Des 10:297–304
Furnham A, Zhang J, Chamorro-Premuzic T (2006) The relationship between psychometric and self-estimated intelligence, creativity, personality and academic achievement. Imagin Cogn Personal 25:119–145
Garcia MU (2005) Training and business performance: the Spanish case. Int J Hum Resour Man 16:1691–1710
Gardner H (1988) Creativity: an interdisciplinary perspective. Creativity Res J 1:8–26
Garland H (1990) Throwing good money after bad: the effect of sunk costs on the decision to esculate commitment to an ongoing project. J Appl Psychol 75:728–731
Garland H, Sandefur CA, Rogers AC (1990) De-escalation of commitment in oil exploration: when sunk costs and negative feedback coincide. J Appl Psychol 75:721–727
George B (2003) Authentic leadership: rediscovering the secrets to creating lasting value. Jossey-Bass, San Francisco
Gibson FW, Fiedler FE, Barrett KM (1993) Stress, babble, and the utilization of the leader's intellectual abilities. Leadersh Q 4:189–208
Goh ALS (2005) Harnessing knowledge for innovation: an integrated management framework. J Knowledge Man 9:6–18
Graversen EK, Schmidt EK, Langberg K (2005) Dynamic research environments: a development model. Int J Hum Resour Man 16:1498–1511
Gustafsson A (1996) Customer focused product development by conjoint analysis and QFD. Ph.D. dissertation No. 418. Linköping Studies in Science and Technology, Division of Quality Technology, Department of Mechanical Engineering, Linköping University, Linköping
Hailey VH, Farndale E, Truss C (2005) The HR department's role in organisational performance. Hum Resour Manag J 15:49–66
Hales C (1994) Managing engineering design. Longman Scientific & Technical, Harlow
Hamel G, Prahalad CK (2002) Competing for the future. In: Henry J, Mayle S (eds) Managing innovation and change. Sage, London, pp 23–31
Hart S, Jan Hultink E, Tzokas N, Commandeur HR (2003) Industrial companies' evaluation criteria in new product development gates. J Prod Innovat Manag 20:22–36
Hasan H, Al-hawari M (2003) Management styles and performance: a knowledge space framework. J Knowledge Manag 7:15–28
Hauck WE, Thomas JW (1972) The relationship of humor to intelligence, creativity, and intentional and incidental learning. J Exp Educ 40:52–55
Haykin S (1994) Neural networks. Macmillan College, New York
Heilman KM, Nadeau SE, Beversdorf DO (2003) Creative innovation: possible brain mechanisms. Neurocase 9:369–379
Herling D, Ullman D, D'ambrosio B (1995) Engineering decision support system (EDSS). In: ASME design engineering division DE, vol 83, pp 619–626

Herschel RT, Nemati H, Steiger D (2001) Tacit to explicit knowledge conversion: knowledge exchange protocols. J Knowledge Manag 5:107–116
Howe V, Mathieu RG, Parker J (2000) Supporting new product development with the internet. Ind Manag Data Syst 100:277–284
Hubka V (1987) Principles of engineering design WDK1. Heurista, Zurich
Hubka V, Eder WE (1988) Theory of technical systems. Springer, Berlin
Hubka V, Eder WE (1996) Design science. Springer, Berlin
INCOSE (2013) INCOSE Brochure. www.incose.org/symp2013
IPENZ (2005) Practice Note 05: sustainability and engineers. iPENZ, Wellington
ISO 31000 (2009) Risk management: principles and guidelines. International Organization for Standardization
ISO/IEC 31010 (2009) Risk management: risk assessment techniques. International Organization for Standardization
Jaswal S, Jerath JM (1991) Need for achievement and locus of control as predictors of creativity among males at two levels of intelligence. J Pers Clin Stud 7:137–145
Jeffrey R, Michael X, Shin G-C (2003) Project management characteristics and new product survival. J Prod Innovat Manag 20:104–119
Jiménez-Jiménez D, Sanz-Valle R (2005) Innovation and human resource management fit: an empirical study. Int J Manpower 26:364–381
Johnson WHA (2002) Assessing organizational knowledge creation theory in collaborative R&D projects. Int J Innovat Manag 6:387–418
Kakabadse NK, Kouzmin A, Kakabadse A (2001) From tacit knowledge to knowledge management: leveraging invisible assets. Knowledge Process Manag 8:137–154
Kaplan RS, Norton DP (1992) The balanced scorecard: measures that drive performance. Harvard Bus Rev 70:71–79
Kazelskis R, Jenkins JD, Lingle RK (1972) Two alternative definitions of creativity and their relationships with intelligence. J Exp Educ 41:58–62
Kim KH (2005) Can only intelligent people be creative? A meta-analysis. J Sec Educ 16:57–66
Kuipers B (1994) Qualitative reasoning: modeling and simulation with incomplete knowledge. MIT Press, Cambridge
Lau C-M, Ngo H-Y (2004) The HR system, organizational culture, and product innovation. Int Bus Rev 13:685–703
Lawry K, Pons DJ (2013) Integrative approach to the plant commissioning process. J Ind Eng 2013:1–12
Lee H, Choi B (2003) Knowledge management enablers, processes, and organizational performance: an integrative view and empirical examination. J Manag Inform Syst 20:179–228
Leus R, Herroelen W (2004) Stability and resource allocation in project planning. IIE Trans 36:667–682
Lewis MW, Welsh MA, Dehler GE, Green SG (2002) Product development tensions: exploring contrasting styles of project management. Acad Manag J 45:546–564
Liao Y-S (2006) Task characteristics as a moderator of the relationship between human resource management control and product innovation. Int J Manag 23:348–355
Lin IB (2004) Innovation in the networked world: new corporate mindsets for the cyber age. In: Barker C, Coy R (eds) Innovation and imagination at work. Australian Institute of Management, McGraw Hill Australia, Sydney, pp 1–29
Lindemann MD, Fullagar CJ (1975) Creativity and intelligence in South African adolescents. J Behav Sci 2:179–182
Lopez EC, Esquivel GB, Houtz JC (1993) The creative skills of culturally and linguistically diverse gifted students. Creativity Res J 6:401–412
Marr B, Gray D, Neely A (2003) Why do firms measure their intellectual capital? J Intell Cap 4:441–464
Mårtensson M (2000) A critical review of knowledge management as a management tool. J Knowledge Manag 4:204–216

Martin MV, Kmenta S, Ishii K (1998) QFD and the designer: lessons from 200+ houses of quality. In: Proceedings of world innovation and strategy conference, Sydney
Mavondo FT, Chimhanzi J, Stewart J (2005) Learning orientation and market orientation: relationship with innovation, human resource practices and performance. Eur J Mark 39:1235–1263
McAdams DP (1992) The five-factor model in personality: a critical appraisal. J Pers 60:329–361
McAllum M (2004) Rewiring your head: learning to navigate the future. In: Barker C, Coy R (eds) Innovation and imagination at work. McGraw Hill Australia, Sydney, pp 31–61
McCrae RB, Costa PT (1990) Personality in adulthood. Guilford, New York
McCrae RR, Costa PT (1985) Comparison of EPI and psychoticism scales with measures of the five-factor model of personality. Pers Indiv Differ 6:587–597
McCrae RR, Costa PTJ (1999) A five-factor theory of personality. In: Pervin LA, John OP (eds) Handbook of personality: theory and research. Guilford Press, New York, pp 139–153
Meer B, Stein MI (1955) Measures of intelligence and creativity. J Psychol 39:117–126
Melymuka K (2004) Surviving the sponsor exit. Computerworld 38:40
Meredith JR, Mantel SJ (1995) Project management: a managerial approach. Wiley, New York, NY
Meso P, Smith R (2000) A resource-based view of organizational knowledge management systems. J Knowledge Manag 4:224–234
Meso P, Troutt MD, Rudnicka J (2002) A review of naturalistic decision making research with some implications for knowledge management. J Knowledge Manag 6:63–73
Messica A, Mehrez A (2002) Time-to-market, window of opportunity, and salvageability of a new product development. Manag Decis Econ 23:371–378
Mill HF (1994) Simplifying the implementation of QFD. In: IEE colloquium on customer driven quality in product design, London, pp 5/1–5/4
Mingjie R, Jie Y, Hutchinson J, Jinjun W (2008) Managing knowledge for new product performance in the high technology industry. Int J Technol Manag 41:96–108
Mullen B, Johnson C, Salas E (1991) Productivity loss in brainstorming groups: a meta-analytic integration. Basic Appl Soc Psychol 12:3–23
Nijsse M (1975) Creativity and its relationship with intelligence and personality variables among children approximately 12 yrs old. Ned Tijdschr Psychol 30:657–676
NIST (2013) Criteria for performance excellence. In: Baldrige National Quality Progam. National Institute of Standards and Technology, Department of Commerce, Gaithersburg. http://www.nist.gov/baldrige/. Cited 15 Nov 2013
Nonaka I (1990) Redundant, overlapping organization: a japanese approach to managing the innovation process. Calif Manag Rev 32:27–38
Nonaka I (1991) The knowledge-creating company. Harvard Bus Rev 69:96–104
Nonaka I (1994) A dynamic theory of organizational knowledge creation. Organ Sci 5:14–37
Nonaka I, Toyama R, Konno N (2000) SECI, ba and leadership: a unified model of dynamic knowledge creation. Long Range Plan 33:5–34
Norman WT (1963) Toward an adequate taxonomy of personality attributes: replicated factor structure in peer nomination personality ratings. J Abnorm Soc Psychol 66:574–583
Olin T, Wickenberg J (2001) Rule breaking in new product development: crime or necessity? Creativity Innovat Manag 10:15–25
Oxley NL, Dzindolet MT, Paulus PB (1996) The effects of facilitators on the performance of brainstorming groups. J Soc Behav Pers 11:633–646
Paese PW (2000) Deciding whether to complete or terminate an unfinished project: a strong test of the project completion hypothesis. Organ Behav Hum Dec 81:178–194
Pahl G, Beitz W (1988) Engineering design: a systematic approach. The Design Council, London
Panayotopoulou L, Papalexandris N (2004) Examining the link between human resource management orientation and firm performance. Pers Rev 33:499–520
Panico CR (2004) From risk to revenue. Glob Cosmet Ind 172:2–4
Paulus PB, Van der Zee K (2004) Should there be a romance between teams and groups? J Occup Organ Psychol 77:475–480

PMI (2004) A guide to the project management body of knowledge (PMBOK guide). Project Management Institute, Newtown Square
PMI (2006) Pull the plug. PM Network 20:38–44
PMI (2008) A guide to the project management body of knowledge (PMBOK guide). Project Management Institute, Newtown Square
Polanyi M (1958) Personal knowledge: towards a post critical philosophy. The University of Chicago Press, Chicago
Pons DJ (2008) Project management for new product development. Proj Manag J 39:82–97
Pons DJ (2010) Strategic risk management in manufacturing. Open Ind Manag Eng J 3:13–29
Pons DJ, Raine J (2005) Design mechanisms and constraints. Res Eng Des 16:73–85
Popescu-Neveanu P, Facaoaru C (1972) On the relation between intelligence and creativity: creativity as a dimension of personality. Rev Psiho 18:301–314
Porter ME (1996) What is strategy? Harvard Bus Rev 74:61–78
Pratim Ghosh P, Chandy Varghese J (2004) Globally distributed product development using a new project management framework. Int J Proj Manag 22:669–678
Preckel F, Holling H, Wiese M (2006) Relationship of intelligence and creativity in gifted and non-gifted students: an investigation of threshold theory. Pers Indiv Differ 40:159–170
Rad PF, Levin G (2005) A formalized model for managing a portfolio of internal projects. In: AACE InternTrans, pp 04.01–04.05
Richards RL (1976) A comparison of selected Guilford and Wallach-Kogan creative thinking tests in conjunction with measures of intelligence. J Creative Behav 10:151–164
Rothwell JD (2007) In mixed company: communicating in small groups and teams. Thomson Wadsworth, Belmont
Runco MA, Albert RS (1986) The threshold theory regarding creativity and intelligence: an empirical test with gifted and nongifted children. Creative Child Adult Q 11:212–218
Rushton JP (1990) Creativity, intelligence, and psychoticism. Pers Indiv Differ 11:1291–1298
Saucier G, Goldberg LR (2001) Lexical studies of indigenous personality factors: premises, products, and prospects. J Pers 69:847–879
Scratchley LS, Hakstian AR (2001) The measurement and prediction of managerial creativity. Creativity Res J 13:367–384
Shipton H, Fay D, West M, Patterson M, Birdi K (2005) Managing people to promote innovation. Creativity Innovat Manag 14:118–128
Silverman B (1994) Unifying expert systems and the decision sciences. Oper Res 42:393–413
Simon HA (1981) The sciences of the artificial. MIT Press, Cambridge
Singh U (2006) Novelty and meaning contexts of creativity vis-a-vis Jensen's level I and level II abilities. J Proj Psychol Ment Health 13:140–152
Singh RB, Mathur SR, Saxena S (1977) Creativity as related to intelligence, achievement and security-insecurity. Indian Psychol Rev 14:84–88
Sligh AC, Conners FA, Roskos-Ewoldsen B (2005) Relation of creativity to fluid and crystallized intelligence. J Creative Behav 39:123–136
Smoliar SW (2003) Interaction management: the next (and necessary) step beyond knowledge management. Bus Pros Manag J 9:337–353
Smulders F, de Caluwé L, van Nieuwenhuizen O (2003) The last stage of product development: interventions in existing operational processes. Creativity Innovat Manag 12:109–120
Snell SA (1992) Control theory in strategic human resource management: the mediating effect of administrative information. Acad Manag J 35:292–327
Söderlund J (2002) Managing complex development projects: arenas, knowledge processes and time. R&D Manag 32:419–430
Starr JW, Nicholl C (1975) Creativity and achievement in Nuffield physics. Brit J Educ Psychol 45:322–326
Statman M, Caldwell D (1987) Applying behavioral finance to capital budgeting: project terminations. Finance Manag 16:7–15
Sternberg RJ (2004) Why smart people can be so foolish. Eur Psychol 9:145–150

Sweet M, Michaelsen L (2007) How group dynamics research can inform the theory and practice of postsecondary small group learning. Educ Psychol Rev 19:31–47

Swink M (2005) Exploring new product innovation types and performance: the roles of project leadership, functional influences, and design integration. Int J Prod Dev 1:241–261

Tatikonda MV (1999) An empirical study of platform and derivative product development projects. J Prod Innovat Manag 16:3–26

Tidd J, Bessant J, Pavitt K (2001) Managing innovation: integrating technological, market and organizational change. Wiley, Chichester

Tuckman BW (1965) Developmental sequence in small groups. Psychol Bull 63:384–399

Ullman DG (2001) Robust decision-making for engineering design. J Eng Design 12:3–13

Virgolim AMR (2006) Creativity and intelligence: a study of Brazilian gifted and talented students. Ph.D. dissertation, University of Connecticut, Storrs

Vroman K, Kovacich J (2002) Computer-mediated interdisciplinary teams: theory and reality. J Interprof Care 16:159–170

Wagner R (1991) Managerial problem solving. In: Sternberg R, Frensch P (eds) Complex problem solving: principles and mechanisms. Lawrence Erlbaum Associates, Hillsdale, pp 159–183

Wheelan SA, Davidson B, Tilin F (2003) Group development across time. Small Gr Res 34:223–245

Yassine AA, Falkenburg DR (1999) A framework for design process specifications management. J Eng Design 10:223–234

Chapter 46
Key Factors of Relational Partnerships in Project Management

Hemanta Doloi

Abstract Relational partnership is one of the most widely used procurement mechanisms for construction projects. While the relational procurement is based on principles of contracting parties being cohesive and committed to work with an agreed project outcome, what really drives the success of a good relationship among the parties and the underlying factors are difficult to ascertain. Among many factors highlighted by researchers, the three widely known significant factors are *communication*, *trust and confidence* and *joint risk management*. Based on an empirical study in Australian construction industry, a comprehensive investigation was undertaken by the author to analyse these factors further and thereby to understand the impacts on the success of relational partnerships in construction projects. The results of the investigation identified communication as the single most influencing factor impacting relational partnering success. While the trust and confidence were found to be mutually inclusive for effective communication, both the factors have direct influence on developing capability for joint risk management within the partnering organisations.

Keywords Construction projects • Contract management • Procurement mechanisms • Project management • Relational partnership

46.1 Introduction

Perhaps the most recognizable characteristic in standard forms of project procurement within the construction industry is the existence of the uncompromised culture among the contracting parties towards achieving the target objectives. The idea that "your profit loss is my gain" is entrenched throughout all standard procurement methods and can only be seen as counter-productive as it generates a variety of inefficiencies during project delivery. While the problem is recognized widely

H. Doloi (✉)
Faculty of Architecture, Building and Planning, The University of Melbourne, Parkville, VIC, Australia
e-mail: hdoloi@unimelb.edu.au

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_16

across many countries, this chapter aims to generate a meaningful debate on the issue based on empirical observations in medium to large scale development projects within Australian construction industry.

The adversarial attitudes among the contracting parties often result in a difference of opinion which eventually end up with disputes and claim of compensations (Jones 2001). The conflict will then result in unnecessary cost and time delay which could otherwise be avoided. Numerous studies have shown that errors by the contractor in competitive tender prices can result in sub-standard quality of workmanship and design non-conformance (Naoum 2003). Anecdotally, an earlier involvement of an integrated team potentially assists in reducing conflicts among the project partners. Relationship agreements promote the culture of working together within a cohesive team to achieving an agreed outcome. Rather than penalising non-conformance with the threat of liquidated damages and excessive variation claims, participants in a relationship agreement generally receive a share of profit that is determined by the overall team performance. Some of the key benefits in relationship agreements include accelerated delivery times, reduction in conflict, appropriate risk allocation, informed decision making and a reduction in the overall project cost. Studies have shown that sharing of profit margins ensures "best for the project" outcomes in projects and such philosophy forms the primary theory behind the relationship agreements (Rahman and Kumaraswamy 2002). The remainder of the chapter will analyse the key factors associated with the relational agreement based on empirical evidence collected from Australian construction industry.

In Sect. 46.2, partnering has been discussed in the context of relationships among the parties. Section 46.3 then reviews some of the key literature to identify the gaps within the published domain in order for highlighting the challenges associated with the current practice. In Sects. 46.4–46.7, some of the key factors have been analysed and explored based on statistical analysis of the data collected empirically in Australian construction industry. The findings and contributions of the research have been summarised in Sect. 46.8.

46.2 Partnering Through Relational Agreements

Relational Agreement can be viewed as the core in both Project Alliance and Partnership, where the notion of partnering is well understood. While the project alliancing is relatively a new concept, partnership practice in construction projects is quite common (Walker and Shen 2002). However, the link between the processes of forming the partnerships and achieving success in projects is not quite reported in the project management literature. This chapter proposes to unveil the perceptions of the industry, identifying the pros and cons of relationship agreement in relation to traditional approach and thereby highlight the key distinctions of the critical factors in achieving project success.

Partnering has been viewed as an effective tool in successful delivery of projects across many countries including UK (Wood and Ellis 2005; Naoum 2003), Europe

(Williams and Lilley 1993), Hong Kong (Rahman and Kumaraswamy 2002) and Singapore (Kwan and Ofori 2001). There is an increasing perception that partnering could help managing risks and uncertainties and thereby improve productivity in projects. However, clear understanding of the influencing factors is a critical element for deriving true value in effective partnerships (Chan 2001). To this effect, Bresnen and Marshal (2000) highlighted the significance of long term relational links between contractors and subcontractors in achieving partnering success in projects. Halman and Braks (1999) asserted the importance on the organizational structure of the partners for deriving potential benefits in reduced costs and enhanced profits. Given the nature of modern construction projects where involvement of multitude of contracting parties results in very high risks, partnering based on relationship agreements and cooperative teamwork is perceived to be an effective medium for managing conflicts between diverse participants (Rahman and Kumaraswamy 2002). Over the last few decades, numerous enquiries have been reported highlighting the underlying factors and their impacts on relational partnering success. According to published literature, the three factors critical to the success of relational partnering include the establishment of the joint risk management for effectively managing the project risks (Rahman and Kumaraswamy 2005), trust and confidence (Ngowi 2007) and open and reliable lines of communication between team members (Cheung et al. 2003). Although the theory behind relational partnering remains relatively simple, previous studies including Phua (2006) and Ngowi (2007) have shown that a lack of trust between parties and a difference in opinion on resolving disputes may jeopardize an otherwise successful project and cause an unwarranted market perception of the particular procurement process.

In order for clients, designers and contractors to appropriately contribute to the success of relational partnerships, a clear understanding of these attributes is a difficult task. By establishing the relational links, a better understanding of the effects of these factors for successful delivery of projects can be created objectively. Results of such analysis help the contracting partners to prioritize the factors in terms of their criticality for developing contractual arrangements and assuming responsibilities in order to obtain the desired outcomes. It is anticipated that the contributions of this chapter will provide a meaningful insight to the relative effect of the relational partnering process for effective development and implementation in construction projects. By increasing each discipline's knowledge in meeting requirements and expectations of all the contracting parties, it may be possible to accurately highlight the pros and cons of relational contracting approach and clarify the perceived views within the construction industry.

46.3 Background Reviews

The use of relational partnerships is not a new concept and there has been numerous studies examining the influences of the underlying factors affecting their performance in projects. In an effort to understand the current scholarships

and significance of influencing factors, a comprehensive literature review was undertaken and the summary is presented below.

Williams and Lilley (1993) examined the influencing factors by taking into consideration of individual importance of each factor in selecting suitable partners for developing alliances in project. Based on a UK based case study, their research discussed the criticality and importance of nine different perspectives in the partners' selection process. Among them, 'communications barriers' between contracting parties was reported to be most important perspective in the partner selection process. However, mutual dependency was found to be the least important in the adopted case study. While the relative importance of all nine perspectives was critically discussed, this research did not attempt to quantify the impacts of these perspectives in the context of partnering success.

Halman and Braks (1999) investigated the organizational concepts of project alliancing in the context of reduce project costs and enhance profits for all the contracting parties in the project. Based on an in-depth case study carried out within a contractor company in the offshore oil industry, their research asserted that open communication within the partners at an early stage results in more deliberation and as a consequence more time consuming in senior management decision process. Such assertion certainly contradicts the widely accepted view that communication is one of the key elements in partnering success.

In the UK, Bresnen and Marshal (2000) developed a predictive model to examine the critical issues and links associated with partnering in construction from a cultural perspective at both organizational and inter-organizational levels. Based on a theoretical analysis of the partnering concepts, it was asserted that application of tools and techniques to partnering must be backed up with strong commitment, trust and unity and behavioural compliance among the contracting parties for partnering success. In a separate study, Bresnen and Marshal (2000) investigated the relationships of use of incentives, motivation and commitments among partners in partnering and alliancing success. Based on a semi-structured interview approach across five case projects and qualitative research methods, this research concluded no evidence on any strong or systematic relationships between incentive systems and consolidation of trust and commitments between participants. While this research did not address the interrelated elements and practices of enhanced relationships in partnering success, the need for further investigation had been clearly highlighted.

In a study in Singapore, Kwan and Ofori (2001) reviewed the success of partnerships by developing a synergy from Chinese cultural perspective. Based on a postal survey on twenty-seven Chinese-owned large contractors in Singapore, their research ascertained that trust and friendship, commitment and open communication are some of the fundamental elements for successful partnering implementation. However, such outcome was considered not robust to define the links between each of these elements leading to conducive relationships among partners and the overall partnering success.

In Hong Kong, Rahman and Kumaraswamy (2002) examined the attitudes of contracting parties and their co-operative relationships for joint management of risks in partnering implementation. Based on the transaction cost economy (TCE)

and relational contracting (RC) principles and 47 responsive questionnaire across construction industry in Hong Kong, two conceptual models were developed. The findings clearly highlighted the requirements of better relationships, cooperative teamwork and adaptation of appropriate restoration techniques for collaborative management of risks and successful partnering in projects. Though Rahman and Kumaraswamy (2002) highlighted the advantages of joint risk management (JRM) in partnership arrangement, exclusion of the critical elements such as communication and trust and confidence made the usefulness of the research in the context of establishing the impacts on partnering success incomprehensive. While the importance of adaptive contractual arrangement among the contracting parties is emphasized, significance of the critical drivers for successful development and implementation of partnerships have not been explicitly reported in the research.

Based on the UK construction projects, Naoum (2003) explored the ingredients for good partnering practice from the perspectives of improved productivity, lower costs, satisfactory quality and on time delivery. Based on a theoretical ground and perceived best practices across construction industry in the UK, this research contended relational partnership as the single most value-based procurement approach for success. While cost and time savings have been identified as a result of long term partnering relationships, measure and motivation for successful partnering were not quite well discussed. In a separate study in Hong Kong, Cheung et al. (2003) investigated the behavioral aspects of construction partnering with a specific focus to trust and cooperation among the contracting parties. Based on a major railway project in Hong Kong, the research reported that trust and commitment are the only decisive factors in construction partnering success. A high level of trust and commitment naturally promote the cooperation, open and joint problem solving attitudes among contracting partners leading the partnering success. However, findings of such qualitative research could not be considered exhaustive in establishing relational links among the critical elements.

In a separate study in Hong Kong, Wong and Cheung (2004) examined the relative importance of trust factors contributing to partnering success. Based on a questionnaire survey response from private and public sector developers, consultant firms and contractors, 14 trust attributes were analyzed using statistical factor analysis. Among all 14 attributes, five attributes namely problem solving, competent, unity, communication and respect were reported to be highly influencing in successful partnering implementation. Focusing on UK construction projects and by investigating the contractors' experience of partnerships relationships over the life of the project, Wood and Ellis (2005) concluded co-operation, teamwork and shared vision as most cited factors for a successful partnering process. This research also confirmed the widely expressed view that partnering arrangements are largely cost-driven as gain sharing and pain-sharing are some of the underlying principles in partnership arrangements. Wong and Cheung (2005) investigated the influence on trust in partnering by identifying four major factors: partners' performance, partners' permeability, system-based trust and relational bonding. While the findings of this research highlighted the clear links between partner's trust levels and the first three factors, departure of relational bonding in the equation

made the model incomprehensive. All these findings were found to be insufficient in defining professional obligations, motivation and relational links among the contracting parties in the overall partnering success.

46.3.1 Key Issues in Relational Partnerships

From the above literature review, a few gaps in understanding the impacts and criticality of the widely known factors associated with the partnering process emerge. A large proportion of the existing research in the field of construction partnering notes the difficulties faced by the contracting parties as a consequence of the existence of the cost-driven, gain sharing and pain sharing principles. However, much of the partnering literature tends to report on critical success factors based on anecdotal evidence of the success stories (Bresnen and Marshal 2000). The weakness, however, lies in overlooking the importance of these factors in the context of relational links and influence in successful partnering implementation in projects. Having reviewed the work conducted by the above researchers in the field, it has been evident that trust and confidence, communication and joint risk management factors broadly represent the partnership principles leading to partnering success. Ngowi (2007) noted the influence of partner trustworthiness that can eliminate the needs for contractual clauses to effective operation in partnering process. While Rahman and Kumaraswamy (2002) asserted capability of joint risk management as a result of relational partnering, Naoum (2003) reported on the shortfall of any clear link between success measures and partnering in projects. Jones (2001) and Naoum (2003) considered motivation among the partners as an emerging good practice. However, a clear consensus on the role of each contracting party and distinctive degree of responsibility across the global issues of relational partnering could not be drawn decisively. While the outcomes of the above research aimed to identify many key issues in a disjointed manner, the root causes surrounding the working relationships between all parties and the behavioral implications in relational partnering success remain unexplored.

46.3.2 Scholarships and Contributions

As evidenced by the above research, the field of partnering and underlying processes are reasonably understood. However, it is unclear how each of the elements or attributes associated with the contracting parties in relational partnering contracts relates to one another and how do they impact on successful partnering outcomes. This chapter aims to highlight some of the key inherent attributes associated with the contracting parties involved in relational partnering process and to investigate their relative importance and significance in successful project implementation.

46.4 Empirical Evidence from Australian Construction Industry

Anecdotally, relational agreement is quite a common practice across Australian construction industry. However, there is no any documented evidence on how the relational agreements between the parties perform in relation to the project delivery. While the fragmental Australian construction industry generally supports the relational agreement as one of the key forms of procurement strategies within a relatively smaller market, much investigation is required for developing appropriate understanding of partners' behaviors and their linkages to the performance outcomes using the field data. In an attempt to collect the relevant empirical evidence for analysis, a survey method was considered appropriate as part of this study. A questionnaire survey was designed for respondents to assess the performance of a project they had participated in and to evaluate the influence of trust and confidence, communication and joint risk management in that project. Field data was collected using a Likert scale requesting the respondents to provide the responses with varying degrees of agreement or disagreement.

The preliminary data was based on a total of 43 medium to large construction firms in Australia. The target population of the survey in this study was contractors, architects, consultants and owners involved mostly in infrastructure, residential and commercials projects. A total of 150 questionnaires were mailed out or hand delivered to target participants involved mostly in the senior management teams and 97 valid responses were returned. Among the 97 respondents, 56 are contractors, 10 are architects, 18 are consultants or designers and 13 are owners or developers. The responses of the questionnaires were analyzed using two methods: standard statistical methods and Structural Equation Modeling (SEM). While the statistical significance of the data sample is crucial for deriving any meaningful observations in Standard Statistical Methods, the selection of the indicators is highly significant in the context of true measure of the representative practices across the latent or unmeasured variables for the use in SEM (Doloi 2009). Details of the selection processes of the measured and latent variables are not included in this chapter but can be referred to Doloi (2009).

46.5 Results from Standard Statistical Analysis

Determination of a suitable analytical tool for testing the data is quite an important process for developing appropriate understanding of the impacts of the measured parameters on the outcomes. In order to derive the advantages and limitations in relationship agreements, a number of analytical tools were used namely descriptive analysis, bi-variate correlations, independent t-test and factor analysis. The results of some of these analyses are discussed in the following sections

Table 46.1 Top five mean scores of the five key factors

Rank	Relationship agreements Variable	Mean
1	Team member build a broad range of skills	4.25
2	Team environment facilitating informed decision making	3.94
3	Delivery method for achieving best project outcome	3.89
4	Team members understand risks	3.78
5	Team member pro-active in resolving problems	3.78

46.5.1 Descriptive Observations

Table 46.1 shows the list of the five variables in the survey with the highest mean scores in relationship agreements and thus could be interpreted, on a basic level, as the most important variable for each procurement method. As indicated in the literature by Ross (2006), the most important variable for relationship agreements, with a mean score of 4.25 (above agree), was "team members are able to build on a broad range of skills". This result highlights the benefit of the integrated project team established by relationship agreements and suggests that in removing the adversarial nature present in many traditional procurement methods, team members are able to learn important skills from one another.

The second most important variable was that the "team environment consistently resulted in informed decision making" with a mean score of 3.94 (agree). One would expect that constant interaction between team members is an advantage of relationship agreements as it not only results in informed decision making but also enables team members to build on a broad range of skills. The third ranked variable for relationship agreements was "The project delivery method consistently results a best for project outcome" with a mean score of 3.89 (agree). This variable is an obvious advantage in the agreement structure and the result confirms its importance in comparison to other believed advantages (Ross 2006).

"Team members understanding the project risks" was the fourth most important variable, with "team members being proactive in resolving disputes" the fifth with means of 3.78 and 3.78 respectively. While both variables ranked in the top five of importance for relationship agreements, it is difficult to confirm the findings of Rahman and Kumaraswamy (2002) as members in relationship agreements didn't report to have any greater knowledge of the project risks than participants in traditional procurement approach.

In regards to the limitations, the integrated project team established by relationship agreements does not necessarily lead to reliable communication between the project team, and therefore, could not be identified as an advantage of the delivery method. Furthermore, the relationship agreements are not necessarily more cost efficient than traditional methods and the mean difference indicates that traditional procurement methods may, in fact, be the most cost efficient delivery process. The analysis suggests that both relationship and traditional methods could improve the

level of trust between project team members to increase efficiency. However, a mean difference of 0.46 indicates that lack of trust generates more inefficiency in traditional procurement methods than relationship agreements. The findings regarding lack of trust influencing project efficiency support those of Ngowi (2007), who suggested partners in relationship agreements will be vulnerable to project inefficiencies in project delivery due to lack of trust.

In addition to the above, the results for relationship agreements indicate that the delivery process could be reviewed to be more time efficient and that disputes regularly arise from a difference of opinion by the project team. It is expected that "disputes" in this case may relate to problems that are resolved internally rather than referring to issues that may require legal intervention, to resolve a standoff between two or more parties.

46.5.2 Findings from Factor Analysis

A total of 18 key attributes associated with relational agreements have been reduced to six broad factors by employing the Factor Analysis using Statistical Package for Social Sciences (SPSS). As seen in Fig. 46.1, the first factor (Factor 1) in the relational agreements named as *a cohesive project team*, comprises five key attributes namely *team environment*, *dispute resolutions*, *proactive project team*, *risks and responsibility* and *competent team*. As seen, all five attributes highlight the importance of team performance in achieving success in the project. A positive team environment is highly desirable for proactive dispute resolutions and sharing risks and responsibilities within the project. Willingness to risks and responsibility sharing is highly dependent on the team's technical competency and being proactive in the decision making context. Under the traditional method, the issue of team performance is confined to predominately time and quality performance only. However, focus on time, cost and quality performance was the third factor (Factor 3) under the relational agreements which show a significant shift of focus across both procurement methods. Time and cost performance is usually coupled with some of the key performance measures within the project. Such performance measures are

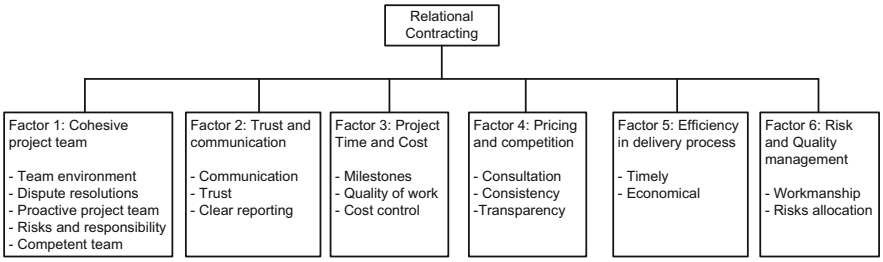


Fig. 46.1 Underlying factors associated with the relationship contracting agreements

usually meeting the milestones, achieving the desired quality outcome or controlling cost within the approved budget in the project.

The second factor (Factor 2) *Trust and Communication* comprising three key underlying attributes has been widely branded by many past researchers as one of the most important factors especially in developing relational partnerships in project (Bresnen and Marshal 2000; Rahman and Kumaraswamy 2005; Smyth and Edkins 2007). The three attributes are *communication*, *trust* and *clear reporting* protocols within the organisation. In contrast, the importance of trust and communication is considered to be not quite important due to the strict contractual obligation among the parties concerned. The risk management in traditional methods is found to be process driven in terms of appropriate risk allocation between the partners. Ensuring high quality workmanship and informed decisions require strict process control in traditional approach. However, increased trust and confidence under relational agreements allow sharing risks and devising appropriate management strategy within a cohesive team environment.

Factor 4, *pricing and competition* under the relationship agreements is closely coupled with the attributes associated with *cost efficiency*, *consistency* and *transparency* in delivery processes. Team members are expected to be consulted in relation to achieving cost efficiency across the supply chain within the project. Similarly consistency across the organisational conducts and transparency in decision making processes are highly desirable within the parties for achieving efficiency in partnering projects.

46.6 SEM Analysis and Outputs

In order to unfold the significance and criticality of the three key factors, *trust and confidence*, *communication* and *joint risk management* in achieving *relational partnering success*, the empirical dataset was then further analyzed using Structural Equation Modelling (SEM). SEM is a co-variance based research methods capable of establishing relational links between the latent variables with hypothetical constructs (Doloi 2009). Table 46.2 shows the four latent factors used in the hypothetical construct on the first column and their corresponding measured attributes or variables on the second column.

The initial structural model was analysed using AMOS 16.0. As discussed by Molenaar et al. (2000), the initial SEM that was based on theoretical expectations and past empirical findings, found to be premature without meeting the standard indices of model-fit (such as t-statistics and R-Squares for model equations). A feasible model should be selected based on the recommended Goodness-For-Fit (GOF) measures and the model that satisfies both theoretical expectations and GOF is finally selected for SEM analysis (Molenaar et al. 2000). Thus, in this research, by employing the GOF measures, the model refinement was performed to improve the fit to its recommended levels (Jin et al. 2007; Molenaar et al. 2000). Based on a number of trials through eliminating some of the attributes across three latent

Table 46.2 Constructs and measurements for SEM analysis

Factors	Attributes/Indicators
Trust/Confidence	• Lack of trust (T1) • Increased confidence and trust (T2) • Importance of trust and confidence (T3) • Lack of confidence (T4) • Effects on dispute resolution and delays (T5) • Mutual confidence among partners (T6) • Long term working relationships (T7) • Likelihood of disputes being erupted (T8)
Communication	• Lack of communication (C1) • Increased communication (C2) • Reliable and frequent communication (C3) • Effect on reduction of conflicts (C4) • Effect on informed decision making (C5) • Effect on improvement in expectations (C6) • Likelihood of disputes being erupted (C7) • Scope changes without causing disputes and delays (C8)
Joint risk management	• Efficiency in managing project risks (R1) • Advantages in relationship agreements (R2) • Effective monitoring and successful project delivery (R3) • Effects on communication (R4) • Importance of trust and confidence (R5)
Relational partnering success	• On time project delivery (P1) • On budget project delivery (P2) • Desired quality outcomes (P3) • Cost savings (P4)

factors and the relational partnering success indicators, a total of nine attributes were required to be eliminated due to their low correlations with other latent factors in the final SEM. Due to sake of brevity, the details of this elimination process have not been discussed which can be found in Molenaar et al. (2000); Wong and Cheung (2005), and Doloi (2009). A final SEM model was established that highlights the relational linkages among the latent variables as expected in the research.

46.7 Findings of the SEM Analysis

The summary of the standardized coefficient of the final SEM is shown in Fig. 46.2. As seen, the test results generally support the relationships between communication, trust and confidence and joint risk management. A significant and positive relationship was found between communication and trust and confidence factors. The two way relationship between these two factors suggests that effective and frequent communication certainly influence in enhancing the trust and confidence among partners and vice versa. According to the measurement model of the final SEM, trust and confidence in the final SEM was measured by four attributes: increased trust

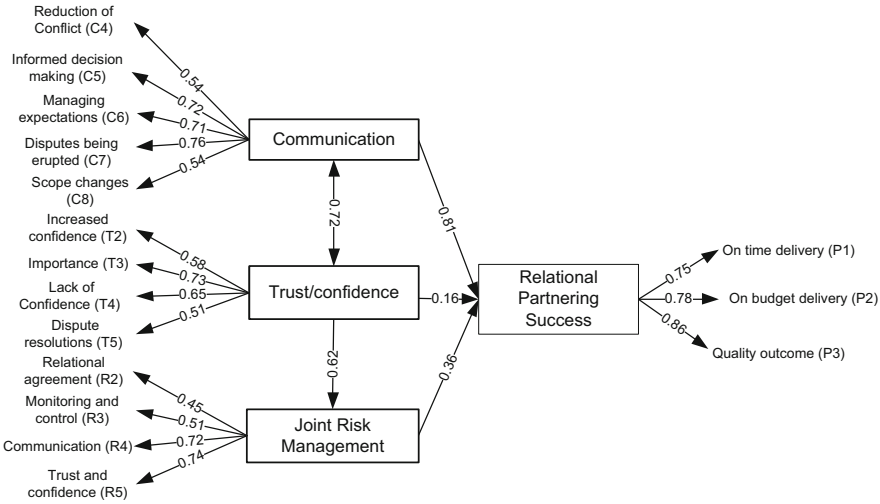


Fig. 46.2 Standardized coefficient estimates (p-value) of the final SEM

and confidence (T1), importance of trust and confidence (T2), lack of confidence (T3) and effects on dispute resolution and delays (T4). All these four attributes with standardized loadings over 0.55 have been found to be sufficient to judge the trustworthiness of the partners.

The final SEM results as depicts in Fig. 46.2 also suggests that only one factor namely communication, has the significant correlation with the relational partnering success Among the three main factors, trust and confidence and joint risk management were found to have least direct influence on the relational partnering success. While role of communication has been found to be the greatest influencing factor for relational partnering success, the trust and confidence were found to be mutually inclusive for effective communication. Consequently, the trust and confidence were found to have direct influence on developing capability for joint risk management within the partnering organisations. This finding alters the widely accepted view across Australian construction industry that partnering is built on trust and confidence only and any risks associated in projects are best dealt with joint responsibility without any problems (Williams and Lilley 1993).

The results of the above model with a small positive standardized coefficient between trust and confidence and relational partnering success ($p = 0.16$) and a marginal standardized coefficient between joint risk management and relational partnering success ($p = 0.36$) direct weaker linkages between the respective factors. While trust and confidence are assumed to be one of the important factors in partnerships arrangement, this is the east important factor directly affecting project success through relational agreements. The result highlights the fact that there is too little or no relationship between the latent variable and the relational partnering success. The research revealed quite a contradiction to some of the

findings highlighted in earlier studies by Wong and Cheung (2004) and Ngowi (2007). Most of the past studies asserted trust and confidence as a critical factor contributing to success of relational partnerships.

The finding revealed that the communication is rather an essential factor in the successful delivery of relational partnering success with the high standardized coefficient ($p = 0.81$). It has been revealed that reliable line communications has the capability to contribute to the success of a relational partnering agreement more than simply relying on trust and confidence building exercises. Therefore projects that experience lack of communication are less likely to achieve their objectives. Previous findings also suggest similar conclusions that preliminary function of adopting partnering is to provide a merely conducive platform for partners for active communication for managing any conflicts and potential disputes in projects (Wood and Ellis 2005)

On the other hand, contradicting to research conducted by Rahman and Kumaraswamy (2002), the practice of joint risk management was also found to be a least contributing factor to the success of relational partnerships. There was a clear assertion that the integrated project team established in relational partnerships collectively better manage the project risks without putting the sole responsibilities on the individuals. Rather, as stated earlier, increased trust and confidence among partners has relatively a greater influence in collective management of risks in projects. While there may be a relationship between joint risk management and project success, the magnitude of the relationship is rather small or even negligible. From the above, it is difficult to state with any conviction, that joint risk management is an important factor directly contributing to the relational partnering success.

46.8 Conclusions

The benefit of relational partnering in achieving project delivery success is acknowledged within the reviewed literature. Previous studies identified communication, joint risk management and trusts among partners as important attributes in partnering success. Amongst all, trust was perceived to be one of the most critical factors influencing relational partnerships and previous research suggests that lack of trust has been a major reason behind inefficiencies in project delivery (Wong and Cheung 2004; Ngowi 2007). However, in contrast, the findings of this research suggest that trust and confidence has little or no effect on project success. While trust and confidence may have been a major reason behind project inefficiencies, the inefficiencies created by lack of trust and confidence do not affect the success of a relationship agreement.

Based on the mean score ranking for relationship agreements, the importance of the key variables were highlighted and discussed in relation to the traditional practices. Variables such as building skills, understanding risks and being pro-active in resolving problems were important to both relational and traditional management techniques. However, cost efficiency, trust and communication between the project

team require improvement under both relationship agreements and traditional procurement methods.

On reviewing the means values, it was concluded that while relationship agreements may not be as time efficient as traditional procurement methods, the integrated management team has a number of advantages. In the relationship agreement, the team environment consistently results in informed decision making and working with other members of the project team enabled the individual to build on a broad range of skills. Current delivery process consistently produces a best for project outcome and the milestones are consistently achieved on program across the projects. Disputes are resolved quickly and efficiently.

The structural equation models developed on the empirical dataset also suggested that joint risk management was not critical in terms of achieving success in relational partnerships in project. However, joint risk management capability is influenced by the level of trust and confidence among the parties. This finding proves that joint risk management becomes better as the perceived trust and confidence among the partners become higher. The latter was also found to be mutually inclusive with the level of communication within the partners in the project. The trust and confidence is also found to be higher as the perceived communication between partners becomes better and vice versa. This finding contrasts to the results published by Rahman and Kumaraswamy (2002) which contended that by collectively managing the risks, the project team can easily prevent the occurrence of undesirable activities interfering with the project objectives. The results found in the SEM, therefore, do not support these previous findings. However, it has been revealed that increased trust and confidence among partners has relatively greater effect on collective risk sharing and effective management in projects. Supporting part of the first hypothesis, the SEM identified communication as the single critical factor to the relational partnering success. This finding confirms the previous assertion that achieving reliable lines of communication will contribute to the success of a relational partnership, while projects that experience lack of communication are less likely to achieve their objectives (DTF DOTAF 2006; Naoum 2003; Cheung et al. 2003).

It has been revealed that the success of relational partnering can only be achieved if all the key partners engage in clear line of communication across all levels. In order to improving the practice of effective communication, firms need to set out the protocols between the associated partners with clear definition of lines of roles and responsibilities in the organisation. A conducive environment for facilitating effective communication evidently leads to developing trust and confidence among partners, which eventually supports developing the collaborative risk management capability for the project. The revelation for the need of free communication among partner's organizations is considered as a major shift towards developing successful partnership relationships and hence achieving the project success.

Acknowledgements Some of the materials in this chapter have been taken from the following published work with appropriate permission for reprint from the publisher, Taylor & Francis Ltd. No further reprint is permitted without written consents of the author and the publisher.

Doloi (2009), Relational Partnerships: the importance of communication, trust and confidence and joint risk management in achieving project success. *Construction Management and Economics*, 27(1):1099–1109.

References

Bresnen M, Marshal, N (2000) Partnering in construction: a critical review of issues, problems and dilemmas. Constr Manag Econ 18:229–237

Chan APC (2001) Time-cost relationship of public sector projects in Malaysia. Int J Proj Manag 19:223

Cheung SO, Ng TST, Wong SP, Suen HCH (2003) Behavioral aspects in construction partnering. Int J Proj Manag 21:333

DTF DOTAF (2006) Project alliancing practitioners guide. Victorian Department of Treasury and Finance, April

Doloi H (2009) Relational partnerships: the importance of communication, trust and confidence and joint risk management in achieving project success. Constr Manag Econ 27(1):1099–1109

Halman JIM, Braks BFM (1999) Project alliancing in the offshore industry. Int J Proj Manag 17:71

Jin XH, Doloi H, Gao SY (2007) Relationship-based determinants of building project performance in China. Constr Manag Econ 25:297–304

Jones D (2001) Keeping the options open: alliancing and other forms of relationship contracting with government. Build Constr Law 17:153–163

Kwan AY, Ofori G (2001) Chinese culture and successful implementation of partnering in Singapore's construction industry. Constr Manag Econ 19:619–632

Molenaar K, Washington S, Diekmann J (2000) Structural equation model of construction contract dispute potential. J Constr Eng M ASCE 126:268

Naoum S (2003) An overview into the concept of partnering. Int J Proj Manag 21:71

Ngowi AB (2007) The role of trustworthiness in the formation and governance of construction alliances. Build Environ 42:1828

Phua FTT (2006) When is construction partnering likely to happen? An empirical examination of the role of institutional norms. Constr Manag and Econ 24:615–624

Rahman MM, Kumaraswamy MM (2002) Joint risk management through transactionally efficient relational contracting. Constr Manag Econ 20:45–54

Rahman MM, Kumaraswamy MM (2005) Assembling integrated project teams for joint risk management. Constr Manag and Econ 23:365–375

Ross J (2006) Project alliancing practioners' guide. Department of Treasury and Finance, State of Victorian, pp 1–150

Smyth H, Edkins A (2007) Relationship managment in the management of PFI/PPP Projects in the UK. Int J Proj Manag 25:232–240

Walker DHT, Shen YJ (2002) Project understanding, planning, flexibility of management action and construction time performance: two Australian case studies. Constr Manag Econ 20:31–44

Williams RG, Lilley MM (1993) Partner selection for joint-venture agreements. Int J Proj Manag 11:233

Wong PSP, Cheung SO (2005) Structural equation model of trust and partnering success. J Manag Eng 21:70

Wong SPP, Cheung SO (2004) Trust in construction partnering: views from parties of the partnering dance. Int J Proj Manag 22:437

Wood GD, Ellis RCT (2005) Main contractor experiences of partnering relationships on UK construction projects. Constr Manag Econ 23:317–325

Chapter 47
Incentive Mechanisms and Their Impact on Project Performance

Xianhai Meng

Abstract Good performance characterizes project success and value for money. However, performance problems are not uncommon in project management. Incentivization is generally recognized as a strategy of addressing performance problems. This chapter aims to explore incentive mechanisms and their impact on project performance. It is mainly based on the use of incentives in construction and engineering projects. The same principles apply to project management in other industry sectors. Incentivization can be used in such performance areas as time, cost, quality, safety and environment. A client has different ways of incentivizing his contractor's performance, e.g. (1) a single incentive or multiple incentives; and (2) incentives or disincentives or a combination of both. The establishment of incentive mechanisms proves to have a significant potential for relationship development, process enhancement and performance improvement. In order to ensure the success of incentive mechanisms, both contractors and clients need to make extra efforts. As a result, a link is developed among incentive mechanisms, project management system and project performance.

Keywords Improvement • Incentive mechanism • Project performance • Reward

47.1 Introduction

Construction and engineering projects are often large and complex. It is not uncommon for these projects to suffer from performance problems, such as time delays, cost overruns and quality defects (Sun and Meng 2009). Performance problems significantly affect the success of a project. According to Bubshait (2003), for example, a 1-day production delay in an industrial project may cost the client millions of dollars and damage his return on investment. Many studies, such as Lam (1999) and Miller and Lessard (2001), have identified unsuccessful completion

X. Meng (✉)
School of Planning, Architecture and Civil Engineering, Queen's University Belfast, Belfast BT9 5AG, UK
e-mail: x.meng@qub.ac.uk

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_17

as a major risk in construction and engineering projects. Introducing incentives provides a strategic response to this major risk (Floricel and Miller 2001). On the other hand, the lack of incentives is often a cause of poor performance in a project (Assaf and Al-Hejji 2006; Doloi et al. 2012). This is because without appropriate incentives the contractor may have less motivation to work together with the client for performance improvement.

Incentive schemes have received an increasing recognition from construction practitioners and researchers. In a project under such schemes, incentivization aligns the client's and the contractor's objectives (Chapman and Ward 2008). According to the incentivization report released by the Construction Industry Research and Information Association (CIRIA) and written by Mouchel Richmond-Coggan in 2001, the objectives of both parties are aligned through the use of appropriate performance measures and the link between the contractor's performance and the client's payment. The contractor is incentivized to increase his efforts towards the client's objectives beyond minimum contractual specifications (Rose and Manley 2010). As a result of incentivization, the contractor works quicker, cheaper and better and therefore there is a greater certainty of delivering the client's desired performance (Hughes et al. 2007). The client may expect the project to be completed in the shortest possible time, at the lowest possible cost and with the best possible quality (Arditi et al. 1997). In this case, incentivizing the contractor's performance is an inevitable choice.

The first section of this chapter explains the reasons for introducing incentive mechanisms. It is followed by the review of previous practices of performance incentives. The third section focuses on the summary of project objectives and performance indicators. How to select appropriate incentives is further discussed in the fourth section. The fifth section presents a case study on the use of incentives in a road project. The impact of incentive mechanisms is analyzed in the sixth section, providing empirical evidence for their application. Finally, this chapter ends with a model to describe how an incentive mechanism works.

47.2 Previous Practices of Performance Incentives

Performance incentivization emerged in line with the development of project management. According to Herten and Peeters (1986), the United States Department of Defense (DOD) tried several types of incentive provisions during World Wars I and II for large military acquisitions. This is probably the earliest practices of performance incentives in the project environment. During the 1960s, the DOD and the National Aeronautics and Space Administration (NASA) developed the policy for performance incentives and the guide for incentive contracting (Hildebrandt 1998). The incentive policy and guide proved to be useful for ensuring the success of large NASA projects, such as Apollo (Morris 1994). In Europe, the European Space Agency (ESA) can be described as an early advocator of establishing incentive mechanisms. One of the ESA examples was the use of incentives for the Highly

Eccentric Orbit Satellite project HEOS-1 in the late 1960s (Herten and Peeters 1986). Subsequent to the successful application in the military and aerospace sectors, performance incentives were gradually introduced into the construction industry and its projects in the global context.

Existing studies demonstrate the successful application of incentives in various construction and engineering projects across different countries. For example, Christiansen (1987) investigated a highway project with incentive provisions in the United States. Berends (2000) analyzed eight incentive projects from the oil and chemical industry in the Netherlands. Richmond-Coggan (2001) discussed 20 incentive schemes in different types of UK construction and engineering projects, such as road, tunnel, power station and waste treatment. Based on four Australian building projects, Rose and Manley (2010) provided recommendations for project clients who design and implement financial incentives. Chan et al. (2010) presented an underground railway station modification project in Hong Kong in which performance was improved through the use of incentives. In addition to various construction and engineering projects, it is possible to apply incentives to other types of projects, such as software development and research development, which can be witnessed in Gopal et al. (2003) and Seston et al. (2003).

47.3 Project Objectives and Performance Indicators

A common understanding in the construction industry is that three major objectives of the client in a project are: time, cost and quality. According to the construction contract, it is the responsibility of the contractor to meet the client's major objectives. For this reason, time, cost and quality are generally described as an "iron triangle" for project management. They are also defined as three key indicators to measure the contractor's performance or the completed project's performance. If completion on time, on budget and with the specified quality is considered as normal performance, it is possible to define completion behind schedule, over budget and with the quality lower than the specifications as underperformance and completion ahead of schedule, under budget and with the quality higher than the specifications as outperformance (Meng and Gallagher 2012). Obviously, outperformance represents performance above "business-as-usual" (Rose and Manley 2011). In contrast, performance below "business-as-usual" characterizes underperformance.

In addition to time, cost and quality, other project objectives or performance indicators have been gradually introduced into project management to reflect different stakeholders' needs and expectations. The change is described as "going beyond the iron triangle" (Toor and Ogunlana 2010). For example, Ofori (1992) identified environment as the fourth project objective or performance indicator. Safety was identified by Toor and Ogunlana (2010) as another important project objective or performance indicator. The recognition of the time-cost-quality triangle and the movement of going beyond the iron triangle are the same for projects in other industry sectors as for those in construction. The new paradigm illustrates the

necessity of measuring project success from a wider perspective. It also illustrates the importance of customizing performance measures, varying from one industry sector to another, from one project to another, and from one stakeholder to another.

47.4 Selection of Incentive Mechanisms

There are different ways of establishing incentive mechanisms. In the following, they are classified in terms of which project objective is selected for performance incentivization, whether outperformance is financially paid or not, whether incentives are contractually specified or not, whether a single incentive or multiple incentives should be selected, and whether incentives for outperformance or disincentives for underperformance are more preferable.

47.4.1 Incentivization for Different Project Objectives

Both Herten and Peeters (1986) and Bower et al. (2002) stated that an incentive mechanism can be established for time, cost, quality or safety performance. At the first step of incentivization in a project, the client should make clear which project objective needs to be incentivized. In construction practice, time and cost incentives are more commonly used than quality and safety incentives (Bubshait 2003; Meng and Gallagher 2012). Time incentives are particularly important for schedule-driven project management (Yaghootkar and Gil 2012). This means that the contractor is motivated to complete the project earlier than the target date if time objective is of great importance. Generally, a bonus is paid to the contractor for early completion. In order to benefit from early start-up, for example, the client in a case project provided by Abu-Hijleh and Ibbs (1989) set up a $3,000 bonus for each day of early completion and meanwhile there was a cap on the total bonus. When a time incentive is used in a project, the contractor has motivation to complete the project as early as possible. Even if the project cannot be completed early due to any reasons, the time incentive will help to avoid late completion.

Cost incentives represent an important mechanism to encourage cost savings and improve cost efficiency and effectiveness. This mechanism is directly linked with payment methods in construction contracts. Generally, there are three types of payment methods: fixed price contract, cost reimbursement contract and target cost contract. Fixed price contracts and cost reimbursement contracts themselves have no incentives for cost savings. However, incentives can be added to form fixed price incentive contracts and cost reimbursement incentive contracts that enable contractors to either share cost savings with clients or receive incentive fees from clients as rewards for cost saving efforts. Among existing publications, Ward and Chapman (1995) paid particular attention to fixed price incentive contracts, whereas

Al-Harbi (1998) and Berends (2000) focused on cost reimbursement incentive contracts. On the other hand, target cost contracts have cost incentives/disincentives that allow contractors to share both savings and overspendings with clients according to sharing formulas (Chan et al. 2010). This is usually called gain/pain sharing or reward/risk sharing (Bresnen and Marshall 2000). Target cost contracts have attained a growing popularity in the construction industry, which can be seen from a considerable number of publications. There are two main concerns for target cost contracting: one is the appropriate selection of sharing ratios and the other is the accurate estimation of target costs. The higher a sharing ratio is, the greater impact an incentive mechanism has on cost reduction (Badenfelt 2008). On the other hand, cost incentives only work if target costs are estimated accurately (Hughes et al. 2012).

Quality incentives can be used to encourage project completion with the quality higher than the specifications, e.g. zero defects. In a project under a quality incentive scheme, the contractor is usually awarded a bonus if he can achieve the incentive quality target. Unlike a large number of studies on time and cost incentives, few studies have reported the use of quality incentives in construction practice. Richmond-Coggan (2001) and Meng and Gallagher (2012) are two of the few studies in which empirical evidence is provided for the use of quality incentives. According to Meng and Gallagher (2012), quality incentives are less commonly used than time or cost incentives. Meng and Gallagher (2012) further found that quality incentives are seldom used individually but are often combined with time and/or cost incentives. The phenomenon can also be found in Richmond-Coggan (2001). All these explain why quality-specific incentive studies are quite limited within existing literature.

According to Bubshait (2003), safety incentives can be used for contractors to comply with safety rules and standards established by clients. If the requirements provided in safety acts and regulations are regarded as safety performance at the normal level, safety performance required by clients in their rules and standards may be higher than the normal level. Compared to quality incentives, there are a little more publications on safety incentives, such as Hinze (2002) and Gangwar and Goodrum (2005). However, these publications refer to construction firms incentivizing their workers rather than clients incentivizing contractors. On the other hand, safety incentives were not found in any of the three case projects provided by Bower et al. (2002). Similarly, Meng and Gallagher (2012) did not find any signs of safety incentives in almost all the surveyed projects. Project management today is influenced by increasing health and safety acts and regulations and contractors have to comply with the latest legislations. No need for additional mechanisms to incentivize safety performance may be the primary reason for the limited report of safety incentives in construction practice (Meng and Gallagher 2012).

In addition to time, cost, quality and safety incentives, the following two mechanisms can be established in construction practice to incentivize contractors in other performance areas:

- Environment incentives that motivate contractors to protect the environment from construction pollution (Meng and Gallagher 2012); and
- Innovation incentives that motivate contractors to make innovation for better value for money (Bubshait 2003; Leiringer 2006).

47.4.2 Achievable and Attractive Incentive Targets

Different types of incentives provide project clients with enough flexibility. A client can decide whether or not his project needs incentivization and which project objective should be incentivized: time, cost, quality, safety, environment or innovation. According to Abu-Hijleh and Ibbs (1989), on the other hand, a contractor's motivation to perform will be maximized when:

- The contractor believes that the performance at the desired level is possible;
- The contractor believes that performance improvement efforts will lead to certain positive outcomes; and
- The outcomes attract the contractor.

This means that incentive targets must be achievable and attractive. If incentive targets are not achievable, a contractor will lose the ability to perform even if incentive payment is very attractive. It is important to realize that not all incentive projects are successful. One of the possible reasons for unsuccessful incentive projects is that incentive targets are too high to be achieved. Therefore, a client has to take achievability into consideration when setting up incentive targets.

On the other hand, clients have to ensure the attraction of incentivization to their contractors. In order to achieve incentive targets, contractors need to make extra efforts, both internally and externally. Distributive justice theories suggest that the amount of rewards should be judged by the fairness relative to efforts (Rose and Manley 2011). In incentive practice, this means that rewards must be enough to attract contractors. Matching rewards with efforts explains why Badenfelt (2008) and Hughes et al. (2012) emphasized the appropriate selection of sharing ratios in target cost contracts. This is because too low sharing ratios will make gain sharing unattractive to contractors. As a result, contractors may become unenthusiastic for cost reduction. In other words, mismatch of efforts and rewards will affect the success of incentive mechanisms.

47.4.3 Financial or Non-Financial Incentives

Incentives can be financial, non-financial or a combination of both (Richmond-Coggan 2001). Financial incentives mean that outperformance must be paid through bonus, incentive fee or cost saving sharing. The basic principle of financial incentives is simply to take advantage of a contractor's general objective to maximize

his profits by giving him the opportunity to make greater profits if he can perform the contract more efficiently and effectively (Bower et al. 2002). Although most incentives may be financially based, not all incentives need financial rewards. For example, Fischer and Nunn (1992) identified recognition as an excellent non-monetary incentive. Recognition is even considered by Carrillo (2004) more important than financial rewards. Similarly, Nyström (2005) found that non-financial incentives, such as appreciation, influence and development, can also create motivation and improve efforts. According to Rose and Manley (2011), possible future work is a strong motivator of contractors' behavior. Both financial and non-financial incentives are important in practice. Compared to pure financial incentives, there may be a more positive effect if financial incentives are combined with non-financial incentives.

47.4.4 Contractual or Non-Contractual Incentives

Incentives can be contractual, non-contractual or a combination of both. Formal incentives are built into construction contracts. Generally, contractual incentives are based on financial rewards, e.g. a certain amount of bonus per day or per week for early project completion. On the other hand, non-contractual incentives are informal and usually do not rely on financial rewards. For example, recognition is not specified contractually and not paid monetarily. Rahman and Kumaraswamy (2008) took risk allocation as an example to distinguish contractual incentives from non-contractual incentives: contractual incentives may include clear and fair risk allocation in contracts, whereas non-contractual incentives may include a change in the attitude for such equitable risk allocation. Similar to financial and non-financial incentives, both contractual and non-contractual incentives are important for project management and performance improvement. In incentive practice, a combination of both may be more effective than any single one.

47.4.5 Single or Multiple Incentives

Project clients can provide time incentives for early completion, cost incentives for cost savings, quality incentives for zero defects, safety incentives for complying with stricter safety rules and standards, environment incentives for protecting the environment from construction pollution, or innovation incentives for innovation activities. The use of only one incentive is called a single incentive. For example, Abu-Hijleh and Ibbs (1989) focused on time incentives to reduce project duration. On the other hand, two or more of these incentives can be combined, either dependently or independently, to form multiple incentives (Herten and Peeters 1986). For example, Jaafari (1996) twinned time and cost in an incentive scheme for potential savings in both time and cost. The combination of different incentives

aims to make improvement in more than one performance area. Multiple incentives are complicated to manage (Bower et al. 2002). For this reason, the use of multiple incentives is a challenge to project participants (Meng and Gallagher 2012). In order to ensure the success of multiple incentives, project participants have to make much more efforts compared to a single incentive.

One project objective may have priority over another (Parker and Craig 2008). Compared to a low priority objective, more resources are usually allocated to the achievement of a high priority objective. When a single incentive is established in a project, an objective with the highest priority is selected for incentivization and this objective receives the most resources. The establishment of multiple incentives means that some objectives in a project are much more important than others. Although these important objectives are incentivized in the project, they may still have different priorities. For this reason, it is necessary to trade off these incentives so that their priorities are well reflected (Bower et al. 2002; Meng and Gallagher 2012). The trade-off between these incentives is crucial to optimal resource allocation. On the other hand, multiple incentives with the same importance are not encouraged in a real world of limited resources. Otherwise, it is not impossible to only achieve a partial success of multiple incentives or even suffer a failure of multiple incentives, which can be found in some examples provided by Richmond-Coggan (2001).

47.4.6 Incentives or Disincentives

As discussed above, outperformance or excellent performance, such as early completion, cost savings and zero defects, should be rewarded. That is why there is a need for incentivization. On the other hand, underperformance or poor performance should be penalized. The penalty of underperformance is called disincentivization, e.g. time disincentives for late completion, cost disincentives for budget overruns, and quality disincentives for major defects. According to existing studies, disincentives are also useful in practice. If an incentive is used to motivate the contractor for outperformance, a disincentive is used to demotivate the contractor for underperformance (see Fig. 47.1). The National Museum of Australia Project provided a good example of time disincentives. In this project, there was no reward for early completion, but a significant penalty was awarded even if it was

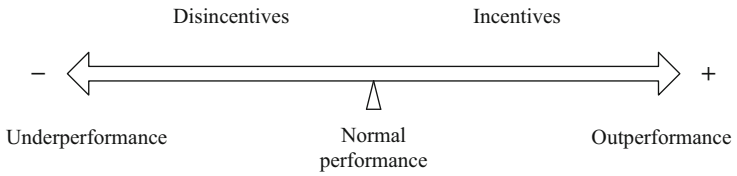


Fig. 47.1 Incentives and disincentives on a spectrum

1 day late (Walker et al. 2002). As aforementioned, fixed price contracts have no incentives for cost savings. Under a fixed price contract, the client pays a fixed amount for the total scope of work and the contractor takes the risk of overspendings completely (Al-Harbi 1998; Berends 2000). For this reason, the fixed price can be regarded as a cost disincentive.

Disincentives can be used separately from incentives. If a client only desires to ensure normal performance, such as on-time completion, pure disincentives may be enough. This can be seen from using a time disincentive but lacking a time incentive in the National Museum of Australia Project illustrated by Walker et al. (2002). On the other hand, it is possible to combine disincentives with incentives (Arditi et al. 1997; Bubshait 2003). The combination of both can be used in any performance areas, such as time, cost, quality, safety and environment. It has a dual effect on the improvement of a contractor's performance: discouraging underperformance and encouraging outperformance (Meng and Gallagher 2012). It aims to complete a project as efficiently and effectively as possible, making the shift of project performance from the negative side to the positive side across the spectrum shown in Fig. 47.1.

When incentives and disincentives are combined, how to balance them appropriately is an important question. An optimal balance depends on the relative importance of outperformance to normal performance. It also depends on the management philosophy, namely, whether rewards or penalties are more preferable. In reality, clients and contractors may have different management philosophies. According to Bubshait (2003), the majority of clients of industrial projects, such as petro-chemical projects and power projects, in Saudi Arabia would assign incentives and disincentives equally or sometimes incline to disincentives, whereas the majority of contractors would expect more incentives than disincentives. Based on the investigation of highway projects in the United States, Arditi et al. (1997) found that disincentives were generally larger in amount and accompanied incentives.

47.5 Case Study

A case study is provided in Meng and Gallagher (2012) to demonstrate the successful application of multiple incentives/disincentives in construction practice. It was based on a road project whose client was a county council in Ireland. This project was scheduled to take 22 months to complete. It was budgeted at €81,000,000. The design-build approach was chosen in this project for its delivery. Based on tender competition, a joint venture was selected by the client to take the overall responsibility for design and construction. The selected design-build contractor did the construction work by himself and subcontracted the design work to a design firm. On the other hand, another firm was employed by the client to act as the project management consultant.

Working in the public sector, the client sought cost certainty and value for money. For this reason, the client adopted target cost contracting in this project.

The target cost was finalized through the modification of the initial target cost based on value engineering in design and the negotiation between the client's management team and the design-build contractor. The use of target cost contracting established a gain/pain sharing arrangement and provided both a cost incentive and a cost disincentive. The contractor was encouraged to reduce costs as much as possible during the project. The sharing ratio was determined through the identification of each other's risks and the mutual negotiation. The determined sharing ratio was acceptable to both parties.

In addition to cost incentivization/disincentivization, incentives/disincentives were established for time, quality and environment in this project. However, there was no safety incentive/disincentive. According to the contract, a bonus would be offered to reward the contractor for early completion. On the other hand, late completion was regarded as delay damages and therefore the contractor would be charged on a daily basis. The quality disincentive was mainly reflected by defect liability clauses in the contract. If there were any avoidable defects after the defect liability period, the contractor was still required to pay the cost of rework. In addition to the defect liability, the payment to the contractor would be affected in the case of quality defects. In order to minimize the impact of construction activities on the environment, an environment incentive/disincentive mechanism was established, which encouraged the contractor to avoid the noise pollution and the pollution of nearby watercourses.

Particular attention was paid to the balance between incentives and disincentives and the trade-off among time, cost, quality and environment incentive targets. This was because the client wanted to maintain a good working relationship with the contractor and encourage the contractor to pursue best practice. During the project, the contractor made enormous efforts. The contractor selected management staff carefully so that they were experienced enough. The contractor also made full use of management techniques, such as critical path method, earned value analysis and risk register, to respond to multiple incentives/disincentives. A performance monitoring and control system was well established to ensure successful performance in different incentive areas. In response to time incentive/disincentive, the contractor introduced a 6-day workweek into site management. Bonus was passed down to staff and workers on site for overtime work or night shift.

During the project, the client and the contractor worked collaboratively together. They maintained a good working relationship throughout the project, which could be seen from the absence of claims and disputes. As a result of incentivization/disincentivization, this project was completed 2 months ahead of the schedule and €500,000 below the target cost. There were also no defects and no environmental pollution. The client was satisfied with the contractor's overall performance and benefited from better value for money. On the other hand, the contractor received incentive payment at the end of the project. The excellent performance also helped the contractor to gain a good recognition from the client. The client would like to develop a long-term business relationship with the contractor. Clearly, the use of incentives/disincentives achieved a win-win result in this project.

47.6 Impact of Incentive Mechanisms

Existing studies have provided clear evidence for the significant impact of incentive mechanisms in construction practice. Based on these studies, the impact of incentive mechanisms is summarized in this section.

47.6.1 Potential Benefits of Incentive Mechanisms

As mentioned earlier in this chapter, particular benefits of incentive mechanisms, such as aligning clients' and contractors' objectives and addressing performance problems, have been identified by some studies. They are recognized as the two primary reasons for introducing incentivization. On the other hand, Richmond-Coggan (2001) summarized potential benefits of incentive mechanisms more systematically, including:

- Alignment of each other's objectives to create a better working relationship;
- Incorporation of a structured management process;
- Encouragement of gain/pain sharing;
- Development of performance-focused contract documentation system;
- Greater chance to achieve expected outcomes for both parties; and
- Better performance in terms of time, cost, quality, safety and environment.

The summary of these benefits is based on 20 construction incentive schemes provided in the CIRIA's report written by Richmond-Coggan (2001).

47.6.2 Contribution to Relationship Development

Relationship management used to be a part of business management at the corporate level, but it becomes a new focus of project management in today's practice (Meng 2012). It describes project management from the "soft" perspective. Working relationships can be divided into inter-personal relationships within a project team and inter-organizational relationships between a project team and other project parties. Inter-personal relationships within a project team represent internal relationships, whereas inter-organizational relationships between a project team and other project parties characterize external relationships. Based on existing studies, it is found that incentivization contributes to the development of both internal and external relationships.

Traditionally, project parties treat each other as adversaries. The adversarial working relationship between construction organizations pulls their efforts in different directions. On the other hand, the use of incentives is a trigger of aligning each other's objectives. The reason behind the alignment of objectives

is that incentivization enables a contractor to share gains with his client if he can perform excellently. The alignment of objectives creates a more proactive, cooperative relationship between the parties and produces a cultural shift away from the reactive, adversarial approach (Bower et al. 2002). It also helps to reduce or avoid conflicts between the parties. Incentive mechanisms have three key roles in inter-organizational relationships: sources of extrinsic motivation, symbols of mutual trust, and generators of effective communication (Kadefors and Badenfelt 2009). With the appropriate use of incentives, an inter-organizational relationship is harmonized and the parties work together as partners (Ling et al. 2006).

Extrinsic incentivization only works when it is successfully translated into intrinsic motivation. In order to achieve the incentive targets in a project, the contractor must improve the efficiency and effectiveness and increase the productivity and predictability during construction. For this reason, it is very important for the contractor to motivate his team members so as to develop a high performance team. Motivation has proven its major influence on a contractor's team and workforce (Bubshait 2003; Meng and Gallagher 2012). A team working environment is characterized by good leadership, strong commitment, team spirit, mutual trust, effective communication between team members, and low inter-personal conflicts (Thamhain 2004). Based on the motivation throughout a contractor's team, a team working environment is well established and the whole team runs very smoothly. As a result, team members concentrate their efforts on overcoming any difficulties and achieving incentive targets.

47.6.3 Enhancement of Process Management

In an incentive project, the contractor has no choice but to improve the efficiency and effectiveness and increase the productivity and predictability during construction. In addition to the creation of a good working environment through internal and external relationship improvement, the challenge requires the contractor to enhance project management processes within his team (Meng and Gallagher 2012). Process management must be enhanced in all the incentive performance areas. From the internal process perspective, extra efforts are needed during construction for effective coordination, material and equipment testing, construction plans and methods, construction procedures, and examination of completed items (Tang et al. 2008).

Process management represents "hard" project management. In addition to the enhancement of process management in all the incentive performance areas, risk management and contract management have a great potential for strengthening the project management system. Both contractors and clients need project management. Experienced clients manage projects by themselves, whereas inexperienced clients employ independent consultants for project management on their behalf (Bennett 2003). In addition to the contractor in an incentive project, it is also important for the client to enhance project management processes. Although the client is not

involved in site management, he must pay close attention to risk management, contract management, information management and other management processes during the project.

47.6.4 Impact on Performance Improvement

Existing studies have provided empirical evidence for a significant impact of incentive mechanisms on performance improvement in construction. For example, Arditi et al. (1997) compared 28 incentives/disincentives (I/D) highway projects and 29 non-I/D highway projects in terms of time performance. The comparative analysis results shows that 90.3 % of I/D projects were completed on time or earlier, whereas 41.4 % of non-I/D projects were completed on or ahead of schedule. By comparison, the impact of time I/D on time performance improvement is evident. Based on a survey with 60 responses, Meng and Gallagher (2012) compared I/D projects and non-I/D projects in terms of time, cost and quality performance. It is found in this study that the significant impact of incentive mechanisms on performance improvement in different areas is at different levels. For example, the impact of time I/D on time performance improvement is more significant than the impact of quality I/D on quality performance improvement.

Existing studies on the use of multiple incentives are quite limited. Among the limited number of studies on the use of multiple incentives, case study is the main methodology. For example, Bower et al. (2002) provided three case projects and Walker et al. (2002) provided one case project, in which the use of multiple incentives was successful. On the other hand, Meng and Gallagher (2012) contributed to the performance comparison between a single incentive and multiple incentives in the surveyed projects. The findings of this study include (1) a single incentive is more effective for performance improvement in a certain area compared to multiple incentives; and (2) multiple incentives are useful for the overall improvement of project performance although they are not as effective as a single incentive for performance improvement in a certain area. The findings suggest that project clients should think about the selection of a single incentive or multiple incentives carefully before making incentivization decisions.

47.7 How an Incentive Mechanism Works

Based on the above discussion in this chapter, Fig. 47.2 is developed to illustrate how an incentive mechanism works. As shown in this figure, incentivization/disincentivization is the input of a project. On the other hand, project performance in terms of time, cost, quality, safety and environment is the output. There are two key aspects in a project management system: one is relationship management in soft, and the other is process management in hard. With the appropriate use

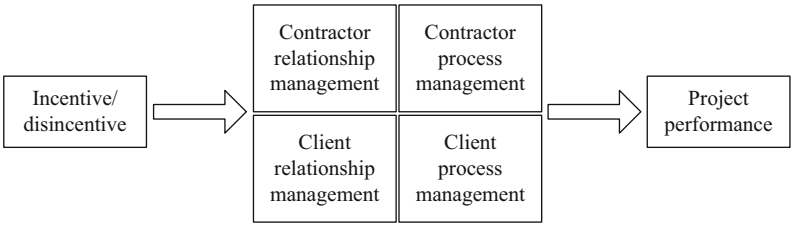


Fig. 47.2 Operation of an incentive mechanism

of incentivization/disincentivization, both the contractor and the client make extra efforts for process management during the project. On the other hand, both parties pay more attention to the working relationship with each other. In addition to the external working relationship with the client, the contractor has to motivate his team and improve the internal working relationship. Consequently, the development of working relationships and the enhancement of management processes contribute to the achievement of excellent performance.

47.8 Conclusions

Incentivization is an important mechanism for improving project performance in terms of time, cost and quality. In addition to time, cost and quality, incentive mechanisms can be used for the improvement of project performance in other areas, such as safety, environment and innovation. The appropriate use of incentives contributes to the alignment of different parties' objectives in a project and the integration of each other's efforts. It is possible for a project client to incentivize his contractor in a single performance area or in multiple performance areas. By comparison, multiple incentives are complicated to manage and therefore there is a need for more efforts. In contrast to incentivization for outperformance, disincentivization is useful to deal with underperformance. A project client can use incentives, disincentives or both to influence his contractor's behavior. Generally, the combination of incentivization and disincentivization has a more significant effect on project performance. Incentives/disincentives do not link project performance directly. Therefore, the use of incentives/disincentives does not necessarily mean a success. In order to ensure the success of incentives/disincentives, two important issues should be highlighted during a project: one is the development of working relationships and the other is the enhancement of management processes.

References

Abu-Hijleh SF, Ibbs CW (1989) Schedule-based construction incentives. J Constr Eng M 115(3):430–443

Al-Harbi KMA (1998) Sharing fractions in cost-plus-incentive-fee contracts. Int J Proj Manag 16(2):73–80

Arditi D, Khisty J, Yasamis F (1997) Incentive/disincentive provisions in highway contracts. J Constr Eng M 123(3):302–307

Assaf SA, Al-Hejji S (2006) Causes of delay in large construction projects. Int J Proj Manag 24(4):349–357

Badenfelt U (2008) The selection of sharing ratios in target cost contracts. Eng Constr Archit Manag 15(1):54–65

Bennett FL (2003) The management of construction: a project life cycle approach. Spon Press, London

Berends TC (2000) Cost plus incentive fee contracting: experiences and structuring. Int J Proj Manag 18(3):165–171

Bower D, Ashby G, Gerald K, Smyk W (2002) Incentive mechanisms for project success. J Manag Eng 18(1):37–43

Bresnen M, Marshall N (2000) Motivation, commitment and the use of incentives in partnerships and alliances. Constr Manag Econ 18(5):587–598

Bubshait AA (2003) Incentive/disincentive contracts and its effects on industrial projects. Int J Proj Manag 21(1):63–70

Carrillo P (2004) Managing knowledge: lessons from the oil and gas sector. Constr Manag Econ 22(6):631–642

Chan DWM, Lam PTI, Chan APC, Wong JMW (2010) Achieving better performance through target cost contracts: the tale of an underground railway station modification project. Fac 28(5/6):261–277

Chapman C, Ward S (2008) Developing and implementing a balanced incentive and risk sharing contract. Constr Manag Econ 26(6):659–669

Christiansen DL (1987) An analysis of the use of incentive/disincentive contracting provisions for early project completion. Trans Res B 212:69–76

Doloi H, Sawhney A, Iyer KC, Rentala S (2012) Analysing factors affecting delays in Indian construction projects. Int J Proj Manag 30(4):479–489

Fischer GW, Nunn NP (1992) Non-monetary incentives: it can be done. J Manag Eng 8(1):40–52

Floricel S, Miller R (2001) Strategizing for anticipated risks and turbulence in large-scale engineering projects. Int J Proj Manag 19(8):445–455

Gangwar M, Goodrum PM (2005) The effect of time on safety incentive programs in the US construction industry. Constr Manag Econ 23(8):851–859

Gopal A, Sivaramakrishnan K, Krishnan MS, Mukhopadhyay T (2003) Contracts in offshore software development: an empirical analysis. Manag Sci 49(12):1671–1683

Herten HJ, Peeters WAR (1986) Incentive contracting as a project management tool. Int J Proj Manag 4(1):34–39

Hildebrandt GG (1998) The use of performance incentives in DOD contracting. Defense Technical Information Center, Fort Belvoir

Hinze J (2002) Safety incentives: do they reduce injuries? Pract Period Struct Des Constr 7(2): 81–84

Hughes W, Yohannes I, Hillig JB (2007) Incentives in construction contracts: should we pay for performance? In: Proceedings of CIB World building congress, Cape Town, pp 2272–2283

Hughes D, Williams T, Ren Z (2012) Is incentivisation significant in enduring successful partnered projects? Eng Constr Archit Manag 19(3):306–319

Jaafari A (1996) Twinning time and cost in incentive-based contracts. J Manag Eng 12(4):62–72

Kadefors A, Badenfelt U (2009) The roles and risks of incentives in construction projects. Int J Proj Organ Manag 1(3):268–284

Lam PTI (1999) A sectoral review of risks associated with major infrastructure projects. Int J Proj Manag 17(2):77–87
Leiringer R (2006) Technological innovation in PPPs: incentives, opportunities and actions. Constr Manag Econ 24(3):301–308
Ling FYY, Rahman MM, Ng TL (2006) Incorporating contractual incentives to facilitate relational contracting. J Prof Issues Eng Educ Pract 132(1):57–66
Meng X (2012) The effect of relationship management on project performance in construction. Int J Proj Manag 30(2):188–198
Meng X, Gallagher B (2012) The impact of incentive mechanisms on project performance. Int J Proj Manag 30(3):352–362
Miller R, Lessard D (2001) Understanding and managing risks in large engineering projects. Int J Proj Manag 19(8):437–443
Morris PWG (1994) The management of projects. Thomas Telford, London
Nyström J (2005) The definition of partnering as a Wittgenstein family-resemblance concept. Constr Manag Econ 23(5):473–481
Ofori G (1992) The environment: the fourth construction project objective? Constr Manag Econ 10(5):369–395
Parker D, Craig M (2008) Managing projects, managing people. Palgrave MacMillan, South Yarra
Rahman MMR, Kumaraswamy MM (2008) Relational contracting and teambuilding: assessing potential contractual and non-contractual incentives. J Manag Eng 24(1):48–63
Richmond-Coggan M (2001) Construction contract incentivisation schemes: lessons from experience. CIRIA, London
Rose T, Manley K (2010) Client recommendations for financial incentives on construction projects. Eng Constr Archit Manag 17(3):252–267
Rose T, Manley K (2011) Motivation toward financial incentive goals on construction projects. J Bus Res 64(7):765–773
Seston E, Hassell K, Cantrill J, Nicolson M, Noyce P, Schafheutle E (2003) Experiences of establishing and maintaining a community pharmacy research network. Prim Hlth C Res Dev 4(3):245–255
Sun M, Meng X (2009) Taxonomy for change causes and effects in construction projects. Int J Proj Manag 27(6):560–572
Tang W, Qiang M, Duffield CF, Young DM, Lu Y (2008) Incentives in the Chinese construction industry. J Constr Eng M 134(7):457–467
Thamhain HJ (2004) Linkages of project environment to performance: lessons for team leadership. Int J Proj Manag 22(7):533–544
Toor SR, Ogunlana SO (2010) Beyond the 'iron triangle': stakeholder perception of key performance indicators (KPIs) for large-scale public sector development projects. Int J Proj Manag 28(3):228–236
Walker DHT, Hampson K, Peters R (2002) Project alliancing vs project partnering: a case study of the Australian National Museum Project. Supply Chain Manag 7(2):83–91
Ward SC, Chapman CB (1995) Evaluating fixed price incentive contracts. Int J Manag Sci 23(1):49–62
Yaghootkar T, Gil N (2012) The effects of schedule-driven project management in multi-project environments. Int J Proj Manag 30(1):127–140

Chapter 48
Drivers of Complexity in Engineering Projects

Marian Bosch-Rekveldt, Hans Bakker, Marcel Hertogh, and Herman Mooi

Abstract This chapter investigates drivers of complexity in engineering projects. Based upon literature and empirical data, the TOE (technical, organizational, external) framework is developed, which captures the drivers of complexity in engineering projects. The empirical data was gathered by means of case studies in which interviews were held with three persons of six different projects. The resulting TOE framework consists of elements related to technical aspects, organizational aspects and external aspects of the project, all potentially contributing to project complexity. This chapter shows that organizational aspects can be considered as the particular drivers of project complexity. The interviewees seem to be well educated to deal with technical aspects; external aspects seem harder to recognise.

Keywords Complexity drivers • Engineering projects • Project complexity • Project management

48.1 Introduction

Complexity of projects played an important role in the project management debate in literature and at project management conferences the last decade (Bosch-Rekveldt 2011; Williams 2002; Geraldi and Adlbrecht 2007; Hass 2007; Dombkins and Dombkins 2008; Bosch-Rekveldt and Mooi 2008; Maylor et al. 2008; Vidal and Marle 2008; Bosch-Rekveldt et al. 2011, 2009; Hertogh et al. 2008). The interest in "complexity" is fed by the assumption that one of the reasons for project failure would be the increasing complexity of projects and underestimation of this complexity (Williams 2002, 2005; Neleman 2006). The complexity of projects is assumed to increase as a result of rapid changes in environment, increased product complexity

M. Bosch-Rekveldt (✉) • M. Hertogh
CITG, Delft University of Technology, Delft, The Netherlands
e-mail: m.g.c.bosch-rekveldt@tudelft.nl; m.j.c.m.hertogh@tudelft.nl

H. Bakker • H. Mooi
3ME, Delft University of Technology, Delft, The Netherlands
e-mail: h.l.m.bakker@tudelft.nl; h.g.mooi@tudelft.nl

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_18

and increased time pressure (Williams 1999) and this complexity heavily hampers the successful completion of projects (Hertogh and Westerveld 2010). Therefore research has been undertaken to better understand project complexity. A common understanding of the concept of project complexity is however lacking and research even expressed the need for "*the development of new models and theories which recognize and illuminate the complexity of projects and project management, at all levels*" (Winter et al. 2006, pp. 642).

This chapter describes drivers of complexity in engineering projects according to literature and according to project professionals.[1] After presenting the notion of complexity in Sect. 48.2, a case study set-up is described in Sect. 48.3. Results of the case study are presented in Sect. 48.4, after which the drivers of complexity in engineering projects are discussed by means of a framework to grasp project complexity in Sect. 48.5.

48.2 Project Complexity According to Literature

This section introduces and details concepts of project complexity from literature. First, project complexity definitions are explored. Next, concepts on project complexity are presented. This section concludes with a discussion on the literature findings and explains our view on project complexity.

48.2.1 Defining Project Complexity

Van der Lei et al. (2010) provide an overview of complexity in multi-actor systems research. In their research, they distinguish the concept of complexity in the context dimension and the concept of complexity in the actor dimension. This distinction was based on systems theory (Waldrop 1992), which says that complex systems consist of many actors that continuously interact with a physical/technical environment with an emergent character. Earlier, complex systems were already defined as systems that consist of a large number of components that heavily interact with each other (Simon 1962). Following these definitions of a complex system, a project can be considered, and often is considered a complex system (Whitty and Maylor 2009).

However, in line with the work of Geraldi, no clear, unambiguous definition of complexity of projects, or projects in a complex environment, was found in literature at this stage (Geraldi 2008). Although the complexity of projects and their environment obviously influences important decisions on and in project management,

[1]This chapter is largely based on the partial content of a dissertation and a preceding conference paper (Bosch-Rekveldt 2011; Bosch-Rekveldt et al. 2009).

complexity as such is often taken intuitively or from previous experiences. Project complexity by definition has a subjective character (Hertogh and Westerveld 2010). Despite the inherent difficulty of defining complexity and the different views on complexity (Flood 1990), a high level definition of project complexity should include structural, dynamic and interaction elements (Whitty and Maylor 2009). Describing projects as complex adaptive systems or socially constructed entities (Cicmil et al. 2006), complexity in projects could then be considered to be related to such structural elements, dynamic elements and interaction of these; broader than the technical or technological domain.

According to the College of Complex Project Managers and Defence Materiel Organisation (DMO) of Australia complex projects can be distinguished from traditional projects in the following aspects (DMO 2006): disorder, instability, uncertainty, irregularity and randomness. In one word: dynamics. A distinction however should be made between complex projects and project complexity, also referred to as the complexity of a project; the first is a specific class of projects (namely the complex ones) and the latter focuses on what aspects make a project complex.

48.2.2 Concepts of Project Complexity

What are well-known literature concepts of project complexity? The goals and methods concept (Turner and Cochrane 1993) classifies projects according to whether the goals of the project are well defined or uncertain and whether the methods to achieve these goals are well defined or uncertain. Baccarini published a review on the concept of project complexity in the construction industry, proposing the following objective measure of project complexity: *"Project complexity consists of many varied interrelated parts and can be operationalized in terms of differentiation and interdependency"* (Baccarini 1996, pp. 202). Complexity as a project characteristic is distinguished from other project characteristics such as size and uncertainty. Both organizational and technological complexities are further elaborated by differentiation and interdependencies (Williams 1999).

Williams (1999) further operationalized the concepts of Baccarini and Turner. To investigate aspects of project structural complexity, measures for product complexity which influence project complexity are described. Project complexity is influenced differently by various types of interdependency, such as pooled, sequential and reciprocal. It is suggested that concurrent engineering is causing more reciprocal interdependency, adding to a project's complexity. Further dimensions of structural complexity include multi-objectivity and multiplicity of stakeholders. Williams assumes that uncertainty adds to the complexity of a project and therefore can be considered as a dimension of project complexity. Adding the concept of Turner and Cochrane (uncertainty in goals' and methods' definition) to Baccarini's concept of project complexity, as well as adding interactions between complexity

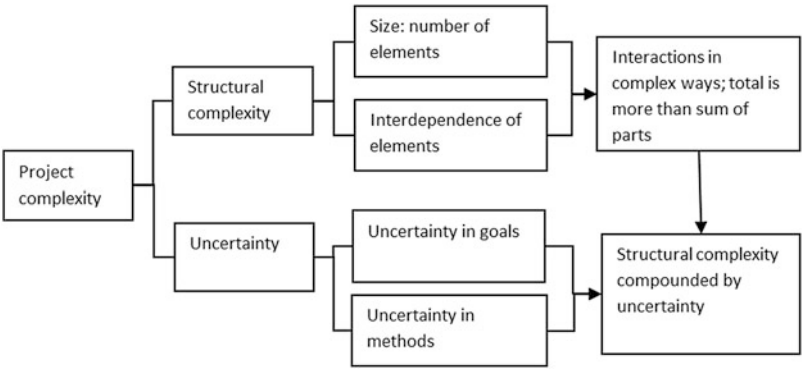


Fig. 48.1 Overview of dimensions of project complexity (Williams 2002)

and uncertainty, leads to the model given in Fig. 48.1 (Williams 2002). Note the similarity of structural complexity and the definition of a complex system as provided by Simon about half a century ago (Simon 1962) .

Xia and Lee (2004) measured the complexity of information systems (IS) development projects along two dimensions: organizational/technical and structural/dynamic. They concluded amongst others that complexity in IS development projects has a multidimensional nature.

Whereas the authors mentioned above focused on "structural complexity" and "uncertainty", also softer aspects and influences from the environment are assumed to influence project complexity (De Bruijn et al. 1996; Jaafari 2003; Geraldi and Adlbrecht 2007). Geraldi further developed the Williams concept earlier described and distinguished the complexity of fact and the complexity of faith (Geraldi and Adlbrecht 2007) as well as the complexity of interaction. The complexity of interaction, taking place at the interfaces between people and organizations, includes aspects like politics, ambiguity and empathy (Geraldi 2009), which are considered the softer aspects that contribute to the overall project complexity.

De Bruijn et al. (1996) also paid explicit attention to softer aspects related to project complexity. They assumed that project complexity would break down into technical, social and organizational complexity. Here technical complexity was assumed to be related to amongst others technological uncertainty, dynamics and the uniqueness of the project. Organizational complexity was assumed to be related to amongst others the organization structure, the project team, and the actors involved. Social complexity referred to (again) actors involved, their interests and the risks and consequences of the project in relation to its environment. Also other studies indicated the environment as an important contributor to project complexity (Jaafari 2003; Xia and Lee 2005; Mason 2007). Also Antoniadis et al. (2011) favour an approach in which complexity is considered broader than just technical complexity: they showed the importance of recognizing socio-organizational complexity.

The Williams' model, introduced above (Williams 2002), primarily covers the technical aspects and to a lesser extent the organizational aspects and aspects related to the environment, although the number of stakeholders involved, their diversity and interdependency could be considered as being part of structural complexity. Integrating Williams' model in a broader model taking into account these "softer" aspects is therefore suggested.

48.2.3 Discussion

The leading theoretical concept of project complexity is built upon the concepts of uncertainty in goals and methods, structural complexity by differentiation and interdependency and interactions between the different aspects (Fig. 48.1). Literature seems to dominantly focus on aspects of technical complexity, although the importance of the softer aspects of complexity is recognized. Literature mostly describes complexity at relatively high, abstract level. Although it expresses common sources of complexity, such as uncertainty and number and interdependencies of elements (Williams 2002), it lacks a real operationalization of project complexity, which would be helpful in order to better manage project complexity.

In the current research, project complexity is broken down into at least technical complexity, organizational complexity and external complexity (T,O,E). With external complexity we refer to complexity related to the direct environment of the project, physical as well as relational (such as location and stakeholders).

48.3 The Case Study Setup

In order to further operationalize project complexity, empirical research was undertaken to investigate the concept of project complexity from the perspective of project professionals working on engineering projects in the process industry. Exploratory case studies were performed to answer the research question: *What is project complexity as experienced by project professionals?*

48.3.1 Case Study Design

The chosen unit of analysis was a completed project in the process engineering industry: projects with the aim to develop and/or construct and/or modify a certain asset or facility. The project was taken in its wide definition: it covers all activities from initiation to close-out (including the feasibility (or scouting) phase, front-end development, implementation and close-out/handover, but excluding operations and maintenance of the facility built).

A multiple-cases embedded design was followed (Yin 2002): six cases were studied, each consisting of interviews with three different persons involved. Findings were combined with the study of written project documentation (such as official reports and project archives).

The six projects were selected from within one major company in the Dutch process industry, active member of the NAP network.[2] This company was selected because of its size, which enabled inclusion of very different types of projects from within one company, the well-developed project management procedures and the positive attitude towards further professionalization of project management. The choice to perform the case study within one company, with a well-developed project management process, limits variation across the cases in the standard front-end activities to be applied. Thereby the main phenomenon under study (project complexity and how it was dealt with) can be better explored.

Semi-structured interviews were held to investigate what elements in a project contributed to the project's complexity and how project complexity was dealt with. These interviews were held with the project managers of these six projects, a project team member (lead process engineer, lead project engineer, control manager, or engineering manager) and an owner representative [future site/plant owner, asset development manager (ADM), managing director]. For one of the cases (case 6), no owner representative was involved in the interviews. Instead, for case 6 two project managers were included since the first was replaced during the project. In total, eighteen interviews were held.

48.3.2 Case Study Protocol

To increase validity of the study, a case study protocol was followed. The foreseen participants were asked to participate in the interview with a short letter of information. All project professionals approached were willing to participate in the research. Before performing the interviews, project documentation such as progress reports and close out reports were studied.

All interviews were taped with permission of the interviewees. The interviews were following the same list of base questions, but there was room to further deepen the answers of the participants. The interviews included questions related to project performance, activities during front-end development and project complexity. The transcripts of the interviews, made by the interviewer, were approved by the interviewee before starting further analysis. In this chapter only the questions and answers related to "project complexity" are discussed.

[2]The NAP network is a competence network of the Dutch process industry, see http://www.napnetwork.nl/.

Regarding project complexity, interviewees were asked to express their ideas on project complexity and whether or not they considered their project as complex (and in what way). Only after obtaining their initial ideas on project complexity, in subsequent questions several potential areas of project complexity were introduced (commercial, economical, organizational, political, technical, and health, safety and environment) and it was asked whether any elements in the project had contributed to project complexity in that area. These potential areas of project complexity, serving as wide categories, were following the in-company risk identification model because this framework allowed a broad approach to the concept of complexity. Even more important; the interviewees were already familiar with the framework. Also, the interviewees were asked to assess the project's complexity on a 1–5 scale (1 = least complex, 5 = most complex) in the three main areas (technical, organizational, external).

Finally, the participation of the interviewee in the review process was asked for and any other business could be discussed, if this was felt appropriate. Some questions were considered more important than others and these questions got preferred attention during the interviews and the analysis.

48.3.3 Case Selection

Based on Yin a replication logic was used for case selection (Yin 2002). This information oriented strategy was chosen in order to "*maximize the utility of information from small samples and single cases*" (Flyvbjerg 2006, pp. 230). The cases together, summarized in Fig. 48.2, covered both successful *and* less successful projects in terms of meeting budget and schedule estimates and delivering

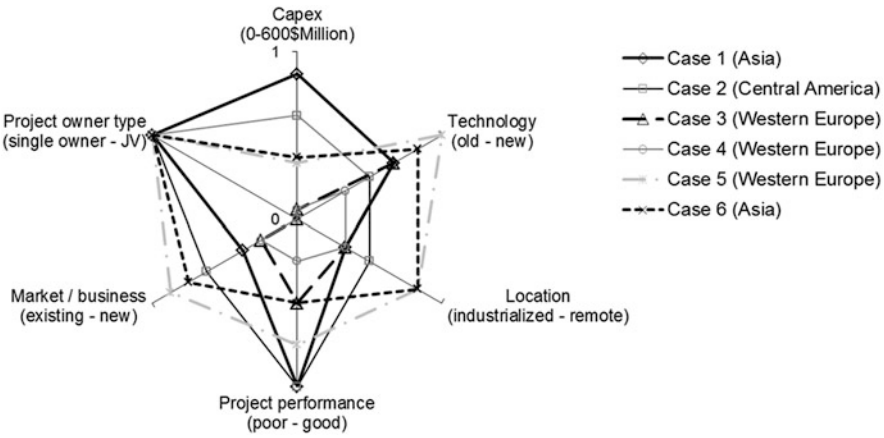


Fig. 48.2 Summary of the selected cases 1–6

according to technical specifications (*project performance: poor–good)*. A range of project types in existing or new markets was included, such as innovative projects, construction of facilities and new businesses (*market/business: existing–new*). Technology involved in the projects ranged from old/proven technology to new/unproven technology (*technology: old–new*). The capital expenditure (*Capex: 0–600 $Million)* of these projects ranged between 20 and 600 M$. Different geographical areas were covered (Europe, Asia and Middle-America) and the project location varied between industrialized and remote areas (*location: industrialized–remote*). The projects differed in project ownership; e.g. from 100 % owned to Joint Venture (JV) partnerships with partial ownership (*project owner type: single owner–JV*). Figure 48.2 shows the broad variety in project characteristics for the projects included in the case studies. The selected cases were all completed, rather recent and data availability was checked upfront.

48.3.4 Data Analysis

Qualitative analysis of the cases was done per case as well as across the different cases, focusing on the different perspectives of the interviewees on the project's complexity. All data was gathered in one database, combining the answers of the interviewees with information regarding the background of the participants and information regarding the project.

Per case, a general picture of the project was sketched based on the written information and an overview of the interview results. After the writing of these narratives (about four A4s per interview), the actual analysis took place by comparing mentioned complexity elements. The answers on the interview questions were analysed by a qualitative comparison, and differences and similarities were explained. Interviewees were not aware of the answers of their colleagues. After the in-depth single case analysis, a cross case analysis was performed focusing on an overall comparison of the results and exploring trends across the single case results (Miles and Huberman 1994).

48.4 Case Results

A summary of the partial case results is given in Table 48.1. Subsequent sections present and explain the case results in more depth.

Table 48.1 Summary of the case results

Project	Perspective of	Project complex?	Scores (1–5)[a]		
			Technical	Organizational	External
1	Project manager	Yes	1	2	2
1	Project controls manager	Yes	2	2	2
1	Future site owner	Yes	3.5	4	2
2	Project manager	Yes	2.5	4	2.5
2	Engineering manager	No	2	3	4
2	Project owner	No/yes	3	4	5
3	Project manager	Yes	1	5	3
3	Process design/ADM	No	2	4	2
3	Future plant owner	No	2	5	1
4	Project manager	Yes	4	4	4
4	Lead process engineer	Yes	3	4	1
4	Asset development manager	Yes/no	4.5	3.5	2.5
5	Project manager	Yes	4	3	4
5	Lead engineer	Yes	4	2–5[b]	2
5	Managing director	No	3	4	2–4[c]
6	Project manager	Yes	4	4	3–4
6	Project manager front-end	Yes	3	2	4
6	Control manager	Yes	5	5	4

[a]1 = low complexity, 5 = very high complexity
[b]Perspective owner's organization: 2, perspective contractor: 5
[c]Project execution phase: 2, front-end phase: 4

48.4.1 Case 1: Design, Construct and Start-Up of a Chemical Plant—Good Project Performance

The objective of project 1 was to design, construct and start up a new chemical plant in South-East Asia, in an industrialized environment. The plant was a copy of an existing plant and mainly proven technology was used, with some smaller unproven parts. The project was owned by a two partner JV that was established for this project. The project team consisted of about 55 members in the owner's team. There was close co-operation with the engineering contractor, who worked under a lump sum plus incentive fee EPCM (Engineering, Procurement, Construction, Management) contract. The contractor's team consisted of about 200 members. A maximum of around 4,000 workers were onsite during peak time. The planned capital expenditure of the project was about 520 M$ and it was a cost driven project. The project was considered as existing business for the company. Based on project documentation, it was concluded that the project performance was very good in terms of meeting budget, schedule, quality and health, safety and environment (HSE) requirements.

In view of the project manager, project 1 was complex in terms of size: the number of people involved in the project but also the physical size of the construction site. He however indicated that it was still manageable and as such, he didn't experience the project as being complex while executing. Further, he stressed the importance of the people in the project. He considered the organizational aspects in the project most complex. Because it was a JV project, there were two home offices and the project manager had to communicate with the two company boards.

In view of the project controls manager, project 1 was complex in terms of being organized in a JV which required approval of the other JV partner on allocation models, agreed budgets, etc. The construction of the facility as such he did not consider complex since it was just a copy of an existing facility. The aspect contributing most to the project complexity was the fact that the project was done with a partner in a JV structure.

In view of the future site owner, project 1 was complex in terms of having a JV partner, resulting in extra interfaces. Also at the contractor side, a JV structure was in place with split responsibilities, introducing additional interfaces. Further, complex arrangements had to be made with feedstock suppliers, resulting in extra logistic interfaces and dependencies. The aspect contributing most to the project's complexity was related to the many parties involved at both owner and contractor's side.

All three interviewees indicated that the organizational aspects contributed most to project complexity, particularly the fact that the project was run by a JV. Based on their scores, they did not really consider the project complex, with the majority of their scores being 2. Note that all three were very experienced employees and that they were involved in both front-end development and project execution.

Although at first sight, project 1 could appear to be very complex [working with a JV partner, large CAPEX (capital expenditure), a large number of people involved]; this wasn't experienced as such by the experienced interviewees. They indicated that the project was complex, but only to a small extent. From the interviews, the value of the integrated team became apparent. In this project a well-integrated project team was realized, with people from the contractor and the two owner's organization working closely together in the FED (front-end development) phase as well as during project execution. Note the (accepted but apparent) dominance of the project manager's company in the JV. Still the presence of the JV did considerably contribute to the complexity of project 1: additional effort was spent to align the parties involved.

48.4.2 Case 2: Development and Construction of a New Facility—Good Project Performance

The objective of project 2 was to develop and build a large facility in Central America, in a rather industrialized environment. The facility consisted of proven technology. The project was owned by a two partner JV that was established for

this project. The two partners both had a background in the oil and gas industry. The project team consisted of about 22 members (12 expats and 10 locals) in the owner team. The maximum number of workers on site was around 1,500. The planned capital expenditure was about 370 M$ and the project was schedule driven. The owner's organization had neither experience with building such a facility in that country nor with the engineering contractor involved, although they did have experience with this type of asset. The project was therefore considered as considerable new business for the company. The contractor worked under an EPCC (Engineering, Procurement, Construction and Commissioning) lump sum contract. Based on project documentation, it was concluded that the project performance was very good in terms of meeting budget, schedule, quality and HSE requirements.

The project manager considered project 2 as complex in several aspects: multiple shareholders with non-aligned objectives, technically complex because of the physical size of the site location and its consequences for logistics. Further, in his view the project included a complex technical process, in which different steps had to be taken in a specific order with different materials and specialisms involved, high quality requirements and limited specialized resources available. The aspect that contributed most to the project complexity, according to the project manager, was related to the different stakeholders involved.

In contrast to the project manager, in view of the engineering manager, project 2 was not complex: the technical scope he didn't consider as complex. He considered the interfaces little more complex; particularly the relations with stakeholders and clients in the country where project 2 was performed. He did not consider the project as complex, but still the most contributing element to the complexity of the project would be some political aspects together with the local stakeholders that he found were difficult to influence.

In view of the project owner project 2 was not complex in comparison with other projects. He would not consider project 2 complex because the players in the project were known and there were no interlinked investment decisions to be made, which he would consider really complex. The timing and joining of all project parts together just before the final investment decision, he found the most difficult.

Comparing the answers of the three interviewees, a major difference in perspective regarding project complexity was observed. The project manager, most experienced of the three interviewees for this project, found the different stakeholders with non-aligned objectives the most complex, and the project owner, who just considered the project as not being complex, nevertheless scores the project relatively high on its complexity. The opinion of the engineering manager is somewhere in between; he agreed with the project owner that the project as such was not complex, but the aspect that contributed most to the (limited) complexity was similar to the project manager, in the area of stakeholder relations.

Based on the project characteristics this project could have been perceived as considerably complex: working in a JV, new business, neither experience in the country nor with the contractor involved. The interviewees, however, showed considerable differences in their opinion on the project's complexity, ranging from not complex at all to considerably complex. Also their ideas on the direction of changes

in project complexity across the different project phases differed considerably. Their different perceptions might be related to years of experience, role in the project or involvement in the different project phases. Not all of the interviewees were involved throughout the entire project (project initiation, FED and project execution), but still the importance of continuity within the team across the different project phases was emphasised in the interviews. This might be related to the extremely late mobilisation of the owner's organization project team (apart from a few project developers) and the absence of such continuity in project 2, resulting in a slow start after FID. Maybe also the perceived complexity of the stakeholders' relations can be explained by the absence of continuity in the project team: discontinuity in the team negatively affects longer term relationships. Although this project was performed as a JV, this was not mentioned as an element contributing to the complexity of the project by any of the three interviewees: potentially because the JV partners had a similar background in oil and gas.

48.4.3 Case 3: Design and Construct of Chemical Plant—Marginal Project Performance

The objective of project 3 was to design and construct a chemical plant in an industrial area in Western Europe. The plant was a copy of an existing plant, only the layout had to be adapted for this project. Although some improvements were made with respect to the existing plant, proven technology was used, but the owner's organization didn't have much experience in this specific field. The project was fully owned by a subsidiary company of the owner's organization. The project team only consisted of the project manager (owner's organization), a project engineer (subsidiary company) and a process engineer (subsidiary company). The contractor worked under a reimbursable EPCM contract with incentive scheme. The planned capital expenditure was about 34 M$ and the project was cost driven. The project was considered as existing business for the company. Based on project documentation, it was concluded that the project performance was just acceptable: very good quality and HSE (health, safety, environment) performance was achieved against poor budget and schedule performance.

In view of the project manager, project 3 was complex in terms of bringing the different parties involved together, not in terms of technology. He considered the organizational part complex: he had experienced major differences in the ways of working, particularly within the owner's organization. The final objectives of the project were aligned, in his view, but the way to achieve those objectives was not. Further a mismatch in personal characters contributed to project complexity. The aspect that contributed most to the project complexity, according to the project manager, was related to these organizational aspects.

For the process designer [and in later stages, the project asset development manager (ADM)], project 3 would not be complex, assessed against his background and experience in the field. Still, aspects contributing most to this project's complexity would be organizational aspects such as poor co-operation and communication as well within the company as with the engineering contractor.

In view of the future plant owner, project 3 was not complex because of the many years of experience the subsidiary company had in the business. He however could imagine the project could be considered complex by somebody without experience, particularly because of its non-standard character. In view of the future plant owner: it is all in the eyes of the beholder and compared to a billion dollar project, project 3 was peanuts. The aspect that still contributed most to the project complexity, according to the future plant owner, was related to organizational aspects; particularly the link between the owner's organization, the subsidiary company and the engineering contractor. In his view, the complexity of the interfaces between these parties was underestimated. Further, he mentioned the complexity of dealing with the non-standard technical process; new for the owner's organization as well as for the engineering contractor.

The three interviewed project members fully agreed about the organizational aspects that contributed most to project complexity, although not all of them would consider project 3 complex. The aspects that all three indicated are related to management of interfaces within the project (communication and co-operation). Note that all three interviewees were very experienced and involved in both front-end development and project execution.

Looking generally at the project characteristics, this project would not be considered very complex. There was only one company involved (the owner's organization, albeit with a subsidiary company), the plant was a copy of an existing plant (although non-standard for the project owner) and the CAPEX was rather small. This opinion was shared by two of the three interviewees. However, the complexity of managing the interfaces within the company was underestimated, which became a major source of organizational complexity. Partly, the character of the organizational complexity was different than shown in the previous cases: in this project 3, complexity was also induced by the poor co-operation between the people involved. From this case, the influence of the relations in the project team on project complexity became clear: obviously there was some tension between the owner's organization and the subsidiary company, with the owner's organization providing the project manager. This tension might have influenced the perception of the interviewees on how the project's complexity developed across the different project phases. Lots of the complexities faced in this project were related to inexperience with and between parties and differences in working procedures between the parties involved.

48.4.4 Case 4: Modification of Current Facility—Poor Project Performance

The objective of project 4 was to improve operational performance at a large site in an industrialized area in Western Europe. The project was a typical brownfield project consisting of modification and extension of current equipment. Only proven technology was included in the project, but not all partners did have experience with the technology involved. A subsidiary company of the owner's organization was the owner of the project. Several departments within the company were involved including technical disciplines and more project related ones. The project team consisted of the project manager, the cost controller, the construction manager of the engineering contractor and the project manager of the engineering contractor. The contractor worked under an EPCM contract as part of a broader alliance contract. The workforce typically ranged between ~20 and ~70 people on site. The planned capital expenditure of the project was about 35 M$ and the project started as a schedule driven project. The project schedule drive, however, decreased because of unforeseen changes in the market. The project was considered existing business for the company. Based on project documentation, it was concluded that the project performance was poor in terms of meeting budget, schedule, quality and HSE requirements.

In view of the project manager, project 4 was complex in team aspects, in terms of underestimation of the technical complexity and organizational aspects like multi-phased project execution and the brownfield character. The aspect that contributed most to the project complexity, according to the project manager, was related to technical organizational aspects, particularly in the project execution phase.

The lead process engineer would not consider project 4 as technologically complex, rather he would call the project complex because of the large variety and diversity of items involved and the need to monitor all work processes related to these items. The aspect that contributed most to the project complexity, in view of the lead process engineer, was related to the involvement of multiple internal customers, again in the organizational area.

In view of the asset development manager, the project was complex because of the inclusion of lots and lots of different scope elements, all interrelated and spread over large areas of the site. Particularly brownfield projects like project 4 she considered more complex than green field projects, because of the linkages with existing systems in case of a brownfield project. The aspect most contributing to project complexity in view of the asset development manager was related to the scope definition; she indicated the scope was not complete and not well enough understood in terms of interrelations. She also mentioned the weakness of the operational implementation plan at the moment of the final investment decision. Further she mentioned the, contradicting, good front-end loading score received in IPA benchmarking (IPA 2009).

Comparing the answers of the three interviewees again organizational complexity arose as the aspect contributing heavily to project complexity. Besides, the importance of a thorough and well understood scope definition and corresponding operational implementation plan became clear: technical complexity was also rated high and should not be underestimated for this type of brownfield project. By his role, which is mostly internally focused, the lead process engineer probably had less attention for the external complexity. Underestimation of the complexity of the project scope seems a shared opinion amongst the interviewees.

Based on the project characteristics, some complexity could have been expected in the project. It was a brownfield project, i.e. there could be complicating interactions with the current site. Not all partners involved did have experience with the technology involved. On the other hand, it was existing business for the company and there were not so many people involved. Still the complexity of the project was considerably underestimated in the FED phase, particularly because of all different (small) scope elements and their unforeseen interactions, and the project showed serious underperformance. The complexity of the project was perceived to increase during the lifetime of the project, but it seems this increase is related to underestimation of complexity in earlier project phases. Also underestimation of the interaction with the current site might have played a role, e.g. the brownfield character of the project in contrast to a greenfield project in which less interaction problems would be expected.

48.4.5 Case 5: Development of a New Offshore Energy Facility—Good Project Performance

The objective of project 5 was to gain experience with the development of new offshore energy sources in Western Europe, hence in a remote environment. Unproven technology was involved. The project can be characterized as a demonstration project for companies as well as governmental parties involved. The project was owned by a two partner JV, established for this project. The two partners were complementary in terms of expertise. The contractor worked under a lump sum turnkey contract. The owner's project team consisted of 5–6 members, the project team at the contractor had about 50 members. Including the workers, at the peak there were about 130 people working on the project. The planned capital expenditure of the project was about 200 M$, partly financed by governmental subsidies. The project was both cost and schedule driven; there was a fixed price as well as a fixed schedule. Note that this twofold drive (cost & schedule!) is remarkable, but in this project it was related to the obtained subsidies. This project was considered new business for the company. Based on project documentation, it was concluded that the project performance was good in terms of meeting budget, schedule, quality and HSE requirements.

In view of the project manager, project 5 was complex in terms of the scale of the overall project. The number of parties involved was high (lots of subcontractors) and the arrangements for corporate governance were numerous (it was a JV structure both at owner's and contractor's side). Because of the contract type, the contractor had to control most of the complexity, but still the project manager considered the project complex. He could not indicate what aspect contributed most to the complexity of the project; in his view it was the combination and the number of issues that made the project complex.

In view of the lead engineer, project 5 was complex, although merely for the contractor, since all activities were outsourced and all risks were transferred to the contractor. The project complexity in project 5 resulted from the number of parties involved, parallel activities that took place with a variety of tasks, high safety demands, difficult logistics and the lack of experience of most parties involved. The aspect that contributed most to the project complexity was related to the differences in background and safety culture of the parties involved, requiring extra effort of the project team to create more HSE awareness. Also the technical complexity of the equipment was heavily contributing to the project complexity, in view of the lead engineer.

In view of the managing director, project 5 was not complex. Some elements in the project he found difficult because of dependencies and he indicated quite some elements had to come together, but all together it did not result in a complex project, despite the innovative character of the project, the low cost requirement and the hard contract negotiations. He considered the relation with the governmental parties the most difficult because of their different roles in the project.

The three interviewees were non-aligned about which aspect contributed most to project complexity. Either they thought it was highly interrelated (project manager) not complex at all (managing director) or in some aspects complex, particularly organizationally complex in perspective of the engineering contractor (according to the lead engineer). Not all were involved in all project phases, and also their years of work experience considerably differs, which might partially explain this difference. The lead engineer indicated that the project was technically complex, which was not the case in view of the managing director. This difference could be explained by the managing director having a more external focus, compared to a lead engineer.

Looking at the project characteristics, this project would be characterized as complex with complexities in various areas: new technology, new business, working in a JV, government involvement and the fact that the project was driven by *both* cost and schedule. However, because of the lump sum turnkey contract, most of the complexities were faced by the contractor, not by the project owner's employees who were interviewed. Still, the interviewees indicated the areas from which they perceived considerable complexity: political, technical, and the non-expected differences in safety culture between the parties involved. The complexity as a result of the large number of dependencies was indicated by all three interviewees.

48.4.6 Case 6: Construction of a New Facility—Marginal Project Performance

The objective of project 6 was to build, own and operate a new plant that would act as the gas supplier for an existing plant in a rural area in Asia. Implementing a new technological process was seen as a sustainable solution to keep that existing plant economically viable. New and unproven technology was included in the project. The project was owned by a two partner JV. The partners of the JV had different backgrounds (international oil company and a local, government owned company). The main contract type with the engineering contractor was EPC (Engineering, Procurement, Construction) lump-sum. The project team consisted of about 20–25 members in the owner's team. The majority of the project team members were local employees, also because of the required local content. In total, there were about 1,000 workers onsite. The planned capital expenditure of the project was about 220 M$ and it was a cost driven project. The project was considered rather new business to the company. Based on project documentation, it was concluded that the project performance was just acceptable in terms of meeting budget, schedule, quality and HSE requirements. Schedule delays and cost overruns were compensated by a successful start-up and handover.

In view of the project manager, project 6 was complex because of the non-alignment between the business objectives of the JV partners. One partner was focused on having a profitable project, run as efficient as possible, whereas the other was more focused on getting as many local people employed as possible. Further, she considered the project technically complex, with a lot of moving parts involved, resulting in some (expected) iterations in the start-up process. The aspect contributing most to the project complexity, according to the PM, was related to HSE, particularly because of the major difference in HSE standards between the owner and the local organizations. Because of the scarcity of skilled resources onsite, this required additional HSE awareness building and training.

In view of the front-end project manager, project 6 was complex due to a lack of local experience of the owner's company and the complexity of the technology. The aspect that contributed most to the project complexity in his view, was related to operating in this specific environment, with much less influence for the owner's organization than he was used to have, requiring on-going compromising between the owner's standards and what could be achieved in that environment.

The control manager considered project 6 complex in terms of not having experience with the JV partner, being the first execution project in this technical area at this location, including novel technology and a novel technical process. He could not indicate which aspect contributed most to the project complexity. In his view, the aspects were quite interlinked and influencing each other, such as cultural differences, novel technology, organization of the project team, unfamiliarity with the JV partner and experience with the contractor.

For the front-end project manager the external complexity was most contributing to project complexity. His opinion was slightly different than the similar opinions

from the two others, but they both could speak the local language, hence eliminating potential communication problems. Still they valued organizational complexity as equally high adding to project complexity as technical complexity. Note that the opinion of the front-end project manager was heavily based on his experiences in front-end.

Based on the project characteristics, this project would be assessed as potentially complex with complexities in various areas: the project took place in Asia in a rural area, involvement of a JV, involvement of new technology, considerable number of workers, required local content and new business for the company. This first view was confirmed by all the interviewees. One of the major complexities to overcome was the lack of local experience and the corresponding language problems. Once the FED PM was replaced by a PM who did have local experience and could speak the local language, this aspect of complexity was overcome. In the FED phase, the FED project manager clearly relied on the contract with the local JV partner that was deliberately drawn up to deal with this. Although contracts are there to act accordingly, this emphasis on the contractual aspects suggests that the relation between the JV partners could have been better and more "easy-going". This suggestion is supported by the fact that the business drivers (or objectives) across the JV partners were non-aligned, in view of the project manager. In this project, public and private interests conflicted.

48.5 Drivers of Project Complexity

This section analyses the interviews, grouped into perspectives of the project managers (7), the team members (6) and the owner representatives (5) on the complexity of their projects, in order to find drivers of project complexity.

48.5.1 Influence of Project Role

The view of the project manager, having main responsibility for the project, is considered most important. All project managers considered their project "complex". The aspect they considered most complex was for five of them related to organizational complexity, for one related to operating in the specific environment and for one a combination of different aspects, including organizational complexity. Hence organizational complexity prevails, in perspective of the project managers. Technical complexity is not mentioned by the project managers as contributing most to project complexity, even though some (parts) of the projects could be considered as highly innovative. The project managers, all having an engineering background, seem so well trained in the technical area that they have full grip on, and understanding of, the technical aspects.

Four of the six team members considered their project "complex". Similar to the project managers, the aspect they considered most complex was for the majority of the team members (four) related to organizational aspects. One team member stressed the linkage of different aspects being the most important and one team member considered the difficulty of influencing the local stakeholders as contributing most to project complexity. Again technical complexity was not referred to as contributing most to project complexity.

The owner representatives tended to consider their projects not complex: twice a frank "no" was scored, twice they couldn't make a clear decision and once a "yes" was scored. Despite this overall impression, they unexpectedly scored the different aspects of project complexity relatively high, compared to the other two groups (project managers and team members). This might be related to a sort of "strategic" behaviour: the written outcome was "more conservative" (higher complexity) than from the oral discussion could be concluded. It emphasizes the difficulty of an absolute interpretation of the complexity scores given. They also had very different opinions about which aspect contributed most to project complexity; some mentioned organizational aspects, some mentioned external aspects, but also technical complexity was mentioned. The latter is little surprising since the owner representatives were expected not to be concerned about the technical aspects. On the other hand, because they were more "on a distance" (and less involved in technical aspects), they might have perceived it as more technically complex.

All project managers did consider their project complex, which was not the case for all team members, neither for the owner representatives. Some sort of defence mechanism might play a role here; admitting that something is complex, protects in case of project failure (see for example case 4). However, also the opposite might play a role; your image as a project manager boosts in case you successfully deliver a highly complex project (see for example case 5).

48.5.2 The Need for a Complexity Framework

Table 48.1 summarizes the scores given by the interviewees for the areas contributing most to project complexity. These can only be interpreted per interviewee; absolute scores have no value since these are amongst others coloured by previous experience and role in the project. Thus analysing, 13 interviewees did score the organizational aspects highest, 8 interviewees did score the external aspects highest and 6 interviewees did score the technical aspects highest. This emphasized the important contribution of organizational aspects to project complexity, partly related to working in a JV, in view of the project professionals. Note that 16 of the 18 interviewees scored technical, organizational and external aspects differently, hence indicating the usefulness of distinguishing different categories.

Every person has an own view on project complexity and an own definition of project complexity—"*it is all in the eyes of the beholder*". Such a definition might be coloured by one's experiences, skills or role in the project under concern. Different perspectives make it difficult to objectify a definition of project complexity. Moreover, also the dynamic character of project complexity complicates the situation. Over different project phases, the complexity of the projects could change, which should be taken into account when assessing project complexity.

What more can we conclude from the case studies? The cases showed that a large project including lots of employees (case 1), was not perceived as complex, whereas a small brownfield project (case 4) was perceived as complex by experienced professionals. Hence the project context is important in assessing a project's complexity, as well as other than technical aspects that at first sight might be overlooked (Sauser et al. 2009; Antoniadis et al. 2011). Drivers of complexity are different for different participants and these drivers come from very different areas.

An extensive framework, listing all potential drivers of complexity, could support an inventory of project complexity taking into account different perspectives (dependent on work experience, role in the project, . . .). Such a framework could be the starting point of complexity "footprints" of projects, allowing different views of different project participants and stimulating constructive discussions.

48.5.3 A Complexity Framework: TOE

Based on the findings of the case studies as described in this chapter and subsequent research gathering additional data, amongst others a quantitative survey (Bosch-Rekveldt 2011), such a complexity framework capturing the drivers of project complexity was developed: the TOE framework, see Sect. 48.6.

The TOE framework is a framework to grasp project complexity: to create awareness of potential project complexities that could be faced in the project (in near future). The framework is to be used in early project phases, preferably by the project team rather than just the project manager and preferably more often than just at the beginning, since project complexity is (a) highly subjective, and (b) highly dynamic.

In the TOE framework, the T-elements represent the potential complexity causes in the project related to the project scope or the content of the project. The O-elements represent the potential complexity causes in the project related to the project internal organization. The E-elements represent all the potential external complexity causes in the project, related to external issues or external organizational complexities (Fig. 48.3).

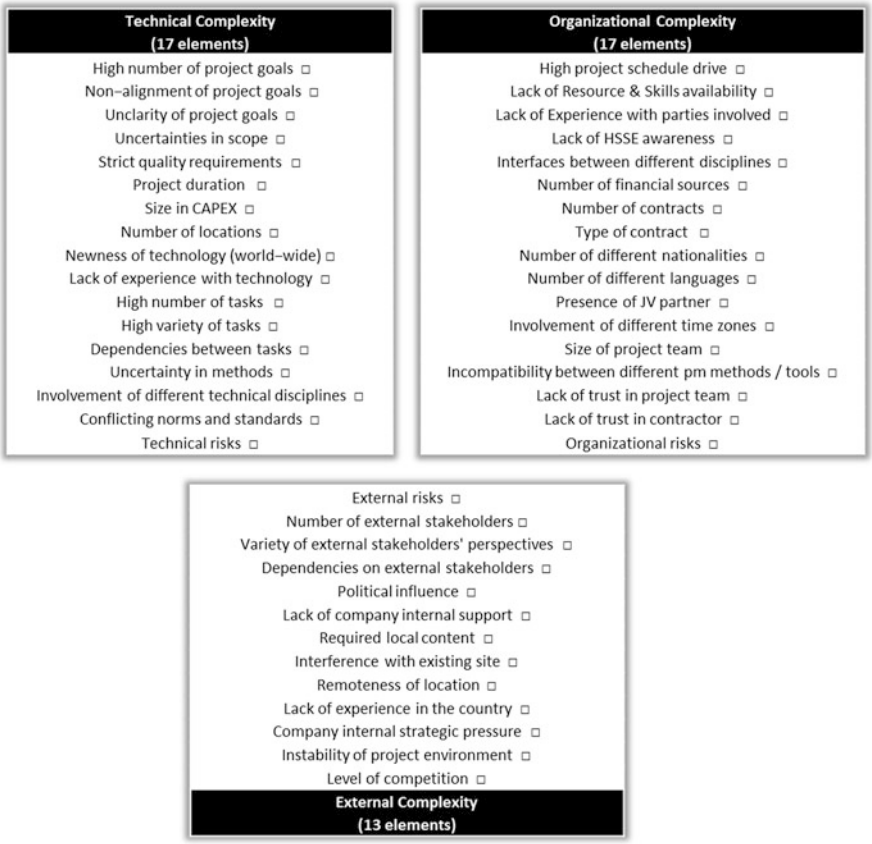


Fig. 48.3 TOE Complexity framework (Bosch-Rekveldt 2011)

48.6 Conclusions

This chapter searched for drivers of project complexity. From the case studies, it is concluded that various aspects contributed to the complexity of the projects under investigation. Organizational complexity prevailed over external complexity and technical complexity in view of the project managers and the team members. The owner representatives seemed less outspoken: next to organizational complexity also external complexity and even technical complexity were mentioned.

Overall, it was concluded that the technically well-educated interviewees seemed well prepared to deal with technical complexities, did recognize external complexities to a lesser extent and particularly faced organizational complexities in their project.

What then are the drivers of project complexity? Based on the cases discussed in this chapter and on additional research, the TOE framework presents the drivers

of complexity in engineering projects clustered in three dimensions: technical, organizational and external. Elements of the framework that were mentioned most frequently in the cases include: a lack of experience with the technology (T), uncertainties in scope (T), unavailability of resources and skills (O), incompatibilities between different project management methods/tools (O), contract types (O), lack of experience with the parties involved (O), presence of JV partner (O), number of stakeholders and variety of their perspectives (E).

The TOE complexity framework could be used to create complexity footprints of a project in order to identify the specific drivers of complexity in a particular project. It could help in identifying the different perspectives on project complexity of several involved parties, hence stimulating constructive discussion in the project. Finally, the gained knowledge and awareness could be used to take adequate management actions to improve the project's performance.

References

Antoniadis DN, Edum-Fotwe FT, Thorpe A (2011) Socio-organo complexity and project performance. Int J Proj Manag 29 (7):808–816

Baccarini D (1996) The concept of project complexity—a review. Int J Proj Manag 14(4):201–204

Bosch-Rekveldt MGC (2011) Managing project complexity. A study into adapting early project phases to improve project performance in large engineering projects. Ph.D. dissertation, Delft University of Technology, Delft

Bosch-Rekveldt MGC, Mooi HG (2008) Research into project complexity classification methods. Paper presented at the IPMA 22nd World Congress 2008, Rome

Bosch-Rekveldt MGC, Mooi HG, Verbraeck A, Bakker HLM (2009) Perspectives of project professionals on project complexity in the process and energy industry. Paper presented at the IRNOP IX, Berlin

Bosch-Rekveldt MGC, Jongkind Y, Bakker HLM, Mooi HG, Verbraeck A (2011) Grasping project complexity in large engineering projects. Int J Proj Manag 29 (6):728–739

Cicmil S, Williams TM, Thomas J, Hodgson D (2006) Rethinking project management: researching the actuality of projects. Int J Proj Manag 24(8):675–686

De Bruijn JA, De Jong P, Korsten A, van Zanten W (1996) Grote projecten: besluitvorming & management. Samson HD Tjeenk Willink, Alphen aan de Rijn

DMO (2006) Acquisition categorisation framework - policy for the categorisation of projects (Defence Materiel Organisation). Australia Government.

Dombkins D, Dombkins P (2008) Contracts for complex programs: a renaissance of process. Booksurge Publishing, Charleston

Flood RL (1990) Liberating systems theory. Plenum Press, New York

Flyvbjerg B (2006) Five misunderstanding about case-study research. Qual Inq 12(2):219–245

Geraldi JG (2008) Reconciling order and chaos in multi-project firms: empirical studies on CoPS producers, Ph.D. dissertation, University of Siegen, Siegen, Germany

Geraldi JG (2009) What complexity assessments can tell us about projects: dialogue between conception and perception. Technol Anal Strateg 21(5):665–678

Geraldi JG, Adlbrecht G (2007) On faith, fact, and interaction in projects. Proj Manag J 38(1):32–43

Hass K (2007) Introducing the project complexity model—a new approach to diagnosing and managing projects (part 1 of 2). PM World Today IX(VII):1–8

Hertogh M, Westerveld E (2010) Playing with complexity. Ph.D. dissertation, Erasmus University Rotterdam, Rotterdam

Hertogh MJCM, Baker SK, Staal-Ong PL, Westerveld E (2008) Managing large infrastructure projects. Netlipse, Baarn

IPA (2009) Independent project analysis - value improving practices workshop. http://www.ipainstitute.com/home/programs/description.aspx?id=15. Cited 10 Oct 2011

Jaafari A (2003) Project management in the age of complexity and change. Proj Manag J 34(4):47–57

Mason RB (2007) The external environment's effect on management and strategy—a complexity theory approach. Manag Decis 45(1):10–28

Maylor H, Vidgen R, Carver S (2008) Managerial complexity in project based operations: a grounded model and its implications for practice. Proj Manag J 39(suppl.):S15–S26

Miles MB, Huberman AM (1994) Qualitative data analysis: an expanded sourcebook. Sage, London

Neleman J (2006) Shell gaat diep. FEM Bus 9(4):30–34

Sauser BJ, Reilly RR, Shenhar AJ (2009) Why projects fail? How contingency theory can provide new insights—a comparative analysis of NASA's Mars climate orbiter loss. Int J Proj Manag 27(7):665–679

Simon HA (1962) The architecture of complexity. Proc Am Philos Soc 106(6):467–482

Turner JR, Cochrane RA (1993) Goals-and-methods matrix: coping with projects with ill defined goals and/or methods of achieving them. Int J Proj Manag 11(2):93–102

Van der Lei TE, Kolfschoten GL, Beers PJ (2010) Complexity in multi-actor system research: towards a meta-analysis of recent studies. J Des Res 8(4):317–342

Vidal L-A, Marle F (2008) Understanding project complexity: implications on project management. Kybernetes 37(8):1094–1110

Waldrop MM (1992) Complexity: the emerging science at the edge of order and chaos. Simon and Schuster, New York

Whitty SJ, Maylor H (2009) And then came complex project management (revised). Int J Proj Manag 27(3):304–310

Williams TM (1999) The need for new paradigms for complex projects. Int J Proj Manag 17(5):269–273

Williams TM (2002) Modelling complex projects. Wiley, London

Williams TM (2005) Assessing and moving on from the dominant project management discourse in the light of project overruns. IEEE Trans Eng Manag 52(4):497–508

Winter M, Smith CD, Morris PWG, Cicmil S (2006) Directions for future research in project management: the main findings of a UK government-funded research network. Int J Proj Manag 24(8):638–649

Xia W, Lee G (2004) Grasping the complexity of IS development projects. Commun ACM 47(5):69–74

Xia W, Lee G (2005) Complexity of information systems development projects: conceptualization and measurement development. J Manag Inform Syst 22(1):45–83

Yin RK (2002) Case study research: design and methods, vol 5. Applied social research methods series. Sage, London

Part XVI
Project Risk Management

Chapter 49
A Framework for the Modeling and Management of Project Risks and Risk Interactions

Chao Fang and Franck Marle

Abstract Nowadays, projects are facing a growing complexity and are thus exposed to numerous and interdependent risks. However, existing methods have limitations for modeling the real complexity of project risks. In this chapter, a four-phase framework for project risk management is proposed. It not only deals with project risks in terms of their probability and impact, but also brings in the modeling of risk interactions. Through identifying and assessing risks and risk interactions, a project risk network is constructed to represent the complexity of project risks. A quantitative model is then developed to describe the propagation behavior in the risk network for refining the risk analysis results. A numerical example is given to illustrate how to apply the framework in practice. The proposed approach provides more insights on project risks and risk interactions for their management, and can be used as a powerful complement to the classical methods for subsequent decision support.

Keywords Project risks • Risk interactions • Risk management • Risk network • Risk prioritization

49.1 Introduction

Project Risk Management (PRM) is an important aspect of project management. It is crucial and indispensable to the success of projects. The objectives of PRM are to increase the probability and impact of positive events, and decrease the probability and impact of negative events in the project (PMI 2008). Because of the

C. Fang (✉)
Economics and Management School, Wuhan University, Wuhan, China
e-mail: chao.c.fang@gmail.com

F. Marle
Laboratoire Genie Industriel, Ecole Centrale Paris, Chatenay-Malabry, France
e-mail: franck.marle@ecp.fr

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_19

uncertainty and potential change inherent to the nature of projects, the PRM process is iterative and goes through progressive elaboration throughout the project's life cycle. Classical PRM process is generally comprised of four major phases: risk identification, risk analysis, risk response planning, and risk monitoring and control.

Nowadays, projects are facing a growing complexity, in both their structure and context. In addition to the organizational and technical complexities described in Baccarini (1996), project managers have to consider a growing number of parameters (e.g., environmental, social, safety, and security) and a growing number of stakeholders, both inside and outside the project. The existence of numerous and diverse components which are strongly interrelated is one of the main characteristics of complexity (Chu et al. 2003). Project systems are then in essence complex and this complexity is undoubtedly a major source of risk, since the project organization may not be able to cope with it. As a consequence, the complexity of projects leads to the increasing complexity of project risks which are associated with the components.

Many risk management methods and associated tools have been developed until now in the context of project management, in both academia and industry. A typology of project risk management is introduced in Zhang (2011). The methods are qualitative and/or quantitative approaches, often based on the two concepts of probability and impact (or severity) of the risky event. For example, Chap. 51 of this handbook discusses different ranking indices based on risk characteristics for risk mitigation. However, many of these methods independently evaluate the characteristics of risks, and focus on analysis of individual risks. Risks are usually listed and ranked by one or more parameters (Baccarini and Archer 2001; Chapman and Ward 2003; Ebrahimnejad et al. 2010). To comprehensively understand a risk, it is helpful to identify its causes as well as its effects. Several tree-structure methods include this principle, but they still concentrate on a single risk for simplifying the problem (Carr and Tah 2001; Heal and Kunreuther 2007). For instance, Failure Modes and Effects Analysis (FMEA) consists in a qualitative analysis of dysfunction modes followed by a quantitative analysis of their effects, in terms of probability and impact (Bowles 1998); Fault Tree and Cause Tree Analyses determine the conditions which lead to an event and link them through logical connectors in a tree-structure which clearly displays causes and effects of the particular risk analyzed (Pahl et al. 2007). These existing methods are unable to model complex interactions among different risks.

According to Fang and Marle (2012), the complexity of project risks can be represented in terms of a risk network, describing how risks interact and propagate from one to another. For instance, there might be propagation from one "upstream" risk to numerous "downstream" risks; on the other side, a "downstream" risk may arise from the occurrence of several "upstream" risks which may belong to different categories. In such network, local risk occurrences may trigger global phenomena like the chain reaction or the "domino effect". In practice, propagation effects throughout the project structure are likely to notably reduce the performance of the risk management process (Eckert et al. 2004). Particular attention should be paid to this performance since poor or delayed risk mitigation decisions may have

great potential consequences in terms of crisis, underachievement of objectives and avoidable waste (Kloss-Grote and Moss 2008). In this regard, we argue that risk propagation behavior should be modeled and analyzed in the process of project risk management. Therefore, to manage project risks with growing complexity, it is necessary to first integrate the characteristics of risks like probability and impact, and then bring the modeling of risk interactions into the PRM process. The main contribution of this chapter is to introduce an innovative framework for the modeling and management of project risks and risk interactions

The remainder of this chapter is organized as follows. Section 49.2 presents the structure of the framework. Section 49.3 introduces the process and method for modeling the project risk network. A risk propagation model is developed in Sect. 49.4 to analyze the built network for decision support. A numerical example is provided to illustrate how to use the proposed approach. Section 49.5 concludes the chapter and discusses some extensions of this study.

49.2 A Framework for Project Risk Management

In order to manage a project with interdependent risks, it is necessary to bring the modeling of risk interactions into the PRM process. Risk interactions should be modeled with a network structure instead of a classical list or tree structure for representing the real complexity of the project. Certainly, this involves using classical risk characteristics like probability and impact as inputs for this network (the nodes for individual risks).

With the purpose of managing the complexity of project risks and based on the classical process of project risk management, we present an original framework for the modeling and management of project risks and risk interactions. This framework includes four phases:

- risk network identification;
- risk network assessment;
- risk network analysis;
- risk management implementation

Figure 49.1 illustrates this framework. First, the network is built. Details are provided in Sect. 49.3. Project risks are identified by classical methods and the output is a project risk list. Based on this list, risk interactions are identified and represented using a matrix-based method. In the phase of assessing the risk network, the probability and impact of identified risks are evaluated by classical PRM methods; then the strength of risk interactions is assessed directly by experts or through pairwise comparisons, in terms of the cause-effect probability between risks. In addition to project risks, the identification and evaluation of risk interactions makes it possible to construct the project risk network.

Next, the risk network is analyzed. Details are given in Sect. 49.4. A matrix-based risk propagation model is developed to quantitatively describe risk propagation

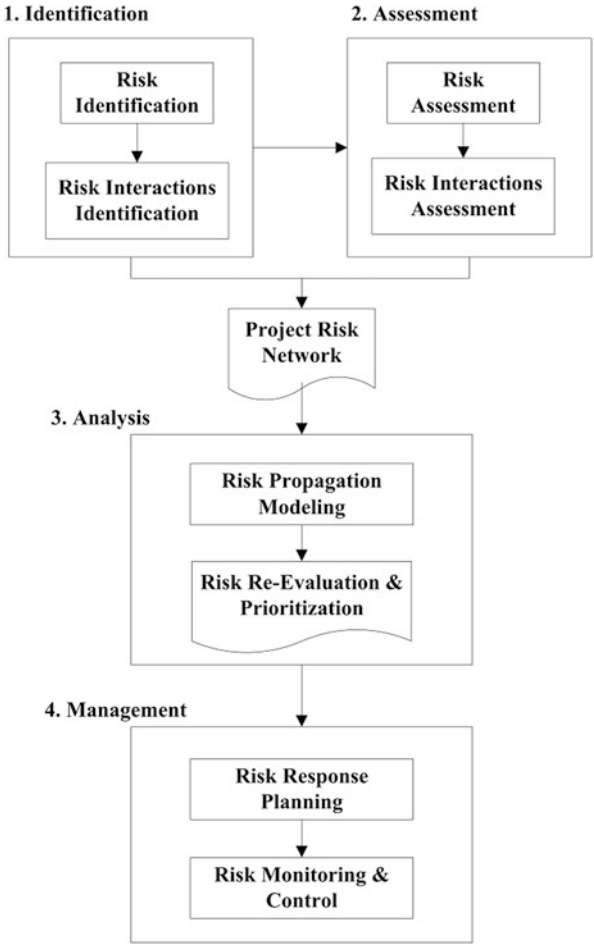


Fig. 49.1 A framework for project risk management

behavior. It enables risk characteristics to be re-evaluated, for updating possibly the risk prioritization. This innovative project risk analysis serves as a powerful complement to classical project risk analysis. The outcomes provide project managers with re-evaluation and new insights on risks and risk interactions for supporting subsequent decision-making.

The decision makers may design and plan risk response actions based on the previous risk network model and risk analysis results. Finally, the evolution of the risk network is monitored and the effectiveness of the actions is evaluated to keep the project under control. The phase of monitoring and control provides feedback for the previous phases, which allows the modification and improvement of their results. In this chapter, we only deal with the first three phases concerning risk network modeling and analysis, while not showing the planning and effects of risk response actions.

49.3 Modeling of Project Risk Network

To understand and manage the complexity of project risks, a network is built by modeling project risks and their interactions. Project risks are identified and represented as the nodes, and their potential interactions are directed edges in the network.

49.3.1 Identification of Project Risk Network

Risk identification is the process of determining which risks may potentially affect the project and documenting them for the next step of analysis. In our study, classical tools and techniques such as documentation reviews, brainstorming, interviewing, and checklist analysis are used to identify project risks. The output of risk identification is a conventional project risk list.

The next step is to determine the dependency relationship between the identified risks. Risk interaction is considered as the existence of a possible precedence relationship between two risks. The nature of risk interactions can be classified into several categories. Research on this subject has appeared in several papers, for example, ALOE model developed by Vidal and Marle (2008) defines different kinds of relationship of links between project risks such as:

- Hierarchical link
- Contribution link
- Sequential link
- Influence link
- Exchange link

Several links with different natures might exist between two risks. In this study, they are all expressed with potential cause-effect relation.

The Design Structure Matrix (DSM) method introduced by Steward (1981) has proven to be a practical tool for representing and analyzing relations and dependencies among system components (Browning 2001; Danilovic and Browning 2007). In our work, we use the concept of DSM with project risks, first introduced in Marle and Vidal (2008). The interrelations between project objects such as tasks, actors and product components help identifying the correlations between the risks attached to these objects. For instance, the project schedule gives information about task-task sequence relationships. This helps to identify the correlation between two risks of delay for these tasks. A component-component relationship (functional, structural or physical) means that the risks, which may be related to product functions, quality, delay or cost, can be linked, since one problem on one component may have an influence on another (e.g., budget or mass limits, or energy or heat flow).

We define the Risk Structure Matrix (RSM) as a binary and square matrix with $RSM_{ij} = 1$ when there is a link from R_j to R_i. It does not address concerns

about the probability or impact assessment of this interaction. We put a sanity check between R_i and R_j. Suppose we know that R_i declared R_j as a cause, if R_j did not declare R_i as a consequence, then there is a mismatch. Each mismatch is studied and solved, like the analogous works by Sosa about the interactions between project actors (Sosa et al. 2004).

49.3.2 Assessment of Project Risk Network

Besides evaluating risk characteristics such as risk probability and risk impact, we also assess the strength of risk interactions, which is interpreted as transition probability between risks.

Risk impact may be assessed on a qualitative scale (ordinal or cardinal scale with 5 or 10 levels for instance) or on a quantitative scale (financial loss for instance). Risk impact is assessed by classical methods, based on previous experience in project management and expert judgment.

For the probability assessment, we make a distinction between the probability of a risk to be triggered by another risk inside the network and its probability caused by external events or risks which are outside the system. Spontaneous probability can be interpreted as the evaluated likelihood of a risk, which is not the effect from other activated risks inside the system. On the other hand, transition probability is the evaluation of direct cause-effect relation between two risks. For the example in Fig. 49.2, Risk 1 occurs only in accordance with its spontaneous probability; and Risk 4 may arise from both its spontaneous probability and the transition probability between Risk 3 and Risk 4.

A numerical structure matrix can provide more detailed information than a binary one about the risk network for assisting decision-making. Thus the RSM can be converted into the Risk Numerical Matrix (RNM) through assessing the risk interaction. Assessment is the process of measuring and estimating the strength of the link between risks. Two ways can be used for the estimation: direct assessment

	Risk 1	Risk 2	Risk 3	Risk 4	Risk 5	Risk 6	Risk 7
Risk 1							
Risk 2						1	
Risk 3					1		1
Risk 4			1				
Risk 5	1			1			1
Risk 6	1	1			1		
Risk 7				1			

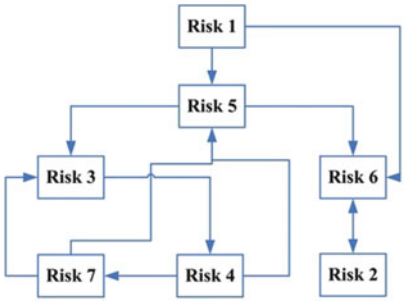


Fig. 49.2 Using DSM method to build the risk network structure

and relative assessment. As a result, RNM_{ij} is defined as the strength value of the cause-effect interaction from R_j to R_i. With regard to the project risk network, values in the RNM can also be interpreted as the transition probability between risks. For example, if the element *RNM(4,3)* is equal to 0.25, then the probability of Risk 4 originating from Risk 3 is considered to be 25 % under the condition that risk 3 is activated. More details about the assessment of risk interactions like using an AHP-based method is discussed in Fang and Marle (2012).

49.3.3 A Numerical Example

A simple example is introduced to illustrate the framework. Let us consider an example of a project with 7 identified risks.

Figure 49.2 displays the risk network of this project, using the matrix-based approach (on the left) or the classical graph approach (on the right).

After the modeling of binary risk interactions as described in the RSM, we get the RNM of the example. The RNM is denoted by matrix A in Eq. (49.1):

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.08 & 0 \\ 0 & 0 & 0 & 0 & 0.15 & 0 & 0.125 \\ 0 & 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0.32 & 0 & 0 & 0.28 & 0 & 0 & 0.09 \\ 0.42 & 0.39 & 0 & 0 & 0.22 & 0 & 0 \\ 0 & 0 & 0 & 0.22 & 0 & 0 & 0 \end{bmatrix} \tag{49.1}$$

To interpret this matrix, for example, $A(4,3) = 0.25$ indicates that if Risk 3 is activated, then there is a transition probability of 25 % originating from Risk 3 to trigger Risk 4. The spontaneous probability vector and gravity vector are obtained through evaluation by classical methods. s and G are given as follows:

$$s = [0.350 \quad 0.220 \quad 0.220 \quad 0.170 \quad 0.080 \quad 0.010 \quad 0.010]^T \tag{49.2}$$

$$G = [20.0 \quad 25.0 \quad 100.0 \quad 10.0 \quad 10.0 \quad 125.0 \quad 50.0]^T \tag{49.3}$$

Here the gravity values in G can be understood as potential impact of risks, such as capitalized loss.

These values will serve as inputs for the next step of analysis and management of project risk network.

49.4 Analysis of the Project Risk Network

After constructing the project risk network, in this section we introduce how to analyze this network. Particularly, the propagation behavior through the risk interactions will be modeled to refine the project risk analysis.

49.4.1 A Risk Propagation Model

In the project risk network, the nodes (risks) are assessed in terms of spontaneous probability and impact (or severity); the edges (risk interactions) are assessed as the probability of transition from one risk to another. As described in Sect. 49.3.2, a distinction between the spontaneous probability of a risk, for example, caused by an external reason or by undefined risks outside the system, and the probability of this risk triggered by any other identified risk inside the system has been made during the assessment process. Thus in this matrix-based propagation model, we assign the spontaneous probability evaluated by classical methods without considering interactions as the initial risk probability.

Some assumptions are made in order to calculate risk propagation in the network:

- A risk may occur more than one time during the project (this does accord with the situation in reality). Risk frequency is thus accumulative if arising from different causes or if arising several times from the same cause.
- The structure and values of RNM do not vary during the analysis time. In other words, there is no added or removed risk, and the transition probability between risks will not change during the analysis.

Suppose there are N identified risks in the network. We use vector s to represent the spontaneous probability of risks. Let the N-order square matrix A denote the RNM of transition probabilities. $P(R)$ is the vector of risk probabilities after propagation analysis.

Vector s also represents the initial vector of risk probabilities. After m steps, the probability vector of risks propagated from initial state is thus equal to $A^m \cdot s$. If we only consider m steps of propagation and according to the assumption of accumulative risk frequency, the re-evaluated risk probability vector can be obtained by the following equation:

$$P(R) = s + \sum\nolimits_{i=1}^{m} A^i \cdot \mathrm{s} = \left(I + \sum\nolimits_{i=1}^{m} A^i\right) \cdot \mathrm{s} = \left(\sum\nolimits_{i=0}^{m} A^i\right) \cdot \mathrm{s} \tag{49.4}$$

where I is the N-order identity matrix. Multiplying both sides of Eq. (49.4) by $(I - A)$ we obtain

$$(I - A) \cdot P(R) = (I - A) \cdot \left(\sum\nolimits_{i=0}^{m} A^i\right) \cdot \mathrm{s} = (I - A^{m+1}) \cdot s \tag{49.5}$$

If not considering the limit of stages in project, the steps m will be infinite. It is not guaranteed that the infinite power of the matrix A would converge to 0, as shown in the following equation:

$$lim_{m\to\infty} A^{m+1} = 0 \tag{49.6}$$

Here 0 is the zero matrix or null matrix in linear algebra. Some research papers established sufficient conditions for the convergence of infinite product of matrix, e.g., in Thomason (1977), Holtz (2000), Daubechies (1992), and Bru et al. (1994). In practice of project risk management, for example, if a risk is involved in several loops and the sum of the products of all the transition probabilities along these loops is greater than 1, the occurrence of this risk leads to chain reactions which will come back and trigger itself again with a probability of more than 100 %. In this way, the risk propagation process does not converge. This type of risk propagation is not likely to occur in practice and is outside the scope considered by the proposed model.

Nevertheless, since A is the risk numerical matrix which is usually sparse and composed of transition probabilities at small values less than 1, usually the condition of Eq. (49.6) is satisfied. Thus, risk probability can be re-evaluated by the following equation:

$$P(R) = (I - A)^{-1} \cdot s \tag{49.7}$$

Moreover, it is possible to predict the consequences of the occurrence of one or more initial risks. In this model, we assign for instance 100 % to the spontaneous probability of R_i, while all the other risks have 0 % initial values. That is to say, the initial vector $s = I^i$, where I^i is the i-th column of the identity matrix I. We can then anticipate the occurrence of the rest of the network, and thus evaluate the global consequences of R_i. Criticality is another important indicator used for prioritizing risks and usually defined as the product of risk probability and impact. Similar to risk probability, we can refine risk criticality by integrating all the potential consequences in the network of a given risk. Giving R_i with its re-evaluated probability (risk frequency) instead of 100 %, we redefine its criticality by:

$$C(R_i) = \sum\nolimits_{j=1}^{n} G(R_j) \cdot P_{R_i}(R_j) \tag{49.8}$$

where $C(R_i)$ is the criticality of R_i; $G(R_j)$ is the original evaluated impact (G for gravity) of R_j; and $P_{R_i}(R_j)$ denotes the probability of R_j as the consequence of $P(R_i)$. According to Eq. (49.7), the re-evaluated risk criticality is expressed by the equation:

$$C(R_i) = G^T \cdot (I - A)^{-1} \cdot (I^i \cdot P(R_i)) \tag{49.9}$$

The vector of risk criticalities can be calculated by the following equation:

$$C(R) = (I - A^T)^{-1} \cdot G^{\circ} P(R) \tag{49.10}$$

Here A^T represents the transpose matrix of A; and the symbol "°" denotes the array multiplication or the Hadamard product (Johnson 1974) of matrices. For example, the Hadamard product $c = a^{\circ}b$ of two vectors $a = [a_1, a_2, \ldots, a_n]$ and $b = [b_1, b_2, \ldots, b_n]$ is still an n-order vector and its elements are defined as:

$$c_i = a_i \cdot b_i \tag{49.11}$$

The re-evaluation of risk characteristics such as probability and criticality enables us to update the risk prioritization results and then to design risk response plans.

49.4.2 The Numerical Example

Based on the numerical example given in Sect. 49.3.3, we are able to calculate the risk propagation according to Eq. (49.7), and then get the re-evaluated risk probability vector:

$$P(R) = (I - A)^{-1} \cdot s = \begin{bmatrix} 0.350 & 0.245 & 0.267 & 0.237 & 0.264 & 0.311 & 0.062 \end{bmatrix}^T \tag{49.12}$$

Equally, risk criticalities are calculated using Eq. (49.10). Risks are prioritized according to different indicators. These refined results are consolidated and compared with original estimates, as shown in Table 49.1.

From the results in Table 49.1, we can see that the probability of some risks has significantly increased after re-evaluation, such as R6 and R5. This kind of risks has little probability to happen spontaneously, but some other events may lead to

Table 49.1 Risk re-evaluation and prioritization results of the example

	By spontaneous probability		By re-evaluated probability		By classical criticality		By re-evaluated criticality	
Ranking	Risk ID	Value	Risk ID	Value	Risk ID	Value	Risk ID	Value
1	R1	0.350	R1	0.350	R3	22.0	R6	40.7
2	R2	0.220	R6	0.311	R1	7.0	R1	32.5
3	R3	0.220	R3	0.267	R2	5.5	R3	29.5
4	R4	0.170	R5	0.264	R4	1.7	R2	18.6
5	R5	0.080	R2	0.245	R6	1.3	R5	14.6
6	R6	0.010	R4	0.237	R5	0.8	R4	9.6
7	R7	0.010	R7	0.062	R7	0.5	R7	4.3

them. The risk prioritization results have changed after taking into account risk interactions in the network. For example, in classical method, R3 was considered to be the most critical risk, but the one with the highest re-evaluated criticality is R6. Moreover, in the new prioritization results, the value gap between risks becomes different from that in the results of classical method. For instance, R5 and R7 are two risks with low criticalities and R5 is ranked superior to R7. After re-evaluation, R5 is still ranked superior to R7, but the gap between their relative criticality values becomes much larger. This is the opposite for R3 and R2. R2 is still behind after re-evaluation, but closer.

Important insights of this complementary analysis are priority swaps between risks. For instance, in terms of probability, R4 and R5 have a ranking swap. Similarly, R3 and R1 are ranked differently considering or not risk interactions. This implies a significant potential difference in future risk management decisions, because they are often based on rankings. Without considering interactions, priority may be given to R3. On the contrary, taking into account risk interactions, R3 drops to the third place, R6 becomes the most important and R1 remains second.

49.5 Conclusions

For dealing with the increasing complexity in projects, we propose an innovative framework for modeling and management of project risks and risk interactions. First, project risks and the interactions among them are identified with the help of expertise and experience. This enables the risk network structure to be built. It can represent the complexity of project risks and provide more insights on the relationship among risks for understanding and managing them.

Then, the parameters in the project risk network can be assessed so that we are able to model and analyze the risk propagation behavior in the risk network in a quantitative manner. Based on some assumptions, the RNM can be regarded as a specific stochastic matrix describing the risk propagation process as the project progresses. In this chapter a risk propagation model based on matrix calculation is developed. Thanks to this model, risk characteristics such as probability and criticality are re-evaluated, which may lead to updated priorities.

A simple example of seven-risk network illustrates how to use the framework and the propagation model. Under the original risk propagation model for risk analysis, some risks have been upgraded in terms of criticality ranking. The proposed approach serves as a complement to the classical project risk analysis. It may support project managers in designing more effective risk response actions and making subsequent managerial decisions. Readers may refer to Fang et al. (2012), which introduces a complementary approach based on network theory for identifying key factors in the risk network.

However, budget and resources are always tight for project implementation and particularly for managing risks as potential loss or threat to the project. For this reason, risk response actions should be selected in order to minimize the negative

risk exposure while keeping the project within budget. In this chapter we do not deal with the fourth phase of the framework concerning risk response planning and implementation of risk management actions. For further details, readers may refer to Fang et al. (2013), which discusses how to optimize the risk response plan under resource constraints.

References

Baccarini D (1996) The concept of project complexity – a review. Int J Proj Manag 14(4):201–204

Baccarini D, Archer R (2001) The risk ranking of projects: a methodology. Int J Proj Manag 19(3):139–145

Bowles J (1998) The new SAE FMECA standard. In: Proceedings of the annual reliability and maintainability symposium. IEEE, Anaheim, pp 48–53

Browning T (2001) Applying the design structure matrix to system decomposition and integration problems: a review and new directions. IEEE Trans Eng Manag 48(3):292–306

Bru R, Elsner L, Neumann M (1994) Convergence of infinite products of matrices and inner–outer iteration schemes. Electron Trans Numer Anal 2:183–193

Carr V, Tah JHM (2001) A fuzzy approach to construction project risk assessment and analysis: construction project risk management system. Adv Eng Softw 32(10–11):847–857

Chapman C, Ward S (2003) Project risk management: processes, techniques and insights. Wiley, Chichester

Chu D, Strand R, Fjelland R (2003) Theories of complexity: common denominators of complex systems. Complexity 8(3):19–30

Danilovic M, Browning T (2007) Managing complex product development projects with design structure matrices and domain mapping matrices. Int J Proj Manag 25(3):300–314

Daubechies I, Lagarias JC (1992) Sets of matrices all infinite products of which converge. Linear Algebra Appl 161:227–263

Ebrahimnejad S, Mousavi SM, Seyrafianpour H (2010) Risk identification and assessment for build-operate-transfer projects: a fuzzy multi attribute decision making model. Expert Syst Appl 37(1):575–586

Eckert C, Clarkson PJ, Zanker W (2004) Change and customisation in complex engineering domains. Res Eng Des 15(1):1–21

Fang C, Marle F (2012) A simulation-based risk network model for decision support in project risk management. Decis Support Syst 52(3):635–644

Fang C, Marle F, Zio E, Bocquet J-C (2012) Network theory-based analysis of risk interactions in large engineering projects. Reliab Eng Syst Safe 106:1–10

Fang C, Marle F, Xie M, Zio E (2013) An integrated framework for risk response planning under resource constraints in large engineering projects. IEEE Trans Eng Manag (99):1–13

Heal G, Kunreuther H (2007) Modeling interdependent risks. Risk Anal 27(3):621–634

Holtz O (2000) On convergence of infinite matrix products. Electron J Linear Algebra 7:178–181

Johnson CR (1974) Hadamard products of matrices. Linear Multilinear Algebra 1(4):295–307

Kloss-Grote B, Moss MA (2008) How to measure the effectiveness of risk management in engineering design projects? Presentation of RMPASS: a new method for assessing risk management performance and the impact of knowledge management—including a few results. Res Eng Des 19(2):71–100

Marle F, Vidal L (2008) Potential applications of DSM principles in project risk management. In: 10th international design structure matrix conference, DSM'08, Stockholm, pp 157–168

Pahl G, Beitz W, Feldhusen J, Grote K (2007) Engineering design: a systematic approach. Springer, London

PMI (2008) A guide to the project management body of knowledge (PMBOK). Project Management Institute, Newtown Square

Sosa M, Eppinger S, Rowles C (2004) The misalignment of product architecture and organizational structure in complex product development. Manag Sci 50(12):1674–1689

Steward D (1981) The design structure matrix: a method for managing the design of complex systems. IEEE Trans Eng Manag 28(3):71–74

Thomason MG (1977) Convergence of powers of a fuzzy matrix. J Math Anal Appl 57(2):476–480

Vidal L, Marle F (2008) Understanding project complexity: implications on project management. Kybernetes 37(8):1094–1110

Zhang H (2011) Two schools of risk analysis: a review of past research on project risk. Proj Manag J 42(4):5–18

Chapter 50
A Reassessment of Risk Management in Software Projects

Paul L. Bannerman

Abstract In software projects, risk management has long been recognized as the junior partner to project management in improving performance outcomes. This chapter reassesses fundamental aspects of software project risk management to highlight what we currently know from empirical research and uncover opportunities for improvement. The chapter considers evidence of the relationship between risk management and project performance; the adoption of risk management in practice, and barriers and enablers to risk management in practice. It then introduces six risk management perspectives and their related schools of thought as a basis for framing future research opportunities. It concludes with a consideration of implications for future research.

Keywords Contingency theory • Management science • Project performance • Risk management • Schools of thought • Software projects

50.1 Introduction

Is risk management worth practicing in software projects? The initial reaction is probably a conditioned: "Yes!" Upon reflection, however, there may be equivocal answers. Conceptually, we know that risk management is a good, indeed, necessary, thing to do. However, actual experience with risk management in practice may be less forthright. Also, the movement towards Agile practices has tended to mitigate some risks through the iterative nature of the development process itself, seemingly obviating the need for an explicit add-on risk management process.

This chapter re-examines software project risk management from research and practice perspectives (the scope is limited to commercial software and systems projects, excluding specialist domains such as safety critical systems). It builds on prior work in Bannerman (2008c). It aims to highlight key aspects of what we

P.L. Bannerman (✉)
National ICT Australia & School of Computer Science and Engineering, University of New South Wales, Kensington, Sydney, NSW 2052, Australia
e-mail: paul.bannerman@nicta.com.au

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_20

currently know from empirical research and uncover opportunities for developing further knowledge through future research. A central contribution of the chapter is the description of six current and emerging risk management perspectives—and their underlying theoretical schools of thought—that frame current research approaches and promising areas of future work. The central argument is that software project risk management is still in its infancy, but evidence suggests that benefits are real. Specific opportunities are identified to extend risk management in practice and diversify research.

First, the next section reviews the *status quo* of software project risk management in research and practice, focusing on evidence relating to its impact on project performance, adoption in practice, and implementation barriers and enablers. Then the perspectives and foundational schools of thought that frame current approaches to risk management are outlined. Finally, implications of this research-practice landscape and conclusions are drawn for the future.

50.2 State of Play

Prior study has found a research-practice anomaly: risk management research lags the needs of practice and risk management practice lags the prescriptions of research (Bannerman 2008c). In this (and the next) section we examine the current state of risk management research and practice in software projects to determine if progress has been made and seek to uncover roadblocks that need to be cleared by researchers and practitioners going forward. First, this section considers evidence of risk management's contribution to project outcomes; trends in the adoption/non-adoption of risk management in practice; and barriers and enablers to risk management utilization in practice.

50.2.1 Does Risk Management Improve Project Performance?

We usually take for granted that risk management is a good thing; that it helps to remove or reduce potential problems that could affect the successful outcome of a project. Considering the significant cost and effort in applying risk management in software projects, a pivotal question, therefore, is whether it does actually make a difference: does empirical research show that use of risk management is positively related to project performance?

Overall, evidence suggests that risk management is related to project success. However, the answer to the question is complicated by the multidimensionality of the concept of project success (Bannerman 2008b), the multiple categories of risks that can impact a project, and the range of available risk response treatments (Bannerman 2008c). As a consequence, achievement of a particular type of project success (such as process success or product success) will likely require appropriate

management of particular types of project risks that are contingent on the focal success criterion. Several research examples illustrate this finding (see also Na et al. 2007 for further background).

In a study of 86 experienced information systems (IS) project managers, Jiang and Klein (1999, 2000) found that three risk categories (lack of clarity of role definitions among team members, application complexity, and lack of user experience on applications) were most significantly related to system success. Further, in a study of 194 IS project managers, it was found that success also requires matching risk reduction strategies to the risks identified in the project (Jiang et al. 2001). System success was significantly related to behavioral-related risks, behavioral-based strategies (most influential), and technology-based strategies.

Similarly, Wallace et al. (2004b) found that social subsystem risk influences technical subsystem risk, which influences the level of project management risk and, subsequently, project performance. The latter was viewed separately as product performance and process performance. Product success was critically dependent on managing customer/user risks, scope and requirements, and project execution, while process success was dependent upon managing scope, requirements and project execution—the risks over which project managers feel that they have the most control (Wallace and Keil 2004).

Further, in a study of over 100 projects, Raz et al. (2002) found that risk management practices are more correlated with success in meeting time and budget goals than with product performance success. Overall, however, they conclude that higher levels of risk management are associated with higher levels of project success.

Finally, recent case study research supports this conclusion with the finding that individual risk management activities contribute to project success (as defined by stakeholder opinion) via direct instrumental effects and communicative effects that create a shared view of the project situation (de Bakker et al. 2012).

50.2.2 Risk Management Adoption/Non-adoption in Practice

Few studies have specifically focused on the adoption of risk management in software projects. Most evidence is provided as secondary observations in studies with other foci. Overall, this evidence suggests that risk management awareness is high but practices are not widely or consistently used; adoption of early process model practices is quite common (such as risk identification and assessment); adoption occurs early in the project and risk management life cycles but then tends to drop away; and integration with other project management and software development practices is varied but generally under-developed. In general, risk management practice adoption lags the prescriptions of research. Several examples of research illustrate support for these conclusions.

First, a study of 83 project managers found that while a large number were engaged in some form of risk management activity, 75 % of the project managers

did not follow any detailed risk management approach, and only vaguely understood the software risk concept and its managerial implications (Ropponen and Lyytinen 1997; Ropponen 1999).

Further, a multi-industry study of project management maturity based on PMI's knowledge areas found that the risk knowledge area had the lowest maturity of all knowledge areas in the IS industry, and the risk maturity level of the IS industry was the lowest of the four industries in the study (Ibbs and Kwak 2000).

More recently, a study of 37 software organizations compared the industrial risk models used against a best-practice model synthesized from the literature (Nyfjord and Kajko-Mattsson 2008a). In support of the earlier findings, it was found that most of the organizations had defined a risk management process model; most used the risk identification and analysis phases but use dropped off for the other phases; and only some of the core activities in the comparison model were used in the phases that were used. Process integration was found to be still in its infancy. Most organizations had integrated their risk management processes with the software processes to some degree but in an ad hoc manner, without a defined integrated process model (Nyfjord and Kajko-Mattsson 2008b).

50.2.3 Risk Management Barriers/Enablers in Practice

If research evidence suggests that risk management is worthwhile, contributing to successful project outcomes, why is adoption of risk management practices so limited? The answer may lie, at least in part, in an array of barriers that can reduce or make implementing risk management unattractive in practice (Gemmer 1997; McGrew and Bilotta 2000; Dedolph 2003; Kwak and Stoddard 2004; Bannerman 2008c; Kutsch and Hall 2009; Kutsch et al. 2012). Risk management implementation barriers include:

- *Cost*: Formally and explicitly practicing risk management can add a significant process overhead to software projects and detract attention from the main game of developing or implementing quality software-based systems. Applying risk mitigation strategies and developing/testing contingency plans can be expensive in resource effort and schedule time, as well as financial cost. Projects may also be considered too small to justify these overheads. Further, some software development methods, such as Spiral and Scrum, may be considered to have sufficient inherent risk management features that an additional process is not justified. Combined with other barriers, stakeholders may conclude that the costs of risk management outweigh the benefits.
- *Culture*: Companies vary in their position on the risk acceptance–risk aversion scale, which can influence the extent to which risk management is practiced (if at all). Some companies view risk management as negative thinking, while some managers view practicing risk management as impugning their ability to manage organizational activities through to successful completion. Others methodically

manage risks because it is a company practice standard to do so. Organizational culture can work for or against effective risk management.

- *Uncertain benefits*: This is also referred to as causal ambiguity or the 'Y2K effect'. As noted above, overall, it is difficult to show that risk management improves project performance (most empirical evidence points to specific risk factor, risk strategy or project component performance effects). Risk management interventions may change the course of a project but it is difficult to determine whether these changes actually reduced the threat and avoided a negative impact on project outcomes and whether the risk response was appropriate (or an overkill) for each treated risk. This causal ambiguity can greatly dampen enthusiasm for risk management in practice.
- *Capability*: While risk management awareness is high, expertise in practicing risk management processes and techniques is often underdeveloped and a secondary priority to other project-related activities. Consequently, adoption and effectiveness of risk management practices can be constrained by limited individual and organizational capabilities in applying risk management.
- *Ownership*: In software projects, responsibility and process ownership of risk management are often not explicitly defined, resulting in an absence of proprietorship of the process. Project managers tend to view risk management as the responsibility of project governance while governance bodies see themselves as too remote from the action to accept full accountability. Ultimately, however, it is the role of governance to ensure that the operational responsibility for risk management is assigned and overseen.
- *Reviews*: Post-project reviews are one of the least performed project management activities and, when they are held, the role and contribution of risk management to the project is even less likely to be considered. Without a review, the value of risk management is unlikely to be understood and the process is unlikely to be continuously improved to deliver increasing benefits. Compounding this problem, project success is usually claimed by the project manager or sponsor as the result of good project management rather than attributed to good risk management. Monitoring and reviewing the process can provide visibility into its value and effectiveness.
- *Reward structures*: Typically, project reward structures are directed towards 'on-time, within-budget, to-specification' delivery, not identification and mitigation of potential barriers to success. The view tends to be that if risk management contributes to project success then well and good, but it is not an objective in itself.

Key enablers of risk management adoption and utilization avoid these barriers and build a climate that encourages proactive risk management thinking and practice. Enablers include:

- *Senior management commitment and support*: Executive support provides a strong signal to the project team that managing risks is aligned to organizational priorities, practices and expectations.

- *Governance*: Holding governance bodies ultimately accountable for process improvement and benefits realization from project and risk management activities ensures that project practices are aligned to expected standards.
- *Empowerment*: Empowering and developing individuals and project teams to deliver and defend project objectives fosters risk awareness and buy-in to risk management processes.
- *Culture*: Institutionalizing risk management into the culture of how software projects are run makes risk management the normal way of smoothing and securing business-as-usual and project activity outcomes in the organization.
- *Continuous improvement*: Monitoring, evaluating and continuously improving risk management practices and techniques sharpens skills and increases returns on investment in practices over time.
- *Highlighting successes*: Acknowledging and celebrating risk management successes communicates and reinforces the benefits derived from risk management practices throughout the project and organizational communities.

50.3 Risk Management Perspectives

This section overviews the research foundations of six risk management perspectives, namely, risk management as: factor analysis; a rational process; modeling; a social process; a capability; and data analytics. The first three reflect common current practice, while the last three are emergent approaches that offer potential additional contributions in diversifying risk management in the future. The aims of this analysis are to highlight the current and possible potential scope of risk management in software projects, and provide a framework within which to encourage and promote further research and practice development.

50.3.1 Risk Management as Factor Analysis (Contingency School)

Arguably the most recognizable approach to risk management in software projects seeks to identify and mitigate contextual variables that might threaten a successful project outcome. The perspective reflects the logic of the contingency school of management which argues that situational factors—inherent to the activity and its environmental context—may influence the outcome of the activity (in this case, a software project). Therefore, from a risk management perspective, to ensure successful project performance, it is necessary to identify the specific factors that might impact the project and manage them away to avoid or minimize their impact on outcomes.

Within this perspective, three broad research approaches are evident from the literature. The first and most common approach aims to identify and mitigate individual risk factors that might impact a particular project or project type. These factors are commonly cited in 'top ten' checklists of risk/success/failure factors that must be managed to ensure the project succeeds. Example lists are provided by Alter and Ginzberg (1978), Keil et al. (1998), Boehm (1991), Heemstra and Kusters (1996), Sumner (2000), Houston et al. (2001), Johnson et al. (2001), Schmidt et al. (2001), Addison and Vallabh (2002), and Han and Huang (2007). Risk factors include, for example, clear objectives, firm requirements, user involvement, senior management commitment, and skilled resources. According to this approach, each factor is individually assessed to determine its probability of occurrence, likely impact, and relative priority (in comparison to other identified risk factors), and is individually mitigated to manage the threat. Depending on the complexity of the project and the organization's risk tolerance (or aversion), risk management under this approach can carry a substantial effort overhead and cost for the project.

The second factor analysis approach seeks to identify categories of sources of risk factors that could impact the project. The rationale here is that risk factors often arise from common sources of exposure or threat (Barki et al. 1993), and may be treated in common by one or more specific control measures, rather than treating each factor separately. This approach can leverage risk management effort and reduce cost. Risk identification, therefore, seeks to uncover the critical categories of threats that might affect the project. Examples of source categories are provided by Boehm and Ross (1989), Barki et al. (1993), Carr et al. (1993), Chittister and Haimes (1993), Heemstra and Kusters (1996), Keil et al. (1998), Cule et al. (2000), Ropponen and Lyytinen (2000), Jiang et al. (2002), Tiwana and Keil (2004), Wallace and Keil (2004), Wallace et al. (2004b), and Han and Huang (2007). Examples of source categories include requirements, technology, client, people, organization, and environment.

The third factor analysis approach aims to identify and apply higher level project dimensions that might have risk implications for the project, particularly in contexts characterized by high levels of change and/or uncertainty (hence the focus on broader dimensions of risk). This typically involves an explicit contingency approach to factor analysis that identifies and applies environmental dimensions (high level factors) to determine an overall risk profile for each project. Each dimension is usually scaled to reflect various levels of risk exposure. Assessing the project against the scale of each dimension—often in the form of a radar (or spider) chart—enables the risk profile of the project to be plotted for analysis of where risk treatments might be best applied. Examples of this approach are provided by McFarlan (1981), Ropponen and Lyytinen (2000), Wallace et al. (2004a), Shenhar and Dvir (2004), Han and Huang (2007), Howell et al. (2010), and Taylor et al. (2012). Shenhar and Dvir, for example, identify four key project dimensions against which project risk can be assessed: product novelty, technological uncertainty, complexity, and pace.

The main benefit of *risk management as factor analysis* is that it focuses attention on potential threats to a particular project. Its key limitation, however, is maintaining the effort and discipline in identifying, treating, and tracking relevant factors throughout the course of the project.

50.3.2 Risk Management as a Rational Process (Process School)

Closely associated with factor analysis is *risk management as a rational process*, which prescribes processes for managing project risk. Processes typically specify predefined (planned) stepwise tasks to follow throughout the risk management life cycle. Identifying, assessing and treating risk factors (the focus of the previous approach) are common steps within such processes. This approach draws on the logic of the process school of management which argues that organizational performance (including in project organizations) can be improved by performing work according to defined, efficient, repeatable and managed processes. Processes can incorporate accepted 'best practices' to improve efficiency and effectiveness in completing routine operational tasks. Such processes can be embedded into the routines of the organization as operational capabilities, improving the prospects of consistent, strong performance outcomes. According to this view, risk management is best approached as a managed process.

Risk management processes tend to lie on a continuum that reflects the level of certainty/uncertainty (or ease of identifying and responding to risks) in the project environment. The most common processes are plan-based, which are positioned at the certainty end of the continuum where it is assumed that risk factors can be explicitly identified, assessed and treated. Conceptually, these processes share similar steps that are executed iteratively throughout the project. These typically include: risk strategy; risk identification; risk analysis (or assessment); risk response; and risk control (or monitoring and review). Examples of such process approaches include: Boehm (1991), ISO 31000 (2009), PRINCE2 (2009), PRAM (2010), CMMI (2010), and PMBOK (2013).

At the other end of the continuum, where knowledge of the project and its environment is incomplete or highly uncertain due to high levels of change, complexity and/or ambiguity, the assumptions of plan-based processes no longer apply. In particular, relevant contingent factors cannot be readily identified within sufficient time for mitigations to be determined and applied to manage the risks. Two alternative process solutions that draw from the innovation management literature are relevant here: trial and error learning and selectionism. *Trial and error learning* refers to iterative incremental adjustments to project activities and targets in response to emerging new information. *Selectionism* refers to trying out several different solution strategies in parallel and selecting the best variation ex post. Examples of this approach are provided by: Pender (2001), Pich et al.

(2002), Sommer and Loch (2004), and Loch et al. (2006). A related third approach, *scenario analysis*, is drawn from the strategy management literature. It confronts uncertainty by considering alternative possible futures (scenarios), the likelihood and consequences of each scenario, and the risk mitigation strategies that might be applied to avoid the negative effects of each scenario (for example, Ahn and Skudlark 2002; Bahi and Rivard 2003).

The major benefit of *risk management as a rational process* is that it provides a defined map to follow to manage risks under different environmental conditions. The main limitation, however, is the potential for complacency in executing the process mechanically and/or assuming that it will guarantee a successful outcome.

50.3.3 Risk Management as Modeling (Management Science School)

Less commonly used in software projects (other than in software-based safety critical systems and hazard analysis applications), *risk management as modeling* aims to represent the risk management context as an abstraction of a phenomenon that can be analyzed to determine and/or assess threats to its successful operation. In essence, this perspective represents a loosely related collection of analysis techniques that are unified by a modeling approach to managing risk. It is influenced by the traditions of management science which apply advanced analytical methods—usually but not always quantitatively-based—to help make better decisions and produce better outcomes. Applied to software projects, risk management modeling can help 'flesh-out' complex project and/or system dynamics and dependencies to clarify risks, responses, decision options and/or paths forward based on probabilistic and non-probabilistic analysis techniques in a structured manner. Examples of analysis techniques used in modeling risk include: fault tree analysis; event tree analysis; root cause analysis; cause-consequence analysis; failure mode and effects analysis; decision analysis; expected value analysis; Bayesian network analysis; system dynamics, Markov analysis; Petri net analysis; Monte Carlo simulations; fuzzy logic; and genetic algorithms (Prichard 2005; Borison and Hamm 2010; Kwan and Leung 2011; Li, Chen and Feng 2012; Xiao et al. 2013). Chapter 49 of this handbook, for example, proposes a framework for modeling complex project risk interactions for analysis in a network structure diagram. When used in software projects, because of the overheads involved in these techniques, *risk management as modeling* tends to be used in high stakes projects involving complex software-based systems.

A particular benefit of *risk management as modeling* is in contexts that permit and require precise mathematical analysis to manage risk. By contrast, the risk analysis techniques used in the *risk management as a rational process* perspective tend to be less sophisticated and more qualitative. This, however, does not make them inferior. The 'best' risk management approach is likely to be from the

perspective that aligns closest to the needs of the project and its context. The main limitations of *risk management as modeling* are the measurement/data, analysis overhead, and analytical expertise required. Careful justification of the chosen approach is needed to ensure beneficial returns.

The following three perspectives are emergent, with isolated support in the literature. They are profiled here as promising software project risk management perspectives that may attract wider acceptance in research and practice as their respective benefits are confirmed.

50.3.4 Risk Management as a Social Process (Behavioral School)

Project management has a long history of recognizing and balancing the key roles of technical elements and people's behavior in projects (Slevin and Pinto 2004). Similarly, behavior in social (team, project and organizational) contexts is both an important enabler of risk management and a source category of potential project risks. Therefore, it is not surprising that *risk management as a social process* has attracted research attention. This perspective is influenced by the organizational behavioral school of thought which seeks to improve organizational effectiveness by understanding the impact of individual and group behaviors.

March and Shapira (1987) set the stage for a behavioral perspective on risk management by arguing that managerial approaches to risk follow a different process to that assumed by classical conceptions. By contrast, managers are quite insensitive to probability estimates of possible outcomes; their decisions are driven by a focus on current performance targets; and they sharply distinguish between taking risks and taking a bet on a particular course of action. Lyytinen et al. (1998) extend the departure from a "rational calculus" approach to risk management by integrating a behavioral view embodying organizational attention shaping routines in a socio-technical model of organizational change. According to this approach, "risk management forms a continuous exercise where project managers engage in multiple maneuvers to master their environment" (page 236).

Other research within this perspective includes a focus on desirable functional behavior (Gemmer 1997); organizational culture and human behavior (Kwak and Stoddard 2004; Krivkovich and Levy 2013); project risk as a subjective social construction (Zhang 2011; Lim et al. 2011); and how project managers define and react to risk (Moeini and Rivard 2012).

Research on behavioral approaches to organizational work has a strong foundation in other domains (such as organizational theory) as well as in software project management more generally. Application and development of this knowledge could be beneficially extended to the software project risk management domain. Its key contribution is in highlighting the role of people, as individuals and in socio-cultural

settings, as the primary actors in identifying and managing project risks. Its key challenge is the tendency for IS practitioners to fixate on technical tasks and solutions at the expense of human agency.

50.3.5 Risk Management as a Capability (Learning School)

This perspective focuses on identifying and developing personal skills and organizational capabilities that are important in successfully managing projects. It has attracted research attention since the turn of the century (Bannerman 2013). It recognizes that managing risk is about the ability to 'do it', not just 'plan it'—particularly in dynamic, uncertain and complex environments. We have seen that while there is high awareness of risk management in practice, the level of understanding and application of risk practices is low. This capability gap limits a project's ability to practice effective risk management as well as to be responsive to unforeseen disruptive events. This perspective is influenced by resource-based theory and organizational learning theory which argues that an organization's ability to perform well (and better than industry competitors) is critically dependent on its ability to learn from experience and accumulate distinctive capabilities (Levitt and March 1988; Barney and Clark 2007).

Examples of research from this perspective include contributions on: real time management of risks (Jaafari 2001); learning as a strategy to cope with information inadequacy (Pich et al. 2002); trial and error learning as a strategy to manage innovation under uncertainty and complexity (Sommer and Loch 2004); capability-based project performance (Bannerman 2008c, 2012, 2013); and managing unforeseen project contingencies (Thamhain 2013).

Underpinning this perspective is recognition that risk response strategies cannot always be planned. Sometimes project threats arise quickly, clouded in an information vacuum. What is needed in these situations is a risk response capability, similar to that used in crisis management, which cannot follow traditional processes or permit time for detailed analyses (see Bannerman 2008a, 2008c for further discussion on this perspective). The key benefits of this perspective are that it focuses attention on the ability to respond to project threats quickly and do risk management well. The main limitation is that it involves complex intangibles (such as individual and organizational learning and capability development) that are difficult to examine in research and measure in practice.

50.3.6 Risk Management as Data Analytics (Business Analytics School)

For software projects, this perspective is more speculative and may ultimately emerge as a subset of the modeling perspective. However, its scope is potentially

broader than modeling so it is considered as a separate, emergent perspective. The approach reflects the increased interest in business intelligence, business analytics, and big data that has emerged over the last decade (Davenport et al. 2010). The underpinning logic of this approach is that current and accumulated organizational and project data can be a rich source of information to support risk identification, risk strategy formulation and decision making. Data analytics can help understand what happened in the past that needs to be changed (historical view), what problems exist now that need action (current view), and what is likely to happen in the future that needs to be managed (future view).

Of course, adoption of this approach assumes the availability of, and ability to generate, relevant project-related data and risk-related metrics, as well as skilled analysts and a tool-box of data analytics techniques. These are not insignificant hurdles to overcome. As such, they may limit application of the perspective to large, high stakes projects and complex software-based systems.

50.4 Conclusions

This chapter has briefly reassessed the state of risk management in software projects with the aims of highlighting what we currently know from empirical research and uncovering opportunities for further knowledge development through future research. Key implications and conclusions from the reassessment include:

- Development of risk management in software projects remains slow, in both research and practice. Risk management research still lags the needs of practice and risk management practice still lags the prescriptions of research. Opportunities exist to both extend the application of existing research-based knowledge in practice and broaden how risk management is viewed and studied in research.
- Further research is needed into the causal linkages between risk(s), risk strategies, and project performance. Understanding risk-related paths to project performance and the boundaries of risk management are foundational to legitimizing a role for risk management in software projects and achieving demonstrable benefits from risk management practices.
- Current perspective-based approaches to risk management in software projects are narrow, reflecting a domain discipline still in its infancy. Opportunities exist for further development within the core current approaches as well as in the alternative emergent perspectives. In particular, to further apply *risk management as modeling* in software projects and diversify research into the role of people and social interactions in risk management; develop capability-based as well as plan-based risk management; and business intelligence/data analytics to identify source categories of risks.

The path forward would likely benefit from closer interaction between the research and practice communities, perhaps via joint participation in focused research-industry conferences and workshops, as well as in industry-based studies.

Such interaction could mutually reinforce development of the interests of both parties.

As the use of software-based systems continues to expand, software development methods evolve, and project management improves, the need for a strong and effective risk management capability is likely to remain. The most capable organizations are likely to adopt multiple risk management perspectives and match them to the project context, rather than default to a single approach. When this is common practice in industry, benefits and progress will likely be palpable.

Acknowledgements NICTA research is funded by the Australian Government via the Department of Broadband, Communications & Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

Addison T, Vallabh S (2002) Controlling software project risks: an empirical study of methods used by experienced project managers. In: Proceedings of the South African institute of computer scientists and information technologists. ACM, New York, pp 128–140

Ahn J-H, Skudlark A (2002) Managing risk in a new telecommunications service development process through a scenario planning approach. J Inf Technol 17:103–118

Alter S, Ginzberg M (1978) Managing uncertainty in MIS implementation. Sloan Manage Rev 20:23–31

Bahli B, Rivard S (2003) The information technology outsourcing risk: a transaction cost and agency theory-based perspective. J Inf Technol 18:211–221

Bannerman PL (2008a) Toward an integrated framework of software project threats. In: Proceedings of the 19th Australian software engineering conference. IEEE, Perth, pp 139–148

Bannerman PL (2008b) Defining project success: a multi-level framework. In: Proceedings of the project management institute research conference. PMI, Warsaw, pp 1–14

Bannerman PL (2008c) Risk and risk management in software projects: a reassessment. J Syst Softw 81:2118–2133

Bannerman PL (2012) Why good project management is not enough: liabilities of incumbency and newness. In: Proceedings of the project management institute (PMI) research and education conference. PMI, Limerick, pp 1–21

Bannerman PL (2013) Barriers to project performance. In: Proceedings of the Hawaii international conference on system sciences. IEEE, Maui, pp 4324–4333

Barki H, Rivard S, Talbot J (1993) Toward an assessment of software development risk. J Manage Inf Syst 10:203–225

Barney JB, Clark DN (2007) Resource-based theory: creating and sustaining competitive advantage. Oxford University Press, Oxford

Boehm BW (1991) Software risk management: principles and practices. IEEE Softw 8:32–41

Boehm BW, Ross R (1989) Theory-W software project management: principles and examples. IEEE Trans Softw Eng 15:902–916

Borison A, Hamm G (2010) How to manage risk (after risk management has failed)? MIT Sloan Manage Rev 52:51–57

Carr M, Konda S, Monarch I, Ulrich F, Walker C (1993) Taxonomy-based risk identification. Carnegie Mellon, Software Engineering Institute, Technical Report, CMU/SEI-93-TR-6

Chittister C, Haimes Y (1993) Risk associated with software development: a holistic framework for assessment and management. IEEE Trans Syst Man Cybern 23:710–723

CMMI (2010) CMMI for Development. Version 1.3: improving processes for better products and services (CMMI-DEV, V1.3). Carnegie Mellon, Software Engineering Institute, CMU/SEI-2010-TR-033

Cule P, Schmidt R, Lyytinen K, Keil M (2000) Strategies for heading off project failure. Inf Syst Manag 17:65–73

Davenport TH, Harris JG, Morison R (2010) Analytics at work: smarter decisions, better results. Harvard Business Press, Boston

de Bakker K, Boonstra A, Wortmann H (2012) Risk managements' communicative effects influencing IT project success. Int J Proj Manag 30:444–457

Dedolph FM (2003) The neglected management activity: software risk management. Bell Labs Tech J 8:91–95

Gemmer A (1997) Risk management: moving beyond process. Computer 30:33–43

Han W-M, Huang S-J (2007) An empirical analysis of risk components and performance on software projects. J Syst Softw 80:42–50

Heemstra FJ, Kusters RJ (1996) Dealing with risk: a practical approach. J Inf Technol 11:333–346

Houston DX, Mackulak GT, Collofello JS (2001) Stochastic simulation of risk factor potential effects for software development risk management. J Syst Soft 59:1247–1257

Howell D, Windahl C, Seidel R (2010) A project contingency framework based on uncertainty and its consequences. Int J Proj Manag 28:256–264

Ibbs CW, Kwak YH (2000) Assessing project management maturity. Proj Manag J 31:32–43

ISO 31000 (2009) 31000:2009 risk management: principles and guidelines. International Organization for Standardization

Jaafari A (2001) Management of risks, uncertainties and opportunities on projects: time for a fundamental shift. Int J Proj Manag 19:89–101

Jiang J, Klein G (1999) Risks to different aspects of system success. Inf Manag 36:263–272

Jiang J, Klein G (2000) Software development risks to project effectiveness. J Syst Softw 52:3–10

Jiang J, Klein G, Discenza R (2001) Information systems success as impacted by risks and development strategies. IEEE Trans Eng Manag 48:46–55

Jiang J, Klein G, Ellis TS (2002) A measure of software development risk. Proj Manag J 33:30–41

Johnson J, Boucher KD, Connors Y, Robinson J (2001) Project management: the criteria for success. Softw Mag 21:S3–S11

Keil M, Cule PE, Lyytinen K, Schmidt RC (1998) A framework for identifying software project risks. Commun ACM 41:76–83

Krivkovich A, Levy C (2013) Managing the people side of risk. McKinsey & Co., New York, pp 1–6

Kutsch E, Hall M (2009) The rational choice of not applying project risk management in information technology projects. Proj Manag J 40:72–81

Kutsch E, Denyer D, Hall M, Lee-Kelley E (2012) Does risk matter? Disengagement from risk management practices in information systems projects. Eur J Inf Syst 21:1–13

Kwak YH, Stoddard J (2004) Project risk management: lessons learned from software development environment. Technovation 24:915–920

Kwan TW, Leung HKN (2011) A risk management methodology for project risk dependencies. IEEE Trans Softw Eng 37:635–648

Levitt B, March JG (1988) Organizational learning. Annu Rev Sociol 14:319–340

Li Y, Chen J, Feng L (2012) Dealing with uncertainty: a survey of theories and practices. IEEE Trans Knowl Data Eng 99:1–20

Lim W-K, Sia SK, Yeow A (2011) Managing risks in a failing IT project: a social constructionist view. J Assoc Inf Syst 12:414–440

Loch CH, De Meyer A, Pich MT (2006) Managing the unknown: a new approach to managing high uncertainty and risk in projects. Wiley, Hoboken

Lyytinen K, Mathiassen L, Ropponen J (1998) Attention shaping and software risk: a categorical analysis of four classical risk management approaches. Inf Syst Res 9:233–255

March JG, Shapira Z (1987) Managerial perspective on risk and risk taking. Manag Sci 33:1404–1418
McFarlan FW (1981) Portfolio approach to information systems. Harv Bus Rev 59:142–150
Moeini M, Rivard S (2012) A behavioral model of software project risk management. In: Proceedings of the JAIS theory development workshop. Sprouts: Working Papers on Information Systems 12(10)
Na K-S, Li X, Simpson JT, Li X, Singh T, Kim K-Y (2007) Software development risk and performance measurement: evidence in Korea. J Syst Softw 80:596–605
Nyfjord J, Kajko-Mattsson M (2008a) Software risk management: practice contra standard models. In: Proceedings of the 2nd international conference on research challenges in information science, Marrakech. IEEE, New York, pp 65–70
Nyfjord J, Kajko-Mattsson M (2008b) Integrating risk management with software development: state of practice. In: Proceedings of the international conference on engineers and computer scientists, vol I. IMECS, Hong Kong, pp 878–884
McGrew JF, Bilotta JG (2000) The effectiveness of risk management: measuring what didn't happen. Manag Decis 38:293–300
Pender S (2001) Managing incomplete knowledge: why risk management is not sufficient. Int J Proj Manag 19:79–87
Pich MT, Loch CH, de Meyer A (2002) On uncertainty, ambiguity, and complexity in project management. Manag Sci 48:1008–1023
PMBOK (2013) A guide to the project management body of knowledge (PMBOK guide). Project Management Institute, Newtown Square
PRAM (2010) Project risk analysis and management guide. Association for Project Management, Princes Risborough
PRINCE2 (2009) Managing successful projects with PRINCE2. The Stationery Office, London
Pritchard CL (2005) Risk management: concepts and guidance. ESI International, Arlington
Raz T, Shenhar AJ, Dvir D (2002) Risk management, project success, and technological uncertainty. R&D Manag 32:101–109
Ropponen J (1999) Risk assessment and management practices in software development. In: Willcocks LP, Lester S (eds) Beyond the IT productivity paradox. Wiley, Chichester, pp 247–266
Ropponen J, Lyytinen K (1997) Can software risk management improve system development: an exploratory study. Eur J Inf Syst 6:41–50
Ropponen J, Lyytinen K (2000) Components of software development risk: how to address them? A project manager survey. IEEE Trans Softw Eng 26:98–112
Schmidt R, Lyytinen K, Keil M, Cule P (2001) Identifying software project risks: an international delphi study. J Manag Inf Syst 17:5–36
Shenhar AJ, Dvir D (2004) How projects differ, and what to do about it. In: Morris PWG, Pinto JK (eds) The Wiley guide to managing projects. Wiley, Hoboken, pp 1265–1286
Slevin DP, Pinto JK (2004) An overview of behavioral issues in project management. In: Morris PWG, Pinto JK (eds) The Wiley guide to managing projects. Wiley, Hoboken, pp 67–85
Sommer SC, Loch CH (2004) Selectionism and learning in projects with complexity and unforeseeable uncertainty. Manag Sci 50:1334–1347
Sumner M (2000) Risk factors in enterprise-wide/ERP projects. J Inf Technol 15:317–327
Taylor H, Artman E, Woelfer JP (2012) Information technology project risk management: bridging the gap between research and practice. J Inf Technol 27:17–34
Thamhain H (2013) Managing risks in complex projects. Proj Manag J 44:20–35
Tiwana A, Keil M (2004) The one-minute risk assessment tool. Commun ACM 47:73–77
Wallace L, Keil M (2004) Software project risks and their effect on outcomes. Commun ACM 47:68–73
Wallace L, Keil M, Rai A (2004a) Understanding software project risk: a cluster analysis. Inf Manag 42:115–125

Wallace L, Keil M, Rai A (2004b) How software project risk affects project performance: an investigation of the dimensions of risk and an exploratory model. Decis Sci 35:289–321

Xiao J, Osterweil LJ, Chen J, Wang Q, Li M (2013) Search based risk mitigation planning in project portfolio management. In: Proceedings of the international conference on software and systems process. ACM, San Francisco, pp 146–155

Zhang H (2011) Two schools of risk analysis: a review of past research on project risk. Proj Manag J 42:5–18

Chapter 51
Ranking Indices for Mitigating Project Risks

Stefan Creemers, Stijn Van de Vonder, and Erik Demeulemeester

Abstract The goal of project risk management is to mitigate the impact of risks on project objectives such as budget and time. A popular approach to determine where to focus mitigation efforts, is the use of so-called "ranking indices". Ranking indices produce a ranking of activities (or even better, risks) based on their impact on project objectives. In turn, this ranking can be used to determine the risks that are to be mitigated. Different ranking indices, however, produce different rankings. Therefore, one might wonder which ranking index is best? In this chapter, we provide an answer to this question.

Keywords Project risk • Ranking indices • Risk analysis • Risk management • Risk mitigation

51.1 Introduction

A recent study shows that projects worldwide are still struggling to meet their objectives (Standish Group 2009). During project execution, unforeseen events occur that disrupt plans and that give rise to substantial budget overruns. Risk management is widely recognized as an essential tool to deal with this kind of project uncertainty (see for instance Chap. 49 of this handbook).

The Project Management Institute (PMI 2008) defines risk management as the process that deals with the planning, identification, analyzing, responding, monitoring, and controlling of project risks. In this chapter, we focus on the risk analysis process and its effect on the risk response process. The risk analysis

S. Creemers (✉)
IESEG School of Management, Lille, France

Department of Decision Sciences and Information Management, KU Leuven, Leuven, Belgium
e-mail: s.creemers@ieseg.fr; stefan.creemers@kuleuven.be

S. Van de Vonder • E. Demeulemeester
Department of Decision Sciences and Information Management, KU Leuven, Leuven, Belgium
e-mail: stijn.vandevonder@kuleuven.be; erik.demeulemeester@kuleuven.be

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_21

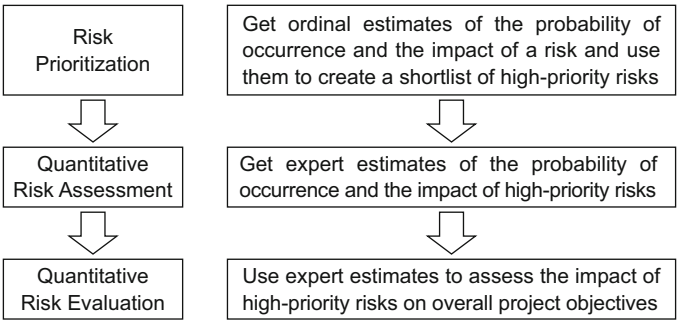


Fig. 51.1 Risk analysis process

process can be divided into three subprocesses: risk prioritization, quantitative risk assessment, and quantitative risk evaluation. Risk prioritization is a qualitative procedure that allows to prioritize the risks that have been identified in an earlier stage of the risk management process. Using ordinal estimates of both probability of occurrence and impact of a risk, a shortlist of high-priority risks can be created. During risk assessment, experts provide detailed estimates of the probability of occurrence and the impact of high-priority risks. These estimates are used in the quantitative risk evaluation procedure to analyze the impact of the shortlisted risks on overall project objectives. Figure 51.1 provides an overview of the risk analysis process.

Good risk management requires a risk analysis process that is scientifically sound and that is supported by quantitative techniques (Hubbard 2008). A wide body of knowledge on quantitative techniques has been accumulated over the last decades. Monte Carlo simulation is the predominant quantitative risk evaluation technique in both practice and in literature. Alternative techniques include neural networks, fuzzy logic, and decision-tree analysis. Their advocates, however, have so far failed to persuade most project schedulers of their practical use (refer to Hubbard 2008 and Chap. 49 of this handbook for an evaluation of different risk analysis techniques).

Risk analysis aims to provide insight into the risk profile of a project as to facilitate and to drive the risk response process (PMI 2008). The generated insights include: the probability of achieving a specific project outcome, the distribution of the project completion time etc. The risk response process will use these insights to come up with practical risk responses that allow project managers to mitigate risks (i.e., to reduce the impact of risks on project objectives). A popular approach to determine where to focus mitigation efforts is the use of so-called "ranking indices" (e.g., the criticality index and the significance index). Ranking indices allow the ranking of project activities (or risks) based on the impact they have on project objectives. A distinction needs to be made between activity-based ranking indices (i.e., those that rank activities) and risk-driven ranking indices (i.e., those that rank risks). Note that the ranking of the impact of an activity (or risk) may differ

depending on the ranking index used. Therefore, one might wonder: which ranking index is best? In the remainder of this chapter, we will address this question.

This chapter is organized as follows: in Sect. 51.2 we review the basic principles of stochastic project scheduling. Section 51.3 compares the activity-based and the risk-driven approach. In Sect. 51.4 we present the ranking indices. Their performance is discussed in Sect. 51.5. Section 51.6 concludes.

51.2 Stochastic Project Scheduling

The Critical Path Method (CPM) has been developed in the fifties and it provides the foundations of modern project scheduling. CPM represents a project as an activity network which is a graph $G = (V, E)$ that consists of a set of nodes $V = \{1, 2, \ldots, n\}$ and a set of arcs $E = \{(i, j)|i, j \in V\}$. The nodes represent project activities whereas the arcs represent precedence relationships. Each activity i has a deterministic activity duration p_i and can only start when all its predecessors have finished. CPM uses an early-start schedule in which activities are scheduled to start as soon as possible. The early-start schedule ES is represented by a vector of earliest start times $ES = \{ES_1, ES_2, \ldots, ES_n\}$. The earliest start time of an activity i is defined as follows:

$$ES_i = \max\left\{EC_j|(j, i) \in E\right\} \tag{51.1}$$

Where EC_j is the earliest completion time of activity j and equals:

$$EC_j = ES_j + p_j \tag{51.2}$$

The project starts at time instance 0 and completes at time instance C. C is given by:

$$C = \max(EC_i|i \in V) \tag{51.3}$$

A path of scheduled activities is the longest path if its length equals C. A longest path is also called a critical path and the activities on the path are referred to as critical activities.

Since the fifties, many extensions of the basic model have been proposed: resource constraint project scheduling, multi-mode scheduling, generalized precedence relationships, etc. We refer to Demeulemeester and Herroelen (2002) for an extensive overview of the field. In this chapter we are mainly interested in what is called "stochastic project scheduling" or "stochastic CPM". Stochastic CPM acknowledges the stochastic nature of activity durations. The duration of an activity i may be represented as a random variable $\tilde{p}_i$. Because activity durations are random variables, the earliest start time and the earliest completion time are random variables as well. Let $\widetilde{ES}_i$ and $\widetilde{EC}_i$ denote the random variable of the earliest start time and the earliest completion time of an activity i respectively. The project

completion time is a random variable $\tilde{C}$ that is a function of $\tilde{p}_i$. Calculating the distribution function of $\tilde{C}$ is shown to be $\#\mathscr{P}$-complete (Hagstrom 1988) and thus requires approximative methods such as Monte Carlo simulation (Van Slyke 1963). Monte Carlo simulation is used to virtually execute a project a large number of times, providing insights that can be used to enhance the actual execution of the project.

We use Monte Carlo simulation to generate random variates of $\tilde{p}_i$. More formally, let $\mathbf{p}_i = \{p_{i1}, p_{i2}, \ldots, p_{iq}\}$ denote the vector of q random variates of $\tilde{p}_i$. We refer to $\mathbf{p}_i$ as the vector of realized durations of $\tilde{p}_i$. In addition, define $\boldsymbol{ES}_i$, the vector of realized earliest start times of an activity i:

$$\boldsymbol{ES}_i = \max\left\{\boldsymbol{EC}_j | (j, i) \in E\right\} \tag{51.4}$$

Where $\boldsymbol{EC}_j$ is the vector of realized earliest completion times of an activity j and equals:

$$\boldsymbol{EC}_j = \boldsymbol{ES}_j + \mathbf{p}_j \tag{51.5}$$

The vector of realized completion times $\mathbf{C}$ is defined as follows:

$$\mathbf{C} = \max(\boldsymbol{EC}_i | i \in V) \tag{51.6}$$

$\boldsymbol{ES}_i$, $\boldsymbol{EC}_i$, and $\mathbf{C}$ are vectors of random variates of random variables $\widetilde{ES}_i$, $\widetilde{EC}_i$, and $\tilde{C}$ respectively.

51.3 Activity-Based or Risk-Driven?

One of the biggest challenges in project risk management is to estimate and to model the uncertainty of activity durations. Often, it is assumed that the duration of an activity follows a distribution that captures all uncertainty that originates from the occurrence of risks (popular distributions include: the triangular distribution, the beta distribution and the normal distribution). As such, risk assessment boils down to providing the estimates of activity duration distribution parameters. We refer to this approach as the activity-based approach.

It has been argued that the activity-based approach is inherently flawed (Creemers et al. 2013). As pointed out by Hulett (2009), there is no clear link between the impact of identified risks on the duration of an activity and the distribution of the activity duration itself (i.e., the activity-based approach is unable to identify the root causes of the uncertainty in activity durations). In addition, it is our experience that practitioners have a hard time assessing uncertainty by estimating the parameters of an activity duration distribution.

To resolve the problems of the activity-based approach, Creemers et al. (2013) have devised a risk-driven approach in which the impact of each risk is assessed individually and is mapped to the duration of an activity afterwards. The approach is based on previous work by Schatteman et al. (2008) and Van de Vonder (2006), and is similar to the risk-driver approach of Hulett (2009). Figure 51.2 illustrates the difference between the activity-based and the risk-driven approach. The activity-based approach uses Monte Carlo simulation to generate random variates of activity durations. The risk-driven approach, on the other hand, uses Monte Carlo simulation to obtain random variates of risk impacts. These random variates are then used to determine the activity durations.

The following example further supports the risk-driven approach. Consider an activity whose duration is impacted by two risks. The first risk has a small impact but a large probability of occurrence whereas the second risk has a large impact yet a small probability of occurrence. The probability distribution function of the duration of the activity is given in Fig. 51.3.

Figure 51.3 also shows the best fit of the triangular distribution (i.e., the dotted line). It is clear that such a fit would result in significant errors. In addition, it would

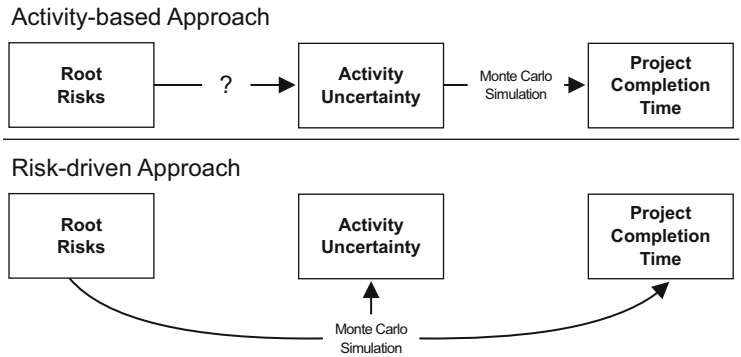


Fig. 51.2 Activity-based versus risk-driven approach

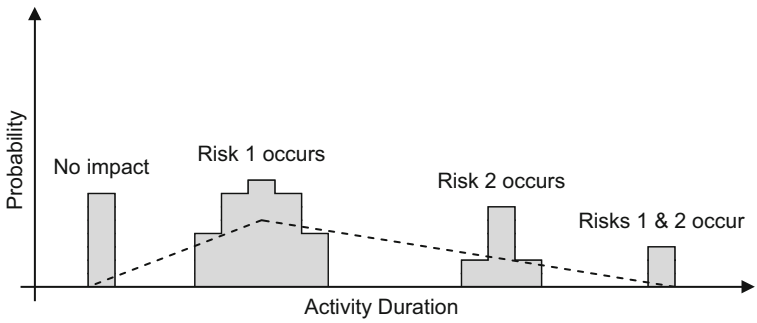


Fig. 51.3 Example probability distribution function of the duration of an activity that is impacted by two risks

be very hard for practitioners to assess the parameters of the fitted distribution. Estimating the probability of occurrence and the impact of both risks, however, would be a manageable task.

51.4 Ranking Indices

Most risk analysis software packages provide the functionality to generate insight into the source of project overruns. The activities (or risks) that contribute most to the project delay are identified using ranking indices. Let $(\cdot)_i^{(W)}$ and $(\cdot)_w^{(W)}$ denote the ranking values of a ranking index $(\cdot)$ for an activity i and a risk w when activity durations are subject to a set of risks W. A large ranking value indicates that the activity (or risk) contributes a lot to the delay of the project. The ranking of activities (or risks) is typically visualized using a ranked bar chart (see Fig. 51.4 for an example of a ranked bar chart).

In the remainder of this section, we define how risks impact activity durations. Next, we present the ranking indices themselves. For a more detailed discussion on the ranking indices presented in this section, refer to Elmaghraby (2000), Demeulemeester and Herroelen (2002), and Creemers et al. (2013).

51.4.1 Definitions

In order to formally define risks and their impacts, let $R = \{1, 2, \ldots, r\}$ denote the set of risks and let $M = \left\{\tilde{M}_{iw} | i \in V \wedge w \in R\right\}$ denote the set of risk impacts, where $\tilde{M}_{iw}$ is the random variable of the risk impact of a risk w on the duration of

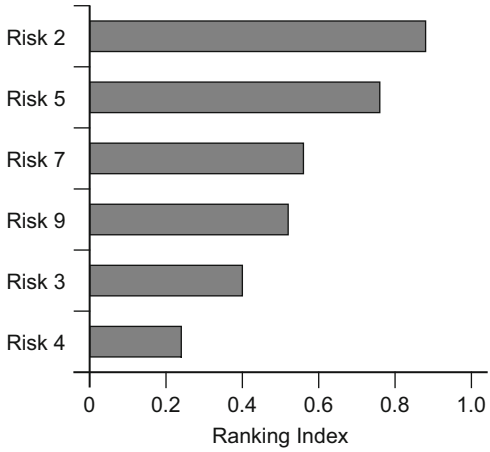


Fig. 51.4 Example ranked bar chart

an activity i. Let $\mathbf{M}_{iw} = \{M_{iw1}, M_{iw2}, \ldots, M_{iwq}\}$ represent the vector of random variates of $\tilde{M}_{iw}$, and define $\mathbf{p}_i^{(W)} = \left\{p_{i1}^{(W)}, p_{i2}^{(W)}, \ldots, p_{iq}^{(W)}\right\}$, the vector of random variates of the duration of an activity i when subject to a set of risks $W \subseteq R$. The entries of $\mathbf{p}_i^{(W)}$ are given by:

$$p_{ix}^{(W)} = p_i + \sum_{w \in W} M_{iwx} \quad (x \in \{1, 2, \ldots, q\}) \tag{51.7}$$

Where p_i is the deterministic, risk-free duration of an activity i. From $\mathbf{p}_i^{(W)}$, we obtain $\boldsymbol{ES}_i^{(W)} = \left\{ES_{i1}^{(W)}, ES_{i2}^{(W)}, \ldots, ES_{iq}^{(W)}\right\}$, $\boldsymbol{EC}_i^{(W)} = \left\{EC_{i1}^{(W)}, EC_{i2}^{(W)}, \ldots, EC_{iq}^{(W)}\right\}$, and $\mathbf{C}^{(W)} = \left\{C_1^{(W)}, C_2^{(W)}, \ldots, C_q^{(W)}\right\}$ by generalizing Eqs. (51.4)–(51.6):

$$\boldsymbol{ES}_i^{(W)} = \max\left\{\boldsymbol{EC}_j^{(W)} | (j, i) \in E\right\} \tag{51.8}$$

$$\boldsymbol{EC}_j^{(W)} = \boldsymbol{ES}_j^{(W)} + \mathbf{p}_j^{(W)} \tag{51.9}$$

$$\mathbf{C}^{(W)} = \max(\boldsymbol{EC}_i^{(W)} | i \in V) \tag{51.10}$$

The expected project delay over q simulation iterations is defined as follows:

$$\Delta^{(W)} = \frac{1}{q} \sum_{x=1}^{q} C_x^{(W)} - C \tag{51.11}$$

Where C is the risk-free project completion time and is obtained using Eq. (51.6).

51.4.2 Activity-Based Ranking Indices

In what follows, we discuss the activity-based ranking indices. These indices produce a ranking of activities that may be used to determine where to focus mitigation efforts.

51.4.2.1 Critical Activities (*CA*)

It is common practice to focus mitigation efforts on the critical activities of the deterministic early-start schedule (Goldratt 1997). The Critical Activities (*CA*) ranking values are computed as follows:

$$CA_i^{(W)} = \delta_i \tag{51.12}$$

Where δ_i equals 1 if i is critical in *ES* and 0 otherwise.

While easy to implement, *CA* does not recognize the uncertain nature of a project. In addition, the discriminative power of *CA* is limited because all activities on the critical chain have an equal ranking value.

51.4.2.2 Activity Criticality Index (*ACI*)

In stochastic CPM, the critical path may change. The Activity Criticality Index (*ACI*) recognizes that almost any path and any set of activity can become critical (Van Slyke 1963). If Monte Carlo simulation is used, the *ACI* of an activity is the proportion of simulation iterations during which the activity is critical:

$$ACI_i^{(W)} = \frac{1}{q}\sum_{x=1}^{q}\delta_{ix}^{(W)} \tag{51.13}$$

Where $\delta_{ix}^{(W)}$ equals 1 if activity i is critical in $ES_x^{(W)}$ and 0 otherwise ($ES_x^{(W)}$ is the early-start schedule during simulation iteration x when activity durations are subject to a set of risks W).

Whereas *ACI* takes into account the criticality of an activity, it ignores the variance of the activity durations. Therefore, *ACI* cannot identify the activities that effectively contribute to the delay of the project (e.g., activities that are not impacted by risks can have a larger *ACI* than activities that become critical only when impacted by a risk).

51.4.2.3 Significance Index (*SI*)

The Significance Index (*SI*) was developed by Williams (1992) as a reaction to the criticism on *ACI*. If Monte Carlo simulation is used, *SI* is given by:

$$SI_i^{(W)} = \left(\frac{1}{\sum_{x=1}^{q} C_x^{(W)}}\right)\left[\sum_{x=1}^{q}\left(\frac{p_{ix}^{(W)}}{p_{ix}^{(W)} + TF_{ix}^{(W)}}C_x^{(W)}\right)\right] \tag{51.14}$$

Where $TF_{ix}^{(W)}$ is the total float of an activity i during a simulation iteration x when activity durations are subject to a set of risks W.

SI relates both the criticality of an activity and the project completion time. *SI*, however, does still not take into account the variance of the activity durations and is therefore also flawed.

51.4.2.4 Cruciality Index (*CRI*)

The Cruciality Index (*CRI*) is the absolute value of the correlation between the duration of an activity and the total project duration. If Monte Carlo simulation is used, *CRI* is given by:

$$CRI_i^{(W)} = \left| corr\left(\mathbf{p}_i^{(W)}, \mathbf{C}^{(W)}\right) \right| \tag{51.15}$$

Although rather intuitive, *CRI* has some major drawbacks. First, it measures the linear relationship between the duration of an activity and the project completion time. It is, however, well known that the relationship between these two entities does not have to be linear at all (Elmaghraby 2000). Second, *CRI* does not take into account whether or not activities are critical. As such, an activity that is not critical can have a larger *CRI* than a critical activity that has a small duration variability.

51.4.2.5 Spearman Rank Correlation (*SRCA*)

Cho and Yum (1997) have criticized *CRI* because of its assumption of a linear relationship between activity durations and the project completion time. They propose the use of a non-linear correlation measure such as the Spearman correlation coefficient. The Spearman Rank Correlation Index (*SRCA*) is given by:

$$SRCA_i^{(W)} = \left| corr\left(rank\left(\mathbf{p}_i^{(W)}\right), rank\left(\mathbf{C}^{(W)}\right)\right) \right| \tag{51.16}$$

Where $rank(\cdot)$ transforms a vector $\cdot$ into a vector of ranking numbers.

SRCA is an improvement upon *CRI* as it allows for monotonic relationships rather than linear relationships. *SRCA*, however, still does not take into account whether or not activities are critical.

51.4.2.6 Schedule Sensitivity Index (*SSI*)

The PMI (2008) and Vanhoucke (2010) define a ranking index that combines (1) *ACI*, (2) the variance of activity durations, and (3) the variance of the project completion time. If Monte Carlo simulation is used, the Schedule Sensitivity Index (*SSI*) is given by:

$$SSI_i^{(W)} = ACI_i^{(W)} \sqrt{\frac{var\left(\mathbf{p}_i^{(W)}\right)}{var\left(\mathbf{C}^{(W)}\right)}} \tag{51.17}$$

SSI captures the variance of activity durations as well as the variance of the project completion time. However, it ignores the covariance that might exist between both entities.

51.4.2.7 Critical Delay Contribution for Activities (*CDCA*)

Creemers et al. (2013) propose a ranking index that redistributes the project delay $\Delta^{(W)}$ over the combinations of activities and risks that cause the delay. More formally, the Critical Delay Contribution (*CDC*) of an activity i and a risk w may be expressed as follows:

$$CDC_{iw}^{(W)} = \frac{1}{q} \frac{\sum_{x=1}^{q} M_{iwx} \delta_{ix}^{(W)} \left(C_x^{(W)} - C \right)}{\sum_{i \in V} \sum_{w \in W} \sum_{x=1}^{q} M_{iwx} \delta_{ix}^{(W)}} \tag{51.18}$$

$$= E \left(\frac{\mathbf{M}_{iw} \mathbf{y}_i^{(W)}}{\sum_{i \in V} \sum_{w \in W} \mathbf{M}_{iw} \mathbf{y}_i^{(W)}} \right) \Delta^{(W)} \tag{51.19}$$

Where $\left(\mathbf{y}_i^{(W)} = \left\{ \delta_{i1}^{(W)}, \delta_{i2}^{(W)}, \ldots, \delta_{iq}^{(W)} \right\} \right)$. From $CDC_{iw}^{(W)}$ it is easy to obtain an activity-based ranking index. The Critical Delay Contribution for Activities (*CDCA*) is given by:

$$CDCA_i^{(W)} = \sum_{w \in W} CDC_{iw}^{(W)} \tag{51.20}$$

51.4.3 Risk-Driven Ranking Indices

All prior ranking indices have been criticized in the literature (refer to Williams 1992; Elmaghraby 2000; Cui et al. 2006; Creemers et al. 2013) and are primarily designed to rank activities, not risks. To the best of our knowledge, Hulett (2009) and Creemers et al. (2013) are the only references that explicitly refer to a risk-driven ranking index. In what follows, we introduce the risk-driven ranking indices proposed by Hulett (2009) and Creemers et al. (2013).

51.4.3.1 Cruciality Index for Risks (*CRIR*)

Hulett (2009) proposes a simple adaptation of *CRI*. The Cruciality Index for Risks (*CRIR*) calculates the correlation between the impact of a risk and the project completion time. If Monte Carlo simulation is used, *CRIR* is given by:

$$CRIR_w^{(W)} = \left|corr\left(\mathbf{M}_w, \mathbf{C}^{(W)}\right)\right| \tag{51.21}$$

Where $\mathbf{M}_w = \left\{M_{w1}, M_{w2}, \ldots, M_{wq}\right\}$ and $\left(M_{wx} = \sum_{i \in V} M_{iwx}\right)$ for all $w \in W$.

51.4.3.2 Spearman Rank Correlation for Risks (*SRCR*)

Creemers et al. (2013) propose an adaptation of *SRCA* which is similar to the suggestion made by Hulett (2009) with respect to *CRI*. The Spearman Rank Correlation for Risks (*SRCR*) is given by:

$$SRCR_w^{(W)} = \left|corr\left(rank\left(\mathbf{M}_w\right), rank\left(\mathbf{C}^{(W)}\right)\right)\right| \tag{51.22}$$

51.4.3.3 Critical Delay Contribution for Risks (*CDCR*)

In order to compute the Critical Delay Contribution for Risks (*CDCR*), we use the *CDC*-values that were discussed in Sect. 51.4.2.7. *CDCR* is given by:

$$CDCR_w^{(W)} = \sum_{i \in V} CDC_{iw}^{(W)} \tag{51.23}$$

51.5 Computational Results

We perform an extensive computational experiment in order to evaluate the resilience of the ranking indices in a wide variety of settings. At the core of our experiment are the 600 projects of the PSPLIB J120 data set (Kolisch and Sprecher 1996). For each of the networks, we evaluate the mitigation potential of the ranking indices discussed in Sect. 51.4. A similar approach is followed in Vanhoucke (2010) and Creemers et al. (2013).

In what follow, we first discuss the experimental design and the experimental setup. Next, we define the performance measures and present the results of the computational experiment.

51.5.1 Experimental Design

For each of the projects of the PSPLIB J120 data set we introduce uncertainty by means of risks. We use five parameters to characterize the risks: (1) risk uniformity, (2) risk quantity, (3) risk probability, (4) risk impact, and (5) risk correlation. The settings of the parameters are based on our experience in the field of risk management.

Risk uniformity determines the number of activities that are impacted by a single risk. Often, clusters of activities have a similar task content and hence are subject to the same risks. We refer to these clusters of activities as activity groups (Schatteman et al. 2008). If risk uniformity is low, the number of activities impacted by any risk $w \in R$ follows a discrete uniform distribution with minimum and maximum equal to 1 and 3 activities respectively. A low risk uniformity setting results in an average of 60 activity groups and the average number of activities in an activity group equals 2. When risk uniformity is high, the number of activities impacted follows a discrete uniform distribution with minimum and maximum equal to 1 and 11 activities respectively. A high risk uniformity setting corresponds to an average of 20 activity groups and an average activity group size of 6 activities.

Risk quantity determines the number of risks that are identified during the risk identification process. A low risk quantity corresponds to a setting in which activities are impacted by 25 risks. When risk quantity is high, 50 risks impact the project activities. Risks are randomly assigned to a single activity group.

Risk probability determines the probability of occurrence of a risk whereas *risk impact* determines the impact of a risk on the duration of an activity. We define two types of risks: (1) risks that have a small probability of occurrence yet a large impact and (2) risks that have a large probability of occurrence but a small impact. Risks are randomly assigned to one of both risk types, where each risk has a 25 % chance of being of type 1. Table 51.1 summarizes the risk probability and the risk impact settings. Note that the probability of occurrence and the risk impact are modeled using a continuous uniform distribution and a triangular distribution respectively. We opt for the use of uniform and triangular distributions because our experience learns that practitioners find it easier to assess the parameters that correspond to these distributions (e.g., a practitioner is able to assess the worst, best, and most likely impact of a risk).

Risk correlation determines whether the occurrences of a risk (on activities in the impacted activity group) are correlated. We investigate three possible scenarios. A first scenario deals with the setting in which there is perfect correlation (i.e., either all activities in the activity group are impacted or none are). In a second scenario,

Table 51.1 Parameter settings for risk probability and risk impact

Risk probability	Risk impact	Risk type	Probability		Impact		
			Min	Max	Min	most likely	Max
High	High	Type 1	0.05	0.05	1.0	2.0	9.0
		Type 2	0.1	0.7	0.0	1.0	2.0
High	Low	Type 1	0.05	0.05	0.5	1.0	4.5
		Type 2	0.1	0.7	0.0	0.5	1.0
Low	High	Type 1	0.025	0.025	1.0	2.0	9.0
		Type 2	0.05	0.35	0.0	1.0	2.0
Low	Low	Type 1	0.025	0.025	0.5	1.0	4.5
		Type 2	0.05	0.35	0.0	0.5	1.0

we assume that the risk correlation is random, indicating that the occurrences of a risk are correlated with a random correlation factor that is drawn from a continuous uniform distribution with minimum and maximum equal to 0 and 1 respectively. The third scenario, assumes that risk occurrences are independent (i.e., there is no correlation between risk occurrences).

The settings of the five parameters combine to 48 different risk profiles that are to be evaluated. For each risk profile and over all projects in the PSPLIB J120 data set, we evaluate and compare the mitigation efficiency of the ranking indices. In this experiment, we assume that the mitigation of a risk results in the elimination of that risk.

51.5.2 Performance Measures

In order to compare the performance of the ranking indices, define the Relative Residual Delay (*RRD*) after mitigation of z risks using ranking index $(\cdot)$:

$$RRD^{(\cdot)_z} = \frac{\Delta^{(\cdot)_z}}{\Delta^{(\cdot)_0}} \tag{51.24}$$

Where: (1) $\Delta^{(\cdot)_z}$ is the expected project delay after mitigation of z risks using ranking index $(\cdot)$ and (2) $\Delta^{(\cdot)_0}$ is the expected project delay before any mitigation takes place. Smaller values of $RRD^{(\cdot)_z}$ correspond to a more effective ranking index.

Another measure that allows to assess the performance of a ranking index is the Mitigation Efficiency Index (*MEI*):

$$MEI^{(\cdot)} = 1 - 2\frac{\sum_{z=1}^{r} RRD^{(\cdot)_z}}{r-1} \tag{51.25}$$

The details of the dynamics of this measure may be found in Creemers et al. (2013). In short, $MEI^{(\cdot)}$ is supported on the $[-1, 1]$ real interval, where a value of 0 indicates that the performance of the ranking index equals that of a random procedure (i.e., a procedure that randomly mitigates risks). A value of 1 on the other hand, corresponds to the optimal case in which the mitigation of a single risk is sufficient to resolve all project uncertainty. In general, it is not possible to obtain a value of 1.

51.5.3 Experimental Setup

We test the mitigation potential of each ranking index using a stepwise procedure. In each step, the selected ranking index is used to identify the risk that contributes

Algorithm 51.1 Computational Experiment

```
for all Project networks in the PSPLIB J120 data set do
    for all Risk uniformity settings do
        Assign activities to activity groups
        for all Risk quantity settings do
            Set r and define W := {1, 2, ..., r}
            for all Risk probability settings do
                for all Risk impact settings do
                    for w = 1 to r do
                        Set the probability and impact of each risk
                    end for
                    for all Risk correlation settings do
                        Set the correlation of risk occurrences
                        for i = 1 to n do
                            for w = 1 to r do
                                for x = 1 to q do
                                    For the current project and risk profile, generate common risk impact M_iwx
                                end for
                            end for
                        end for
                        for all Ranking indices (·) do
                            use Monte Carlo simulation to obtain Δ^(·)z for all z : z ∈ {0, 1, ..., r}
                            for all z = 0 to r do
                                compute performance measure RRD^(·)z using Eq. 51.24
                            end for
                            For the current project and the current risk profile, compute performance measure MEI^(·)
                            using Eq. 51.25
                        end for
                    end for
                end for
            end for
        end for
    end for
end for
```

most to the delay of the project. Next, this risk is eliminated (i.e., is fully mitigated). After mitigation, we rerun the simulation and recalculate the expected project delay. Once more, the selected ranking index is used to identify and to mitigate the risk that has the largest impact on the project delay. This process continues until all risks have been mitigated. An outline of the procedure is provided in Algorithm 51.1.

We evaluate a total of 12 ranking indices. Next to the ten ranking indices discussed in Sect. 51.4 (i.e., *CA*, *ACI*, *SI*, *CRI*, *SRCA*, *SSI*, *CDCA*, *CRIR*, *SRCR*, and *CDCR*), we introduce two additional ranking indices: (1) *RAND* randomly selects a risk from those risks still active and (2) *OPT* is a greedy-optimal procedure that evaluates the elimination of each risk and selects the risk that yields the largest reduction in project delay. *RAND* may be considered as a worst-case scenario whereas *OPT* represents the best-case scenario. *OPT*, however, has limited practical value due to its computational requirements.

With respect to the activity-based ranking indices, selecting the largest risk is a two-step procedure. In a first step, the highest-ranked activity is selected. In a second step, the risk that has the largest expected impact on the selected activity is identified as the highest-ranked risk. For instance, observe the matrix of realized (during a given simulation iteration) and expected risk impacts presented in Table 51.2.

Table 51.2 Example of realized and expected risk impacts

	Realized impacts				Expected impacts			
Risk	1	2	3	Total	1	2	3	Total
Activity 1	2	1	1	4	1	2	1	4
Activity 2	0	0	2	2	0	0	1	1
Activity 3	0	0	1	1	0	0	1	1
Total	2	1	4	7	1	2	3	6

It is clear that activity 1 has the largest realized impact over all simulations. Risk 2 has the largest expected impact on activity 1 and hence is selected as the risk that contributes most to the project overrun (i.e., risk 2 is the highest-ranked risk). It is obvious, however, that risk 3 in fact has the most severe impact on the durations of the different activities.

In order to evaluate the performance of the different ranking indices, we use Monte Carlo simulation. The details of the simulation model are given in Creemers et al. (2013). We simulate the execution of each of the 600 projects in the PSPLIB J120 data set: (1) for each of the 48 risk profiles, (2) for each of the 12 ranking indices, and (3) for each step in the mitigation process (i.e., for each number of risks mitigated).

51.5.4 Results

Figure 51.5 gives an overview of the average performance of the activity-based ranking indices with respect to measure $RRD^{(\cdot)_z}$ for the range starting from $z = 0$ (i.e., no risks have been mitigated) until $z = 10$ (i.e., ten risks have been mitigated). The data are aggregated over all 600 project networks in the PSPLIB J120 data set and over all 48 risk profiles. We observe that the mitigation of risks results in a decrease of the expected project delay for each ranking index. Because *RAND* randomly selects risks, its improvement is linear with the number of risks mitigated. For all other ranking indices, the improvement is convex, implying that risks with a larger impact on the project delay are selected first. One might conclude that *SRCA* is the best activity-based ranking index, closely followed by *CDCA*. It is clear, however, that there still exists a gap between the performance of the activity-based indices and the *OPT* procedure.

Figure 51.6 is similar to Fig. 51.5 and presents the performance of risk-driven ranking indices with respect to measure *RRD*. We observe that *SRCR* outperforms *CRIR* as well as the activity-based ranking indices. More importantly, however, is the observation that *CDCR* easily outperforms *CRIR* and *SRCR*, and even matches the performance of the *OPT* procedure. It is clear that *CDCR* sets a new standard in the field of ranking indices.

Table 51.3 presents the *MEI* of the different ranking indices. For each ranking index, Table 51.3 shows: (1) the *MEI* for each risk profile and (2) the average *MEI* over the 16 risk profiles that correspond to a given risk correlation setting.

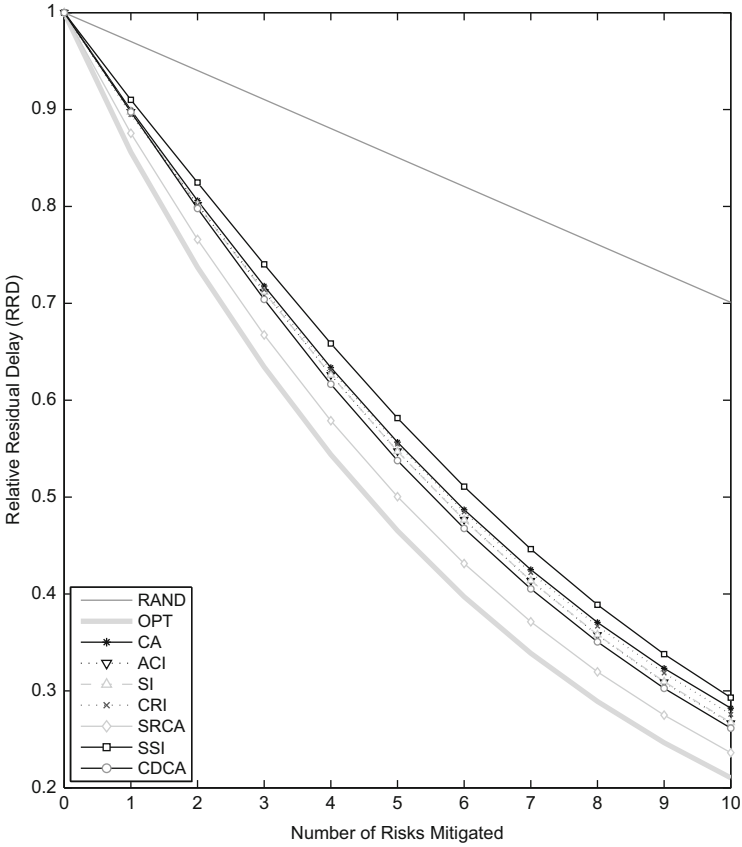


Fig. 51.5 Mitigation potential of activity-based ranking indices

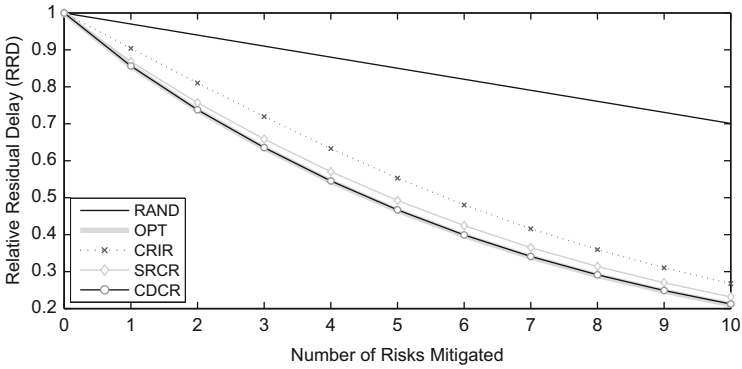


Fig. 51.6 Mitigation potential of risk-driven ranking indices

Table 51.3 Mitigation efficiency of the different ranking indices

Index	Avg	Corr	*MEI*															
CA	0.621	0 %	0.46	0.50	0.49	0.52	0.48	0.52	0.50	0.53	0.69	0.75	0.72	0.76	0.73	0.77	0.74	0.78
	0.619	100 %	0.44	0.49	0.47	0.50	0.47	0.51	0.49	0.52	0.70	0.75	0.72	0.76	0.73	0.77	0.74	0.78
	0.614	RND	0.45	0.50	0.48	0.52	0.48	0.52	0.50	0.52	0.70	0.75	0.72	0.76	0.73	0.77	0.74	0.78
ACI	0.643	0 %	0.49	0.52	0.51	0.53	0.51	0.53	0.52	0.54	0.74	0.76	0.75	0.77	0.77	0.78	0.77	0.79
	0.640	100 %	0.48	0.51	0.50	0.52	0.50	0.52	0.51	0.53	0.75	0.77	0.75	0.77	0.76	0.78	0.77	0.79
	0.637	RND	0.49	0.52	0.50	0.52	0.50	0.52	0.51	0.53	0.75	0.77	0.76	0.77	0.76	0.78	0.77	0.79
SI	0.641	0 %	0.49	0.52	0.51	0.53	0.51	0.53	0.52	0.53	0.74	0.76	0.75	0.77	0.76	0.78	0.77	0.79
	0.639	100 %	0.48	0.51	0.49	0.52	0.50	0.52	0.51	0.53	0.74	0.77	0.75	0.77	0.76	0.78	0.77	0.79
	0.637	RND	0.49	0.52	0.50	0.52	0.50	0.52	0.51	0.53	0.74	0.77	0.75	0.77	0.76	0.78	0.77	0.79
CRI	0.612	0 %	0.45	0.49	0.47	0.50	0.45	0.49	0.47	0.50	0.71	0.74	0.73	0.75	0.74	0.76	0.75	0.77
	0.636	100 %	0.49	0.53	0.52	0.55	0.50	0.54	0.53	0.56	0.73	0.76	0.75	0.77	0.75	0.77	0.76	0.78
	0.643	RND	0.49	0.53	0.51	0.55	0.49	0.52	0.52	0.54	0.73	0.75	0.74	0.76	0.75	0.77	0.76	0.77
SRCA	0.657	0 %	0.49	0.53	0.51	0.54	0.52	0.55	0.53	0.56	0.75	0.78	0.77	0.79	0.78	0.81	0.79	0.81
	0.677	100 %	0.53	0.57	0.55	0.58	0.56	0.59	0.57	0.60	0.77	0.79	0.78	0.80	0.79	0.81	0.80	0.82
	0.680	RND	0.53	0.56	0.55	0.58	0.55	0.58	0.56	0.59	0.76	0.79	0.78	0.80	0.79	0.81	0.79	0.81
SSI	0.616	0 %	0.46	0.48	0.48	0.50	0.46	0.48	0.48	0.49	0.73	0.75	0.74	0.76	0.75	0.77	0.76	0.77
	0.614	100 %	0.44	0.47	0.46	0.49	0.45	0.47	0.47	0.48	0.73	0.75	0.74	0.76	0.75	0.77	0.75	0.77
	0.610	RND	0.45	0.48	0.48	0.50	0.46	0.48	0.47	0.49	0.73	0.75	0.74	0.76	0.75	0.77	0.76	0.77
CDCA	0.646	0 %	0.47	0.51	0.49	0.52	0.50	0.53	0.51	0.54	0.75	0.78	0.76	0.79	0.78	0.80	0.79	0.81
	0.644	100 %	0.45	0.50	0.47	0.51	0.48	0.52	0.50	0.53	0.75	0.78	0.76	0.79	0.78	0.80	0.78	0.81
	0.640	RND	0.46	0.51	0.48	0.53	0.49	0.53	0.50	0.54	0.75	0.78	0.76	0.79	0.78	0.80	0.79	0.81
CRIR	0.639	0 %	0.49	0.53	0.51	0.54	0.52	0.55	0.53	0.55	0.72	0.75	0.74	0.76	0.75	0.77	0.76	0.78
	0.638	100 %	0.48	0.51	0.51	0.54	0.50	0.53	0.53	0.55	0.73	0.75	0.74	0.76	0.75	0.77	0.76	0.77
	0.635	RND	0.49	0.53	0.52	0.55	0.50	0.54	0.53	0.55	0.73	0.75	0.74	0.76	0.75	0.77	0.76	0.77
SRCR	0.674	0 %	0.52	0.56	0.54	0.57	0.56	0.58	0.57	0.60	0.75	0.78	0.77	0.79	0.78	0.80	0.80	0.81
	0.684	100 %	0.50	0.54	0.56	0.59	0.55	0.58	0.58	0.60	0.75	0.78	0.78	0.80	0.79	0.81	0.80	0.82
	0.676	RND	0.54	0.58	0.56	0.60	0.56	0.59	0.58	0.60	0.76	0.78	0.78	0.80	0.79	0.81	0.80	0.82
CDCR	0.697	0 %	0.57	0.60	0.58	0.61	0.59	0.61	0.60	0.62	0.77	0.79	0.78	0.80	0.79	0.81	0.80	0.82
	0.695	100 %	0.56	0.59	0.57	0.60	0.57	0.60	0.59	0.61	0.78	0.80	0.78	0.80	0.79	0.81	0.80	0.82
	0.692	RND	0.56	0.60	0.58	0.61	0.58	0.61	0.59	0.61	0.77	0.80	0.78	0.80	0.79	0.81	0.80	0.82
RAND	0.000	0 %	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.000	100 %	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.000	RND	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
OPT	0.698	0 %	0.57	0.60	0.59	0.61	0.59	0.61	0.60	0.62	0.77	0.79	0.78	0.80	0.79	0.81	0.80	0.82
	0.697	100 %	0.56	0.59	0.58	0.60	0.58	0.60	0.59	0.61	0.78	0.80	0.79	0.80	0.79	0.81	0.80	0.82
	0.695	RND	0.57	0.60	0.58	0.61	0.58	0.61	0.59	0.62	0.78	0.80	0.79	0.80	0.79	0.81	0.80	0.82
Risk profile			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Risk impact			H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L
Risk probability			H		L		H		L		H		L		H		L	
Risk quantity			H				L				H				L			
Risk uniformity			H								L							

We observe that the *MEI* of the *RAND* procedure is close to zero, indicating that it has no real mitigation potential. The *OPT* procedure has the highest values of *MEI* and is rivalled only by *CDCR*. Virtually no difference exists between the performance of the *OPT* procedure and the *CDCR* ranking index. With respect to the activity-based ranking indices, it is clear that *SRCA* takes the pole position, followed by *CDCA*, *ACI*, and *SI*.

Furthermore, we observe that risk correlation seems to have a limited impact on the performance of the ranking indices. A similar conclusion holds for risk probability. Risk uniformity on the other hand has a significant effect on the mitigation efficiency of the ranking indices. A higher risk uniformity results in lower performance (i.e., it is easier to distinguish between risks that only impact a small number of activities). With respect to risk quantity, we observe that an increase in the number of risks leads to a decrease of ranking index performance (i.e., if there are more risks, the mitigation of a single risk tends to be less effective). Lastly, it is clear that ranking index performance increases if risk impacts become less severe (i.e., the relative effect of mitigating a risk increases if there are only a few risks that impact project objectives).

51.6 Conclusions

Project risk management deals with the planning, identification, analyzing, responding, monitoring, and controlling of risks. In this chapter we have focussed on the risk analysis process and its effect on the risk response process. When it comes to analyzing risks, we have shown that a risk-driven approach is better than an activity-based approach. Therefore project risk management should focus on assessing the uncertainty on the level of risks (i.e., the root cause) rather than on the level of activities themselves.

In addition, we performed a large computational experiment in order to compare a number of ranking indices. Ranking indices are used to rank activities (or risks) in order to determine where to focus mitigation efforts. We compared both activity-based ranking indices (i.e., those that ranks activities) and risk-driven ranking indices (i.e., those that ranks risks). The Spearman Rank Correlation (*SRCA*) is the best activity-based ranking index. *SRCA*, however, is still far from optimal. The Critical Delay Contribution (*CDC*) may be used to devise both an activity-based and a risk-driven ranking index. The Critital Delay Contribution for Activities (*CDCA*) is second only to *SRCA* among the activity-based ranking indices. The Critital Delay Contribution for Risks (*CDCR*), on the other hand, nearly matches the performance of a greedy-optimal procedure. Therefore, we conclude that *CDCR* is the state of the art in the field of ranking indices.

References

Cho JG, Yum BJ (1997) An uncertainty importance measure of activities in PERT networks. Int J Prod Res 35(10):2737–2758

Creemers S, Van de Vonder S, Demeulemeester E (2014) A new approach for quantitative risk analysis. Ann Oper Res 213(1):27–65

Cui W, Jiajun Q, Chaoyuan Y (2006) Criticality measurement in PERT networks. In: Proceedings of the IEEE international conference on systems, man and cybernetics, Taipei, pp 703–706

Demeulemeester E, Herroelen, W (2002) Project scheduling: a research handbook. Kluwer, Boston

Elmaghraby S (2000) On criticality and sensitivity in activity networks. Eur J Oper Res 127(2):220–238

Goldratt EM (1997) Critical chain. The North River Press Publishing, Great Barrington

Hagstrom JN (1988) Computational complexity of PERT problems. Networks 18(2):139–147

Hubbard DW (2008) The failure of risk management: why it's broken and how to fix it. Wiley, New York

Hulett DT (2009) Practical schedule risk analysis. Gower Publishing, Farnham

Kolisch R, Sprecher A (1996) PSPLIB - A project scheduling problem library. Eur J Oper Res 96(1):205–216

Schatteman D, Herroelen W, Boone B, Van de Vonder S (2008) A methodology for integrated risk management and proactive scheduling of construction projects. J Constr Eng M Asce 134(11):885–895

The Project Management Institute (2008) Guide to the project management body of knowledge (PMBOK guide). The Project Management Institute, Newtown Square

The Standish Group (2009) Chaos report. The Standish Group, Boston

Van de Vonder S (2006) Proactive-reactive procedures for robust project scheduling. Ph.D. disseration. KU Leuven, Leuven

Vanhoucke M (2010) Using activity sensitivity and network topology information to monitor project time performance. Omega 38(5):359–370

Van Slyke RM (1963) Monte Carlo methods and the PERT problem. Oper Res 11(5):839–860

Williams TM (1992) Criticality in stochastic networks. J Oper Res Soc 43(4):353–357

Part XVII
Project Scheduling Applications

Chapter 52
Scheduling Tests in Automotive R&D Projects Using a Genetic Algorithm

Jan-Hendrik Bartels and Jürgen Zimmermann

Abstract For each car model an automotive manufacturer has to perform hundreds of tests on prototype vehicles before mass production can be started. In this chapter we present heuristic methods for scheduling the individual tests in automotive R&D projects such that the number of required experimental vehicles and hence the testing costs are minimized. The problem at hand can be interpreted as a multi-mode resource-constrained project scheduling problem with minimum and maximum time lags and cumulative resources. We present forward and backward variants of a priority-rule based method as well as a genetic algorithm based on an activity list representation. The presented methods are examined in a comprehensive computational study.

Keywords Genetic algorithm • Multiple execution modes • Priority-rule based methods • Project scheduling • Renewable and cumulative resources • Scheduling tests

52.1 Introduction

In the last decades the automotive market has changed from a seller's to a buyer's market accompanied by a shortening product life cycle (cf., e.g., Henseler 2004). Consequently, the number of cars produced throughout the life-span of a model has decreased, which has led to an increase in the portion of indirect costs. In particular, it is still a challenge to reduce development cost in interaction with a decreasing time-to-market (Gembrys 1998). The product development process in the automotive industry generally consists of two alternating stages. First,

J.-H. Bartels
Consultant at McKinsey & Company, Düsseldorf, Germany
e-mail: jan.hendrik.bartels@gmail.com

J. Zimmermann (✉)
Institute of Management and Economics, Clausthal University of Technology, Clausthal-Zellerfeld, Germany
e-mail: juergen.zimmermann@tu-clausthal.de

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_22

new components are constructed with the help of computer aided engineering techniques. Subsequently, these components are tested using *experimental vehicles* (prototypes) that have to be built up by the prototype section. Testing is necessary to reveal further demand for engineering in order to reach the level of quality customers expect. Additional tests verifying certain product attributes are prescribed by law. While engineering costs have decreased throughout the last few years due to the successful implementation of platform strategies, testing costs have risen because of increasing product complexity and variety (Risse 2002). Since the construction of one experimental vehicle costs up to 1.5 Million Euros, the majority of testing costs is caused by the prototype section. All tests that have to be carried out are specified in advance with the consequence that the demand for experimental vehicles depends only on the schedule of these tests. Thus, we consider a scheduling problem where we have to determine a start time for each test such that the number of required experimental vehicles is minimized and several constraints concerning, e.g., the capacity of the prototype section and the destructive effect of some tests are met.

To the best of our knowledge there exist only a few approaches for scheduling tests in automotive R&D projects with the aim to reduce costs for the production of prototypes. Assuming that each test imposes several requirements regarding the equipment of the used vehicle, Chelst et al. (2001) focus on minimizing the number of differently equipped prototypes by solving a set covering problem. For the underlying problem, it was sufficient to consider only due dates for the individual tests whereas resource constraints and constraints for destructive tests are neglected. This approach was enhanced by Lockledge et al. (2002) using a multi-stage mathematical programming model to optimize the prototype fleet for the Ford Motor Company, where the planners have to specify the characteristics of the fleet and have to assign the prescribed tests to the vehicles of the created fleet simultaneously. Two steps are performed to minimize the number of vehicles built subject to the constraint that every test must be completed on an appropriate vehicle until a specific deadline. In the first step a classic set-covering problem is solved with the aim to determine a minimum number of specific vehicle configurations that are required to cover each test. For the solution of this set-covering problem, test durations and deadlines are ignored. In the second step a minimum number of corresponding prototype variants is determined such that all tests can be executed with respect to their due dates. A basic version of the model implemented on a complex vehicle program lead to a 25 % reduction in fleet size as compared to the forecast originally made by the company.

In Scheffermann et al. (2005) a parallel machine scheduling problem is considered where the prototypes are associated with the machines and the tests with the jobs. The authors search for an allocation of tests to prototypes and an appropriate sequence for tests assigned to some prototype. Ignoring possible differences between various prototypes, two objectives are taken into account: firstly, the minimization of the number of prototypes used, and secondly, the minimization of the makespan of the schedule. The authors propose a heuristic approach to support the decision process of a manufacturer. Limtanyakul and Schwiegelshohn (2007) as well as Limtanyakul (2008, 2009) consider the same

problem and propose a constraint programming and integer programming approach. They consider the problem as a variant of the classical parallel machine scheduling problem $Pm|r_j, d_j, M_j, prec|C_{max}$, where M_j describes the prototypes eligible for test j. Their procedure starts with a fixed number of prototypes while the makespan is minimized. Subsequently, the number of prototypes is reduced successively to find a minimal number of prototypes sufficient to perform all prescribed tests.

Bartels and Zimmermann (2006a,b, 2009) as well as Zimmermann and Bartels (2006) contemplate the minimization of prototypes in the test phase of research and development projects. Each test stage is considered as a single project which must be scheduled taking into account temporal and resource constraints as well as destructive tests and so-called "partial ordered destructive relations". Besides a mixed-integer linear programming formulation a priority-rule based method is presented.

Recently, Limtanyakul and Schwiegelshohn (2012) treat the problem outlined by Bartels and Zimmermann (2006a) and develop a hybrid method based on Benders decomposition. The master problem which provides the resource allocation and estimates the number of prototypes and corresponding variants needed is modeled as a mixed-integer linear program (MILP) and solved by a standard solver. The energetic reasoning principle and a lower bound on the number of required prototypes based on pairs of tests which must be executed on different prototypes (cf. Bartels 2009) are used to strengthen the MILP formulation. The slave problem that complies with the original problem to determine an allocation of tests to prototypes and a start time for each test is solved by constraint programming (CP), where a so-called nogood constraint is used to remove infeasible solutions. Three alternative CP models and corresponding Bender cuts are considered which differ in whether or how it is allowed to partially change the pre-specifications of the master problem. The results obtained are quite competitive to the results obtained by Bartels (2009).

In this contribution we report on results for the problem of scheduling tests devised by Bartels (2009) and Bartels and Zimmermann (2009). In particular we will treat the problem as a multi-mode resource-constrained project scheduling problem (see Chaps. 21 and 22 in the first volume of this handbook), where renewable and cumulative resources are taken into account (see Chaps. 1 and 9 in the first volume of this handbook), what leads to a variant of problem $MPSs \mid temp, \overline{d} \mid \sum c_k \max r_{kt}$. Since some project activities cannot make use of the same resources if they are carried out in a certain sequence, so-called partially ordered destructive relations are considered. Moreover, experimental vehicles needed to perform the different tests must be established before they can be used in test activities. Note that, therefore, the overall number of required experimental vehicles is to be minimized and not prescribed.

The remainder of this chapter is organized as follows. Section 52.2 gives an introduction to the problem of scheduling tests in automotive R&D projects and presents an appropriate project scheduling model. Section 52.3 is devoted to a priority-rule heuristic based on a serial schedule-generation scheme. In Sect. 52.4 different variants of the schedule-generation scheme including backward and bidirectional

planning are discussed and in Sect. 52.5 we develop a genetic algorithm where each individual is represented by a list of test activities. Section 52.6 presents a comprehensive computational study that shows that the proposed methods are promising and suitable to solve large sized instances. Finally, Sect. 52.7 is devoted to concluding remarks and directions for further research.

52.2 Problem Description

Automotive R&D projects consist of m *development stages*, each containing an *engineering stage* and a corresponding *test series*. The m stages differ by the development status of the automobile under construction such that each test series makes use of a different type of experimental vehicle. In an early phase, vehicles of the preceding car model with incorporated components of the currently developed automobile are used. In a later stage, the first prototype vehicle, which is almost totally hand-made and contains all new components, is available for testing. Finally, vehicles that were built under conditions of the series production process are tested. Most kinds of tests are executed in every stage. As there are no dependencies between tests of different test series, each of the m series can be treated independently. In what follows, we restrict our considerations to a single test series which, for simplicity, will be called a *project*. Note that the processing times of the tests are specified a priori. After a test has been executed for a constant duration, it is decided whether the test was successful or needs to be repeated. The repetition of the test is usually done in the subsequent test series, after the engineering sections were able to alter the failing components. Thus, uncertainty usually does not influence a single project significantly and an independent near-term planning of the individual test series that can make use of up-to-date data is possible.

Each test series consists of a set $V^t := \{1, \ldots, n^t\}$ of n^t tests each of which has to be executed on an experimental vehicle. Before an experimental vehicle can be used for testing, it must be built up. The process of building up a vehicle is represented by a so-called *building up activity*. Let $V^b := \{1, \ldots, n^b\}$ be the set of all possible building up activities, with n^b being an appropriate upper bound for the number of required prototypes. In order to execute all tests of the test series at hand with minimal testing costs, all activities $i \in V^t \cup V^b$ have to be scheduled such that the number of vehicles used is minimized.

When performing the different types of activities the following constraints must be observed:

- *Temporal constraints*: there are precedence relations between tests, release and due dates for some tests, and a prescribed duration for each test series
- *Variant and mode feasibility*: each test has to be carried out on a suitable variant of experimental vehicle

- *Destructive tests*: several tests destroy the used vehicle, which, therefore, cannot be used for further tests afterwards
- *Partially ordered destructive relations*: some test i disables the used vehicle to perform some other test j afterwards
- *Prototyping feasibility*: an experimental vehicle cannot be used to perform a test before it has been built up by the prototype section

Temporal constraints between the start times of activities can be described by an activity-on-node network (Roy 1964; Zimmermann et al. 2010). Such a network contains a set $V := \{0, 1, \ldots, n, n+1\}$ of nodes, each of which represents an activity. In the case of our test series project each activity, $i \in \{1, \ldots, n\}$, represents a building up or test activity thus $n := n^t + n^b$ gives the total number of real activities with a processing time $p_i > 0$. Activities 0 as well as $n+1$ are fictitious activities with processing time zero that specify the start and the end of the project, respectively. For each minimum time lag, claiming that activity j has to start at least d_{ij}^{min} time units after the start of activity i, the project network contains an arc (i, j) with weight $\delta_{ij} := d_{ij}^{min}$. Likewise, for each maximum time lag, claiming that j starts at most d_{ij}^{max} time units after the start of i, a backward arc (j, i) with weight $\delta_{ji} := -d_{ij}^{max}$ is introduced (Neumann et al. 2006). Let $S := (S_i)_{i \in V}$ be a *schedule* assigning a start time S_i to each activity $i \in V$. We assume that the first n^b entries after the project start represent the start times of the building up activities followed by the n^t start times of the tests and the project end time. Minimum as well as maximum time lags lead to restrictions of the form $S_j - S_i \geq \delta_{ij}$, $(i, j) \in E$. A schedule S for which none of these restrictions is violated is called *time feasible*.

Figure 52.1 shows an example for a project network containing five tests, where the processing time p_i of activity i is depicted by a label on the respective

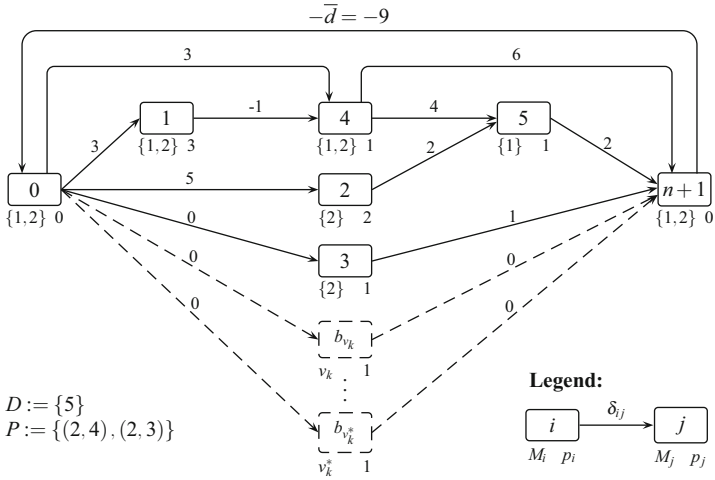


Fig. 52.1 Activity-on-node project network for the project of a test series

node. A prescribed maximum project duration $\overline{d}$ is represented by the maximum time lag between activities 0 and $n + 1$. Precedence relations between pairs of tests are ensured by appropriate minimum time lags. Furthermore, there exist dependencies between testing activities and engineering tasks. As the latter activities are not part of the problem of scheduling tests, the dependencies between testing and engineering activities are observed by release and due dates for the tests. Accordingly, release and due dates for some activity i are modeled by minimum time lags between nodes 0 and i and between nodes i and $n + 1$, respectively. In the example project test 4 can be started at point in time 3 at the earliest (release date) and must be started at point in time 3 at the latest (due date) due to the time lags between the project start and test 4 as well as the time lags between test 4 and the project completion. Test 5 is destructive and partially ordered destructive relations $(2, 4)$ and $(2, 3)$ have to be observed.

Since several variants of a model-line are developed simultaneously, different *variants* v of experimental vehicles have to be distinguished. Set M contains all those variants v, which differ, e.g., by their chassis, engine, or body. Some tests, like the comfort level testing of seats, can be processed on some or all variants of experimental vehicles, whereas the execution of other tests requires a specific variant, e.g., a variant with a diesel engine. By $M_i \subseteq M$ we denote the set of variants that are suitable to perform a test $i \in V^t$. In Fig. 52.1 set M_i of test i is shown by an additional label on the respective node.

Let $\overline{n}_v$ be an appropriate upper bound for the required number of vehicles of variant v. For the subsequent constraints it is not only important on which variant v a test i is processed, but also which individual vehicle v_k $(k = 1, \ldots, \overline{n}_v)$ of this variant v is used. According to the variants $v \in M$, we call $v_k \in \mathscr{M}$ an *execution mode* and $\mathscr{M}_i \subseteq \mathscr{M}$ the set of modes that are feasible to execute test $i \in V^t$, where $\mathscr{M}$ denotes the set of all existing modes. That is, each experimental vehicle v_k is identified with execution mode v_k and vice versa. Let x_{iv_k} be a binary variable that is 1 if test i is executed in mode v_k, and 0 otherwise. Then vector $x := \left(x_{iv_k}\right)_{i \in V^t, v_k \in \mathscr{M}}$, that assigns each test i to an experimental vehicle $v_k \in \mathscr{M}_i$ is termed a *mode assignment*. Note that modes of the same test differ only by their resource allocation, whereas—in contrast to the usual use of modes (cf., e.g., Heilmann 2000, 2001)—processing times and temporal constraints for the tests do not depend on the underlying mode.

A *destructive test* $i \in D \subseteq V^t$, e.g., a crash test, is taken into account by occupying the used vehicle v_k from the start of i to the end of the project. Moreover, we must cope with *partially ordered destructive relations* $(i, j) \in P$ which imply that a test i disables the used vehicle to perform a certain test j afterwards. Thus, either tests i and j are performed on different experimental vehicles (i.e., $x_{iv_k} x_{jv_k} = 0, v_k \in \mathscr{M}_i \cap \mathscr{M}_j$) or test j ends before test i starts (i.e., $S_j + p_j \leq S_i$). Each partially ordered destructive relation $(i, j) \in P$ leads to a restriction

$$\left(S_j + p_j\right) \sum_{v_k \in \mathscr{M}_i \cap \mathscr{M}_j} x_{iv_k} x_{jv_k} \leq S_i \tag{52.1}$$

Finally, an experimental vehicle v_k must be built up before it can be used to perform tests. The corresponding building up activity is denoted by b_{v_k}. Building up any vehicle v_k lasts, independently from the variant of the vehicle, p_b time units. The number of required vehicles and consequently the number of building up activities that have to be processed is initially unknown. In our model, the set of building up activities V^b contains an activity b_{v_k} for each possible vehicle v_k ($v \in M, k = 1, \ldots, \overline{n}_v$). The building up activities b_{v_k} of those vehicles v_k not being used in a solution of the underlying problem start at time $\overline{d}$. Therefore, the project network in Fig. 52.1 contains minimum time lags of zero time units between the building up activities $b_{v_k} \in V^b$, which are represented by the dashed nodes, and the project's end.

Due to the limited capacity $R_b(t)$ of the prototype section at each point in time $t \in [0, \overline{d} + p_b]$, a *renewable resource* (cf., e.g., Neumann et al. 2003, 2006) is introduced that limits the number of building up activities $b_{v_k} \in V^b$ that are simultaneously in execution at point in time $t \in [0, \overline{d}[$. For time $t \in [\overline{d}, \overline{d} + p_b]$ we set $R_b(t) := n^b$ in order to enable that, like described before, all building up activities b_{v_k} of not required vehicles v_k may take place at time $\overline{d}$. Let S_{v_k} be the start time of building up activity b_{v_k}. Given schedule S, $A^b(S,t) := \left\{b_{v_k} \in V^b \mid S_{v_k} \leq t < S_{v_k} + p_b\right\}$ is the set of building up activities that are processed at time t. The number of those activities that are simultaneously carried out at time t is

$$r_b(S,t) := \mid A^b(S,t) \mid \tag{52.2}$$

Since the experimental vehicles are provided successively, testing and building up activities may overlap in time, and the determination of the sequence in which the vehicles are built up becomes part of the optimization problem. Unfortunately, our definition of modes prevents us from linking building up and testing activities by temporal constraints in order to ensure that no test can start before the used vehicle has been build. Instead, we ensure the "prototyping feasibility" by a concept making use of so-called *cumulative resources*. Cumulative resources are typically used to represent *inventories* that are depleted and replenished by the project activities over time (cf., e.g., Neumann and Schwindt 2002). In our case, each vehicle $v_k \in \mathscr{M}$ is considered as a cumulative resource, where its "inventory" $r_{v_k}(S,t,x)$ at point in time t depends on the selected modes and start times of the activities $i \in V$. Initially, the inventory of each cumulative resource comprises zero units in order to indicate that the respective vehicle v_k is not available for testing. At the end of a building up activity b_{v_k}, the inventory of the respective cumulative resource is incremented by one unit, indicating that vehicle v_k is again available for testing. Each test depletes the inventory of the used resource by one unit at its start and, provided that the test is not destructive, replenishes the inventory by one unit at its end. Let $\delta_{v_k}(t)$ be a binary variable that is 1 if the building up activity for vehicle v_k has been finished until t (i.e., $S_{v_k} + p_b \leq t$), and 0 otherwise. Moreover, for a given schedule S, $A^t(S,t) := \{i \in V^t \mid S_i \leq t < S_i + p_i\}$ is the set of tests being in execution at

time t. Only a test $i \in A^t(S,t) \setminus D$ or $i \in D : S_i \leq t$ may influence the inventory of a vehicle v_k at point in time t. Thus, given a mode assignment x and a schedule S, the inventory of vehicle v_k at time t is given by

$$r_{v_k}(S,t,x) := \delta_{v_k}(t) - \sum_{i \in A^t(S,t) \setminus D} x_{iv_k} - \sum_{i \in D : S_i \leq t} x_{iv_k} \tag{52.3}$$

All activities $i \in V$ must be scheduled such that the inventory of any cumulative resource v_k does not fall below a minimum inventory $\underline{R}_{v_k} := 0$ at any point in time.[1] Schedule S is called resource-feasible with respect to assignment x if it fulfills constraints $r_b(S,t) \leq R_b(t)$ and $r_{v_k}(S,t,x) \geq \underline{R}_{v_k}$. A schedule that is time and resource-feasible with respect to mode assignment x is termed feasible with respect to x. The concepts of building up activities and cumulative resources for the problem of scheduling tests are illustrated in Fig. 52.2 which is borrowed from Bartels and Zimmermann (2009). For our problem instance with five tests and two modes (cf. Fig. 52.1), a gantt chart is given, which shows, for a time feasible solution with two vehicles 1_1 and 2_1 (one of each variant) and, respectively, two building up activities, a time feasible schedule $S = (0,1,0,3,5,1,3,7,9)$ as well as mode assignments where the modes $(1_1, 2_1, 2_1, 2_1, 1_1)$ are assigned to tests $(1,2,\ldots,5)$. Furthermore, the inventory profile $r_{1_1}(S,t,x)$ for vehicle $v_k = 1_1$ and the renewable resource profile $r_b(S,t)$ of the prototype section are presented.

Having introduced all relevant constraints, we can summarize the problem of scheduling tests by the following project scheduling model (Bartels and Zimmermann 2009):

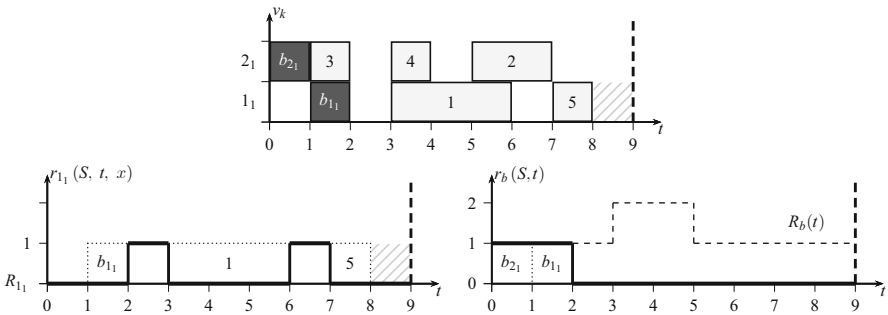


Fig. 52.2 Gantt chart representing a feasible solution; inventory profile $r_{1_1}(S,t,x)$ of vehicle 1_1 and (renewable) resource profile $r_b(S,t)$ of the prototype section

[1] A maximum inventory $\overline{R}_{v_k} := 1$ cannot be exceeded as each test must have depleted the inventory of the used resource before replenishing it and at most one prototyping activity per resource exists.

$$\text{Min.} \quad f(S,x) = \sum_{v_k \in \mathscr{M}} \delta_{v_k}(\overline{d}-1) \tag{52.4}$$

$$\text{s.t.} \quad S_j - S_i \geq \delta_{ij} \quad ((i,j) \in E) \tag{52.5}$$

$$\left(S_j + p_j\right) \sum_{v_k \in \mathscr{M}_i \cap \mathscr{M}_j} x_{iv_k} x_{jv_k} \leq S_i \quad ((i,j) \in P) \tag{52.6}$$

$$r_b\,(S,t) \leq R_b(t) \quad (t \in [0,\overline{d}]) \tag{52.7}$$

$$r_{v_k}\,(S,t,x) \geq \underline{R}_{v_k} \quad (v_k \in \mathscr{M}; t \in [0,\overline{d}]) \tag{52.8}$$

$$\sum_{v_k \in \mathscr{M}_i} x_{iv_k} = 1 \quad (i \in V^t) \tag{52.9}$$

$$S_0 = 0 \tag{52.10}$$

$$S_i \geq 0 \quad (i \in V) \tag{52.11}$$

$$x_{iv_k} \in \{0,1\} \quad (i \in V;\ v_k \in \mathscr{M}) \tag{52.12}$$

$$\delta_{v_k}(t) \in \{0,1\} \quad (v_k \in \mathscr{M}; t \in [0,\overline{d}]) \tag{52.13}$$

Equation (52.4) indicates that the number of used experimental vehicles is to be minimized. As previously described, an experimental vehicle can only be used if the corresponding building up activity has been completed no later than $\overline{d}-1$ (i.e., $\delta_{v_k}(\overline{d}-1) = 1$), that is, the model serves to decide whether a vehicle has to be built or not. The building up activities b_{v_k} of all vehicles v_k that are not used to execute tests in an optimal solution to problem (52.4)–(52.13) start at time $\overline{d}$ and do not affect the objective function value. Restrictions (52.5) ensure that the solution is time feasible and inequalities (52.6) regard the partially ordered destructive relations. Due to constraints (52.7) the capacity of the renewable resource (prototype section) must not be exceeded and because of restrictions (52.8) no inventory conflict for the cumulative resources may occur. Furthermore, constraints (52.9) make sure that each test is executed in an admissible mode. Finally, by (52.10)–(52.13) the domains for the decision variables are defined. In Bartels and Zimmermann (2009) a corresponding mixed-integer linear program is presented which can be used to solve our test scheduling problem using a standard solver.

In Bartels and Zimmermann (2009) the following properties of problem (52.4)–(52.13) are proved.

Lemma 52.1. *Problem (52.4)–(52.13) is $\mathscr{NP}$-hard in the strong sense. The corresponding feasibility problem is $\mathscr{NP}$-complete.*

Due to Lemma 52.1 we must assume that problem instances of practical size can not be solved to optimality within an acceptable amount of time. Thus, we will report on heuristic solution methods in Sects. 52.3–52.5 that are able to solve large problem instances in acceptable time and make use of the following Theorem (Bartels and Zimmermann 2009).

Theorem 52.1. *i) Let (S, x) be an optimal solution to problem (52.4)–(52.13). Then each time-feasible schedule S' implying the same precedence relations between the activities $i \in V$ as S, i.e., for all $(i, j) \in V \times V$ it holds that $S'_i + p_i \leq S'_j$ exactly if $S_i + p_i \leq S_j$, is optimal as well.*

ii) There always exists an optimal schedule where all building up activities start as early as possible, i.e., no building up activity can be started earlier without delaying at least one other building up activity.

Theorem 52.1 enables a different consideration of the building up activities $b_{v_k} \in V^b$. A priori we can determine an ordered multiset of possible start times S^P at which the building up activities must begin. Multiset S^P contains $n^b + 1$ elements. The first $\mu = 1, \ldots, n^b$ elements $S^P_\mu \in S^P$ are derived by starting the (possible) building up activities, which all require the same amount of resources from the prototype section and have the same processing time p_b, one after another such that they begin as early as possible and no renewable resource constraint is violated. Moreover, we set $S^P_{n^b+1} := \overline{d}$. Having predetermined the possible start times $S^P_\mu \in S^P$, we afterwards assign them to the start times S_{v_k} of the real prototyping activities $b_{v_k} \in V^b$. In other words, we prescribe time slots in which building up activities must be processed and afterwards assign the activities $b_{v_k} \in V^b$ to these time slots. Again, the building up activities of vehicles v_k not being used in a solution are started at $S_{v_k} = S^P_{n^b+1} = \overline{d}$. Note, that assigning the predetermined start times of set S^P to the building up activities always ensures that the renewable resource constraints are met.

52.3 A Priority-Rule Based Approach

To solve large instances with up to 600 test activities of problem (52.4)–(52.13), a *priority-rule based heuristic* to determine "good" solutions may be used which brings together a serial generation scheme and some priority rules. The generation scheme is based on an approach of Ballestín et al. (2007), Selle and Zimmermann (2003), Zimmermann (1997) that has been devised in order to solve project scheduling problems with non-regular objective functions. The basic idea of this approach is to schedule the individual activities successively such that the respective increase in the objective function value of the extended partial solution is minimal.

Let C be the set of activities that have already been scheduled. A partial solution is represented by pair (S^C, x^C) with S^C and x^C being the partial schedule and the partial mode assignment of tests $i \in C \cap V^t$, respectively. By $f^C(S^C, x^C)$ we denote the objective function value of some partial solution (S^C, x^C). Moreover, let ES^C_j and LS^C_j be the schedule dependent earliest and latest start times of the activities $j \in V^t \setminus C$ that result from the temporal constraints. In each main step of the proposed serial generation scheme (cf. Fig. 52.3), we first schedule those tests $i \in V^t \setminus C$ that are critical (i.e., for which $ES^C_i = LS^C_i$ holds). If no such activity

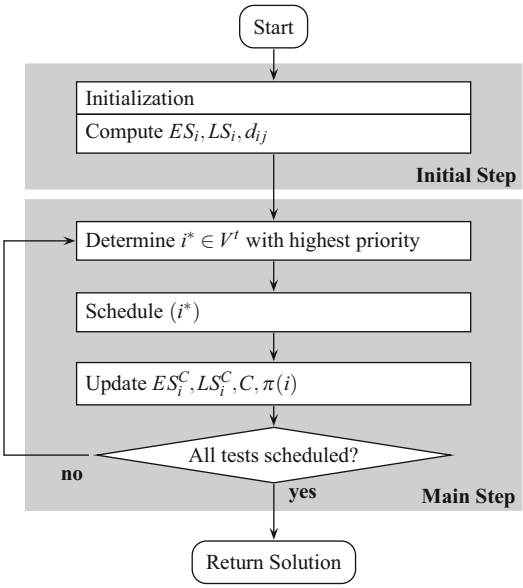


Fig. 52.3 Generation scheme (Bartels and Zimmermann 2009)

exists, test $i^* \in V^t \setminus C$ with highest *priority value* $\pi(i^*)$ subject to some *priority-rule* is determined to be scheduled next. Afterwards, we determine a start time S_{i^*} as well as an execution mode $\mathscr{M}_{i^*}$ for test i^* and add i^* to C. Having assigned a start time S_{i^*} to test i^*, the earliest and latest start times of all tests $j \in V^t \setminus C$ have to be updated according to

$$ES_j^{C \cup \{i^*\}} := \max(ES_j^C, S_{i^*} + d_{i^*j})$$

$$LS_j^{C \cup \{i^*\}} := \min(LS_j^C, S_{i^*} - d_{ji^*})$$

where d_{i^*j} (d_{ji^*}) is the length of a longest path between tests i^* and j (j and i^*) in the project network. The longest path lengths d_{ij} between any two nodes $i, j \in V^t$ are computed in the initial step of the procedure by the Floyd-Warshall-Algorithm in $\mathscr{O}(n^3)$ time (cf., e.g., Ahuja et al. 1993). In the same step, the initial earliest and latest start times are determined by $ES_i := d_{0i}$ and $LS_i := -d_{i0}$, respectively. Initializing the procedure by setting $C := \emptyset$, the procedure terminates once $V^t \subseteq C$.

A start time S_{i^*} and an assignment $m_{i^*} \in \mathscr{M}_{i^*}$ for selected test i^* are determined by procedure *Schedule*(i^*) (cf. Fig. 52.4). Note that m_i represents a vehicle v_k for which $x_{iv_k} = 1$ in the current partial mode assignment x^C holds. First of all let us present a summary of this procedure. A more detailed description is given subsequently.

As already mentioned, a selected test i^* is scheduled such that the increase in the objective function value is minimal. The objective function is initialized with 0 and increases by 1 if and only if for test i^* an assignment $m_{i^*} \neq m_i$ for all $i \in C \cap V^t$ is chosen, i.e., no scheduled test $i \in C \cap V^t$ is performed on the selected vehicle

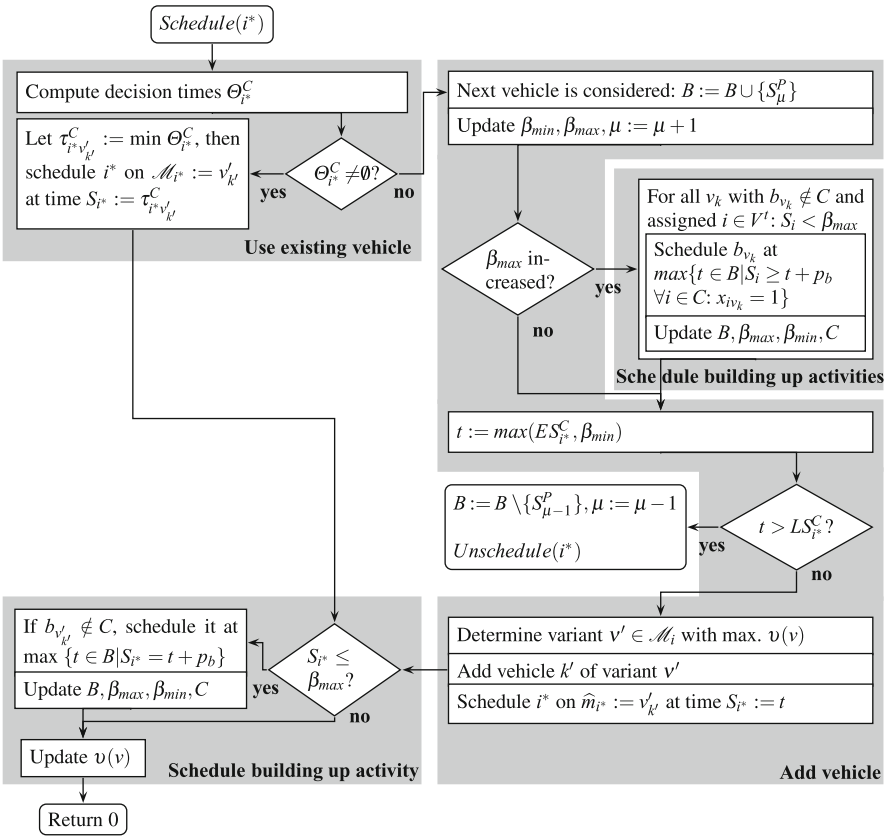


Fig. 52.4 Procedure *Schedule*(i^*)

$v_k = m_{i^*}$. For simplicity, we then speak of adding an experimental vehicle to the partial solution. If $m_{i^*} = m_i$ for any $i \in C \cap V^t$, we say an existing vehicle is used. Thus, in order to avoid an increase in the objective function value, we first try to schedule test i^* on an existing vehicle v_k. To this end, for each existing vehicle $v_k \in \mathscr{M}_{i^*}$ we determine appropriate start times $S_{i^*} \in [ES^C_{i^*}, LS^C_{i^*}]$ that neither violate any partially ordered destructive relation $(i^*, j) \in P$ or $(j, i^*) \in P$ $(j \in C \cap V^t)$ nor lead to an inventory conflict with the considered cumulative resource v_k. However, if no such opportunity exists at all, we start test i^* on an additional vehicle accepting that the objective function value increases by one. Next, we consider how to treat the building up activities. According to Theorem 52.1 ii) we initially predetermine the ordered multiset of possible start times S^P for the building up activities which are assigned to the activities $b_{v_k} \in V^b$ in some later iterations of the solution procedure. When adding an experimental vehicle to the current partial solution, we have to assign a start time $S_{v_k} \in S^P$ to the corresponding building up activity b_{v_k}. Therefore, we denote a possible start time S^P_μ "assignable" $(\mu = 1, \ldots, n^b)$ in partial solution

(S^C, x^C), if $\mu \leq f^C(S^C, x^C)$ and S^P_μ has not been assigned to a building up activity b_{v_k}. This means that the number of assignable start times S^P_μ equals the number of experimental vehicles v_k used in partial solution (S^C, x^C), for which building up activity b_{v_k} has not been scheduled so far. Multiset $B \subseteq S^P$ contains all those assignable start times S^P_μ. Moreover, we introduce two parameters $\beta_{max} := \max\{S^P_\mu \in B\} + p_b$ and $\beta_{min} := \min\{S^P_\mu \in B\} + p_b$ that represent the minimum and maximum assignable completion time for a building up activity, respectively. Each time a vehicle is added to the partial solution or a building up activity has been scheduled, β_{max} and β_{min} are updated. If $B = \emptyset$, we set $\beta_{max} := -1$ and $\beta_{min} := \overline{d}$. Parameter β_{max} serves to decide when the building up activity b_{v_k} of an existing vehicle v_k has to be scheduled. If there exists an experimental vehicle v_k without scheduled building up activity b_{v_k} that is used by a test $i \in C \cap V^t$ with $S_i \leq \beta_{max}$, building up activity b_{v_k} is started at the maximum $S^P_\mu \in B$ with $S^P_\mu + p_b \leq S_i$ $(i : m_i = v_k)$. This may occur, if either selected test i^* is started before β_{max} or β_{max} is increased due to the next possible start time S^P_μ becoming available with adding an experimental vehicle to the current solution. In the first case, only vehicle $v_k = m_{i^*}$ needs to be considered, whereas in the latter case all existing vehicles without scheduled building up activity are examined. Note that up to its end the procedure does not schedule the building up activities b_{v_k} of all those vehicles v_k that are not used by a test $i \in V^t$ before β_{max}. However, for all these building up activities the remaining possible start times $S^P_\mu \in B$ can be assigned arbitrarily.

The second parameter β_{min} reveals if even an additional vehicle may not be used to perform a test i^*. For $LS^C_{i^*} < \beta_{min}$ the building up activity b_{v_k} of an additional vehicle v_k could not be finished before the start of test i^* and a procedure *Unschedule(i*)* is called.

Next, we will give some additional details on the determination of start time S_{i^*} in procedure *Schedule*(i^*) if i^* is assigned to an established vehicle. Let multiset $\Theta^C_{i^*}$ contain all appropriate start times for test i^*, for which a mode assignment exists such that i^* can feasibly be scheduled on an existing vehicle v_k. To determine multiset $\Theta^C_{i^*}$, we examine all vehicles $v_k \in \mathscr{M}_{i^*}$ that are used by at least one test $i \in C \cap V^t$. The previously selected test i^* must start between its schedule dependent earliest start time $ES^C_{i^*}$ and latest start time $LS^C_{i^*}$ (see Fig. 52.5).

Given partial solution (S^C, x^C), let $\vartheta^C_{i^* v_k} \subseteq \left[ES^C_{i^*}, LS^C_{i^*}\right]$ be the time domain in which test i^* can start on vehicle v_k such that neither the described minimum

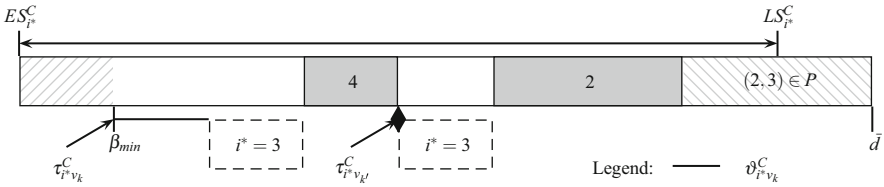


Fig. 52.5 Appropriate start times $\tau^C_{i^* v_k} \in \Theta^C_{i^*}$ to schedule test i^* on a vehicle v_k

inventory $\underline{R}_{v_k}$ of v_k nor any partially ordered destructive relation with a scheduled test $i \in C \cap V^t$ is violated (cf. Fig. 52.5). Note that it is not necessary that a start time S_{v_k} has already been assigned to building up activity b_{v_k} when scheduling test i^* on vehicle v_k. Then, for vehicle v_k we set $\vartheta^C_{i^*v_k} := \vartheta^C_{i^*v_k} \setminus [0, \beta_{min}]$ to ensure that at least the smallest assignable start time $S^P_{\mu'} \in B$ (i.e., $S^P_{\mu'} + p_b = \beta_{min}$) can be assigned to building up activity b_{v_k} such that it can be finished before test i^* may start.

Recall that Theorem 52.1 i) says that the objective function value does not change within the set of schedules S' implying the same precedence constraints. Thus, within this set of schedules we may always select, for instance, the schedule for which all activities start as early as possible. This means that only a number of *appropriate start times* $\tau^C_{i^*v_k} \in \vartheta^C_{i^*v_k}$ need to be examined for feasibly scheduling i^* on some vehicle $v_k \in \mathscr{M}_{i^*}$ which is already used by some tests $i \in C \cap V^t$. In particular, it is sufficient to consider only those points in time $\tau^C_{i^*v_k} \in \vartheta^C_{i^*v_k}$ at which a scheduled test i ends on vehicle v_k as well as point in time $\tau^C_{i^*v_k} := \min\{\tau \in \vartheta^C_{i^*v_k}\}$. For each vehicle $v_k \in M_{i^*}$ that is already used by a test $i \in C \cap V^t$ all appropriate start times $\tau^C_{i^*v_k}$ of test i^* are calculated and added to multiset $\Theta^C_{i^*}$. Within non-empty multiset $\Theta^C_{i^*}$ we may choose any appropriate start time, e.g., the smallest one $\tau^C_{i^*v'_{k'}} := \min \Theta^C_{i^*}$. By selecting start time $S_{i^*} := \tau^C_{i^*v'_{k'}}$ for test i^*, we simultaneously fix the mode assignment $\mathscr{M}_{i^*} := v'_{k'}$.

Referring to the example we introduced in Sect. 52.2, Fig. 52.5 illustrates the appropriate start times for test $i^* = 3$ on vehicle $v_k = 2_1$, where we assume that tests 2 and 4 have already been scheduled on vehicle 2_1, but no start time has been assigned to building up activity b_{2_1} thus far.

For $\Theta^C_{i^*} = \emptyset$ the execution of test i^* on an additional vehicle is examined as follows. First of all, considering an additional experimental vehicle implies that the next possible start time S^P_μ of ordered multiset S^P is added to B and after the necessary update of β_{max} and β_{min} it is checked, whether a building up activity must be scheduled on any existing vehicle like described before. Next, we determine the earliest time t at which test i^* may start. Since a possible start time $S^P_\mu \in B$ must be assignable to the building up activity of the added vehicle, i^* must not start before β_{min} and we set $t := \max(\beta_{min}, ES^C_{i^*})$.

If $t \leq LS^C_{i^*}$ holds, the variant v' of the additional vehicle is chosen such that test i^* can be performed on it (i.e., $v' \in M_{i^*}$) and a criterion called the *unsatisfied workload of variant* v, which is calculated by $\upsilon(v) := \sum_{\{i \in V^t \setminus C : v \in M_i\}} p_i$, is maximum.

If $t > LS^C_{i^*}$, it is not sufficient to add a vehicle to the underlying partial solution in order to find a feasible start time for activity i^*. In this case, we remove the possible start time S^P_μ previously added to multiset B and call procedure *Unschedule*(i^*) illustrated in Fig. 52.6. Due to the relative small latest start time of i^*, no possible start time $S^P_\mu \in B$ can be assigned to the building up activity of an additional vehicle. Hence, we first try to enlarge $LS^C_{i^*}$. To this end, we remove a set U of tests from the current partial solution which restrict $LS^C_{i^*}$ to a value smaller than

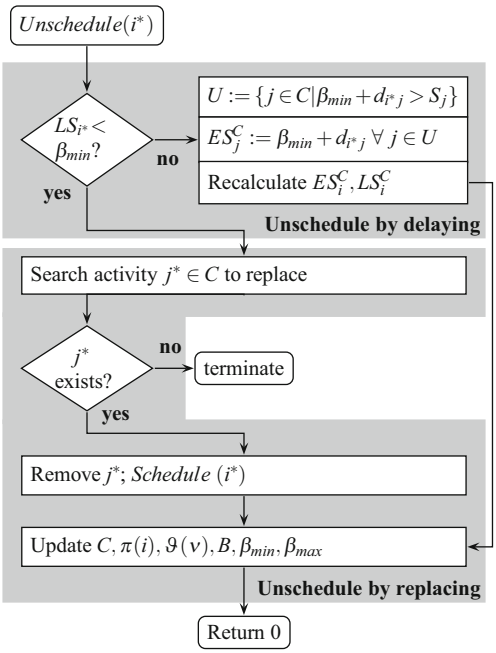


Fig. 52.6 Procedure *Unschedule*(i^*)

β_{min} (cf., e.g., Valls et al. 2006). Afterwards, we increase the earliest start times of all tests $j \in U$ such that, when these tests are scheduled again, LS_{i*}^C cannot become smaller than β_{min} and the building up activity of an additional vehicle can be processed before test i^* must start. However, if the initial latest start time $LS_{i^*} = -d_{i^*0}$ is smaller than β_{min}, this approach is not expedient and we search for a test $j^* \in C \cap V^t$ that can be replaced by test i^*. That is, we remove some test j^* from a vehicle v_k which enables us to schedule test i^* feasibly on v_k. Subsequently, test j^* is scheduled on a different vehicle or at a different time by calling procedure *Schedule*(j^*). If no j^* is found that can be replaced by test i^*, the procedure terminates without finding a feasible solution. By restricting the number of times a test j^* can be replaced, we prevent the procedure from cycling.

In our generation scheme (see Fig. 52.3) the priority values $\pi(i)$ are calculated using one of the following priority-rules (cf., e.g., Neumann et al. 2003)

- Minimum latest start time first (*LST*)
- Most total successors first (*MTS*)
- Least not scheduled total predecessors first (*LNTP*)
- Minimum number of modes first (*MNM*)
- Minimum slack time first (*MST*)

which displayed superior performance in a preliminary computational study. For the last rule *MST* let ST_i^C be the *schedule dependent slack time* of activity i that is calculated by $ST_i^C := LS_i^C - ES_i^C$.

We distinguish *static* and *dynamic* priority-rules. For static priority-rules like *MTS* and *MNM*, the priority values are never changed throughout the solution procedure, whereas for the dynamic versions of the rules *LST*, *LNTP* and *MST*, the priority values are updated each time a test is scheduled. On average, the dynamic versions of *LST*, *LNTP*, and *MST* lead to better results than their static versions. Thus, we report only on the dynamic versions of these priority-rules.

If two activities have the same priority with respect to the chosen priority rule a second rule or the activity number can be used to break ties. If a second rule is used, we speak of a priority rule combination.

Next, two variants of the aforementioned described generation scheme namely backward and bidirectional planning are outlined. In addition, we sketch a multi-start procedure where some generation scheme is applied repeatedly, each time using different priority values, and which holds a lot of promise compared to the single-pass heuristics for details we refer to Bartels and Zimmermann (2009).

52.4 Variants of the Priority-Rule Based Method

Because the basic variant presented in Sect. 52.3 fails to schedule efficiently in some special situations, it can be enhanced by some modifications in the schedule-generation scheme as described subsequently.

52.4.1 Backward Planning

In the basic schedule-generation scheme destructive and partially ordered destructive tests are scheduled as early as possible. This tends to be inefficient, because the used vehicle cannot be occupied by other tests afterwards. For some resource-constrained project scheduling problems *backward planning* turned out to be expedient to provide good heuristic solutions (cf., e.g., Klein 2000). A backward planning approach schedules the selected activity as late as possible, which should be advantageous for (partially ordered) destructive tests. That is, we either start selected test i^* at its latest start time $LS_{i^*}^C$ or such that it is completed at the beginning of a scheduled test $i \in C \cap V^t$ (i.e., $S_{i^*} + p_{i^*} = S_i$). If several appropriate start times leading to a feasible partial schedule exist, we select the latest one. Compared to the forward planning approach, we also make use of different priority-rules—e.g., instead of *LNTP* we now apply "Least not scheduled total successors first" (*LNTS*).

52.4.2 Bidirectional Planning

We may also combine the forward and the backward planning approach to a *bidirectional planning procedure* (cf., e.g., Klein 2000). Set $\mathscr{E}^C$ then contains all

activities for which all predecessors or all successors have already been scheduled. A test that has no preceding test $i \in V^t \setminus C$, is scheduled forward by means of the serial schedule-generation scheme. A test that has no succeeding activity $i \in V^t \setminus C$ is planned backward as indicated above.

52.4.3 Multi-Start Procedure

All proposed variants of our solution procedure are so called *single-pass heuristics* which either compute a single or no feasible solution. Usually, generating a single solution by means of the proposed heuristics requires less than 0.1 s for large problem instances with 600 tests and 25 variants of experimental vehicles. Thus, it is advisable to make use of a so called *multi-pass heuristic* (Valls et al. 2001), where the priority-rule method is applied repeatedly, each time using different priority values $\pi(i)$. However, since the number of adequate priority-rules is limited, we make use of *compound priority rules* and *values*. Let $\pi_{PR}(i)$ be the priority value for test i that is calculated by priority-rule PR, $\mathscr{P}$ be the set of priority-rules that should be applied within the multi-start procedure, and rnd_{PR} be a [0,1]-distributed random number for each rule PR. Within the multi-start procedure, the random numbers rnd_{PR} are generated individually for each solution that has to be computed. The compound priority value for test i is defined as

$$\pi(i) := \sum_{PR \in \mathscr{P}} rnd_{PR}\, \pi_{PR}(i). \tag{52.14}$$

Note that the priority values $\pi_{PR}(i)$ of each rule PR are normalized by dividing them by their maximum possible value.

52.5 A Genetic Algorithm

In this Section we suggest a *genetic algorithm* that makes use of the generation scheme of the priority-rule based heuristic described before. The genetic algorithm we developed to solve problem (52.4)–(52.13) is based on the so-called Standard Genetic Algorithm of Holland (1975). Applying principles from evolution theory, the Standard Genetic Algorithm works on a *population* of *individuals*, each of them representing a solution to the underlying scheduling problem. By iteratively selecting two promising individuals and recombining good properties of each individual, the algorithm improves the population and, thus, the solution. To this end, the Standard Genetic Algorithm consists of three elements, which are described below:

- *Selection strategy* which determines how the population evolves over time by selecting individuals to be eliminated or to be reproduced.

- *Crossover* that determines how two selected individuals are combined to generate new individuals.
- *Mutation* to randomly change the properties of selected individuals.

For more details on the theory of genetic algorithms we refer to Goldberg (1989) and Michalewicz (1996). Before we describe how the three elements are applied within our genetic algorithm, we first need to explain how a solution for our problem is represented by an individual. Based on concepts developed by Hartmann (1998), an individual is a list of activities that contains each test $i \in V^t$ exactly once. This list determines in which sequence the activities have to be scheduled by the generation scheme explained in Sect. 52.3. That is, the scheduling sequence of the tests is given by their order in the activity list, which hence replaces the function of the priority-rule. We also speak of *decoding an individual* when the generation scheme is applied to calculate a solution based on an activity list. Note that not every activity list necessarily represents a feasible solution.

Let us now present the selection strategy and also provide more detail on how the generation scheme is applied. Therefore, we have to introduce the term of a *gene pool*. A gene pool is a subset of the population, containing individuals that have been selected for reproduction. This also means that individuals that have not been selected cannot reproduce and, thus, are eliminated. The selection of individuals for the gene pool is driven by some selection strategy. Let us consider a population Φ of individuals ℓ (activity lists) with population size σ_{pop}. To determine which individuals ℓ of a population are selected for the gene pool, we calculate their so-called *fitness* $\overline{f}(\ell)$. Fitness $\overline{f}(\ell)$ of individual ℓ is represented by the value of the objective function of the solution it represents, adjusted by a corrective term taking the solution's "degree of feasibility" into account. To determine the fitness of an individual, we decode it into a solution using a slightly modified version of the generation scheme introduced in Sect. 52.3. The modification of the generation scheme is restricted to its Unschedule function. Remember that this function tries to unschedule activities in two different ways in case the generation scheme cannot find a feasible start time and execution mode for the current activity i^* to be scheduled. Firstly, it tries to unschedule predecessors or successors of i^* that restrict its scheduling window $\left[ES_i^C, LS_i^C\right]$. Secondly, it tries to remove a previously scheduled activity j on a vehicle which could be used to time-feasibly execute activity i^* if j would not be run on it. The latter is not applied for the genetic algorithm since we want to make sure that individuals "survive" where i^* and j are in an appropriate order and, thus, do not require a time consuming unscheduling step. Moreover, we limit the total number of unscheduling steps for decoding an individual to $\overline{u}$. In case an activity cannot be scheduled because we cannot find an unschedule option or the limit for unscheduling steps for the individual is reached, we do not interrupt the generation scheme as described in Sect. 52.3. Instead, we leave the activity not-scheduled and proceed with the next candidate in the activity list. At the end of the decoding procedure, the number of activities that could not be scheduled is added to the value of the objective function for the (partial) solution. This reflects the assumption that all not-scheduled activities had to be run on separate vehicles which

represents a kind of worst case consideration and therefore penalizes infeasible solutions. Let $\overline{V^t} \subseteq V^t$ be the set of activities that could not be scheduled. Then, the fitness of an individual ℓ that was decoded to solution (S^C, x^C) is

$$\overline{f}(\ell) := f(S^C, x^C) + |\overline{V^t}| \tag{52.15}$$

To select individuals for the gene pool, we make use of a so-called *tournament based selection* (Pohlheim 1999). The tournament based selection works as follows. We conduct $\mu = 1, \ldots, \sigma_{pop}$ binary tournaments that compare the fitness of individual ℓ with the fitness of a randomly selected individual ℓ' of the population with $\mu \neq \mu'$. The individual with the better fitness, i.e., the smaller value $\overline{f}(\ell)$, is considered winner of the tournament and selected for the gene pool. In case of both individuals having the same fitness the individual with the smaller order number is chosen. Given that every individual of the population is participating in at least one tournament, we ensure that the individual with the best fitness is part of the gene pool and the individual with the worst fitness is eliminated. Moreover, the better the fitness of an individual the higher the probability that it has multiple occurrences in the gene pool. On the other hand, the poorer the fitness of an individual the higher the probability that it will be eliminated. The tournament based selection is a non-fitness-proportional selection strategy and leads to sufficient selection pressure even for problems where the spread of the fitness across the individuals of a population is small. The latter is true for the problem under consideration.

Let us now describe the crossover which recombines individuals of the gene pool to generate new individuals. We stepwise select two arbitrary individuals of the gene pool which we call *father* and *mother*. A *crossover-probability* π_{cross} determines whether the individuals will actually be recombined (probability π_{cross}) or whether father and mother will become part of the new population without recombination (probability $1-\pi_{cross}$). In case of a recombination, we apply the following crossover-operators to generate a so-called *son* and a *daughter*. The *one-point crossover* works as follows. Let q be an integer random number from set $\{1, \ldots, n^t - 1\}$. To generate a daughter we copy the first q tests from the father's to the daughter's activity list and then fill up the remaining $n^t - q$ activities based on the order of tests in the mother's activity list. Of course, we only copy those activities from the mother that have not already been added from the father, i.e., the daughter's activity lists contains each test $i \in V^t$ exactly once. This procedure is depicted in Fig. 52.7. Additionally, we make use of an alternative crossover-operator, the *two-point-crossover*. For this operator, we determine two integer random numbers q_1 and q_2 with $q_1 < q_2$ from set $\{1, \ldots, n^t - 1\}$. The first q_1 tests for the daughter's activity list are copied from the father, the next $q_2 - q_1$ tests result from the mother, and the remaining $n^t - q_2$ tests are again copied from the father's activity list (see Fig. 52.7). For both crossover-operators a son is generated by exchanging the role of father and mother.

Next to the crossover-operator, we need to define the *evolution strategy* that controls how the next population is created based on the generated individuals. Three different strategies were tested. The first strategy builds the new population solely based on the generated daughters and sons. The second strategy considers

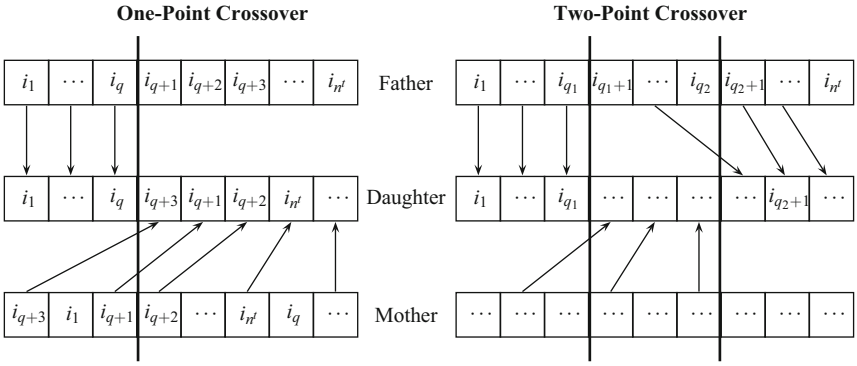


Fig. 52.7 One-point and two-point crossover. *Source*: Neumann et al. (2003)

all four individuals being part of a crossover—father, mother, son, and daughter—and keeps those two with the best fitness for the new generation. The third evolution strategy firstly lets all four individuals involved in a crossover become part of the new population. That is, once all crossovers have been conducted, the population consists of $2\sigma_{pop}$ individuals. To restore the original population size σ_{pop}, half of the population is eliminated by selecting those individuals with the worst fitness. Computational results have shown that the first evolution strategy leads to the best results. Hence, we are going to restrict our considerations to that strategy in the remainder of this contribution. Let us mention that we do not apply an elite strategy which would guarantee that the fittest individual of the current population automatically becomes part of the next solution. In computational studies we realized that an elite strategy leads to premature convergence of the genetic algorithm. Instead we put the best solution found throughout the course of the calculations on storage.

Finally, we describe the third element of our genetic algorithm—the *mutation* which randomly modifies individuals of the population. Parameter π_{mut} controls the probability of an individual being mutated. For the mutation itself we make use of a *swap-operator*. This operator exchanges the position of two activities being adjacent on the activity list. Please note, that any activity list can be generated by using this swap operator multiple times consecutively.

Having introduced all relevant elements of our genetic algorithm, we now can describe the overall procedure that is depicted in Fig. 52.8. In an initial step we generate a population consisting of σ_{pop} individuals that are randomly generated by a permutation of the tests $i \in V^t$. The purpose of generating the start population randomly is to find individuals that represent different areas of the solution space. In each main step of the algorithm we create a new generation of the population. Therefore, we first calculate the fitness of all individuals in the current population using the generation scheme described in Sect. 52.3 with the slightly modified *Unschedule()* function and then generate the gene pool by applying the tournament selection. The individual with the best fitness is put on storage in case it is better

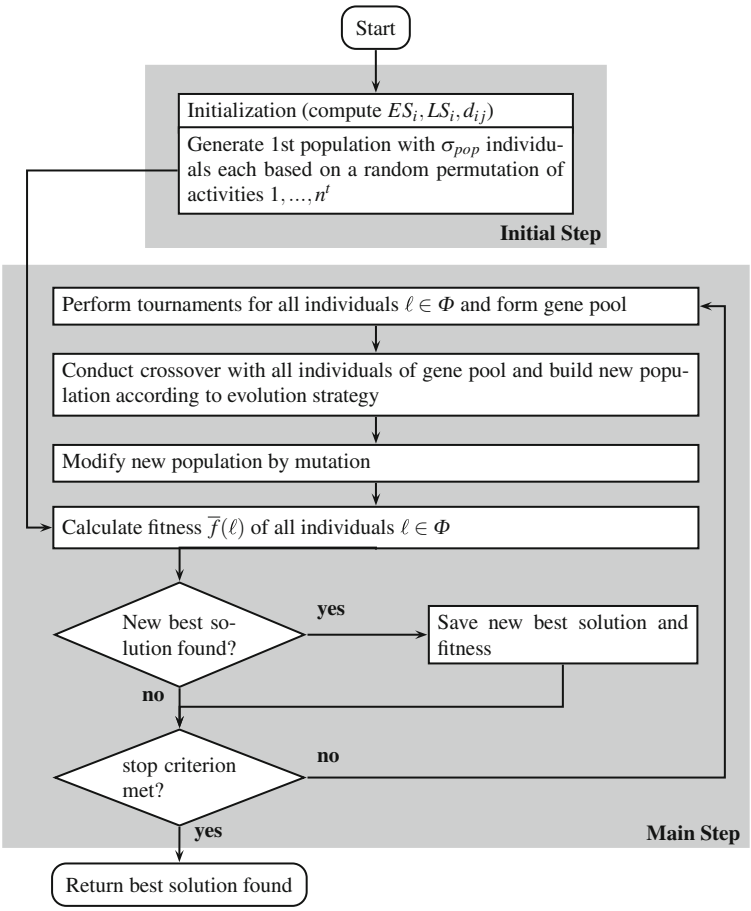


Fig. 52.8 Genetic algorithm

than any individual found before. Then, with the help of the crossover (either the one-point or the two-point version) we generate daughters and sons that, according to our evolution strategy, form the next generation of our population. Finally, we apply the described mutation to the individuals of the new population and proceed with the next main step until one of the following stop criteria is met:

- The value of the objective function of the best solution found so far is equal to a lower bound (for the determination of lower bounds to our problem see Bartels 2009)
- A maximum number of generations of the population has been built
- A maximum computation time has been reached
- No feasible solution has been found within a maximum number of populations

- The value of the objective function has not been improved within a maximum number of iterations
- All individuals of a generation have the same fitness

Let us make one final remark. The presented approach—even though in experimental pre-analysis it led to the best results—does not fulfill one criterion that is often claimed for genetic algorithms. Our activity-list based representation of solutions together with the generation scheme described in Sect. 52.3 cannot make sure that for any instance of problem (52.4)–(52.13), which has a feasible solution, an optimal solution can be found. Therefore, we experimented with different representations for a solution of problem (52.4)–(52.13).

In what follows, we show that based on our generation scheme two additional lists can ensure that an existing optimal solution can be coded for any problem instance. A *precedence list* with variable length prescribes additional precedence relationships between two activities that have to be considered in addition to the given temporal constraints. A *variants list* guides the generation scheme which variant of a vehicle to use when an additional prototype is needed. That is, this list replaces the criterion of the "unsatisfied workload" to control the variant selection as described in Sect. 52.3. Moreover, by constructing appropriate cases, one can show that any two of these three lists (activity, precedence, and variants list) are not sufficient for coding an optimal solution. However, an important disadvantage of this representation by three lists is that it leads to an exponential increase of potential individuals compared to solely using the activity list, while many individuals redundantly represent the same solution. Thus, the genetic algorithm has to calculate much more solutions before it starts converging. Pre-studies have shown that any combination of at least two lists led to significant higher computation times (50 % higher) but did not lead to significantly improved results compared to the representation by a simple activity list.

Finally, we have experimented with different versions of the genetic algorithm in terms of evolution strategy, selection strategy, and generation scheme, using forward, backward, and bidirectional generation schemes as well as a combination of them. In the end, even allowing large maximum computation times, the algorithm using solely the activity list and applying the forward planning generation scheme worked best in terms of solution quality and feasibility. This is due to the fact that it enables very fast calculations for decoding and crossover operations. Therefore, we restrict our consideration to the described version in the remainder of this chapter.

52.6 Computational Results

In this section we present the results of an experimental performance analysis for the genetic algorithm introduced in Sect. 52.5. To illustrate the effectiveness of the evolutionary elements of the genetic algorithm, we compare the results to those of selected priority-rule based methods described in Sects. 52.3 and 52.4. Detailed

results of an experimental performance analysis for priority-rule based methods applied to our problem are provided in Bartels (2009).

For our performance analysis, we made use of one real-world problem instance with 133 tests and 8 different variants of experimental vehicles. Moreover, we analyzed the performance for two different standard test sets each of which containing 100 problem instances. These problem instances were generated with the problem generator *ProGen/max* (Kolisch et al. 1999 and Schwindt 1998) based on parameters deducted from the real-world problem instance. *Test set A* (*Test set B*) contains problem instances with 20 (600) tests and 4 (25) different variants of experimental vehicles. In each instance, 10 % of tests are destructive and 30 % of tests lead to partially ordered destructive relationships, where on average a test destroys a vehicle partially for 50 % of the remaining tests.[2] A more detailed description of the instances' properties and how they were generated is provided in Bartels (2009).

The experimental performance analysis was conducted using a PC with Athlon-3.2 GHz-Processor and 512 MB RAM operating under Windows XP (32 Bit). All algorithms were coded in ANSI-C using Microsoft Visual Studio .NET 2003.

If not indicated differently in the subsequent explanations, we made use of the following implementation variants and parameters for the individual methodologies. Based on experimental pre-analyses we parametrized the genetic algorithm in a way that it does not converge too early and at the same time leads to the most promising results. It turned out that a population size of $\sigma_{pop} = 20$ individuals for test set A and $\sigma_{pop} = 100$ for test set B, a crossover probability of $\pi_{cross} = 0.8$ combined with the two-point crossover, and a probability for mutation of $\pi_{mut} = 0.05$ led to best results. Moreover, we restricted the number of unscheduling steps $\overline{u}$ for a single-pass priority-rule based method to $\overline{u} = (n^t)^2$, for a multi-pass heuristic to $\overline{u} = n^t$, and for the genetic algorithm to $\overline{u} = \frac{n^t}{2}$. Differentiating the number of maximum unscheduling steps is driven by the following rationale. Since for the single-pass heuristic we only have one pass to come up with a feasible solution, we allow more unscheduling steps to increase the probability of calculating a feasible solution at all. For the genetic algorithm, we restrict the number of unscheduling steps in order to increase its efficiency by generating selection pressure of those individuals that require no or only few unscheduling steps. Please note that based on these parameters all analyzed solution procedures found a feasible solution for every problem instance.

First of all, we analyzed the small problem instances of test set A with 20 tests each. For all these instances, optimal solutions based on a MILP-formulation of the problem solved with CPLEX 10.0 are known (for details see Bartels and Zimmermann 2009). Therefore, we can compare the optimal solutions to the results calculated by the procedures under consideration. For the genetic algorithm (*GA*) we restrict the number of generated populations to 300, i.e., given a population size of $\sigma_{pop} = 20$ we calculate 6,000 solutions for every instance unless the procedure does

[2]The test sets are available at the following URL: https://www.wiwi.tu-clausthal.de/testsets-evt/.

Table 52.1 Comparison of performance results for test set A

Method	*GA*	*SAM*	$MBP_{MNM/MST}$	$FWD_{LST/MNM}$
n_{sol}^{max}	6,000	6,000	6,000	1
$\Delta_{opt}^{\varnothing}[\%]$	1.7	5.1	5.4	16.7
$p_{opt}[\%]$	91	72	69	26

not finish earlier due to one of the other stop criteria. To make results comparable, we let the multi-pass heuristics also calculate 6,000 solutions per instance.

Subsequently, we present the results for two different versions of the multi-pass heuristics sketched in Sect. 52.4. Firstly, we consider the multi-pass bidirectional heuristic (*MBP*) with priority-rule combination *MNM/MST* ($MBP_{MNM/MST}$). Secondly, a sampling multi-pass heuristic (*SAM*) which randomly determines the planning sequence of the individual activities based on a random sampling approach (Kolisch and Hartmann 1999) is treated. That is, it randomly generates activity lists. The genetic algorithm is compared to the sampling approach in order to evaluate the effectiveness of the GA's evolutionary elements. Therefore, for the sampling approach we also make use of the forward planning generation scheme. Finally, we show the results for a priority-rule based single-pass heuristic that just generates a single solution. Here, the forward planning approach with priority-rule combination *LST/MNM* ($FWD_{LST/MNM}$) provided the best solutions for test set A compared to all other single-pass heuristics presented in Sect. 52.3. Table 52.1 shows the results for test set A where $\Delta_{opt}^{\varnothing}$ is the average relative deviation of the objective function value of the calculated compared to an optimal solution, and p_{opt} denotes the percentage of problem instances that have been solved to optimality. The best results for test set A are calculated by the genetic algorithm with 1.7 % average deviation from an optimal solution. For 91 % of the problem instances an optimal solution was found. These results are significantly better than those for the sampling multi-pass heuristic. Thus, the evolutionary elements of the genetic algorithm seem to pay off. Comparing the sampling multi-pass heuristic to the priority-rule based multi-pass heuristic it delivers slightly better results. This seems surprising on a first view but can be explained by the big number of solutions being calculated. While the priority-rules restrict the search to a certain area of the solution space, the sampling approach seems to analyze a much broader spectrum of the solution space given that it can generate any planning sequence of activities.[3] Thus, for the remainder of the analysis we will restrict our comparison of the genetic algorithm to the sampling approach. Finally, let us briefly consider the results for the priority-rule based single-pass method. Even for the small problem instances its best version $FWD_{LST/MNM}$ cannot compete with the genetic algorithm at all. Compared to the genetic algorithm, it finds optimal solutions for less than a third of the problem

[3]Nota bene: A deeper analysis of this phenomenon has shown that for smaller sample sizes (e.g., 100 calculated solutions) the priority-rule based method leads to better results than the sampling algorithm.

instances and the average deviation from an optimal solution's objective value is almost ten times higher.

For test set B no proven optimal solution is known so far. However, at least a good lower bound for the problem at hand is given that was developed by Bartels (2009) based on workload considerations. For the instances of test set A the average deviation between an optimal solution and the lower bound amounts to 13.6 %. In what follows, we compare the calculated results for test set B to its lower bound, where $\Delta_{LB}^{\varnothing}$ denotes the average relative deviation of the calculated objective function value from the corresponding lower bound. Moreover, we calculate the relative error $\Delta_{best}^{\varnothing}$ within the set of compared heuristics. For a solution of a problem instance let f_X be the objective function value that is calculated by heuristic X and f_{best} be the best objective function value that is computed by any of the heuristics that shall be compared. Then for heuristic X $\Delta_{best}^{\varnothing} = \frac{f_X}{f_{best}} - 1$ is the relative error with respect to the set of compared heuristics. Furthermore, for the problem instances of test set B the computation time becomes important, in particular with respect to the applicability of the individual methods in practice. Thus, we have also analyzed the average computation time per instance t_{CPU} for the different heuristics.

For the genetic algorithm (*GA*), we first of all analyzed how the number of generated populations affects the solution quality. Therefore, we restricted the number of populations to 4,000, 10,000, and 30,000. Given the size of the population with $\sigma_{pop} = 100$ this corresponds to a maximum of 0.4, 1, and 3 million solutions generated per problem instance, respectively. Like for test set A, we compared the results to a sampling multi-pass heuristic (*SAM*) that also generated 0.4 million solutions per instance based on randomly generated activity lists.

Table 52.2 shows the results revealing two main insights.

Firstly, the solution quality increases with the maximum number of populations the genetic algorithm is allowed to generate. With 30,000 populations we found a best solution for every problem instance across all considered heuristics and deviate 15.7 % from a lower bound. Given that for the small test set A the deviation between optimal solution and lower bound is 13.6 % and the genetic algorithm converges at about 30,000 populations, this might be a hint that we are not far away from optimality. However, to obtain these results we require approximately 4.5 h computation time. Accepting somewhat worse solutions, it is possible to reduce computation time to 2 or even 1 h by terminating after 10,000 or 4,000 populations, respectively. Here, one interesting insight is that the calculation of the first 4,000 populations takes almost as long as calculating the next 6,000 populations to reach 10,000 populations, where the next 20,000 populations take only slightly longer

Table 52.2 Comparison of performance results for test set B

Method	*GA*	*GA*	*GA*	*SAM*
n_{pop}^{max}	30,000	10,000	4,000	–
n_{sol}^{max} [Mil]	3	1	0.4	0.4
$\Delta_{LB}^{\varnothing}$[%]	15.7	17.8	20.5	32.8
$\Delta_{best}^{\varnothing}$[%]	0	1.8	4.2	15.0
$t_{cpu}^{\varnothing}(min)$	275.3	113.7	54.8	90.3

Table 52.3 Results for a real-world problem instance

Method	*MAN*	*GA*
# maximum generated solutions	1	300,000
$\#_{EV}$	35	24
$\Delta_{MAN}[\%]$	–	−31.4
t_{CPU}	∼1.5 weeks[a]	772 s

[a]Time span during which a manual planner generated a solution (with about 50–60 % of his capacity)

than the first 10,000. This illustrates the effectiveness of the selection pressure to foster individuals requiring no or only few unscheduling steps.

Secondly, comparing the results of the sampling multi-pass heuristic with the corresponding genetic algorithm (both calculating up to 0.4 million solutions), shows the effectiveness of the evolutionary elements, again. The average deviation from a lower bound is more than 12 % points worse for the sampling multi-pass heuristic and the computation time is more than 50 % higher. A detailed analysis shows that the additional computation time for the sampling approach is partly explainable by the additional unscheduling-steps required. The other part of the additional computation time is due to the fact that generating the random numbers, which are required for the sampling approach, is quite time consuming.

Finally, we used our genetic algorithm to calculate a solution for the initially mentioned real-world problem instance with 133 tests and 8 variants of experimental vehicles. To this end, we calculated 10,000 populations with $\sigma_{pop} = 30$ individuals each. We compare our results with those having been generated by a manual planner (*MAN*) considering the number of required experimental vehicles $\#_{EV}$, the relative deviation from the manual solution Δ_{MAN}, and the computation time t_{CPU}. The results are shown in Table 52.3. After 772 s, the genetic algorithm comes up with a solution of 24 experimental vehicles. Compared to the solution with 35 experimental vehicles generated by the human planner it saves 31.4 % of the expensive experimental vehicles. Although computation times between a manual planning approach and an algorithmic approach are difficult to compare, we observe that the genetic algorithm can significantly speed up test planning processes in automotive R&D projects. Moreover, Bartels and Zimmermann (2009) calculated a solution with 26 experimental vehicles for the same problem instance using a multi-pass priority-rule based heuristic as described in Sect. 52.3. That is, using the genetic algorithm we improved the best solution known so far by two vehicles or 7.7 %.

52.7 Conclusions

In this contribution we revisit the problem of scheduling tests in automotive R&D projects presented in Bartels and Zimmermann (2009). We extend the set of solution procedures for this problem by a genetic algorithm that was developed in different

variants. The most promising variant is described in detail and is able to significantly improve solutions known before.

For small problem instances with up to 20 tests it finds optimal solutions for nearly all problem instances and outperforms the known priority-rule based multi-pass heuristics. The calculation time of 2.77 s is not prohibitive in practice.

For problem instances with up to 600 tests the evolutionary elements of the genetic algorithm lead to a clear outperformance compared to the multi-pass heuristics. Considering lower bounds for our test instances, we get a hint that the generated solutions might be close to an optimum once our genetic algorithm has converged at approximately 30,000 populations. However, to calculate these 30,000 solutions the genetic algorithm requires about 4.5 h computation time. Nevertheless, given that the problem of planning tests in automotive R&D projects belongs to the class of tactical planning problems, and therefore is conducted only few times per test series, a computation time of 4.5 h does not matter. It is even faster than the current manual approach applied in practice.

For a real world problem instance, we were able to calculate a test plan that improved the manually generated plan of a human planner by more than 30 %. In other words, even for a rather small problem instance with 133 tests the genetic algorithm was able to save 9 experimental vehicles each of which causing costs of up to 1.5 Million Euros. Although this single result cannot be used to make a general conclusion, it gives a hint that our solution procedure may enhance the planning methods currently applied.

Meanwhile, we have tested the approach successfully with a second automotive manufacturer. Going forward, we would suggest embedding the developed algorithm into a holistic planning tool. This tool should facilitate the modeling of the problem instance with all its constraints based on a knowledge database for testing (containing typical test data like duration, interdependencies with other tests, etc.), the visualization of the planning results, and a manual interface for the human planer who likes to have a final say before a test plan is approved.

For further research, we suggest to evaluate the eligibility of our method for related problems involving R&D projects in other industries, e.g., the aircraft industry. Moreover, an automotive manufacturer discussed with us a related problem in the environment of experimental vehicle testing. Objective of this problem is to level the utilization of test resources (experimental vehicles, testing rigs, test staff, etc.) once the number of experimental vehicles and the maximum duration of a test program has been fixed. We believe that this problem can be solved combining resource leveling methods as described in Neumann and Zimmermann (2000), Zimmermann (2001), and Gather et al. (2011) with findings on structural properties and solution approaches for the problem of scheduling tests in automotive R&D projects as described in this contribution, Bartels (2009), and Bartels and Zimmermann (2009).

References

Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows. Prentice Hall, Englewood Cliffs

Ballestín F, Schwindt C, Zimmermann J (2007) Resource leveling in make-to-order production: modeling and heuristic solution method. Int J Oper Res 4(1):1–13

Bartels JH (2009) Anwendungen von Methoden der ressourcenbeschränkten Projektplanung mit multiplen Ausführungsmodi in der betriebswirtschaftlichen Praxis. Gabler, Wiesbaden

Bartels JH, Zimmermann J (2006a) Optimizing the use of experimental vehicles: solution procedures. In: Proceedings of the 10th international workshop on project management and scheduling, Poznań, pp 68–72

Bartels JH, Zimmermann J (2006b) Scheduling tests in automotive R&D projects. In: Haasis HD, Kopfer H, Schönberger J (eds) Operations research proceedings 2005. Springer, Berlin, pp 661–666

Bartels JH, Zimmermann J (2009) Scheduling tests in automotive R&D projects. Eur J Oper Res 193(3):805–819

Chelst K, Sidelko J, Przebienda A, Lockledge J, Mihailidis D (2001) Rightsizing and management of prototype vehicle testing at Ford Motor Company. Interfaces 31:91–107

Gather T, Zimmermann J, Bartels JH (2011) Exact methods for the resource levelling problem. J Sch 14(6):557–569

Gembrys SN (1998) Ein Modell zur Reduzierung der Variantenvielfalt in Produktionsunternehmen. Ph.D. dissertation, Technical University of Berlin, Berlin

Goldberg (1989) Genetic algorithms in search, optimization and machine learning. Kluwer, Boston

Hartmann (1998) A competitive genetic algorithm for resource-constrained project scheduling. Nav Res Log 45:733–750

Heilmann R (2000) Ressourcenbeschränkte Projektplanung im Mehr-Modus-Fall. Gabler, Wiesbaden

Heilmann R (2001) Resource-constrained project scheduling: a heuristic for the multi-mode case. OR Spectr 23:335–357

Henseler P (2004) Die Konfigurations- und Verträglichkeitsmatrix als Beitrag für eine differenzierte Betrachtung von Konfigurierungsproblemen. VDI Verlag, Düsseldorf

Holland J (1975) Adaption in natural and artificial systems. MIT Press, Ann Arbor

Klein R (2000) Scheduling of resource-constrained projects. Kluwer, Boston

Kolisch R, Hartmann S (1999) Heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis. In: Węglarz J (ed) Project scheduling: recent models, algorithms and applications, Kluwer, Boston, pp 147–178

Kolisch R, Schwindt C, Sprecher A (1999) Benchmark instances for project scheduling problems. In: Węglarz J (ed) Project scheduling: recent models, algorithms and applications. Kluwer, Boston, pp 197–212

Limtanyakul K (2008) Scheduling of tests on vehicle prototypes using constraint and integer programming. In: Kalcsics J and Nickel S (eds) Operations research proceedings 2007, Springer, Berlin, pp 421–426

Limtanyakul K (2009) Scheduling of tests on vehicle prototypes. Ph.D. dissertation, Technical University of Dortmund, Dortmund

Limtanyakul K, Schwiegelshohn U (2007) Scheduling tests on vehicle prototypes using constraint programming. In: Proceedings of the 3rd multidisciplinary international scheduling conference: theory and applications, Paris, pp 336–343

Limtanyakul K, Schwiegelshohn U (2012) Improvements of constraint programming and hybrid methods for scheduling of tests on vehicle prototypes. Constraints 17(2):172–203

Lockledge J, Mihailidis D, Sidelko J, Chelst K (2002) Prototype fleet optimization model. J Oper Res Soc 53:833–841

Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs. Springer, Berlin

Neumann K, Schwindt C (2002) Project scheduling with inventory constraints. Math Method Oper Res 56:513–533

Neumann K, Zimmermann J (2000) Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. Eur J Oper Res 127:425–443

Neumann K, Schwindt C, Zimmermann J (2003) Project scheduling with time windows and scarce resources. Springer, Berlin

Neumann K, Schwindt C, Zimmermann J (2006) Resource-constrained project scheduling with time windows: recent developments and new applications. In: Jozefowska J, Węglarz J (eds) Perspectives in modern project scheduling. Springer, New York, pp 375–407

Pohlheim H (1999) Evolutionäre Algorithmen. Springer, Berlin

Risse J (2002) Time-to-Market-Management in der Automobilindustrie: ein Gestaltungsrahmen für ein logistikorientiertes Anlaufmanagement. Ph.D. dissertation, Technical University of Berlin, Berlin

Roy B (1964) Les problèmes d'ordonnancement. Dunod, Paris

Scheffermann R, Clausen U, Preusser A (2005) Test scheduling on vehicle prototypes in the automotive industry. In: Proceedings of 6th Asia-Pacific industrial engineering and management systems (APIEMS), vol 11, pp 1817–1830

Schwindt C (1998) Generation of resource-constrained project scheduling problems subject to temporal constraints. Technical Report WIOR 543, University of Karlsruhe, Karlsruhe

Selle T, Zimmermann J (2003) A bidirectional heuristic for maximizing the net present value of large-scale projects subject to limited resources. Nav Res Log 50:130–148

Valls V, Ballestín F, Barrios A (2006) An evolutionary algorithm for the resource-constrained project scheduling problem subject to temporal constraints. In: Proceedings of the 10th international workshop on project management and scheduling, Poznań, pp 370–376

Valls V, Quintanilla S, Ballestín F (2001) Resource-constrained project scheduling: a critical activity reordering heuristic. Eur J Oper Res 149:282–301

Zimmermann J (1997) Heuristics for resource levelling problems in project scheduling with minimum and maximum time lags. Technical Report WIOR 491, University of Karlsruhe, Karlsruhe

Zimmermann J (2001) Ablauforientiertes Projektmanagement: Modelle, Verfahren und Anwendungen. Gabler, Wiesbaden

Zimmermann J, Bartels JH (2006) Optimizing the use of experimental vehicles: problem description and mathematical model. In: Proceedings of the 10th international workshop on project management and scheduling, Poznań, pp 383–387

Zimmermann J, Stark C, Rieck J (2010) Projektplanung: Modelle, Methoden, Management. Springer, Berlin

Chapter 53
Scheduling of Production with Alternative Process Plans

Roman Čapek, Přemysl Šůcha, and Zdeněk Hanzálek

Abstract This chapter deals with a scheduling problem with alternative process plans that was motivated by a production of wire harnesses where certain parts can be processed manually or automatically by different types of machines. Only a subset of all the given activities will form the solution, so the decision whether the activity will appear in the final schedule has to be made during the scheduling process. The problem considered is an extension of the resource constrained project scheduling problem with positive and negative time-lags and sequence dependent setup times. We extend the classic RCPSP problem by a definition of alternative branchings and for this representation of the problem, an mixed integer linear programming model is formulated. Furthermore, a heuristic algorithm based on priority schedule construction with an unscheduling step is proposed for the considered problem and it is used to solve the large instances of the considered problem.

Keywords Alternative process plans • Heuristic algorithm • Mathematical model • Project scheduling • Resource constraints • Temporal constraints

53.1 Introduction

Production processes often involve more than one way how to complete the product. Such alternative process plans occur in the production of wire harnesses, where operations to produce a wire harness can be performed in various ways, using fully automated machines, semi-automated machines or manually operated ones with special equipment. Not only the resource requirements are different, but the processing times, precedence relations and also the number of activities in each process plan can differ in general, too. The process plan defines a set of activities such that their execution leads to the completion of a product. Each process plan

R. Čapek (✉) • P. Šůcha • Z. Hanzálek
Faculty of Electrical Engineering, Department of Control Engineering, Czech Technical University, Prague, Czech Republic
e-mail: capekrom@fel.cvut.cz; suchap@fel.cvut.cz; hanzalek@fel.cvut.cz

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_23

is formed by a set of disjunctive activities. On the other hand, an activity can be included in more process plans. We use the term *alternative process plans* since there are more process plans in the studied problem while only one of them has to be executed. Hence the goal of the scheduling is to choose a subset of all activities that forms one process plan and schedule them according to the given constraints and criterion.

Traditional scheduling algorithms according to Błażewicz et al. (1996) assume exactly given set of activities to be scheduled, i.e. only one process plan is defined. In this chapter, the traditional scheduling approach is extended by a definition of alternative process plans, i.e. the traditional time scheduling and the decision which process plan will be executed are both integrated into one problem. The problem can be formalized as an extension of the $PS|temp, s_{ij}|C_{max}$ problem. Therefore, we deal with the resource constrained project scheduling problem with positive and negative time-lags, sequence dependent setup times and alternative process plans. Time-lags (also called generalized precedence relations) are useful to specify the relative time position of two activities in general. Sequence dependent setup times serve to cover the time needed to change the equipment or set up a machine between two different operations. The optimality criterion is to minimize the schedule length.

Although the resource constrained project scheduling problem (RCPSP) is a well-studied problem, there were only a few attempts to include the alternatives into the scheduling process. However, the alternative process plans can be found as a natural part of the production processes and, therefore, we have decided to extend the RCPSP problem by the definition of alternative process plans (*RCPSP-APP*). The combination of generalized precedence relations and logical constraints (in form of alternative process plans) makes the problem even more difficult since we have to introduce new decision variables into the problem.

This chapter presents the RCPSP-APP problem motivated by the real production of wire harnesses. A literature overview is given in Sect. 53.2. Section 53.3 contains the statement of the $PS|temp, s_{ij}, nestedAlt|C_{max}$ problem with the representation based on the RCPSP-APP formalism. The model also considers sequence dependent setup times and generalized precedence relations (positive and negative time-lags). Section 53.4 contains the mathematical formulation. A heuristic method, where the choice of process plan and traditional scheduling are executed simultaneously, is described in Sect. 53.5. Computational experiments are discussed in Sects. 53.6 and 53.7 concludes the work.

53.2 Related Works

The problem outlined in the previous section is addressed in the literature as *scheduling with alternative* (*optional*) *activities* (*tasks*) or problem with *alternative process plans* or *scheduling with alternatives* (Beck and Fox 2000; Barták and Čepek 2008; Capacho et al. 2009; Čapek et al. 2012b,a; Čapek 2012). To avoid any misunderstandings, let us assume that the notions *activity*, *operation* and *task*

have the same meaning and we will prefer to use the term *activity* in this chapter. To represent alternative process plans, some type of special graph is usually used in the existing works. Beck and Fox (2000) established the *Modified Temporal Graph* with so called *XorNodes*, *AndNodes* and *ActivityNodes* to model the possibility of choice among the process plans. All process plans are interconnected via the aforementioned nodes. Another approach to model the alternative process plans in scheduling, similar to the Modified Temporal Network methodology, was presented by Barták and Čepek (2008). They use a special type of graph called *Nested Temporal Network with Alternatives* (NTNA), which is a directed acyclic graph where the nodes represent activities and the arcs correspond to temporal constraints. Logical constraints are specified through the input and output labels of each node. The NTNA formalism is used to represent the problem structure in this chapter.

Capacho et al. (2009) studied an assembly line balancing problem with alternatives, where certain parts can be processed in several alternative modes and the goal is to balance the workload of the available resources. Kis (2003) presented a genetic algorithm and a tabu search for the job-shop problem with processing alternatives using a special graph to represent the problem instance. Shao et al. (2009) dealt with the problem of integrated planning and scheduling, which is close to the job shop problem with alternative process plans, since each job includes more alternative ways (process plans) to complete the product. The goal is to select a process plan for each job and schedule job activities such that the schedule length is minimized.

The resource constrained project scheduling problem (RCPSP) is a well known problem with many real applications. Dorndorf et al. (2000) proposed an effective branch and bound method for the RCPSP problem denoted as $PS|temp|C_{max}$ (Błażewicz et al. 1996; Brucker et al. 1999a). The solution method is based on the constraint propagation that reduces the search space. Nonetheless, setup times are not considered and also the introduction of alternative process plans would not be straightforward. Brucker et al. (1999a) summarized the notation of the RCPSP including both single-mode and multi-mode problems with various resource environments and activity characteristics. With the proposed classification, the resource constrained project scheduling problem with positive and negative time-lags and sequence dependent setup times can be formulated as $PS|temp, s_{ij}|C_{max}$. The problem studied in this chapter is an extension of such problem. Hanzálek and Šůcha (2009) published a constructive algorithm for the $PS|temp, s_{ij}|C_{max}$ problem which is extended to solve the problem with alternative process plans in this chapter.

Several solutions methods are described throughout this book. The schedule generation approaches for the RCPSP problem are described in Chap. 1 in the first volume of this handbook, Chap. 2 presents various mathematical formulations of the RCPSP problem, and in Chap. 4, the metaheuristic approaches are discussed. The formulation for the generalized precedence relations is presented in Chap. 5 in the first volume of this handbook. Finally, the overviews for the multi-mode problem extension are given in Chaps. 21 and 22 in the first volume of this handbook.

53.3 Problem Statement

Let the production consist of n indivisible operations performed on the specified machines according to the process plan. Consequently, there is a set of $n + 2$ non-preemptive activities $V = \{0, \ldots, n + 1\}$ to be scheduled on a set of K resources $\mathscr{R} = \{1 \ldots K\}$ where each resource $k \in \mathscr{R}$ has a discrete capacity $R_k \geq 1$, i.e. R_k units are available for resource k. Each activity i is characterized by the processing time p_i and resource demand $r_{ik} \geq 0$ for the resource $k \in \mathscr{R}$. Only mono-resource activities are considered in this chapter, meaning that each activity demands exactly one resource, i.e. $\sum_{k \in \mathscr{R}: r_{ik} > 0} (1) = 1$ for all $i \in \{1, \ldots, n\}$. Activities 0 and $n + 1$ with $p_0 = p_{n+1} = 0$ and $r_{0k} = r_{n+1\,k} = 0$ for all $k \in \mathscr{R}$ denote *dummy* activities such that activity 0 is a predecessor and activity $n + 1$ is a successor of all other activities. Precedence relations together with the definition of alternative process plans are specified using an NTNA formalism (Sect. 53.3.1).

Generalized precedence relations, also called positive and negative time-lags, are defined such that $S_i + d_{ij} \leq S_j$ for all $(i, j) \in V^2$, where $d_{ij} \in \mathbb{R}$ is the length of the time-lag and S_i is the start time of activity i in the schedule. We assume that $d_{ij} \geq 0$ for all $(i, j) \in V^2 : i \in Pred\,(j)$. If there is no temporal constraint from i to j, then $d_{ij} = -\infty$. If activity i has to be constrained by the release time r_i and the deadline $\overline{d_i}$, then $d_{0i} = r_i$ and $d_{i0} = -\overline{d_i}$.

Sequence dependent setup times s_{ij} are considered as an additional time needed for setting up the resource between the activities scheduled consequently on the same resource. We presume that the setup times satisfy the triangular inequality $s_{ij} + s_{jk} \geq s_{ik}$ for all $\{i, j, k\} \in V^3$ (Brucker 2007).

The goal of the scheduling is to select a subset $V' \subseteq V$ of all activities (i.e. one process plan) such that constraints for selection of activities given by the NTNA instance (Sect. 53.3.1) are fulfilled and then to schedule V' to a set of resources while minimizing the schedule length. To represent the schedule, three types of variables $v_i \in \{0, 1\}$, $S_i \in \mathbb{R}_0^+$ and $z_{iqk} \in \{0, 1\}$ are considered. If $v_i = 1$, then activity i is present in the schedule and it is called *selected* activity; if $v_i = 0$, then the activity i is not in the schedule and it is called *rejected* activity. Variable S_i denotes the start time of activity i in the schedule. Finally, variable z_{iqk} denotes whether activity i is assigned to a resource unit q of resource k.

The described problem can be classified as $PS|temp, s_{ij}, nestedAlt|C_{max}$ using $\alpha|\beta|\gamma$ notation (Błażewicz et al. 1996; Brucker et al. 1999a) where *temp* denotes the generalized precedence relations and s_{ij} represents the setup times. We add the term *nestedAlt* to the field describing constraints of the problem to denote the presence of alternative process plans in the nested form (see the following subsection).

53.3.1 Nested Temporal Networks with Alternatives

To define alternative process plans, the formalism of Nested Temporal Network with Alternatives (NTNA) , proposed by Barták and Čepek (2008), is used. NTNA is an acyclic directed graph where nodes represent activities and edges represent temporal constraints. Each node i of the graph (corresponding to activity i) has an input label in_i and an output label out_i, denoting the type of input/output *branching* which can be either *parallel* or *alternative*.

When there is a parallel branching at the input/output of selected activity i ($in_i/out_i = 0$), all its direct predecessors/successors have to be selected. If activity i is rejected, all its direct predecessors/successors have to be rejected. On the contrary, when there is an alternative branching at the input/output of selected activity i ($in_i/out_i = 1$), exactly one of its direct predecessors/successors has to be selected. If activity i is rejected, all its direct predecessors/successors have to be rejected. Finally, the selection rule for a pair of activities i and j constrained by a simple precedence (i has only one successor j and vice versa j has only one predecessor i) is that both activities have to be Selected/rejected simultaneously. For the sake of simplicity, $out_i = in_j = 0$ for such activities.

The term *branch* denotes a set of activities forming a connected component that starts by a direct successor of activity i and ends by a direct predecessor of corresponding activity j. A branch can further contain another parallel or alternative branching and both parallel and alternative branchings can be arbitrary nested one in another. Temporal constraints in the formalism of Nested Temporal Networks with Alternatives are defined as follows. For each edge between nodes i and j, there is a time-lag $d_{ij} \geq 0$ denoting the minimal time distance between start times of activities i and j respectively. If it is needed to define a time-lag for activities i and j which are not connected by the edge in the NTNA, d_{ij} is set to the desired value without modification of the NTNA structure. An example of the NTNA instance is depicted in Fig. 53.1. Parallel branchings are denoted as *PAR* and alternative branchings are denoted as *ALT*. Each edge represents one temporal constraint, for the sake of simplicity only two of them are depicted, and there are two further minimal time-lags $d_{97} = 3$ and $d_{11\,14} = 8$ and one maximal time-lag $d_{90} = -16$.

53.4 Mathematical Model

In this section, the MILP model for the $PS|temp, s_{ij}, nestedAlt|C_{max}$ problem is formulated. There are three types of constraints in the model—constraints for selection of activities (53.1)–(53.4), temporal constraints (53.5) and resource constraints (53.6)–(53.10).

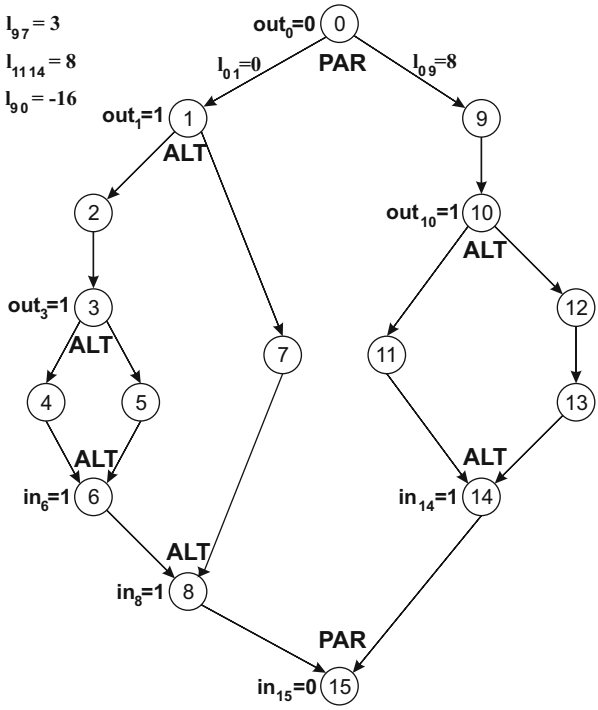


Fig. 53.1 Example of the NTNA instance

$$\underset{i \in V}{\text{Max.}} \ (S_i + p_i)$$

$$\text{s.t.} \quad v_i = \sum_{j \in Succ(i)} v_j \qquad (i \in V : out_i = 1) \tag{53.1}$$

$$v_i = \sum_{j \in Pred(i)} v_j \qquad (i \in V : in_i = 1) \tag{53.2}$$

$$v_i = v_j \qquad \left((i, j) \in V^2 : out_i = 0 \wedge in_j = 0 \wedge j \in Succ\,(i)\right) \tag{53.3}$$

$$\sum_{i \in V} v_i \geq 1 \tag{53.4}$$

$$S_i + d_{ij} \leq S_j + UB \cdot \left(2 - v_i - v_j\right) \qquad \left((i, j) \in V^2\right) \tag{53.5}$$

$$S_j + p_j + s_{ji} \leq S_i + UB \cdot \left(x_{ij} + y_{ij}\right) + UB \cdot \left(2 - v_i - v_j\right) \qquad ((i,j) \in \mathscr{M}) \tag{53.6}$$

$$S_i + p_i + s_{ij} \leq S_j + UB \cdot \left(1 - x_{ij} + y_{ij}\right) + UB \cdot \left(2 - v_i - v_j\right) \qquad ((i,j) \in \mathscr{M}) \tag{53.7}$$

$$-x_{ij} + y_{ij} \leq 0 \qquad ((i,j) \in \mathscr{M}) \tag{53.8}$$

$$z_{iqk} + z_{jqk} - 1 \leq 1 - y_{ij} \qquad ((i,j) \in \mathscr{M}; k \in \mathscr{R}; q \in \{1 \ldots R_k\}) \tag{53.9}$$

$$\sum_{q=1}^{R_k} z_{iqk} = r_{ik} \qquad (i \in V; k \in \mathscr{R}) \tag{53.10}$$

where

$$\mathscr{M} = \left\{(i,j) \in V^2 : i < j \wedge r_{ik} > 0 \wedge r_{jk} > 0 \text{ for any } k\right\}; S_i \in \mathbb{R}_0^+;\ v_i, x_{ij}, y_{ij}, z_{iqk} \in \{0,1\}$$

Formulas (53.1)–(53.4) represent rules for the selection of activities. Formulas (53.1) and (53.2) stand for the start and the end of alternative branching, Eq. (53.3) stands for parallel branchings and direct precedences and Eq. (53.4) eliminates solution corresponding to an empty schedule. Equation (53.5) represents temporal constraint between each pair of activities. If there is no temporal constraint, then $d_{ij} = -\infty$ and the equation is satisfied regardless the values of S_i and S_j. Formulas (53.6)–(53.10) represent the resource constraints. Equations (53.6) and (53.7) ensure that there will be no resource conflict between activities that are assigned to the same unit of the same resource, considering also the sequence dependent setup times. For this purpose, let x_{ij} be a binary decision variable such that $x_{ij} = 0$ if activity j precedes activity i on the same resource and $x_{ij} = 1$ otherwise. Furthermore, let y_{ij} be another binary decision variable such that $y_{ij} = 0$ if activities i and j share at least one resource unit and $y_{ij} = 1$ otherwise. Formulas (53.8) and (53.9) determine whether there can be actual resource conflict for each pair of activities and Eq. (53.10) ensures that each activity has assigned appropriate resource capacity.

53.4.1 Problem Complexity

Let us focus on the complexity of the problem with alternative process plans. The problem $PS|temp, s_{ij}|C_{max}$, i.e. the case without alternative process plans, is $\mathscr{NP}$-hard since it is a generalization of the $1|r_j, \overline{d_j}|C_{max}$ problem (see reduction of this problem from a three-partition problem in Lenstra et al. 1977). If the resource constraints are omitted, we have a $PS\infty|temp|C_{max}$ problem, which can be solved

in polynomial time (e.g. using linear programming while eliminating the resource constraints). On the other hand, the problem $PS\infty|temp, nestedAlt|C_{max}$, is $\mathcal{NP}$-hard, despite the resource constraints relaxation, see Čapek et al. (2012b) for more details. This leads to the observation that the computation of the earliest start times for all activities $i \in V$ is an $\mathcal{NP}$-hard for the problem $PS|temp, s_{ij}, nestedAlt|C_{max}$ since the $PS\infty|temp, nestedAlt|C_{max}$ problem is a subproblem of finding the earliest start times for all activities.

53.5 Heuristic Algorithm

Since $PS|temp, s_{ij}, nestedAlt|C_{max}$ is an $\mathcal{NP}$-hard problem, the optimal solution can be obtained, in reasonable amount of time, only for small instances. For large instances, we propose a heuristic algorithm that does not ensure finding an optimal solution, but it is able to handle instances with a significantly larger amount of activities. The idea of this algorithm, called Iterative Resource Scheduling with Alternatives (*IRSA*), is based on an IRS algorithm for $PS|temp, s_{ij}|C_{max}$ inspired by software pipe-lining and presented by Rau (1994) and extended by Hanzálek and Šůcha (2009) who focused on the acyclic scheduling problem and introduced so called *take-give resources* into the problem. It is a constructive method where activities are being added to the schedule according to their actual priority or being removed if the partial schedule is not feasible. The input of the algorithm is an instance of the $PS|temp, s_{ij}, nestedAlt|C_{max}$ problem. The output of the algorithm is a schedule S determined by the selected activities, their start times and assigned resource units, i.e. $S = [s, v, z]$. The main purpose of the proposed heuristic is to deal with the problems where a feasible schedule cannot be found in polynomial time in general case. The optimization of the C_{max} criterion is achieved by the gradual tightening of the constraint for the schedule length.

53.5.1 Initialization

The algorithm starts with the estimation of the bounds for the length of the schedule. The upper bound is computed as $UB = \sum_{i \in V} \max\left(p_i + \max_{j \in V}(s_{ij}), \max_{j \in V}(d_{ij})\right)$ (Brucker et al. 1999b). The lower bound is computed as $LB = S^{LB}_{n+1}$, i.e. the lower bound of the earliest start time of activity $n + 1$. For this purpose, let G^{temp} be a directed graph with nodes corresponding to activities $V(G^{temp}) = V$ and edges $E\,(G^{temp}) = \left\{(i, j) \in V(G^{temp}) \times V(G^{temp}) : d_{ij} \neq -\infty\right\}$ with weights equal to d_{ij}. Furthermore, let G^{prec} be a directed graph with nodes $V(G^{prec}) = V(G^{temp})$ and $E\,(G^{prec}) = \{(i, j) \in E\,(G^{temp}) : i \in Pred\,(j)\}$. Then the estimated LB is equal to the shortest path length between nodes 0 and $n + 1$ in G^{prec} computed by Dijkstra's algorithm (Korte and Vygen 2000).

In the original IRS algorithm, the priority of an activity is equal to its longest path length to the terminal activity $n + 1$. Due to $\mathcal{NP}$-hardness of the longest path lengths computation in our case, we use only the estimation retrieved from G^{temp}, i.e. negative time-lags are omitted. Moreover, we have to distinguish priorities according to alternative process plans. Therefore, the priority of an activity increases with its estimated distance to the end of the schedule and decreases with the length of the alternative branch in which the activity is included. To compute priorities, we first set $aprior_i = c_1 \cdot d^G_{i,n+1} - c_2 \cdot d^G_{open,close}$ for each activity i where $d^G_{i,j}$ is the longest path length between nodes i and j in G^{temp}, the *open* and *close* are activities that start and terminate the minimal alternative branch containing activity i and c_1 and c_2 are constants. Minimal alternative branch for activity i is the alternative branch (Sect. 53.3.1) containing activity i such that there is no other alternative branch containing activity i with the lower number of activities. In the example in Fig. 53.1, the *open* and *close* for activity 5 are activities 3 and 6 respectively. For activity 2, the *open* and *close* are activities 1 and 8. Based on the algorithm testing on various instances, the best performance is achieved when the longest path length to the end of the schedule is given higher influence on the priority value (we use $c_1/c_2 = 5/3$). Finally, the priority $priority_i$ of each activity i is set to a value equal to the position of its $aprior_i$ value in the ascending order of all *aprior* values. In other words, activity with the lowest *aprior* value will have priority equal to 1, next activity will have priority equal to 2 and the activity with the highest *aprior* value will have priority equal to $n + 2$.

53.5.2 Main Loop

In each iteration of the main loop, the function *findSchedule* tries to find the schedule with the given upper bound while the number of steps is limited by the parameter *budget* that is usually set as a number of activities multiplied by the parameter *budgetRatio*. If a feasible schedule is found, all activities are shifted to the left by the label-correcting algorithm (Brucker and Knust 2006) so that the constraints and the order of activities in S are kept. A new upper bound of the schedule length is computed as $UB = UB - 1$ and the next iteration of the loop is performed. If a feasible schedule S is not found for the given schedule length, the algorithm modifies the priority according to the returned partial schedule.

A general observation for heuristic algorithms is that more incorrect decisions are made at the beginning and, therefore, the priority of the earliest scheduled activities and activities that have been added to the schedule more often is decreased. The function *findSchedule* is then called for the same upper bound *UB* using the modified priorities. If the schedule was not found and the maximum number of priority modification steps determined by the parameter *maxModifications* is exhausted, the algorithm returns the best schedule.

Algorithm 53.1 IRSA(budgetRatio, maxModifications, instance)

```
compute LB and UB;
set initial priorities;
budget := budgetRatio · n;
actualRestarts := 0;
while UB ≥ LB
  S := findSchedule (UB, priority, budget) ;
  if S is feasible
    S := shiftLeft (S) ;
    UB := S_{n+1} − 1;
  else
    if actualModifications < maxModifications
      priority := modifyPriority (priority, S);
      actualRestarts := actualRestarts + 1;
    else
      break;
    end
  end
end
```

53.5.3 Inner Loop

In the inner loop of the IRSA algorithm, priorities are updated in the function *updatePriority* (see Algorithm 53.2) such that the priority is increased for the activities marked as selected and proportionally decreased to the number of inclusions of the activity into the schedule. This update of priorities allows the heuristic to switch between alternative branches instead of staying in the same selection for the whole run of the algorithm. For each activity i, the priority is updated such that $priority_i = priority_i + 0.5 \cdot v_i - 0.5 \cdot nAdds_i$ where $nAdds_i$ denotes the number of inclusions of activity i to the schedule. Activity l with the highest priority is found among the set of not yet scheduled activities and a time window $\langle S_l^{LB}, S_l^{UB} \rangle$ where activity l can be scheduled is computed. The lower bound for start time S_l^{LB} is calculated as the minimum time such that all temporal constraints $S_j + d_{jl} \leq S_l$ for all $j \in scheduled : d_{jl} \geq 0$ are satisfied, where *scheduled* is a set of activities that forms the current partial schedule. The start time upper bound S_l^{UB} is set to the maximal value such that the activity is completed before the given UB.

The function *findSlot* tries to find the earliest time slot within the given time window with respect to the resource constraints. In other words, the time interval given by S_l^{LB} and S_l^{UB} is explored while searching for a time point where the given activity can be scheduled without violating any resource constraint. Sequence dependent setup times are also considered. If no feasible time position is found, then the time slot is set to S_l^{LB} if the activity is being added to the schedule for the first time. If the activity has been already included into the schedule in previous

Algorithm 53.2 Inner loop of IRSA

```
findSchedule (UB, priority, budget)
  scheduled := {};
  nAdds_i := 0 (i ∈ V);
  S_i := 0 (i ∈ V);
  v_i := 0 (i ∈ V);
  while budget ≥ 0
    priority := updatePriority (priority, nAdds, v);
    l := max_{j∈V : j∉scheduled ∧ j∉rejected} (priority_j);
    S_l^LB := max_{j∈scheduled} (S_j + d_jk);
    S_l^UB := UB − p_l;
    [conflicts, S_l] := findSlot (l, scheduled, S_l^LB, S_l^UB);
    nAdds_l := nAdds_l + 1;
    [s, scheduled] := insertActivity (l, S_l, conflicts);
    v := findSelected (v, scheduled);
    if schedule is complete
      return S;
    end
    budget := budget − 1;
  end
  return S;
end
```

step, its time slot is set to $S_l^{LB} + 1$ to avoid cycling of the algorithm. The function *findSlot* then returns all conflicting activities, i.e. activities that cannot be kept in the schedule without violating any resource or temporal constraint with respect to the last included activity.

Activity l is then inserted into the partial schedule and all activities marked as conflicting are removed in order to keep the partial schedule feasible at any time. If an unscheduled activity (i.e. activity actually removed from the schedule) is a member of some alternative branch, then all activities in the same alternative branch are also removed. The list of the Selected/rejected activities is then updated; the scheduled activities are marked as selected, activities belonging to the same alternative branch are also marked as selected activities and all activities that cannot be added to the schedule without violating propagation rules from the MILP model are marked as rejected activities. The selection/rejection of other activities is not decided yet. If each activity is already scheduled or marked as rejected, then the schedule S is complete.

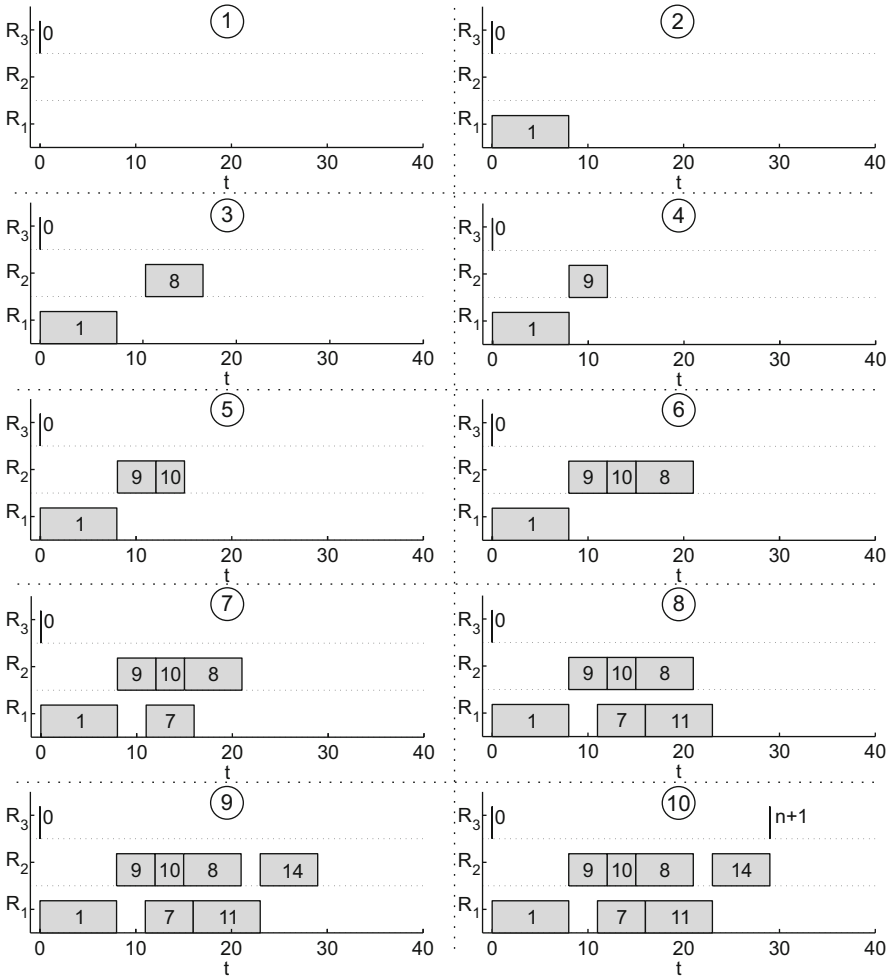


Fig. 53.2 Example of the IRSA algorithm progress

53.5.4 Example of the IRSA Algorithm Progress

Figure 53.2 illustrates one iteration of the IRSA main loop for the instance depicted in Fig. 53.1, considering three resources with capacity equal to one. In the initialization, the algorithm sets *priority* = (16 15 10 9 7 8 6 11 14 13 12 5 3 2 4 1) and consequently it starts with the addition of activity 0 into the schedule. Then activity 1 is added to the schedule and its start time is set to its lower bound, i.e. $S_1 = 0$ (step 1 in Fig. 53.2). Then activity 8 is scheduled and the next not yet scheduled activity with the highest priority is 9, which has to be scheduled to the same resource as activity 8. Its time window is given as $S_9^{LB} = 8$ and $S_9^{UB} = 16$,

resulting from $d_{09} = 8$ and $d_{90} = -16$. Within the given time window, there is no space to schedule activity 9 without violation of resource constraints and therefore activity 8 is marked as conflicting in function *findSlot* and then removed from the schedule in function *insertActivity* (step three). Activity 9 is scheduled instead and its start time is set to 8. In the following step, activity 10 is added and then activity 8 is added back to the schedule. Then the algorithm adds the activities one by one up to the last activity $n + 1$ and the schedule is complete, since each activity is marked as scheduled or rejected.

53.6 Computational Results

In this section, the performance evaluation for the MILP model proposed in Sect. 53.4 and the heuristic algorithm IRSA is shown. Up to our knowledge, there are no standard benchmarks for the $PS|temp, s_{ij}, nestedAlt|C_{max}$ problem, hence randomly generated instances have been used to test both the MILP model and the IRSA algorithm. The heuristic algorithm is further evaluated on instances of integrated process planning and scheduling (IPPS) problem from Shao et al. (2009), which is a specific subproblem of the problem considered in this chapter. Finally, the instances of the job shop scheduling problem with processing alternatives from Kis (2003) are used to test slightly modified version of the IRSA algorithm. Experiments were performed on a PC with 2x Intel Core 2 Quad CPU at 2.83 GHz with 8 GB of RAM. To solve the MILP problems, the ILOG CPLEX 11.2 was used and the IRSA algorithm was implemented in C# language.

53.6.1 Mathematical Model Complexity

Both the MILP model and the IRSA algorithm were tested on 500 randomly generated feasible instances for each number of activities and resources. Each randomly generated instance contains an NTNA instance, where a ratio of alternative and parallel branchings can be specified. The maximum value of the activity processing time, the number and lengths of the positive and negative time-lags and the number of resources can also be set for each generated instance.

The mixed integer linear programming model was tested with the processing time chosen from a uniform distribution on the interval $\langle 1, p_{max} \rangle$ where $p_{max} = 15$, the number of time-lags was set to $|E\ (G^{prec})\ | + n/3$ for the positive values and $n/5$ for the negative values, where $|E\ (G^{prec})\ |$ is the number of edges in graph G^{prec} defined in Sect. 53.5. Figure 53.3 shows the number of solved instances in dependence on the solving time. 500 random instances have been generated for each number of activities n. We measured how many instances the MILP solver was able to solve within a given amount of time. The larger instances can be solved if the number of resources is higher, which can be seen e.g. for the instances with 10 activities

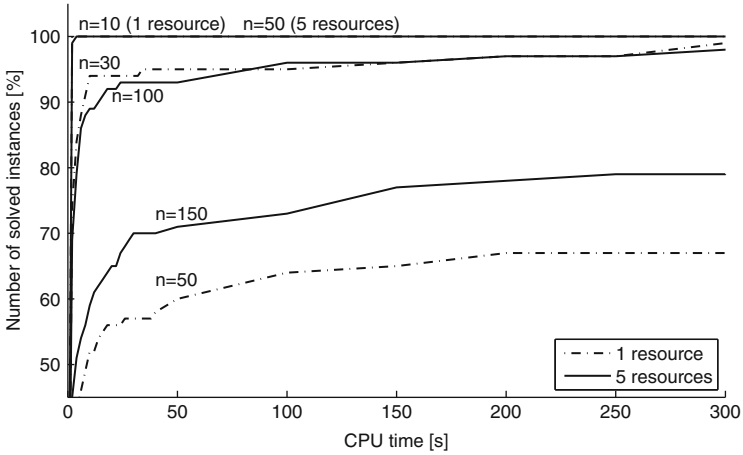


Fig. 53.3 Ratio of solved instances for MILP model

and one resource when compared to instances with 50 activities and five resources. The effectiveness of the MILP model is the same in both cases even though the difference number of activities per instance is considerable. This is obvious due to the smaller amount of resource constraints to be resolved.

53.6.2 Performance Evaluation of IRSA Algorithm

The IRSA algorithm was evaluated using the same set of instances as for the MILP solver. The parameters of the algorithm were set to *budgetRatio* = 6 and *maxModifications* = 2. Figure 53.4a shows the mean difference of the IRSA algorithm from the optimal value given by the MILP solver for feasible instances. Problems with the different number of resources and different maximum processing times are considered. Figure 53.4b demonstrates the number of feasible instances (out of 500) that IRSA was not able to solve to feasibility. Instances with shorter processing times of activities lead to results with a bigger difference from the optimal value. On the other hand, the number of instances for which the IRSA was not able to find a solution is slightly lower in the case with shorter processing times.

The influence of the *budgetRatio* parameter on the results obtained by the IRSA is illustrated in Fig. 53.5a where the mean difference from the optimum and the solution time are depicted in dependence on the given budget for the algorithm. Data were measured on 500 instances with 20 activities and $p_{max} = 15$. Figure 53.5b shows the influence of the *maxModifications* parameter using the same data sets as in the previous case.

The mean solving time for the IRSA algorithm with regard to the number of activities is shown in Table 53.1. For each number of activities, 20 feasible instances

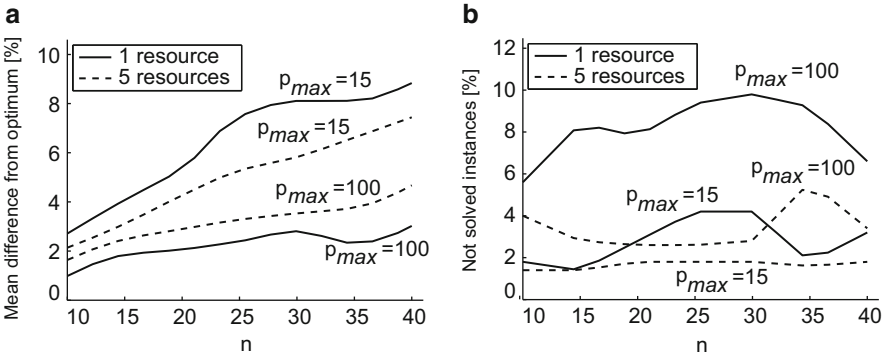


Fig. 53.4 Performance evaluation of IRSA algorithm compared to results of MILP model

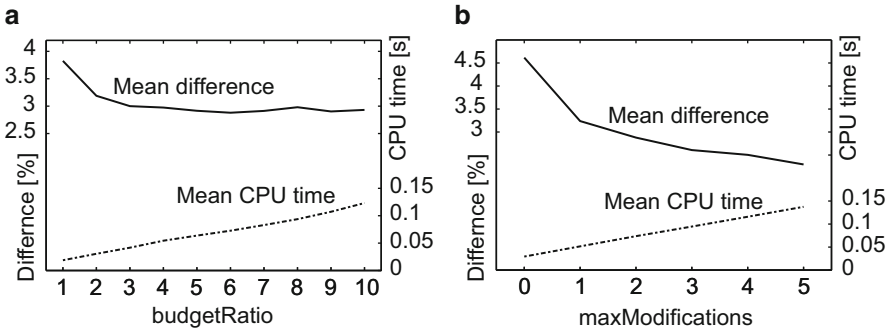


Fig. 53.5 Influence of IRSA settings for the results quality

Table 53.1 Solving time for IRSA algorithm

n	10	50	100	250	500	1,000	2,000
$t_{cpu}^{\varnothing}$ [s]	0.01	0.06	0.15	0.41	1.07	2.55	5.72

with $p_{max} = 15$ were generated. The number of positive and negative time-lags is the same as in the MILP model evaluation. The parameters of the algorithm were the same as in previous paragraphs, i.e. *budgetRatio* = 6 and *maxModifications* = 2.

53.6.3 Integrated Process Planning and Scheduling

The Integrated Process Planning and Scheduling (IPPS) problem studied in Shao et al. (2009) is used to prove the effectiveness of our algorithms for the scheduling problems containing alternatives. IPPS is again a subproblem of the problem considered in this paper. The goal is to select and schedule a subset of all activities based on the precedence graph containing alternative routes and alternative machine

Table 53.2 Comparison of IRSA algorithm with Shao et al. (2009)

Instance	1	2	3	4	5	6	7
Shao et al. (2009)	116	116	95	93	116	116	162
IRSA	117	119	98	93	119	117	171

assignment such that the makespan is minimized. In Shao et al. (2009) there are six small instances (1–6) of IPPS and one bigger instance (7) obtained by joining all small instances into one graph. The comparison of the reported objective values and the values obtained by the IRSA algorithm for all seven instances is depicted in Table 53.2. It should be appointed out that the objective value for the first instance indicated in Shao et al. (2009) is not possible, since the optimal value is 117 instead of 116. The average solution time reported in Shao et al. (2009) is 1 s for small instances, while for the bigger one there is no solution time at all. The algorithm was coded in C++ language and run on a machine with 2.40 GHz Pentium IV. The average running times for the IRSA algorithms is 12 ms for small instances and 2 s for the bigger one.

As can be seen from Table 53.2, the IRSA algorithm is competitive with the evolutionary algorithms proposed in Shao et al. (2009). Therefore we can conclude that the solution methodology is eligible to solve the problems with alternative process plans.

53.6.4 Evaluation on AJSP Instances

Finally, we have evaluated the IRSA algorithm on the instances of the *job-shop scheduling problem with processing alternatives* (AJSP) proposed by Kis (2003). We have decided to solve such instances since our problem is the generalized version of the AJSP problem. The results are depicted in Table 53.3 where columns GA, TABU and RAND contain the results found by algorithms proposed by Kis (2003) and column IRSA contains the results found by the IRSA algorithm, $\Delta_{LB}^{\varnothing}$ is the ratio of the schedule length found by the given algorithm over the lower bound estimated by the MILP solver and $t_{cpu}^{\varnothing}$ is the average computational time in seconds.

As we can see, the results found by the algorithms proposed by Kis (especially TABU algorithm) are superior than the results found by the IRSA algorithm. On the other hand, the increase in the computational time in dependence on the number of activities is more crucial for algorithms proposed by Kis. The total computational time for each instance is also much lower in case of the IRSA algorithm, although the comparison is not straightforward since Kis (2003) reported that C++ language was used and the tests were performed on a machine with Pentium II 400 MHz.

The problem assumed in this chapter is more general than the problem described by Kis. The main difference is that positive time-lags are restricted to be equal to processing times of activities and there are no negative time-lags at all in the AJSP instances. We also assume more general definition of alternative process plans where the alternative and parallel branchings can be arbitrary nested one in another.

Table 53.3 Comparison of IRSA algorithm with Kis (2003)

	GA		TABU		RND		IRSA	
Instances	$\Delta_{LB}^{\varnothing}$	$t_{cpu}^{\varnothing}$ [s]	$\Delta_{LB}^{\varnothing}$	$t_{cpu}^{\varnothing}$ [s]	$\Delta_{LB}^{\varnothing}$	$t_{cpu}^{\varnothing}$ [s]	$\Delta_{LB}^{\varnothing}$	$t_{cpu}^{\varnothing}$ [s]
a01–a03	1.025	3.812	1.021	2.331	1.023	3.308	1.062	0.07
a04–a06	1.042	17.04	1.011	11.38	1.024	16.78	1.096	0.16
a07–a09	1.042	40.52	1.012	30.76	1.095	42.98	1.077	0.39
a10–a12	1.042	78.67	1.005	67.68	1.093	87.04	1.137	0.62
a13–a15	1.020	27.55	1.014	71.29	1.098	29.92	1.251	0.14
a16–a18	1.051	67.57	1.012	49.27	1.135	77.41	1.263	0.27
a19–a21	1.068	124.8	1.015	97.14	1.149	153.1	1.235	0.67
a22–a24	1.072	60.31	1.042	43.23	1.136	72.02	1.299	0.31
a25–a27	1.123	147.8	1.058	131.4	1.203	191.6	1.364	1.11
a28–a30	1.145	274.3	1.025	274.1	1.212	386.3	1.259	1.02
a31–a33	1.152	100.5	1.083	82.76	1.249	130.8	1.341	0.85
a34–a36	1.157	243.6	1.060	253.8	1.261	347.7	1.381	1.93
a37–a39	1.151	457.7	1.036	327.6	1.232	709.5	1.258	2.42

Furthermore, we do not focus on the particular situation where activities are joined in jobs with the specific precedence relations and resource assignment. Finally, there are no sequence dependent setup times in the AJSP problem.

To solve the AJSP instances, we have slightly modified function *findSlot* in the IRSA algorithm. Each job in the AJSP problem is a sequence of activities where at most one activity can be in process at each time but the order of activities in and-subgraphs is not specified. Therefore, the function *findSlot* has to check one more constraint during the search for the feasible time position, i.e. the feasible time position of an activity has to satisfy three types of constraints—temporal constraints, resource constraints and job constraints.

53.7 Conclusions

This chapter presented the resource constrained project scheduling problem with alternative process plans $PS|temp, s_{ij}, nestedAlt|C_{max}$, motivated by the production of the wire harnesses in Styl Plzeň. We have decided to represent the structure of the problem by Nested Temporal Networks with Alternatives and for such representation, the mathematical model able to solve, in a reasonable amount of time, instances with up to 50 activities per resource time is presented. In order to solve larger problems in the nested form, we have developed the heuristic algorithm IRSA. Computational experiments demonstrate good performance of this algorithm with a mean difference from the optimal value of the makespan less than 10 %, while solving time for instances with 100 activities within 20 ms. Instances with up to 2,000 activities can be solved in the order of a few seconds. Moreover, the instances of two related problems have been used for the algorithm evaluation and

the experiments showed that the IRSA algorithm is able to solve much more specific problems with good quality of solutions in very short time.

Acknowledgements This research was supported by the Ministry of Education of the Czech Republic under the Projects OE09004, ME10039, GACR P103/12/1994 and Research Program MSM6840770038. The results of the research have been originally published as the article in European Journal of Operational Research (Čapek et al. 2012b).

References

Barták R, Čepek O (2008) Nested temporal networks with alternatives: recognition and tractability. In: Proceedings of the 2008 symposium on applied computing (SAC). ACM, Ceara, Brazil, pp 156–157

Beck JC, Fox MS (2000) Constraint-directed techniques for scheduling alternative activities. Artif Intell 121(1):211–250

Błażewicz J, Ecker KH, Pesch E, Schmidt G, Węglarz J (1996) Scheduling computer and manufacturing processes. Springer, New York

Brucker P (2007) Scheduling algorithms. Springer, New York

Brucker P, Knust S (2006) Complex scheduling. Springer, New York

Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999a) Resource-constrained project scheduling: notation, classification, models, and methods. Eur J Oper Res 112(1):3–41

Brucker P, Hilbig T, Hurnik J (1999b) A branch and bound algorithm for a single-machine scheduling problem with positive and negative time-lags. Discrete Appl Math 94(1–3):77–79

Capacho L, Pastor R, Dolgui A, Guschinskaya O (2009) An evaluation of constructive heuristic methods for solving the alternative subgraphs assembly line balancing problem. J Heuristics 15(2):109–132

Čapek R (2012) Resource constrained project scheduling problem with alternative process plans and total changeover cost minimization. In: Proceedings of the 25th European conference on operational research, Vilnius, pp 183–183

Čapek R, Van Peteghem V, Hanzálek Z (2012a) Minimization of the total production cost for the RCPSP with alternative process plans. In: Proceedings of the 13th international conference on project management and scheduling, Leuven, pp 110–113

Čapek R, Šůcha P, Hanzálek Z (2012b) Production scheduling with alternative process plans. Eur J Oper Res 217(2):300–311

Dorndorf U, Pesch E, Phan-Huy T (2000) A time-oriented branch-and-bound algorithm for resource-constrained project scheduling with generalised precedence constraints. Manag Sci 46(10):1365–1384

Hanzálek Z, Šůcha P (2009) Time symmetry of project scheduling with time windows and take-give resources. In: 4th multidisciplinary international scheduling conference: theory and applications (MISTA). Dublin, Springer, pp 239–253

Kis T (2003) Job-shop scheduling with processing alternatives. Eur J Oper Res 151(2):307–322

Korte B, Vygen J (2000) Combinatorial optimization: theory and algorithms. Algorithms and combinatorics, vol. 21. Springer, Berlin

Lenstra JK, Rinnooy Kan AHG, Brucker P (1977) Complexity of machine scheduling problems. Ann Discrete Math 1:343–362

Rau BR (1994) Iterative modulo scheduling: an algorithm for software pipelining loops. In: Proceedings of the 27th annual international symposium on microarchitecture (MICRO 27), ACM, pp 63–74

Shao X, Li X, Gao L, Zhang C (2009) Integration of process planning and scheduling: a modified genetic algorithm-based approach. Comput Oper Res 36(6):2082–2096

Chapter 54
Scheduling Computational and Transmission Tasks in Computational Grids

Marek Mika and Grzegorz Waligóra

Abstract Computational grid is a computing environment dedicated to execute applications with large computational requirements. These applications are mainly scientific applications developed for using in many scientific areas. One type of such applications are workflow application which consists of a set of precedence constrained computational tasks. The precedence constraints are caused by transmission of data files and/or the control flow between two computational tasks. Three models of the problem of resource allocation and scheduling workflow application in the computational grid are considered. They differ among themselves in assumptions about workflow applications and computer network. For the first model with distributed resources we present how to adopt metaheuristics developed for MRCPSP. For the second model with setup operations we use the idea of schedule-dependent setup operations and also present the adaptation of a metaheuristic. For the third model with transportation network we present how to find a feasible resource allocation.

Keywords Computational tasks • Grid computing • Project scheduling • Resource allocation • Transmission tasks

54.1 Introduction

The term grid has been used for the first time in the context of the distributed computer resources in the 1990s by Ian Foster. It is commonly believed that this name was used because it expresses the analogy between the computing power available by using a computer network, and the electric power available through the power grid. The term grid has been used in this context, because it expresses the efforts making the access to computing power as easy as the access to an electric power grid.

M. Mika (✉) • G. Waligóra
Institute of Computing Science, Poznań University of Technology, Poznań, Poland
e-mail: Marek.Mika@cs.put.poznan.pl; Grzegorz.Waligóra@cs.put.poznan.pl

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_24

According to Foster and Kesselman (1998) a computational grid is defined as an infrastructure consisting of hardware and software which provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities. Taking into account some specific characteristics of modern grids the computational grid can be defined as a large-scale geographically distributed, dynamically reconfigurable, scalable hardware and software infrastructure composed of heterogeneous resources connected by a computer network. It can be viewed as an virtual organization because resources are owned and shared by multiple administrative organizations which are coordinated to provide transparent, reliable, pervasive, efficient, secure, and consistent computing support to a wide class of computer applications.

These applications are mainly scientific applications developed for using in many scientific areas, such as high-energy physics, bioinformatics, astronomy, biology, climatology, oceanography, meteorology, seismology, and others. Many of them are composed of multiple simpler components (tasks) that process large data sets, execute scientific simulations, communicate and interact with each other over the curse of the application in order to share data and pass the control. The tasks are very often precedence-related, and the precedence constraints usually follow from the data and/or control flow between them. The data flow occurs when data files generated by one task are needed to start another task. In other words, output of one task becomes an input for the next task. Such complex applications consisting of various precedence-related transformations (tasks) performed on certain data between which data files have to be transmitted very often are called workflow applications. In general, two types of workflows can be distinguished: data-intensive, where transmitted files are very large and therefore file transfer times are comparable or greater than the times of computational tasks, and compute-intensive, for which file transmission tasks do not occur or transferred files are so small that the transfer times can be neglected. Workflow applications are usually very time-consuming (even if single tasks are short) and input/output data files for tasks can be large. Execution time of a single workflow application usually ranges from several hours to several days, although it may as well be much larger. For efficient execution of both types of workflows, high computing power is required, due to a large amount of computations and data involved. This computational power can be provided by a computational grid, because the tasks of workflow applications have one interesting feature – they can be executed asynchronously.

There are at least several approaches to grid resource allocation. They differ among themselves depending on the grid architecture, purpose of a particular grid, and grid management policies. Depending on the architecture, two types of grids can be distinguished: peer-to-peer and centralized grids. In the peer-to-peer grid all services are equal and communicate using a peer-to-peer model of the network. In the centralized grid a grid resource management system plays a central role and is surrounded by many other grid services structured in a layered architecture. In such grids, there is usually one common, central grid broker or grid resource manager that serves all users and their jobs. Such a situation is considered in this chapter.

The modern grid resource management involves possibly several layers of schedulers. At the highest level are metaschedulers (or grid-level schedulers), which usually have a more general view of the resources but it does not own any resource of the grid where the application will eventually run. At the lowest level is a local resource management system that manages a specific resource or a set of resources, but it does not know too much about other resources of the grid. In this chapter we consider the metascheduler level only.

It is easy to observe the similarity between a workflow and a project. In both cases we have a set of precedence-related tasks (activities) which are to be executed on a given set of resources in one of several execution modes. Thus, it is justified to describe the considered problem in terms of project scheduling and use the best techniques developed for the MRCPSP (see Chap. 21 in the first volume of this handbook) to approach the considered problem of scheduling computational and transmission tasks in a computational grid.

The chapter is organized as follows. In the next section we present a sample workflow application and a sample structure of the grid as well as the notation which is used in the next sections of this chapter. In Sect. 54.3 a model with distributed resources is introduced. Section 54.4 is devoted to the model with schedule-dependent setup times and finally in Sect. 54.5 the most complex and the most elaborated model with transportation network is discussed. For each of the three considered models two sets of assumptions are presented: the first one with assumptions about the computational grid and the second one with assumptions about the workflow applications. Next the feasible resource allocation is defined and some algorithms for the resource allocation and scheduling are presented.

54.2 Example and Notation

All the models presented in this chapter are deterministic ones. So, they are rather an approximations of real grid resource allocation and scheduling problems. The models assume that resources allocated to computational and transmission tasks will be available for these tasks in time windows sufficient to execute them. In order to meet this assumption a high quality services and an effective prediction mechanism are required. Of course, in general, resource reallocation and/or rescheduling is still possible during the execution of the schedule created.

As a scheduling criterion we choose the makespan, which is very interesting from the client's point of view, i.e., the owner of the applications. However, let us stress here that various performance measures may be considered as the scheduling criterion (e.g., cost, reliability, resource levelling, etc.). Furthermore, a multi-objective approach is well justified which can combine two or even more measures. For multi-criteria approaches to problems of scheduling on a grid, see Kurowski et al. (2004, 2006, 2008). Similar problems with advance reservation of resources and multicriteria and hierarchical approach were considered by Kurowski et al. (2010, 2013).

One of the most common examples of workflow applications is large-scale scientific simulation. A typical framework of the simulation may consists of the following steps: generation of data for simulation according to a given set of parameters, simulation, translation of output data into a described format, postprocessing and finally analysis and comparisons of the obtained results. The general structure of such a workflow is presented in Fig. 54.1.

There are many practical examples of workflow applications built according to the above scheme. One of them is a simulation of the Compact Muon Solenoid (CMS). CMS is one of two large general-purpose particle physics detectors built on the Large Hadron Collider (LHC) at CERN. It is designed to see a wide range of particles and phenomena produced in high-energy collisions in the LHC. The data recorded with the rate of 100 MB/s by different layers of detectors are used to build up a picture of events occurring during the collision. After the data have been recorded, they will be passed through various filter stages, which transform and reduce the data into formats that are more easily analyzed by physicists. In order to better understand the response of the detector to different input signals, large-scale Monte Carlo simulations are performed which typically involve several different computational stages (Deelman et al. 2003). These simulations are long-running, parallel, multi-stage processes that are ideally suited for grid computation. Typically, a single workflow application creates approximately 1 GB of data and requires 10–20 CPU/hours depending on the type of simulation. A typical production run may include thousands executions of workflow application.

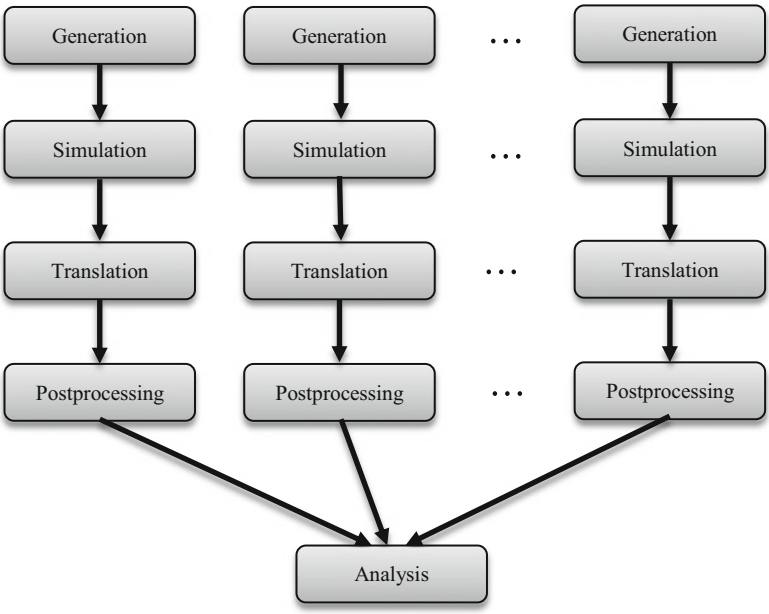


Fig. 54.1 Generic structure of the simulation workflow application

The structure of the workflow application is usually represented by a directed acyclic graph $G = (V, E)$, where set of vertices V represents computational tasks, and set of arcs E represents precedence constraints that follow from control and/or data flow.

Each computational task $i \in V$ is characterized by its size p_i (expressed in assumed computational units, e.g. MIPS) and number of required processors r_i. Let us note that, although there are several types of processors in the grid, it is not necessary to specify the resource requirement for all types of processors, because due to the technological specifications all processor types have the same functionality and differ between themselves only by their speed. So, it is only needed to determine the minimal speed factor ω_i of the processor(s) required for the execution of this task and the function $f_i(p_i, \varpi_k)$ that is used to calculate the actual execution time of computational task i on processor(s) with speed factor ϖ_k. Of course it is necessary to meet the following condition $\varpi_k \geq \omega_i$. We assume for simplicity that $f_i(p_i, \varpi_k) = f(p_i, \varpi_k) = p_i/\varpi_k$.

A transmission task $(i, j) \in E$ that occurs between computational tasks i and j concerns the transfer of the output data file(s) of task i from the computational node where i is executed to another node where task j is executed. The transferred data file(s) is (are) input data file(s) for the computational task j. For each $(i, j) \in E$ the following parameters should be determined: the size F_{ij} of transmitted data file(s), the minimal required bandwidth B_{ij} of the connection between computational nodes in which computational tasks i and j will be executed, and a function $g_{ij}(F_{ij}, B_{ij})$ used to calculate the actual execution time of transmission task (i, j). We assume for simplicity that $g_{ij}(F_{ij}, B_{ij}) = g(F_{ij}, B_{ij}) = F_{ij}/B_{ij}$ if tasks i and j are executed in different nodes, or 0 otherwise (i.e. if in the same node). A very simple example of computational grids presented in Fig. 54.2. This grid consists of six nodes. Three of them denoted by X_1 to X_3 represent computational nodes, i.e. nodes where processors are located. The other three nodes occur in the structure of the grid due to the network topology, because they represent the junction of at least three network links. There are eight processors with the lowest speed factor ϖ_1 in the node X_1, eight processors in X_2 (four with ϖ_2 and four with ϖ_3), and four processors in X_3 (two with ϖ_3 and two with ϖ_4).

The structure of the computational grid is represented by an undirected multi-graph $\Gamma = (\Phi, \Psi)$, where Φ is a set of all nodes in the network, and consists of two disjointed subsets of nodes X and Π. Set X represents computational nodes and is called set of resource nodes, and set Π represents network nodes without processors and is called set of non-resource nodes. Of course, $\Phi = \mathrm{X} \cup \Pi$. Set of edges Ψ contains network links between nodes. Each member of Ψ is a couple $(\mu, \nu)_\psi : \mu, \nu \in \Phi$ where $\psi = 1, 2, \ldots, \Psi^{\mu\nu}$ (ψ denotes alternative links between a given pair of nodes and $\Psi^{\mu\nu}$ is the number of these links). For each resource node $\mathrm{X}_\chi \in \mathrm{X}$ the numbers $R_{k\chi}$ of processors of type k (i.e. processors with the same speed factor ϖ_k) are given, and for each edge $(\mu, \nu)_\psi \in \Psi$ a bandwidth $\Psi_\psi^{\mu\nu}$ of the corresponding network link is given.

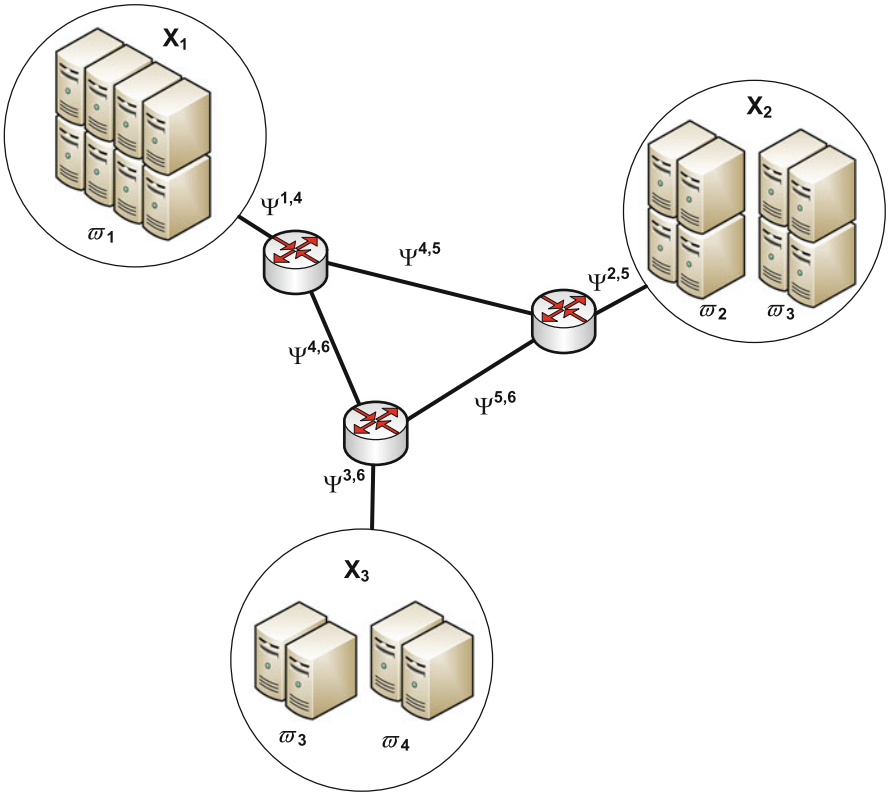


Fig. 54.2 A simple computational grid

54.3 Model with Distributed Resources

The first model of the considered problem is a basic one for two other models described in Sects. 54.4 and 54.5. In this model a workflow application that is executed in the computational grid is the compute-intensive one. We assume that the transmission tasks do not occur during the execution of this application. In practice, it means that if there is any communication between any two consecutive computational tasks, it does not involve the transmission of large data files, but is only used to pass the control or to transfer small data files. In consequence, a computer network does not have a major impact on the schedule, because even if there is a need to transfer some data files between computational tasks, then the transmission time of data files is negligible compared to the execution time of computational tasks. Thus, the only component of the grid which is included in the model are the computational resources. In particular the most important parameters of these resources are: location, type and number of units available in a given location.

54.3.1 Assumptions on the Computational Grid

In the model considered in this section we made the following assumptions about the computational grid:

1. Computational grid is a set of network nodes connected by fast network links
2. There are various resource types in each node for which users' computational tasks may compete (e.g., processors, memory, disk space, I/O devices), but in this model we consider the processors only
3. Processors are divided into types, depending on their power, which is given by a function of some standard unit and is identical for all processors of a given type – we assume a linear form of the processing speed function, as "a speed factor multiplied by the standard unit"
4. For each node a set of resources available in this node is given, which means that the number of units of each resource located in this node is known
5. The structure and the parameters of computer networks are irrelevant to the scheduling problem, due to the limited communication between computational tasks
6. Between the network nodes always exists a connection that allows to pass the control or to transfer small amount of data between the two computational tasks executed in two different network nodes.

54.3.2 Assumptions on the Workflow Application

We also made the following assumptions about workflow applications:

1. Only workflow applications are considered in the model, because of their high practical importance (it is easy to extend it for jobs of other types in the future)
2. For simplicity, there is only one workflow application to be scheduled
3. The structure of a workflow application is represented by a directed acyclic graph G, where each vertex corresponds to a computational task, and each arc represents a precedence relation between two computational tasks
4. Computational tasks are non-preemptable (i.e., once started they have to be completed with no interruptions and without change of the resource allocation)
5. Each computational task is characterized by two values: its size, i.e., the execution time on a standard processor (processors), and the number of processors required for its execution
6. Computational tasks are not scalable, i.e., the number of processors required for the execution of such a task is given a priori by the user and cannot be changed
7. Each computational task is executed by the specified number of processors of the same type (it is not possible to assign a task to processors of different types).

54.3.3 Resource Allocation and Scheduling

It is easy to observe that the considered problem is very similar to the special case of the MRCPSP where only renewable resources are considered, but in the considered model they are distributed over several different locations. Thus, a feasible resource allocation for the workflow application is defined in this model as an allocation of each computational task to a resource node $X_\chi \in X$ that is capable to execute this task, i.e., $R_{k\chi} \geq r_i \wedge \varpi_k \geq \omega_i$. This problem is $\mathcal{NP}$-hard as a generalization of the MRCPSP, and therefore we propose to use metaheuristics to allocate resources and schedule computational tasks. There are many different metaheuristic approaches developed for the MRCPSP, but they cannot be used for the considered problem without any modification. Let us assume that we use the same mechanisms as in simulated annealing proposed by Józefowska et al. (2001). So, the solution is encoded using the most commonly used representation, i.e., the activity list and the mode assignment list. The activity list is a precedence feasible list representing the order (or priorities) of activities. Each element of the mode assignment list represents the execution mode of an activity. If execution modes are assigned to activities according to the mode assignment list, and the priorities of all activities are determined by the order of activities on the activity list, then the schedule is generated using one of the decoding rules, e.g., serial SGS (see Chap. 1 in the first volume of this handbook). In the considered model where computational task correspond to activities such a mechanism is insufficient, because resources are distributed over different locations. So, even if the total capacity of each resource type is known, the more useful information is the one about the capacities of each resource type in each location (resource node), because the computational task may be assigned to exactly one location. So, we need another structure that represents the location to which activity is assigned. Now, when solution is represented by three abovementioned lists a schedule is constructed using, e.g., serial SGS where each pair location-resource type is treated as a separate resource type. With this assumption we do not need to modify this method. Finally, a neighbourhood generation mechanism have to be changed. In order to generate a neighbour solution we may make changes to any of the lists representing the solution. A change on activity list may be implemented in the same way as in other metaheuristic approaches developed for the MRCPSP, e.g., as the shift operation. Changes on the mode assignment list have to take into account the currently assigned location, because not every mode is executable at each location. We need to remove these modes that are not executable in a given location. We have to use similar mechanism for the change on the location list. Before a new location is chosen we need to remove all the locations for which the resource requirements for a corresponding mode assignment are not fulfilled.

54.4 Model with Setup Operations

The second model which was considered by Mika et al. (2004, 2008) is a generalization of the first one described in Sect. 54.3. In this model data-intensive workflow application is considered instead of the compute-intensive one, which means that transmission tasks of very large volumes of data occur between some computational tasks. The size of the data files is so large that the execution time of such transfer, even using high-speed network links, is so large that it cannot be neglected. In addition, this model assumes that neither the structure of the network, nor the quality of the links do not affect the transmission rate. It means, that in practice at least one network link with a guaranteed bandwidth exists between each pair of network nodes. Another assumption is that regardless of the current workload, always exists a connection with required bandwidth between the nodes of the network. In practice, this can be met only when the computer network connecting the nodes with processors is sufficiently extensive (for example, when there is a distinct link between each pair of nodes) and transmission tasks do not appear frequently (for example, when at most one workflow application is performed simultaneously with other types of applications where transmission tasks do not occur, or occur sporadically, and their size is incomparably smaller than the size of transmission tasks of the workflow application). Transmission tasks compete for the resource, which is defined as bandwidth of network connection between selected network nodes, where consecutive computational task are performed. However, it is assumed that this is not a limited resource, taking into account both the structure of the network and the workload of data transmissions occurring in this model. Thus, the structure of the network has no impact on the resulting schedule. However, the data transmission tasks are so time-consuming that they cannot be ignored. Therefore, in the considered model, in addition to computational resources, it is also considered the second aspect of the grid, which are links between pairs of nodes, more precisely the guaranteed bandwidth of these links. In particular, the most interesting in this model are: the location of network nodes with computational resources, the type and the number of computational resources in a given location, as well as bandwidth of network links existing between each pair of such nodes. Due to the assumptions regarding the structure and parameters of computer network, each transmission task (i, j) is always performed immediately after completion of the preceding computational task i, using a direct network connection between two computational nodes where tasks i and j are executed. The execution time of such a task, i.e., the transmission time of a given file depends on the size of the file and the bandwidth of the link. Since each computational task (except the first one) requires as input a file that appears at the output of another directly preceding computational task, the transmission time of this file depends on the location of computational nodes where both precedence related activities are performed. More precisely, it depends on the bandwidth between the two nodes. In this model network resources by definition are not limited. Thus, we do not have to look for the available network resources, we only need to calculate the time of transmission. Next, this transmission

time should be suitably considered during the scheduling. In order to perform the computational task in a given computational node, the node must meet the resource requirements in terms of number of available units of a given types of resources and all the input files must already be there. In other words, the computational node that meets the resource requirements of the computational task will be ready to perform this task only if all the input files will be transmitted to it. Thus, in this case the transmission tasks can be seen as setup operations of computational resources allocated to a given computational task. Such operations are characterized by two parameters: the setup time and setup cost. Considering the makespan as the scheduling criterion the more important is setup time because it may have a direct impact on the schedule length. In this case, setup times depend on the schedule and are called schedule-dependent setup times (Mika et al. 2006).

54.4.1 Assumptions on the Computational Grid

In the model considered in this section we made the following set of assumptions about the computational grid. Assumptions 1–4 are the same as in the previous model presented in Sect. 54.3.1.

5. Bandwidth within the node is unlimited, which means that the computational tasks performed on the same node do not need to wait for the completion of data transmission
6. Network devices and interfaces do not cause any bandwidth limitations, i.e., they are able to handle any incoming or outgoing data transfer—the network bandwidth depends entirely on the bandwidths of links, and not on the characteristics of the network interfaces
7. Breakdowns of physical links are so rare that can be neglected
8. Communication channels are highly reliable, i.e., network links guarantee that, during transmission, packets are not being lost or duplicated
9. Delays do not affect the transmission times between nodes, i.e., transmission lateness is so small in comparison to the transmission time itself that it can be neglected
10. Data transfer does not involve any computational resources, i.e., each processor used to perform a computational task is free immediately after the completion of this task and may be used for the execution of the next computational task, regardless of whether the file transfer of the output data of previous job is still underway or it is already completed
11. Grid structure is represented by the undirected complete graph, which means that between each pair of nodes of a network exist a link characterized by a fixed and guaranteed bandwidth identical in both directions.

54.4.2 Assumptions on the Workflow Application

In this model all assumptions made in Sect. 54.3.2 remain valid. So assumptions 1–7 are the same as in the previous model and there are four new assumptions listed below:

8. A workflow consists of many tasks of two types: computational and transmission ones
9. Transmission tasks are non-preemptable, i.e., they have to be executed with once allocated resources and with no interruptions – once a connection is established between two nodes, the whole transmission must be performed over this connection (a sequence of links)
10. Transmission tasks are not scalable, i.e., the entire transmission is performed using the assigned bandwidth, which may not be changed during the execution of the transmission task
11. Transmission tasks are characterized by a single parameter: the size of the file(s) to be transmitted, which is calculated by dividing the file size by a connection bandwidth or is set to zero if both precedence related computational task are executed in the same node.

54.4.3 Resource Allocation and Scheduling

In this model we additionally consider the transmission times of transmission tasks, but we do not consider the network resources assuming that they are unlimited. Thus, the definition of resource allocation is the same as in Sect. 54.3.3. The transmission times are modeled as schedule-dependent setup times (Mika et al. 2006, 2008), because they depend on the locations where the corresponding computational tasks are executed, and may directly or indirectly depend on some other parameters which depend on the schedule. The representation of the solution and the neighbourhood generation mechanism may be the same as described in Sect. 54.3.3. Only the method used to construct the schedule must be changed. In this case we use a modified version of serial SGS that has been proposed by Mika et al. (2008). Each pair of location and resource type is treated as a separate resource type. Moreover, for each computational task j that requires transmission task (i, j) the transmission time must be taken into account when the computational task j is scheduled. The transmission time of transmission task (i, j) is equal to 0 if both computational tasks i and j are executed in the same resource node or is calculated according to the function g_{ij}.

54.5 Model with Transportation Network

The third model of the considered problem is a generalization of the model presented in Sect. 54.4. In the previous model it is assumed that the transmission tasks can be performed without limitation, i.e., network resources for which transmission tasks compete are not scarce. In the model discussed in this section this assumption is not valid, because the computer network is considered explicitly taking into account its structure and other important parameters. The network links, and more precisely speaking, the bandwidth of these links, are resources for which compete the transmission tasks, which are members of set E that is one of two subsets of tasks composing the workflow application. In the following subsections we present the assumptions concerning the computational grid and workflow application as well as the resource allocation process that is more complicated than those presented in Sects. 54.3.3 and 54.4.3. We do not describe the scheduling phase.

54.5.1 Assumptions on the Computational Grid

As in two previous models we made the set of assumptions on the computational grid. Assumptions 1–10 are the same as in the previous model presented in Sect. 54.4.1, and we extend this set by the following four additional assumptions:

11. There are two types of nodes in the network: resource nodes which contain resources for which computational tasks compete (i.e., processors) and non-resource nodes considered only with respect to the network topology (there may exist various resources in such nodes but they are either not available for some reasons, or not constrained – not scarce, and computational tasks do not have to compete for these resources)
12. Bandwidth between each two connected nodes is given and is identical in both directions
13. Between two given nodes there can be more than one network link, and these links may have different parameters, but these are alternative links and cannot be merged in order to increase the bandwidth
14. The structure of the grid is represented by an undirected multigraph
15. Bandwidth of each link is discretely divisible, i.e., a minimal portion (quant) of bandwidth is assumed.

54.5.2 Assumptions on the Workflow Application

Assumptions 1–10 are the same as in the previous model presented in Sect. 54.4.2 and Assumption 11 is a little bit changed and looks as follows:

11. Transmission tasks are characterized by two values: the size of the data file(s) to be transmitted, and the required bandwidth between the two nodes between which the transmission is to be performed – the user, submitting a workflow to the system, specifies the minimal bandwidth which is required to transmit data between each two precedence-related computational tasks.

We assume that a transmission task gets for its execution a connection with bandwidth equal to the minimal value given by the user. As a result, the transmission time (i.e., the execution time of a transmission task) can take one of two values: the data file size divided by the bandwidth, when successive computational tasks are executed in different resource nodes, or zero, when they are executed in the very same resource node. In our model we assume that the transmission is performed according to the Available Bit Rate (ABR) class of service, as the user specifies a minimum bandwidth needed for his transmission task. However, in this formulation of the model we take this minimal value as the one that the task actually gets for its execution, which makes it closer to the Constant Bit Rate (CBR) class. The assumption on the considered class of service does not have a significant influence during the resource allocation, but it is important in the phase of scheduling.

54.5.3 Resource Allocation

In the previous two models we present the definitions of feasible resource allocation as well as some ideas for using metaheuristics to find a suboptimal schedule. In these models the structure of the network was not taken explicitly into account, and therefore a feasible resource allocation always exists if for each computational task of the workflow application exists at least one resource node that is able to execute this task. The model that was considered by Mika et al. (2011) and is presented in this section is more complicated. The abovementioned condition for computational tasks is insufficient, even if it is fulfilled, because there are also transmission tasks which have their own requirements regarding the bandwidth of the network links. Furthermore, the fulfillment of both the conditions separately may also be insufficient, because it is possible that, even if two consecutive computational tasks are assigned to two different computational nodes that are able to perform these tasks, it may not exist the connection of the required bandwidth between these nodes (even if there are network links in the network with the bandwidth greater than the required one). Thus, the resource allocation for both types of tasks cannot be done separately. In the remaining part of this section we explain how to check if a feasible resource allocation exists, and how to find one or all of feasible resource allocations for a given workflow application submitted to a given computational grid.

Firstly we need to introduce some definitions. Let us stress that we understand resource allocation as assigning computational tasks to resource nodes, and assigning transmission tasks to connections of required bandwidth. We do not assign computational tasks to particular processors in resource nodes, i.e. we do not

perform the scheduling phase, which may be done later when at least one feasible resource allocation will be already known.

Definition 54.1. Resource node $\mathrm{X}_\chi \in \mathrm{X}$ is *capable* for a computational task $i \in V$ if $R_{k\chi} \geq r_i$ and $\varpi_k \geq \omega_i$ ($R_{k\chi}$ is the number of processors with speed factor ϖ_k available in resource node X_χ).

Definition 54.1 determines conditions that have to be met to perform a computational task i in a given resource node. A node is capable for computational task i if it contains at least a given number of processors with a required speed factor or better.
The next two definitions (Definitions 54.2 and 54.3) determine the feasibility conditions for transmission tasks. The first one refers to a single network link, and the second one the chain of network links forming a path between two computational nodes.

Definition 54.2. A link $(\mu, \nu)_\psi \in \Psi$ is called a B_{ij}*-link* if $\Psi_\psi^{\mu\nu} \geq B_{ij}$, otherwise it is called *non-*B_{ij}*-link*.

Let us assume that the computational task i is assigned to resource node $\mathrm{X}_\chi \in \mathrm{X}$, and computational task j is assigned to resource node $\mathrm{X}_\theta \in \mathrm{X}$. Let $P_{ij}(\chi, \theta)$ denote a path in graph $\Gamma(\Phi, \Psi)$ from node X_χ to node X_θ.

Definition 54.3. A B_{ij}*-path* is a path $P_{ij}(\chi, \theta)$ which includes B_{ij}*-links* only.

Notice, that B_{ij}-path is capable of performing a given transmission task (i, j)

In the next definition, we define a feasible resource allocation RA_W for the entire workflow W.

Definition 54.4. Resource allocation RA_W for workflow application W is *feasible* if:

(i) each computational task $i \in V$ is assigned to a capable resource node $X_\chi \in \mathrm{X}$, i.e.:

$$\underset{i \in V}{\forall}\ i \otimes \mathrm{X}_\chi \Leftrightarrow \underset{k}{\exists}(R_{k\chi} \geq r_i \wedge \varpi_k \geq \omega_i) \tag{54.1}$$

(ii) each transmission task $(i, j) \in E$ can be performed over a B_{ij}-path, i.e.:

$$\underset{(i,j)\in E}{\forall} \left(i \otimes \mathrm{X}_\chi \ \wedge\ j \otimes \mathrm{X}_\theta\right) \Rightarrow \underset{P_{ij}(\chi,\ \theta)}{\exists}\ \underset{(\mu,\ \nu)_\psi \in P_{ij}(\chi,\theta)}{\forall} \Psi_\psi^{\mu\nu} \geq B_{ij} \tag{54.2}$$

where $i \otimes X_\chi$ denotes that computational task i is assigned to resource node X_χ.
As it was stated at the beginning of this section, these two conditions are interrelated, but firstly we will discuss them separately.

Checking condition (i) is trivial, since it is sufficient to compare the resource requirements of computational tasks with the computational capability of each resource node. Let $Y_i \subseteq \mathrm{X}$ denote the set of resource nodes capable of executing

computational task i. Each task i then has to be assigned to exactly one node from set Y_i. If for any computational task i, $Y_i = \emptyset$, then there is no feasible resource allocation for the considered workflow.

Checking condition (ii) is more complex. Firstly, we have to remove from graph Γ all non-B_{ij}-links. Then, in the resulting graph $\Gamma_{ij} \subseteq \Gamma$, which may not be a connected graph anymore, all connected components are identified using the Depth-First Search (DFS) method. It is possible, that there will be more than one connected component. We denote the u-th connected component in Γ_{ij} by $\Gamma_{ij}^u \subseteq \Gamma_{ij}$, $\left(u = 1, 2, \ldots, u^\Gamma; u^\Gamma \leq |\Phi|\right)$ and we will call it briefly a subgrid. A transmission task (i, j) can be assigned to any of these subgrids because subgrid Γ_{ij}^u includes B_{ij}-links only. If we denote by Φ_{ij}^u the set of all nodes in subgrid Γ_{ij}^u, then $\mathrm{X}_{ij}^u = \Phi_{ij}^u \cap \mathrm{X}$ is the set of resource nodes in subgrid Γ_{ij}^u. Transmission task (i, j) can be performed in a given subgrid, if and only if both computational tasks i and j can be assigned to their capable nodes from this subgrid. It is possible that it can be the same resource node. Notice that in an extreme case all links from graph Γ may be removed, if they all are non-B_{ij}-links, and each subgrid Γ_{ij}^u consists of one node only (i.e. $u = 1, ..., |\Phi|$). However, it is still possible to execute a transmission task (i, j) by assigning tasks i and j to the same node, if it is capable of executing them both, and the transfer time in this case equals 0.

Definition 54.5. A *tri-task* $\langle i, j \rangle$ is a triple $\{i, (i, j), j\}$, i.e., two consecutive computational tasks and a transmission task between them.

Now we show how to find a feasible resource allocation for a tri-task $\langle i, j \rangle$

Definition 54.6. A *feasible resource allocation* RA_{ij} *for tri-task* $\langle i, j \rangle$ is a pair $\left(X_\chi, X_\theta\right)$ such that $i \otimes X_\chi, j \otimes X_\theta, X_\chi \in Y_i$ and $X_\theta \in Y_j$, and there exists at least one B_{ij}-path $P_{ij}(\chi, \theta)$ between nodes X_χ and X_θ.

In order to find a feasible resource allocation RA_{ij} for tri-task $\langle i, j \rangle$ in subgrid Γ_{ij}^u we need to check if two sets $Z_i^u = Y_i \cap X_{ij}^u$ and $Z_j^u = Y_j \cap X_{ij}^u$ are not empty. If yes it is sufficient to assign task i to any node from the set Z_i^u and task j to any node from the set Z_j^u, otherwise it means that tri-task $\langle i, j \rangle$ cannot be performed in subgrid Γ_{ij}^u. In an extreme situation, if there is no subgrid capable of executing the particular tri-task, it means that there is no feasible resource allocation for the given workflow and, as a result, it cannot be executed on the grid it has been submitted to. So, if there is at least one tri-task in the workflow for which no feasible resource allocation exists, the whole workflow cannot be executed. In all other cases, we are able to define all possible resource allocations for a given tri-task by defining sets Z_i^u and Z_j^u for all subgrids Γ_{ij}^u, $u = 1, 2, \ldots, u^\Gamma$ and enumerating for each subgrid all combinations $\left(\mathrm{X}_\chi, \mathrm{X}_\theta\right)$ such that $\mathrm{X}_\chi \in Z_i^u$ and $\mathrm{X}_\theta \in Z_j^u$. Collecting all those combinations over the whole workflow (all subgrids) gives the set $\mathcal{A}_{ij}$ of feasible resource allocations for tri-task $\langle i, j \rangle$. The following algorithm finds the set $\mathcal{A}_{ij}$:

Algorithm RA-TT (Mika et al. 2011)

1. $\mathcal{A}_{ij} := \emptyset$
2. Find sets Y_i and Y_j. If $Y_i = \emptyset$ or $Y_j = \emptyset$, then no feasible RA_{ij} exists and STOP.
3. Construct graph Γ_{ij} by removing all non-B_{ij}-links from graph Γ.
4. Use the DFS method to find all connected components (subgrids) Γ_{ij}^u, $u = 1, 2, \ldots, u^\Gamma$ of graph Γ_{ij}.
5. For each $u = 1, 2, \ldots, u^\Gamma$ define sets $Z_i^u = Y_i \cap \mathrm{X}_{ij}^u$ and $Z_j^u = Y_j \cap \mathrm{X}_{ij}^u$, where $X_{ij}^u = \Phi_{ij}^u \cap X$. If $Z_i^u = \emptyset$ or $Z_j^u = \emptyset$, then tri-task $\langle i, j \rangle$ cannot be executed in subgrid Γ_{ij}^u
6. For each $u = 1, 2, \ldots, u^\Gamma$ substitute $\mathcal{A}_{ij} := \mathcal{A}_{ij} \cup \left(Z_i^u \times Z_j^u\right)$. If $\mathcal{A}_{ij} = \emptyset$, then no feasible RA_{ij} exists.

Algorithm RA-TT detects two cases when the considered tri-task, and in consequence the entire workflow, cannot be executed on the considered grid:

- if there is at least one computational task for which no capable resource node exists (point 2 of Algorithm RA-TT), or
- if each computational task has at least one capable node, but there is at least one transmission task which cannot be executed because no connection with the required bandwidth between given nodes exists (point 6 of Algorithm RA-TT).

The complexity of the RA-TT algorithm is $\mathcal{O}(\max\{|\mathrm{X}|^3; (|\Phi| + |\Psi|)\})$.

After the execution of Algorithm RA-TT for tri-task $\langle i, j \rangle$ of a workflow application W we obtain set $\mathcal{A}_{ij}$ of all feasible resource allocations RA_{ij} for $\langle i, j \rangle$, i.e., set of pairs $\left(\mathrm{X}_\chi, \mathrm{X}_\theta\right)$ of resource nodes. In order to find a feasible resource allocation for the whole workflow we have to find all feasible allocations RA_{ij} for each tri-task $\langle i, j \rangle$. Thus, we have to run Algorithm RA-TT for each tri-task and as a result we obtain $|E|$ different sets $\mathcal{A}_{ij}$. If any of them is an empty set, then there is no feasible resource allocation for the whole workflow W. Otherwise, we have to check dependencies between tri-tasks, because the condition $\forall \langle i, j \rangle \exists \mathcal{A}_{ij} \neq \emptyset$ is not sufficient for the existence of a feasible resource allocation for the entire workflow. We explain this situation in Example 54.1 where case (1) from Definition 54.7 is discussed.

Definition 54.7. Two tri-tasks $\langle i, j \rangle$ and $\langle i', j' \rangle$ are called *dependent*, if one of the following three cases occurs (Fig. 54.3): 1) $i = i'$, or 2) $j = j'$, or 3) $j = i'$, which is the same as $j' = i$.

Example 54.1. Assume a part of workflow W with three computational tasks i, j, j' and two transmission tasks (i, j) and (i, j'), as shown in Fig. 54.3 (case 1). The required bandwidths for transmission tasks (i, j) and (i, j') are given by B_{ij} and $B_{ij'}$, respectively. Workflow W has been submitted to the computational grid with the simple structure presented in Fig. 54.4. Sets of resource nodes capable for considered computational tasks looks as follows: $Y_i = \{\mathrm{X}_1, \mathrm{X}_3\}$, $Y_j = \{\mathrm{X}_2\}$ and $Y_{j'} = \{\mathrm{X}_4\}$. The bandwidths of networks links are such that $\Psi_{1,2} \geq B_{ij}$ and

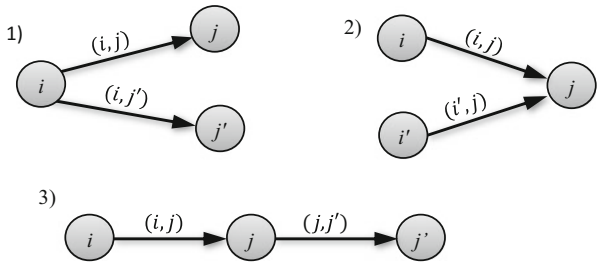


Fig. 54.3 Three cases of dependent tri-tasks

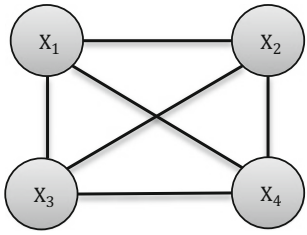


Fig. 54.4 An illustration for Example 54.1—a simple grid structure

$\Psi_{3,4} \geq B_{ij'}$, only. In all other possible pairs of Ψ and B, always $\Psi_{\mu\nu} < B_{ij}$. It means that $RA_{ij} = (\mathrm{X}_1, \mathrm{X}_2)$ is the only feasible resource allocation for tri-task $\langle i, j \rangle$, and $RA_{ij'} = (\mathrm{X}_3, \mathrm{X}_4)$ is the only feasible resource allocation for tri-task $\langle i, j' \rangle$. Thus, after the execution of Algorithm RA-TT for both tri-tasks we obtain the following two sets of feasible resource allocations: $\mathcal{A}_{ij} = \{(\mathrm{X}_1, \mathrm{X}_2)\}$ and $\mathcal{A}_{ij'} = \{(\mathrm{X}_3, \mathrm{X}_4)\}$. Since task i cannot be executed in resource nodes X_1 and X_2 at the same time, there is no feasible resource allocation for workflow W.

Example 54.1 shows that another algorithm is necessary to remove from sets $\mathcal{A}_{ij}$ all resource allocations RA_{ij} which do not guarantee the proper assignment of computational tasks from dependent tri-tasks to resource nodes. Let us first represent a feasible resource allocation RA_W for workflow W with respect to Definition 54.4 as a function $w : V \rightarrow \mathrm{X}$, where $(w(i) = \chi) \Leftrightarrow (i \otimes \mathrm{X}_\chi)$. Then, according to Definition 54.7, for each pair $(\langle i, j \rangle, \langle i, j' \rangle)$ of dependent tri-tasks the following conditions must be hold:

$$i = i' \Rightarrow \underset{\mathrm{X}_\chi}{\exists} \left[(\mathrm{X}_\chi, \mathrm{X}_\theta) \in \mathcal{A}_{ij} \wedge (\mathrm{X}_\chi, \mathrm{X}_\rho) \in \mathcal{A}_{ij'} \right] \tag{54.3}$$

$$j = j' \Rightarrow \underset{\mathrm{X}_\rho}{\exists} \left[(\mathrm{X}_\chi, \mathrm{X}_\rho) \in \mathcal{A}_{ij} \wedge (\mathrm{X}_\theta, \mathrm{X}_\rho) \in \mathcal{A}_{i'j} \right] \tag{54.4}$$

$$j = i' \Rightarrow \underset{\mathrm{X}_\theta}{\exists} \left[(\mathrm{X}_\chi, \mathrm{X}_\theta) \in \mathcal{A}_{ij} \wedge (\mathrm{X}_\theta, \mathrm{X}_\rho) \in \mathcal{A}_{jj'} \right] \tag{54.5}$$

In other words, function w assigns each computational task to a resource node in a way that there is no conflict between dependent tri-tasks.

Let us define the sets $\mathcal{A}_{ij}^{L}$ and $\mathcal{A}_{ij}^{R}$ for each tri-task $\langle i, j \rangle$ by:

$$\mathcal{A}_{ij}^{L} = \left\{ \mathrm{X}_l : \underset{(\mathrm{X}_l, \mathrm{X}_r)}{\exists} (\mathrm{X}_l, \mathrm{X}_r) \in \mathcal{A}_{ij} \right\} \tag{54.6}$$

$$\mathcal{A}_{ij}^{R} = \left\{ \mathrm{X}_r : \underset{(\mathrm{X}_l, \mathrm{X}_r)}{\exists} (\mathrm{X}_l, \mathrm{X}_r) \in \mathcal{A}_{ij} \right\} \tag{54.7}$$

i.e., $\mathcal{A}_{ij}^{L}$ is the set of resource nodes occurring as left elements of pairs RA_{ij} in $\mathcal{A}_{ij}$, and $\mathcal{A}_{ij}^{R}$ as right elements. Then for each computational task j, $j = 1, \ldots, |V|$, we define set $\mathcal{A}_j$ as:

$$\mathcal{A}_j = \prod_j \mathcal{A}_{ij}^{R} \cap \prod_j \mathcal{A}_{jj'}^{L} \tag{54.8}$$

where Π denotes the intersection of multiple sets. A_j is a set of resource nodes where task j can only be executed with respect to both computational and transmission requirements of all tri-tasks in which j occurs. Finally, a new set $\mathcal{A}_{ij}$, denoted $\mathcal{A}_{ij}^{W}$, is constructed thus

$$\mathcal{A}_{ij}^{W} = \mathcal{A}_{ij} \cap \left(\mathcal{A}_i \times \mathcal{A}_j \right) \tag{54.9}$$

Set $\mathcal{A}_{ij}^{W}$ contains all resource allocations RA_{ij}^{W}, which enable to maintain conditions (54.3)–(54.5) over the entire workflow W. Consequently, if none of the sets $\mathcal{A}_{ij}^{W}$ is empty, there must exist at least one feasible resource allocation RA_W for workflow W. Otherwise, i.e., if at least one $\mathcal{A}_{ij}^{W} = \emptyset$, then there is no feasible RA_W.

Now, since each set $\mathcal{A}_{ij}^{W}$ may contain more than one element, the problem of choosing one $RA_{ij} \in RA_{ij}^{W}$ for each tri-task $\langle i, j \rangle$ appears. This is necessary for finding a particular feasible resource allocation RA_W for workflow W. Each function w satisfying the following conditions:

$$\underset{j \in V}{\forall} \; \underset{\theta \in \{1, \ldots, |\mathrm{X}|\}}{\exists} \; w(j) = \theta \tag{54.10}$$

$$\left(\mathrm{X}_\chi, \mathrm{X}_\theta \right) \in \mathcal{A}_{ij}^{W} \Rightarrow w(i) = \chi \tag{54.11}$$

$$\left(\mathrm{X}_\theta, \mathrm{X}_\rho \right) \in \mathcal{A}_{jj'}^{W} \Rightarrow w\left(j'\right) = \rho \tag{54.12}$$

defines a feasible resource allocation RA_W with respect to Definition 54.4.

In other words, each node j, $j = 1, 2, \ldots, |V|$ of the workflow graph representing a computational task must be assigned a number θ indicating a resource node of the grid. Then its incoming arcs (transmission tasks) must be covered by

pairs (resource allocations) with the right element equal to θ, whereas its outgoing arcs must be covered with the left element equal to θ. The number θ does not need to be unique, i.e., more than one node of the workflow may be assigned to the same resource node.

Let us give the following illustration. We can treat pairs $RA_{ij}^{W} = \left(X_{\chi}, X_{\theta}\right)$ from sets $\mathcal{A}_{\mathrm{ij}}^{W}$ as domino bones with χ spots on the left end, and θ spots on the right end (the orientation is important). The problem consists in covering all arcs of workflow W with those domino bones in such a way that arc (i, j) is covered with a bone from set $\mathcal{A}_{\mathrm{ij}}^{W}$ (left end to start of the arc, right end to finish of the arc) and, of course, for each node every attached bone must have the same number of spots on the node-adjacent end.

Summarizing, the following algorithm finds a feasible resource allocation RA_W for workflow W:

Algorithm RA-W (Mika et al. 2011)

1. Execute Algorithm RA-TT for each tri-task $\langle i, j\rangle$ to find the sets $\mathcal{A}_{ij}$. If at least one $\mathcal{A}_{ij} = \emptyset$, then no feasible RA_W exists and STOP.
2. Find for each tri-task $\langle i, j\rangle$ sets $\mathcal{A}_{ij}^{L}$ and $\mathcal{A}_{ij}^{R}$ according to (54.6) and (54.7).
3. Find for each task j, $j = 1, 2, \ldots, |V|$ set $\mathcal{A}_j$ using (54.8). If at least one $\mathcal{A}_j = \emptyset$, then no feasible RA_W exists and STOP.
4. Find for each tri-task $\langle i, j\rangle$ set $\mathcal{A}_{ij}^{W}$ from (54.9).
5. Find a function w satisfying conditions (54.10)–(54.12).

It has been proved by Mika et al. (2011) that Algorithm RA-W always finds a feasible resource allocation for workflow W if it does exist. Moreover, this feasible resource allocation always can be found regardless of which pair of precedence-related computational tasks is allocated first.

The complexity of the Algorithm RA-W is $\mathcal{O}(|V|^2 \cdot \max\{|\mathrm{X}|^4; (|\Phi| + |\Psi|)\})$. It means that finding a feasible resource allocation for a given workflow w submitted to the computational grid $\Gamma = (\Phi, \Psi)$ can be done in polynomial time. Of course, Algorithm RA-W can be also used as a full enumeration scheme to find all possible feasible resource allocations. In such a case step 5 of the algorithm has to be changed in order to find all possible functions w. In consequence the complexity of the algorithm grows to $\mathcal{O}((|\mathrm{X}|^2)^{|V|^2})$.

54.6 Conclusions

In this chapter the problem of scheduling workflow applications in computational grid environments has been considered. Grid resources have been divided into two types: computational resources and network resources. Accordingly, computational tasks of a workflow as well as transmission tasks have been distinguished. The problem consists in allocating grid computational resources to computational tasks,

as well as grid network resources to transmission tasks in such a way that resource demands of all tasks are satisfied and the makespan is minimized. Three models of the problem are presented. In these models processors of different types constitute computational resources, whereas the bandwidth is the network resource for which transmission tasks have to apply. We have shown under which assumptions the defined models of the scheduling and resource allocation problem can be formulated.

For two of the considered models the scheduling algorithms are proposed. For the last one an approach to the problem of finding feasible resource allocations for a given workflow has been presented. The proposed algorithms find a computation- and transmission-feasible allocation of resource nodes of a grid to computational tasks of a workflow in polynomial time. They may also be used, under exponential complexity to find all possible feasible resource allocations.

In future research some algorithms for the scheduling phase of the third model will be developed. Moreover, heuristic algorithms for resource allocation can be proposed and tested along with various strategies for local scheduling. Finally, further extensions of the presented models can be considered.

Acknowledgements This work has been funded by the Polish National Science Centre as a research project in the years 2013–2016 under grant No. 2013/08/A/ST6/00296.

References

Deelman E, Blythe J, Gil Y, Kesselman C, Mehta G, Vahi K, Blackburn K, Lazzarini A, Arbree A, Cavanaugh R, Korranda S (2003) Mapping abstract complex workflows onto grid environments. J Grid Comput 1:25–39

Foster I, Kesselman C (1998) The grid: blueprint for a new computing infrastructure. Morgan Kaufmann Publishers, San Francisco

Józefowska J, Mika M, Różycki R, Waligóra G, Węglarz J (2001) Simulated annealing for multi-mode resource-constrained project scheduling. Ann Oper Res 102:137–155

Kurowski K, Nabrzyski J, Oleksiak A, Węglarz J (2004) Multicriteria aspects of grid resource management. In: Nabrzyski J, Schopf JM, Węglarz J (eds) Grid resource management: state of the art and future trends. Kluwer, Norwell, pp 271–294

Kurowski K, Nabrzyski J, Oleksiak A, Węglarz J (2006) Grid multicriteria job scheduling with resource reservation and prediction mechanisms. In: Józefowska J, Węglarz J (eds) Perspectives in modern project scheduling. Springer, Berlin, pp 345–373

Kurowski K, Nabrzyski J, Oleksiak A, Węglarz J (2008) A multicriteria approach to two-level hierarchy scheduling in grids. J Sched 11:371–379

Kurowski K, Oleksiak A, Węglarz J (2010) Multicriteria, multi-user scheduling in grids with advance reservation. J Sched 13:493–508

Kurowski K, Oleksiak A, Piątek W, Węglarz J (2013) Hierarchical scheduling strategies for parallel tasks and advance reservations in grids. J Sched 16:349–368

Mika M, Waligóra G, Węglarz J (2004) A metaheuristic approach to scheduling workflow jobs on a grid. In: Nabrzyski J, Schopf JM, Węglarz J (eds) Grid resource management: state of the art and future trends. Kluwer, Norwell, pp 295–318

Mika M, Waligóra G, Węglarz J (2006) Modelling setup times in project scheduling. In: Józefowska J, Węglarz J (eds) Perspectives in modern project scheduling. Springer, Berlin, pp 131–163

Mika M, Waligóra G, Węglarz J (2008) Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times. Eur J Oper Res 187:1238–1250

Mika M, Waligóra G, Węglarz J (2011) Modelling and solving grid resource allocation problem with network resources for workflow applications. J Sched 14:291–306

Chapter 55
Make-or-Buy and Supplier Selection Problems in Make-to-Order Supply Chains

Haitao Li

Abstract Time- and cost-effective management of a make-to-order (MTO) supply chain often requires sourcing decisions, such as make-or-buy and supplier selection, to be jointly made with scheduling of supply chain operations. In this chapter, we introduce an RCPSP-based modeling framework to exploit the synergies and interactions between sourcing and scheduling decisions in MTO supply chains under explicit resource constraints. A mixed-integer nonlinear program (MINLP) is formulated to minimize the system wide total supply chain cost. We also present a numerical example to demonstrate the scope and depth of decision-support offered by the model for purchasing and program managers.

Keywords Make-or-buy • Make-to-order • Project scheduling • Resource constraints • Supplier selection • Supply chain configuration

55.1 Introduction

Make-to-order (MTO) is a manufacturing paradigm widely found in defense, airframe, shipbuilding, tooling, commercial transportation, and communication industries (Wisner and Siferd 1995). Each order in MTO manufacturing is often customized and unique, which requires supply chain functionalities ranging from procurement, fabrication to assembly and delivery for fulfilling a customer order. MTO differs from other business models such as assembly-to-order (ATO), build-to-order (BTO) and make-to-stock (MTS) in the so-called customer order decoupling point (CODP, Wouters 1991). MTS typically has the latest CODP toward the order fulfillment end of the supply chain; ATO and BTO often include assembly of standardized components (Gunasekaran and Ngai 2005); while the MTO supply chain considered in this paper also involves sourcing, procurement and fabrication operations.

H. Li (✉)
Department of Logistics and Operations Management, College of Business Administration, University of Missouri, St. Louis, MO 63121, USA
e-mail: lihait@umsl.edu

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_25

Effective management of MTO supply chains requires a well plan and coordination of sourcing, logistics and production operations. Assuming that the engineering design of a product has been completed, purchasing and program managers need to answer questions such as: whether to make the parts/components in-house or procure them from suppliers (the *make-or-buy* decision); if the latter, where to source from (the *supplier selection* decision); and when different operations or processes are competing for limited availability of resources, what is the right schedule (the *project/program scheduling* decision).

These decisions interact with each other and must be simultaneously addressed. For instance, the make-or-buy decisions directly determine the cost and lead time of each individual supply chain *entity*, i.e. a part, component, operation or a process. However, their impact on the system wide performance such as the pipeline stock cost, may also depend on the timing of these operations. Moreover, whether a part can be produced in-house may depend on whether there is sufficient machine capacity and capability. According to Wisner and Siferd (1995), 87.6 % of the MTO manufacturers use process-focused (job shop) layout. Modern manufacturing technologies further diversify the possible settings to flexible job shop (Pinedo 1995) and multi-purpose machines scheduling (Brucker 2001).

Production planning has been well-studied in the MTO research literature. Early work focused on cost estimation and optimization of production rates with learning effect (Walters 1963; Womer 1979). Various resource-constrained project scheduling (RCPSP, Demeulemeester and Herroelen 2002) based models have been proposed to optimize more detailed scheduling and sequencing decisions under limited resources of machine, equipment and manpower. Notably, Demeulemeester and Herroelen (1996) considered setup times, process and transfer batches. Neumann and Schwindt (1997) modeled generalized temporal relationships such as minimal and maximal time lags in a multi-project MTO setting. Kolisch (2000) developed integrated models to simultaneously optimize lot sizing and assembly scheduling decisions. A two-phase approach was proposed by Li and Womer (2007) to optimize an MTO manufacturer's bidding strategies. This line of research emphasizes optimizing a portion of the MTO supply chain, i.e. the production and/or assembly stage, while assuming cost and lead time of supply chain entities are fixed. Therefore, these works do not exploit the interactions and synergies between the procurement and production stages.

Sourcing and supplier selection for MTO supply chains have also been intensively studied. For instance, Ronen and Trietsch (1988) developed a decision-support system to manage purchasing and inventory for multiple large defense programs. Bertrand and Sridharan (2001) proposed heuristic rules for making subcontracting decisions. The model of Murthy et al. (2004) addresses buyer's vendor selection problem in presence of fixed costs, capacity constraints and volume-based discounts. Yue et al. (2009) considered sourcing decisions of MTO supply chains and the tradeoff between cost and reliability of on-time delivery. Multi-objective supplier selection has been studied by Demirtas and Ustun (2008)

and Sawik (2010) among others. Other approaches address supplier disruption due to risk of supply shortage (Li and Zabinsky 2009; Sawik 2011; Wu et al. 2010) and catastrophic events (Berger et al. 2004). These works have focused only on the sourcing and procurement stage of an MTO supply chain. They make the assumption that supply chain entities to be sourced are independent, thus do not consider their demand, time and cost dependencies.

The existing methodologies optimize the MTO sourcing, supplier selection and project scheduling decisions separately. Such decentralized approach may lead to sub-optimal supply chain configuration in terms of system wide performance. Therefore, there is a need of integrated modeling framework for designing an MTO supply chain. The MTO supply chain configuration problem (MTO-SCCP) introduced by Li and Womer (2012) has filled this gap.

We next formally describe the problem setting of MTO-SCCP with a numerical example in Sect. 55.2. Detailed steps to model the key components of MTO-SCCP and a mathematical programming formulation are presented in Sect. 55.3. Section 55.4 discusses solution methods and presents an optimal solution to the numerical example. We summarize conclusions and point out future research opportunities in Sect. 55.5.

55.2 MTO Supply Chain Configuration

This section starts with a description of the MTO-SCCP setting with notations and assumptions made. Then a numerical example of a multi-echelon MTO-SCCP is presented to demonstrate its key components and features.

55.2.1 MTO-SCCP Setting

Consider an MTO manufacturer who just won a set O of new orders. Each order demands Q_o units of end product $o \in O$ before a deadline $\overline{d}_o$.

Assumption 55.1. Engineering design of the products has been completed, so that the bill-of-materials (BOM) of each product o is known.

Let V_o be the set of *entities* involved in the BOM of product $o \in O$. An entity can be a component, a part, or a process/operation, e.g., procurement, fabrication, assembly, transporting, etc. Demand dependency between a pair of entities (i, j) can be described by a constant ρ_{ij}, specifying that one unit of j requires ρ_{ij} units of its predecessor i. Following Kolisch (2000), the BOM structure also defines the precedence relationship among entities. That is, for an entity pair (i, j), it is required that j cannot start before i is finished. Thus the supply chain network described by the BOM can also be viewed as a precedence project network.

Assumption 55.2. The production or procurement of a part/component occurs as an aggregated batch.

Assumption 55.2 means that the execution of entity i fulfills all the units needed for i. It also implies that no preemption is allowed for each entity, i.e. an activity cannot be interrupted once started. Here the manufacturer is more concerned about the tactical level system wide supply chain design, rather than the operational level lot sizing decisions.

Before production starts, the manufacturer must make sourcing decisions for its supply chain entities. *Sourcing* is defined here, in a general sense, as the way to provide or execute a supply chain entity. The company has three sets of entities in consideration: V^{in} representing the set of *in-house only* entities, V^{out} for *procurement only* entities, and V^{io} being the set of *undetermined* entities which can be either outsourced or produced in-house.

For entity $i \in V^{io}$ the manufacturer must make the make-or-buy decision, which may differ in fixed setup cost, direct cost per unit, lead time and resource requirements of i. For instance, in-house production usually incurs significant setup cost, or purchasing/renting cost for new equipment; it also requires certain internal resource such as machines, equipment and skilled workforce. Outsourcing or procurement often incurs less setup cost, and consumes little internal resource.

For each $i \in V^{out}$, several suppliers or vendors may be available, which may differ in fixed ordering cost, direct cost per unit and lead time, reflecting the time-cost tradeoff. For instance, an oversea supplier may supply an item at a lower cost than a domestic supplier, but the transportation lead time may be significantly higher; the next-day delivery is much faster than the ground shipping mode, but is also much more costly.

For $i \in V^{in}$, the manufacturer has the option of *crashing* an entity (a process or operation)by devoting more resources and exploiting the time-resource tradeoff. For instance, a fabrication operation may take one machine and one skilled worker 20 days to complete, but will only take 7 days using two machines and two skilled workers to complete.

Assumption 55.3. Exactly one option (mode) is selected for each entity.

For an entity $i \in V^{out}$, Assumption 55.3 specifies the *single-sourcing* requirement. Incentives for single-sourcing include quantity discounts and maintaining long-term supplier relationship.

The MTO-SCCP simultaneously optimizes the following decisions:

- Mode selection for each supply chain entity
- Scheduling decision to determine the start time of each supply chain entity

The objective function is to minimize the system wide total supply chain cost (TSC) consisting of three terms:

- Total setup cost for all in-house entities
- Total cost of goods sold
- Total pipeline stock cost, or the work-in-process (WIP) inventory cost

The decisions must be made subject to the following constraints:

- Exactly one option is assigned to each entity
- An entity has only one start time (non-preemption)
- The precedence relationships among entities are satisfied
- The order delivery deadline is satisfied
- The resource requirement of all in-house activities cannot exceed the available internal resources

The key feature of MTO-SCCP is that sourcing and scheduling decisions for supply chain entities must be simultaneously made to obtain an optimal, even a feasible solution. For instance, lead times of supply chain entities depend on sourcing decisions, which directly impact the feasibility of a schedule to satisfy the precedence, deadline and resource constraints. In addition, the MTO-SCCP seeks to optimize the global performance of a supply chain, instead of a portion of it. The MTO-SCCP can be compared with other supply chain design and configuration problems (Chandra and Grabis 2007). It addresses the *tactical* level configuration, which differs from the *strategic* level supply chain network design (Daskin 1995), or the *operational* level coordination of supply chain activities (Hall and Potts 2003).

55.2.2 A Numerical Example

We present a numerical example to illustrate the problem setting and data input needed for MTO-SCCP. This example is adapted from the one in the Supplementary Material of Li and Womer (2012).

A manufacturer receives two new orders of 100 Flat Racks and 200 Rear Doors from two different customers. Both orders must be delivered in 120 days. The engineering team quickly comes up with the product design as their bill-of-materials (BOM) shown in Fig. 55.1. The BOM has been simplified by aggregating some parts/components into modules, but this simplification does not eliminate any feature of a typical MTO-SCCP. Note that some parts and components can be shared. This necessitates the need for simultaneously considering the two products. Without loss of generality, a one-to-one relationship is assumed for the demand dependency. That is, one unit of a component requires one unit of its immediate successor down the BOM.

The manufacturer needs to come up with a plan to source some parts from different suppliers. For the in-house entities, a program manager has the option of using more resources to reduce production duration, or less resource to reduce production cost. This reflects the well-known time-cost tradeoff in *project crashing*. For those parts that can be either procured or made in-house, the make-or-buy decision must be made. The delivery to customers is made by third-party logistics providers who have various shipping options such as ground transportation and next-day delivery. The available options for all the entities are provided in Table 55.1.

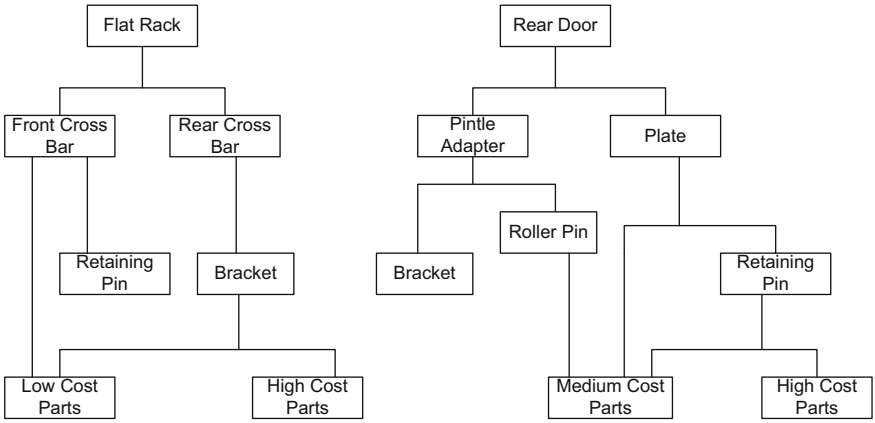


Fig. 55.1 BOM for the two products: flat rack and rear door

Table 55.1 Available sourcing options for all the supply chain entities

	Option 1		Option 2		Option 3	
Outsourced	*p*	*c*	*p*	*c*	*p*	*c*
Low cost parts	40	120	20	125	10	128
High cost parts	30	200	15	208	10	212
Medium cost parts	20	150	10	155	5	158
Delivery of flat rack	15	12	7	22	3	26
Delivery of rear door	15	12	7	22	3	26

	Option 1				Option 2				Option 3			
In-house	*p*	*c*	*r*1	*r*2	*p*	*c*	*r*1	*r*2	*p*	*c*	*r*1	*r*2
Flat rack	20	15	1	1	10	15	2	2	7	15	4	4
Rear door	8	10	1	1	4	10	2	2	3	10	4	4

	In-house					Outsource				
Undetermined entity	*p*	*sc*	*c*	*r*1	*r*2	*p*	*sc*	*c*	*r*1	*r*2
Bracket	30	3,000	20	1	2	25	0	30	0	0
Retaining pin	25	1,500	15	1	2	20	0	20	0	0
Front cross bar	15	500	10	1	2	10	0	15	0	0
Rear cross bar	15	500	10	1	2	10	0	15	0	0
Roller pin	10	1,000	20	1	2	8	0	25	0	0
Plate	10	1,000	15	1	1	8	0	20	0	0
Pintle adapter	10	800	12	1	1	8	0	16	0	0

p lead time (in days), *c* variable cost added (in $), *r*1 units of machines required, *r*2 units of skilled workers required, *sc* fixed setup cost (in $)

For outsourced entities, a purchasing manager must consider the tradeoff between lead time and cost. For instance, one supplier provides the Low Cost Parts at $120/unit in 40 days, while the other supplier can deliver in 20 days (faster)

at $125/unit (more expensive). Regular ground shipping of Flack Rack takes 15 days at $12/unit, but expedited shipping takes 3 days at $26/unit.

Options for in-house entities differ in lead time and resources committed. We assume that the cost of machining and workforce is sunk, so that conserving unused resources does not reduce the cost of the product. For instance, the task of Flat Rack assembly using one machine and one skilled operator takes 20 days, while the lead time can be shortened to 7 days by using four machines and four skilled operators, but the direct cost added remains $15/unit. The company has a total of four machines and four operators available.

Options for the remaining entities differ in setup cost, direct cost added, resources required and lead time, depending upon in-house production or outsourcing. For instance, the in-house subassembly of the Bracket costs $3,000 to start and $20/unit to continue, requiring one machine and two skilled workers; while outsourcing this entity requires no setup cost or resources, but leads to higher direct cost ($30/unit). The outsourcing (25 days) is also a little faster than the in-house production for the Bracket (30 days).

The purchasing and program managers would like to answer the following questions:

- Should bracket, retaining pin, front cross bar, rear cross bar, roller pin, plate and pintle adapter be made in-house or outsourced/subcontracted?
- Which suppliers and vendors should be selected for the low, medium and high cost parts?
- What is the best resource level allocated for the final assembly of flat rack and rear door?
- Which shipping mode should be chosen for delivering the finished flat rack and rear door?
- What is an optimal schedule for the two programs to be delivered on time?

The company has been using a cost accounting based approach, emphasizing cost of goods sold (COGS) minimization, to make sourcing decisions. For each item to be procured, the purchasing department ranks a list of suppliers based on their direct cost, handling and shipping cost, etc. and chooses the one with the lowest total cost. This approach can be viewed as a direct cost minimization (MinCost heuristic) method which minimizes the direct cost of each supply chain activity, thus the COGS. There are exceptions, though, especially when the customer requires a tight delivery deadline. Under such circumstance, each supplier is asked to quote an estimated lead time. Then the purchasing department picks the one with the shortest lead time (MinLT heuristic). For items or activities which can either be outsourced or produced in-house, the make-or-buy decision is made using the well-known break-even analysis for process selection (Jacobs et al. 2009), which compares the fixed setup cost and total variable cost given the demand quantity.

After the sourcing decisions have been made, it is then the program manager's responsibility to plan and monitor supply chain activities to make sure orders are

completed and delivered on time. A number of project management techniques such as the work-breakdown-structure (WBS), critical path method (CPM) and Gantt chart are employed for this purpose. The WBS is defined based on the BOM provided by the operations department. CPM is used to obtain the earliest completion time of contracts and the start-finish time of all the activities involved. Gantt chart is employed to monitor and control the resource utilization of machines and workforce to make sure it does not exceed the available capacities.

Such two-phase sourcing-planning approach works well when the manufacturer has ample resource capacity available to accommodate all in-house production activities. However, when the company's resource capacity is limited, one often struggles to execute the two-phase plan, thus delays and disruptions are not uncommon. The root cause of such deficiency is the lack of a holistic view of the entire supply chain and the available resources when making sourcing decisions. That is, the company may sometimes commit too much in-house production (since it may often appear to be less expensive than outsourcing), for which it will not have enough capacity to support. Other deficiencies of the existing approach include:

- Items in the BOM are treated independently as isolated entities. Their interdependencies in terms of demand and time are not addressed. For instance, if one part becomes more expensive, its successive components and semi-finished goods will also become more expensive, which will lead to higher inventory holding cost; if one part has a longer lead time, its successors may be delayed (depending on whether they are critical activities or not).
- The MinCost or MinLT heuristic focuses on only a single performance metric of the supply chain. Specifically, the MinCost heuristic minimizes the COGS, but not the system wide pipeline stock cost; the MinLT heuristic minimizes the order completion time at the price of higher unit direct cost.
- The CPM itself does not explicitly consider resource constraints when developing the schedule.

55.3 Model Development

This section starts with detailed steps to model the supply chain structure, sourcing decisions and cost components of the MTO-SCCP problem. We then present a mixed-integer nonlinear programming (MINLP) model formulation for the addressed problem.

55.3.1 Modeling Supply Chain Structure

The fundamental idea in our modeling approach is to view an MTO supply chain as a project. While it is a common practice to manage and control MTO activities

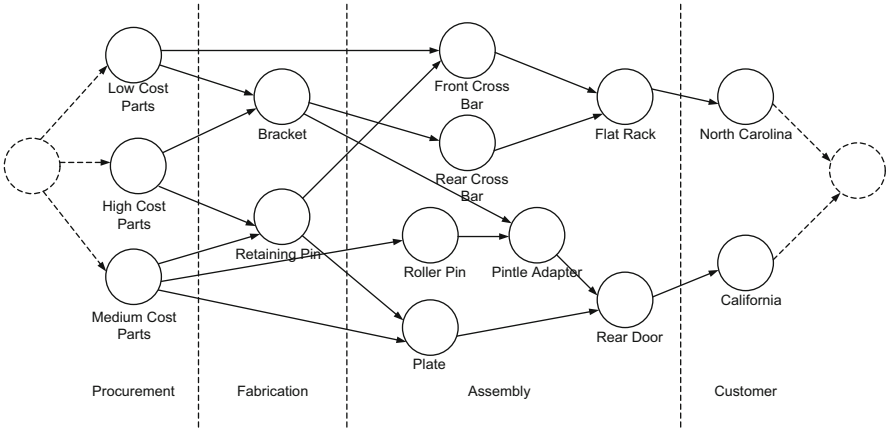


Fig. 55.2 MTO supply chain modeled as a project network

for fulfilling an order as a program/project (Christ 2001), the existing approach only addresses the in-house portion of the chain. Our project network includes all entities in the BOM plus any additional supply chain related activities such as procurement, logistics and delivery.

More formally, the project network $G = (V, E)$ for the set O of orders can be described by the set of all entities with $V = V_1 \cup V_2 \ldots \cup V_{|O|}$, and a set E of arcs. Each arc $(i, j) \in E$ with a weight ρ_{ij} has two meanings: (a) it defines the demand dependency between i and its immediate successor j, i.e. one unit of j requires ρ_{ij} units of i. Thus the demand μ_i of $i \in V$ can be derived by the backward recursion: $\mu_i = \sum_{j:(i,j)\in E} \rho_{ij}\mu_j$, where the demand of the end product $o \in O$ is given as Q_0. (b) Following Kolisch (2001), an arch (i, j) also specifies the precedence relationship between i and j, i.e. j cannot start before i is finished.

Using the above modeling framework, the BOM of the numerical example in Fig. 55.1 can be transformed into the activity-on-node (AON) precedence project network in Fig. 55.2. Here supply chain entities involved in the two orders of Flat Rack and Real Door are represented by project activities (nodes), grouped into stages of procurement, fabrication and assembly in a multi-echelon supply chain. The last layer of nodes can be added to represent the delivery stage. We may further add dummy nodes as start and end of the project. Each arc in the AON network corresponds to the precedence relationship stipulated by the demand dependency in BOM. For example, the subassembly of Pintle Adapter must precede the final assembly of Rear Door. The use of project network offers a convenient platform to model both complex demand- and time-dependencies in an MTO supply chain.

55.3.2 Modeling Sourcing Decisions

We devise a unified framework to model various sourcing decisions including make-or-buy, supplier selection and resource allocation. Let $\mathscr{M}_i$ be a set of sourcing options (modes) available for an entity i, and $\mathscr{R}$ be a set of internal resources such as machines, equipment and skilled workforce. Each mode $m \in \mathscr{M}_i$ is characterized by a tuple: $\mathscr{T}_{im} = \{p_{im}, c_{im}, sc_{im}, r_{ikm}(k \in \mathscr{R})\}$, where p_{im} denotes the lead time of i executed in mode m, c_{im} is the direct cost of i when mode m is chosen for i, sc_{im} is the fixed setup cost of i in mode m, and r_{ikm} represents units of internal resource k required by i in mode m.

For an outsourced/procured entity $i \in V^{out}$, the mode m refers to different available suppliers for i, which may differ in the fixed ordering cost sc_{im}, cost per unit c_{im} and lead time p_{im}. For all $i \in V^{out}$, no internal resource is required, i.e. $r_{ikm} = 0$ for all $k \in \mathscr{R}$.

For an in-house entity $i \in V^{in}$, m represents different resource allocation options with different lead time p_{im}, unit cost c_{im}, and resource requirement r_{ikm} for some resource $k \in \mathscr{R}$. Here we may fix $sc_{im} = 0$ assuming that the company already has the core competence and expertise for entity i, so that there is no significant initial setup cost incurred.

For entity $i \in V^{io}$, m means either in-house production or outsourcing. Here the main tradeoff is between the fixed setup cost sc_{im} (high for in-house and low/zero for procurement) and unit cost c_{im} (low for in-house and high for outsourcing). What differentiate in-house versus outsourcing are also the internal resource requirement r_{ikm} (none if outsourcing and non-zero if in-house) and lead time p_{im}.

55.3.3 Modeling Total Supply Chain Costs

Given the scope of MTO-SCCP, the objective function in the optimization model must address the system wide supply chain performance. We consider the total supply chain costs with three components: cost of goods sold (COGS), total fixed setup cost (TFC) and total pipeline stock cost (TPC).

The TFC is the sum of fixed setup cost sc_{im} of each supply chain entity depending upon the option selected for the entity.

The COGS can be computed based on the cumulative cost of each end product $o \in O$, which is a function of the sourcing decision and demand dependencies in the supply chain network. Consider the example in Fig. 55.3, where entity i has a set of $1, 2, \ldots, m$ available options with different direct cost $c_{i1}, c_{i2}, \ldots, c_{im}$. Suppose the second option is chosen for i (in gray), then the direct cost per unit of i is c_{i2}. To get the cumulative cost per unit of i, one should add up c_{i2} with the cumulative costs of all i's immediately predecessors. If i has only one immediate predecessor j (Fig. 55.3a), i's cumulative cost cc_i equals simply its own direct cost c_{i2} plus j's cumulative cost cc_j; if i has two immediate predecessor j and j' (Fig. 55.3b), cc_i is the sum of c_{i2}, cc_j and $cc_{j'}$.

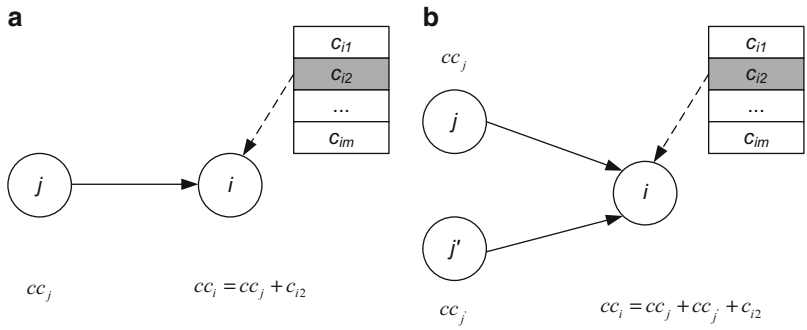


Fig. 55.3 Compute cumulative cost of a supply chain entity. (**a**) Entity i has only one predecessor. (**b**) Entity i has only two predecessors j and j'

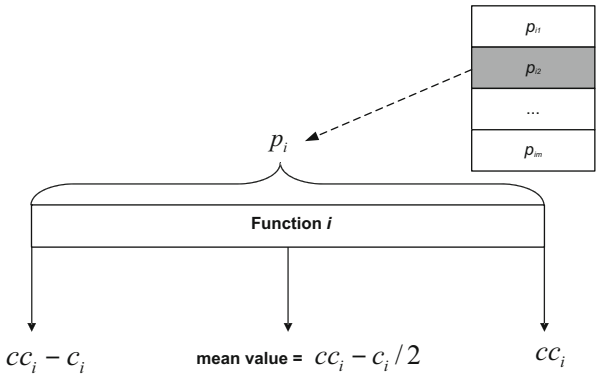


Fig. 55.4 Compute the pipeline stock cost

To see how TPC is computed, consider entity i with lead time p_i dependent upon the option selected for i as shown in Fig. 55.4. Suppose i's direct cost is c_i (resulted from the option selected for i), and its cumulative cost is cc_i (as sum of the cumulative costs of all i's predecessors), then the mean cost of entity i is the average of $cc_i - c_i$ and cc_i. The work-in-process (WIP) or pipeline inventory cost of entity i can be computed as $hc \cdot p_i \mu_i (cc_i - c_i/2)$, where hc is the inventory holding cost per time period. Because both p_i and $(cc_i - c_i/2)$ is a function of the option selection decision, the TPC is clearly nonlinear.

55.3.4 MINLP Formulation

A mixed-integer nonlinear programming (MINLP) formulation is presented next to model the MTO-SCCP.

Define time-indexed binary decision variable x_{imt} for $i \in V$, $m \in \mathscr{M}_i$, $t \in \{1, 2, \ldots, T\}$, where T is the maximum of due dates $\overline{d}_1, \overline{d}_2, \ldots, \overline{d}_{|o|}$ of all the orders. $x_{imt} = 1$ if and only if entity i is assigned with mode m and starts in the t-th time point in the planning horizon. Note that the decision variable x simultaneously makes mode assignment and scheduling decisions.

Several other decision variables are needed for the convenience of formulating the model. Let $p_i \in Z^+$ represent the lead time of entity $i \in V$, $c_i \geq 0$ be the direct cost of i, and $cc_i \geq 0$ denote the cumulative direct cost of i.

The objective function can be written as:

$$\text{Min.} \quad \sum\nolimits_{i \in V} \left[\sum\nolimits_{m \in \mathscr{M}_i} \sum\nolimits_{t=ES_i}^{LS_i} sc_{im} x_{imt} + \mu_i c_i + hc \cdot \mu_i (cc_i - c_i/2) p_i \right] \tag{55.1}$$

It minimizes the total supply chain costs (TSC) consisting of total fixed cost (TFC), cost of goods sold (COGS) and total pipeline stock cost (TPC). ES_i and LS_i represent the earliest and latest starting time of i, respectively. They can be obtained by the classical temporal analysis in project scheduling. The least duration (most expensive) option is chosen for computing ES, and longest duration (least expensive) option is chosen for computing LS.

Since an entity cannot be interrupted once started (non-preemption in Assumption 55.2), it has exactly one start time. An entity is also assigned with exactly one mode due to the single-sourcing requirement (Assumption 55.3). These two requirements can be satisfied by the following constraint (55.2), which assigns exactly one mode and starting time to an entity.

$$\sum\nolimits_{m \in \mathscr{M}_i} \sum\nolimits_{t=ES_i}^{LS_i} x_{imt} = 1 \qquad (i \in V) \tag{55.2}$$

Based on the option selection decision, the lead time of an entity can be computed by (55.3):

$$p_i = \sum\nolimits_{m \in \mathscr{M}_i} \sum\nolimits_{t=ES_i}^{LS_i} p_{im} x_{imt} \qquad (i \in V) \tag{55.3}$$

In a similar way, an entity's direct cost can be computed by (55.4):

$$c_i = \sum\nolimits_{m \in \mathscr{M}_i} \sum\nolimits_{t=ES_i}^{LS_i} c_{im} x_{imt} \qquad (i \in V) \tag{55.4}$$

Next, Constraint (55.5) derives the cumulative direct cost of an entity as the sum of the entity's direct cost and the cumulative direct cost of all its immediately predecessors.

$$cc_i = c_i + \sum\nolimits_{j:(j,i) \in E} cc_j \qquad (i \in V) \tag{55.5}$$

While Constraints (55.2)–(55.5) model the assignment aspect of the MTO-SCCP, the scheduling aspect of MTO-SCCP are taken care of by the following constraints.

The precedence constraints between a pair of entities $(i, j) \in E$ can be enforced by Constraint (55.6), where the left-hand-side of (55.6) is the difference between the starting time of j and the starting time of j's immediate predecessor i.

$$\sum_{m \in \mathscr{M}_j} \sum_{t=ES_j}^{LS_j} tx_{jmt} - \sum_{m \in \mathscr{M}_i} \sum_{t=ES_i}^{LS_i} tx_{imt} \geq p_i \qquad ((i, j) \in E) \tag{55.6}$$

The delivery deadlines of all orders are satisfied by Constraint (55.7), where the left-hand-side of (55.7) is the finishing time of an entity computed as the entity's starting time plus its lead time or duration.

$$\sum_{m \in \mathscr{M}_i} \sum_{t=ES_i}^{LS_i} tx_{imt} + p_i \leq \overline{d}_o \qquad (o \in O; i \in V_o) \tag{55.7}$$

Finally, the resource requirement of all supply chain entities in any period of the planning horizon cannot exceed the company's available internal resource capacities. The inequality (55.8) below models such resource constraint. For each resource type $k \in \mathscr{R}$ and time period t in the planning horizon, Constraint (55.8) ensures that the total requirement of k by all the active entities in period t must not exceed the available capacity R_k. The inner most summation at the left-hand-side identifies those entities being active at t according to their starting time through the time-indexed decision variable x. The outer summation adds up all entities that are possible to be in-house, except those outsourced entities in V^{out}.

$$\sum_{i \in V^{in} \cup V^{io}} \sum_{m \in \mathscr{M}_i} r_{ikm} \sum_{\tau=\max\{t-p_{im}, ES_i\}}^{\min\{t-1, LS_i\}} x_{im\tau} \leq R_k \quad (k \in \mathscr{R}; t \in \{1, \ldots, T\}) \tag{55.8}$$

Given both the assignment and scheduling components in the MINLP formulation (55.1) through (55.8), the MTO-SCCP is an instance of the assignment-type RCPSP (Drexl et al. 1998). The model reduces to a single-mode RCPSP when the mode assignments are fixed. Therefore, the MTO-SCCP is $\mathscr{NP}$-complete, because the single-mode RCPSP is well-known to be $\mathscr{NP}$-complete (Bartusch et al. 1988).

55.4 Solution Methods and Results

The MINLP model for MTO-SCCP can be solved by the general MINLP algorithms (Floudas 1995). For large-size instances, however, such exact methods may not be efficient enough due to the $\mathscr{NP}$-completeness nature of MTO-SCCP. In this section, we first discuss alternative solution approaches, then present an optimal

configuration solution to the numerical example in Sect. 55.2.2. The purpose is to demonstrate various decision-supports provided by the MTO-SCCP model for purchasing and program managers.

55.4.1 Solution Approaches

Li and Womer (2012) have developed a decomposition algorithm to exploit the assignment-scheduling structure of the MINLP formulation. Their algorithm is theoretically grounded in the classical Benders decomposition (Benders 1962) and hybrid Benders decomposition (HBD) framework (cf. Chap. 27 in the first volume of this handbook). It decomposes the MINLP into a relaxed master problem (RMP)containing only the assignment component of model, and a sub-problem (SP) containing only the scheduling component. An optimal solution to the RMP is a partial solution to the original MINLP, and the optimal RMP objective function value provides a lower bound to the MINLP. The feasibility of the optimal RMP solution is checked by solving the resulting scheduling SP (with the RMP assignment solution fixed). If the SP is feasible, an optimal solution has been found; otherwise, the cause of infeasibility is deduced as "cuts" to be added back to the RMP. The algorithm iterates until an optimal solution is found or infeasibility is proved. We refer to Li and Womer (2012) for the detailed algorithm procedure and finite convergence proof.

The advantage of HBD algorithm is its ability to find and prove optimality. However, its performance depends largely on the tightness of order delivery deadlines and effectiveness of handling the RMP. When order deadlines become tight,the number of iterations will increase and the size of RMP will grow, which makes the HBD less effective.

Various metaheuristic algorithms (Glover and Kochenberger 2003) are promising, as they have been successful for variants of RCPSP. The advantage of a metaheuristic is its ability to work for problems with nonlinear objective function as in the MTO-SCCP. This can be done through either a population-based framework such as genetic algorithm (Holland 1975) and scatter search (Glover 1997) etc., or local search based methods including tabu search (Glover and Laguna 1997) and simulated annealing (Kirkpatrick et al. 1983) among others.

55.4.2 Optimal Configuration Results

An optimal solution to the numerical example can be found by the HBD algorithm. Table 55.2 compares quality of the optimal configuration solution with solutions found by the two heuristics, MinCost and MinLT, in Sect. 55.2.2. The MinCost method clearly results in least COGS, but has a higher pipeline stock cost due

Table 55.2 Computational results of different approaches

	MinCost heuristic	MinLT heuristic	Optimization solution
COGS (thousand $)	302.0	330.5	310.3
Total fixed cost (thousand $)	8.3	0	0
Pipeline stock cost (thousand $)	21.0	11.4	17.2
Total supply chain cost (thousand $)	331.3	341.9	327.5
Program makespan (days)	120	56	98

to longer program makespan. The MinLT solution requires the least makespan to complete the orders, but at the price of much higher COGS. The optimal configuration leads to a moderate level of both program makespan and COGS. Insight 1 follows.

INSIGHT 1. An optimal configuration of MTO supply chain exploits the tradeoff between cost and lead time and balances among COGS, pipeline stock cost, fixed cost and program makespan to achieve minimum total supply chain cost.

It is insightful to examine the detailed configurations of the three solutions in Fig. 55.5. The optimal supply chain configuration in Fig. 55.5a suggests low-cost options for all the outsourcing entities, high-cost options for the two in-house entities, and outsourcing to the high-cost (fast) suppliers for the undetermined entities. The MinLT configuration in Fig. 55.5b does not recommend in-house production for the undetermined entities either, and always chooses the fast and high-cost options for all entities. One may also verify that the Flat Rack assembly and Rear Door assembly cannot overlap when they are configured to utilize more resources as in Fig. 55.5a, b. The MinCost configuration in Fig. 55.5c suggests in-house production for undetermined entities, and always chooses the low-cost but slow options for all entities. The optimal configuration leads to a moderate program makespan compared with the shorter makespan of MinLT solution and the longer makespan of MinCost solution.

Figure 55.6 shows the effect of order deadline on total supply chain cost (TSC). As the deadline increases (becomes less tight), the optimal TSC decreases at a diminishing rate. Such relationship may also depend on the available resource capacity. These insights have created opportunities for the company to reduce supply chain costs by negotiating the delivery date with customer. The MTO-SCCP solution has provided an effective way for managers to quantify the monetary value of time, thus reliable decision-support for such negotiation. For example, when the resource capacity is four and the customer's current deadline is 85 days, there is little incentive to extend the delivery date. If the current deadline is 80 days, however, there is a high incentive to negotiate with the customer to extend the deadline to 85 days, which will potentially save the company a total of $1,360 or $272/day.

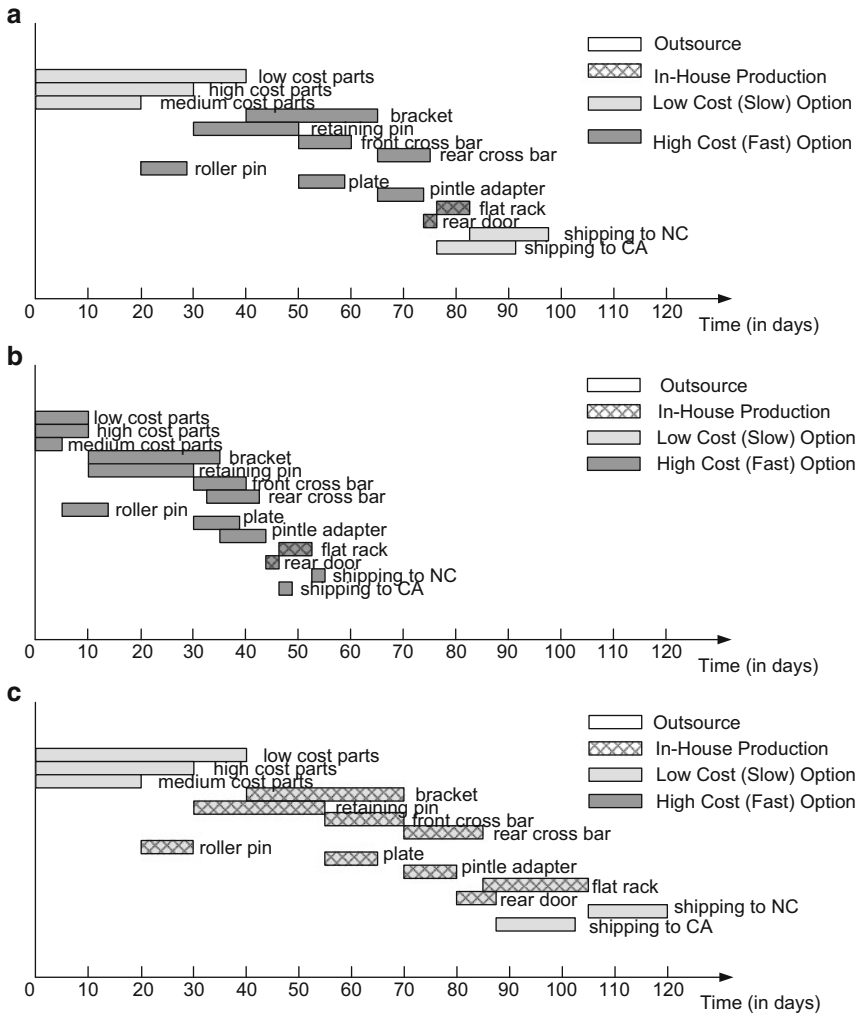


Fig. 55.5 Gantt charts illustrating supply chain configurations of three approaches. (**a**) Activities with optimal configuration. (**b**) Activities with minimum lead time configuration. (**c**) Activities with minimum direct cost configuration

Figure 55.7 further compares the three cost components: total fixed cost, COGS and pipeline stock cost as a percentage of the optimal total supply chain cost when order deadline varies. A large (less tight) deadline makes it feasible and attractive to choose less costly options (but with longer lead time), so that the proportion of COGS in the TSC can be reduced. As a result, however, longer lead times of supply chain entities will potentially increase the proportion of total pipeline stock cost. The observations in Figs. 55.6 and 55.7 can be summarized by Insight 2.

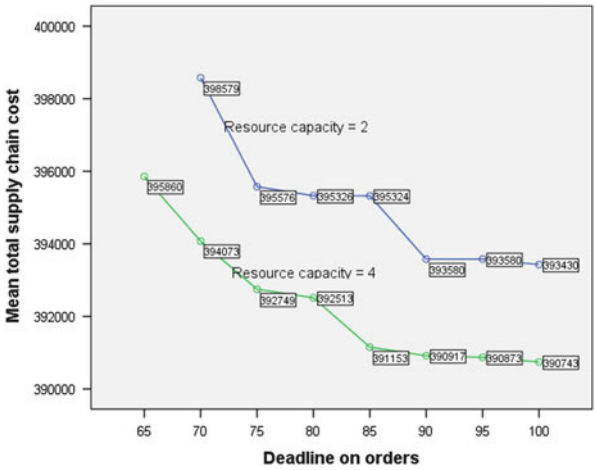


Fig. 55.6 Optimal total supply chain cost when order deadline varies

INSIGHT 2. Increasing order delivery deadline will reduce the optimal total supply chain cost at a diminishing rate. As the deadline increases, the percentage of total pipeline stock cost will likely increase, and the percentage of COGS will likely decrease.

Figure 55.8 depicts the optimal make-or-buy decision of four parts: Bracket, Plate, Front Cross Bar and Pintle Adapter when the demands for the end products (Flat Rack and Rear Door) vary. Both Bracket and Plate show a pattern that can be explained by the break-even analysis: in-house production becomes preferable to outsourcing when demand exceeds the break-even point. Notice that the decision for the Bracket appears to depend on both the demand of Flat Rack and Rear Door, while the decision for Plate seems to have little dependence on the demand of Flat Rack. This can be explained by the BOM structure in Fig. 55.1: it is clear that Bracket is required for both Flat Rack and Rear Door, whereas Plate is a part only required by Rear Door.

The optimal make-or-buy decisions of Front Cross Bar and Pintle Adapter are more interesting. There is no clear break-even point (boundary) for either Front Cross Bar (required solely by Flat Rack) or Pintle Adapter (required solely by Rear Door). The distribution of decisions is fairly even no matter how demand varies. This deviates from the common wisdom as suggested by the break-even analysis, but can be explained by the impact of limited availability of internal resources. Note that the assemblies of Front Cross Bar and Pintle Adapter belong to the same supply chain echelon (Fig. 55.2), thus will compete for their shared machines and skilled operators. With limited resource capacity available, the company will always have to outsource some entities in order to obtain a feasible schedule and ensure on-time order delivery, no matter what the demand quantities are.

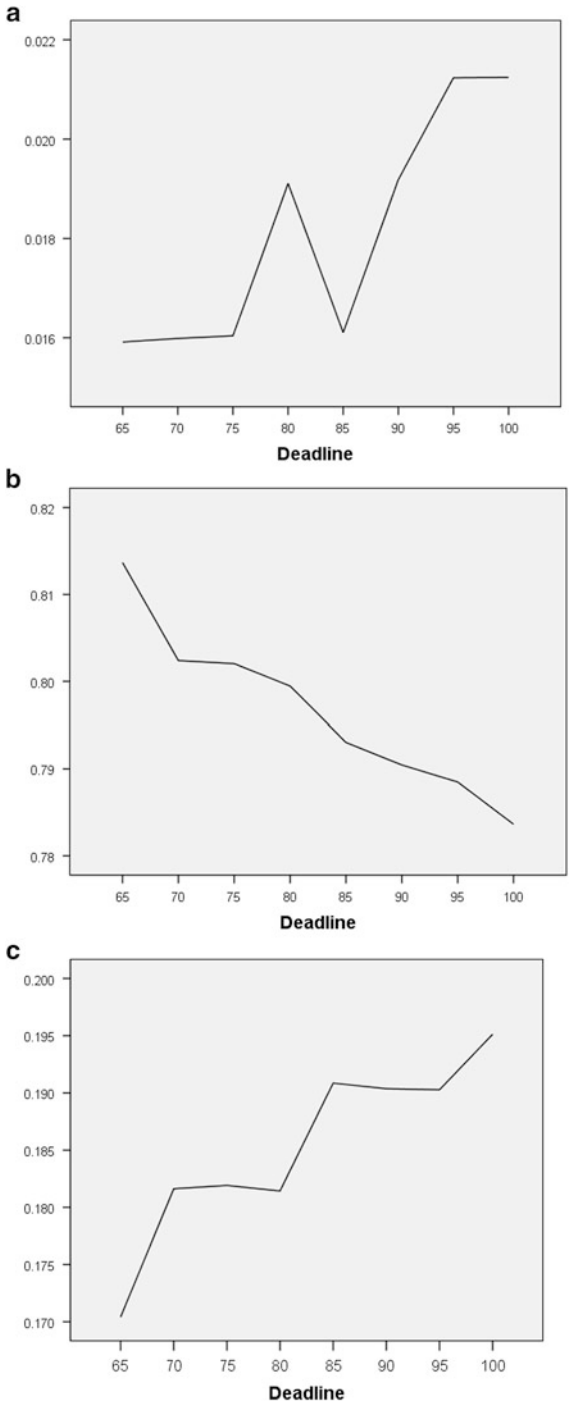


Fig. 55.7 Tradeoffs among different cost components when order due date varies. (**a**) TFC as a percentage of TSC. (**b**) COGS as a percentage of TSC. (**c**) Pipeline stock cost as a percentage of TSC

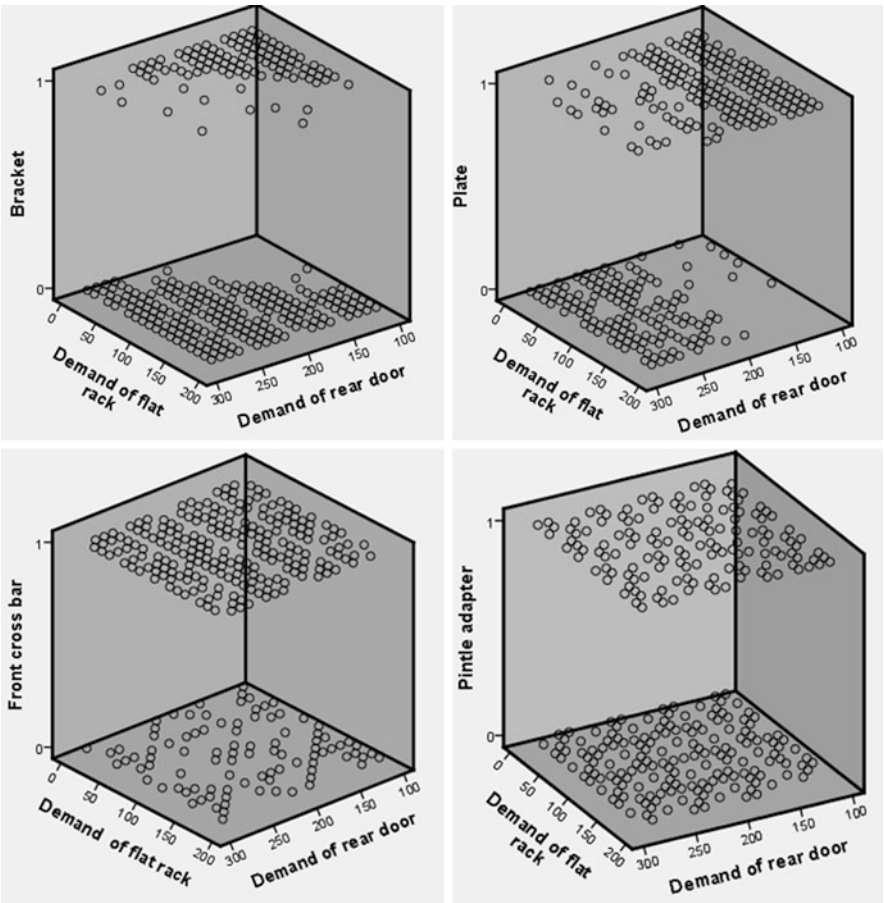


Fig. 55.8 Optimal make-or-buy decisions of four parts when demand varies (0—make, 1—buy)

INSIGHT 3. The optimal make-or-buy decision in an MTO supply chain can be affected by various factors such as demand quantities, BOM structure, order deadline, resource requirement and capacity. Solutions of the naïve break-even analysis may be suboptimal or even infeasible.

55.5 Conclusions

After receiving new orders or contracts, an MTO manufacturer must optimize its make-or-buy and supplier selection decisions for all the relevant supply chain entities: parts, components and operations. These sourcing decisions are impacted by demand quantity, delivery deadline, the products' underlying BOM structure, their

resource requirements and the company's available internal resource capacities. Sourcing also often interacts with scheduling of supply chain entities, especially when there is limited availability of resources for executing operations. Most of the existing research in MTO manufacturing, however, handles sourcing and scheduling decisions separately.

The MTO supply chain configuration problem (MTO-SCCP) considered in this chapter has filled this gap. We model the MTO-SCCP as a variant of RCPSP, i.e. an assignment-type RCPSP, to optimize sourcing decisions including make-or-buy, supplier selection and more general mode selection, in conjunction with the scheduling of operations. The RCPSP-based modeling framework makes it convenient to model the resource-constrained production environment and complex demand dependencies in the BOM structure. The MTO-SCCP also minimizes the system wide total supply chain costs including total setup cost, cost of goods sold (COGS) and pipeline stock cost.

The MTO-SCCP solution assists purchasing and program managers to ensure orders are delivered on time and in a cost-effective way. Specifically, the following decision-supports are provided:

- Exploit the interaction between sourcing and scheduling of supply chain entities
- Optimize make-or-buy and supplier selection decisions in a resource-constrained environment
- Exploit the tradeoff among direct cost, lead time and requirement of internal resources
- Quantify the impact of order delivery deadline on the optimal sourcing decisions and total supply chain costs
- Quantify the impact of internal resource capacity on the optimal sourcing decisions and total supply chain costs

Our MTO-SCCP modeling framework opens an avenue of future research opportunities. Other criteria,such as quality and reliability, can be considered in the sourcing decisions. One may also balance multiple supply chain performance measures of cost, earliness-tardiness in just-in-time (JIT) and responsiveness, through a multi-objective optimization approach. Furthermore, the current model does not allow preemption by assuming one aggregated batch. Such assumption can be relaxed to address more detailed lot sizing decisions. Last but not least, the single-sourcing assumption can be generalized to allow multiple-sourcing, i.e. multiple suppliers or production modes, for supply chain entities.

References

Bartusch M, Mohring RH, Randermacher FJ (1988) Scheduling project networks with resource constraints and time windows. Ann Oper Res 16(1):201–240

Benders JF (1962) Partition procedures for solving mixed variables programming problems. Numer Math 4:238–252

Berger PD, Gerstenfeld A, Zeng AZ (2004) How many suppliers are best? A decision-analysis approach. Omega 32(1):9–15

Bertrand JWM, Sridharan V (2001) A study of simple rules for subcontracting in make-to-order manufacturing. Eur J Oper Res 128(3):509–531

Brucker P (2001) Scheduling algorithms. Springer, Berlin

Chandra C, Grabis J (2007) Supply chain configuration: concepts, solutions and applications. Springer, New York

Christ DK (2001) Theory of constraints project management in aircraft assembly. Technical Report, The Boeing Company

Daskin MS (1995) Network and discrete location: models, algorithms, and applications. Wiley-Interscience series in discrete mathematics and optimization. Wiley, New York

Demeulemeester EL, Herroelen WS (1996) Modeling setup times, process batches and transfer batches using activity network logic. Eur J Oper Res 89(2):355–365

Demeulemeester EL, Herroelen WS (2002) Project scheduling: a research handbook. International series in operations research and management science. Kluwer, Norwell

Demirtas EA, Ustun O (2008) An integrated multiobjective decision making process for supplier selection and order allocation. Omega 36(1):509–521

Drexl A, Juretzka J, Salewski F, Schirmer A (1998) New modeling concepts and their impact on resource-constrained project scheduling. In: Węglarz J (ed) Project scheduling: recent models, algorithms and applications. Kluwer, Boston

Floudas CA (1995) Nonlinear and mixed-integer optimization: fundamentals and applications. Oxford University Press, New York

Glover F (1997) A template for scatter search and path relinking. In: Hao JK, Lutton E, Ronald E, Schoenaurer M, Snyers D (eds) Lecture notes in computer science, vol 1363, pp 13–54, Springer, Berlin

Glover F, Kochenberger G (2003) Handbook of metaheuristics. Kluwer, Boston

Glover F, Laguna M (1997) Tabu search. Kluwer, Boston

Gunasekaran A, Ngai EWT (2005) Build-to-order supply chain management: a literature review and framework for development. J Oper Manag 23(5):423–451

Hall NG, Potts CN (2003) Supply chain scheduling: batching and delivery. Oper Res 51(4):566–584

Holland J (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor

Jacobs FR, Chase RB, Aquilano NJ (2009) Operations and supply chain management. McGraw Hill, Irwin

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680

Kolisch R (2000) Integration of assembly and fabrication for make-to-order production. Int J Prod Econ 68(3):287–306

Kolisch R (2001) Make-to-order assembly management. Springer, Berlin

Li H, Womer K (2007) A model and procedure for competitive bidding under resource constraints. Int J Oper Res 2(4):452–480

Li H, Womer K (2012) Optimizing the supply chain configuration for make-to-order manufacturing. Eur J Oper Res 221(1):118–128

Li L, Zabinsky ZB (2009) Incorporating uncertainty into a supplier selection problem. Int J Prod Econ 134(2):344–356

Murthy NN, Soni S, Ghosh S (2004) A framework for facilitating sourcing and allocation decisions for make-to-order items. Decis Sci 35(4):609–637

Neumann K, Schwindt C (1997) Activity-on-node networks with minimal and maximal time lags and their application to make-to-order production. OR Spectr 19(3):205–217

Pinedo M (1995) Scheduling - theory, algorithms and systems. Prentice Hall, Englewood Cliffs

Ronen B, Trietsch D (1988) A decision support system for purchasing management of large projects. Oper Res 36(6):882–890

Sawik T (2010) Single vs. multiple objective supplier selection in a make-to-order environment. Omega 38(3–4):203–212
Sawik T (2011) Selection of a dynamic supply portfolio in make-to-order environment with risks. Comput Oper Res 38(4):782–796
Walters AA (1963) Production and cost function: an econometric survey. Econometrica 31(1–2): 1–66
Wisner JD, Siferd SP (1995) A survey of U.S. manufacturing practices in make-to-order machine shops. Prod Inv Manag J 36(1):1–6
Womer K (1979) Learning curves, production rate, and program costs. Manag Sci 25(4):312–319
Wouters MJF (1991) Economic evaluation of leadtime reduction. Int J Prod Econ 22(2):111–120
Wu DD, Zhang Y, Wu D, Olson DL (2010) Fuzzy multi-objective programming for supplier selection and risk modeling: a possibility approach. Eur J Oper Res 200(3):774–787
Yue J, Xia Y, Tran T, Chen B (2009) Using frontier portfolios to improve make-to-order operations. Prod Oper Manag 18(2):226–239

Chapter 56
Project Scheduling for Aggregate Production Scheduling in Make-to-Order Environments

Arianna Alfieri and Marcello Urgo

Abstract Production planning of highly customised and complex products is a difficult task and cannot be tackled efficiently by using well-known hierarchical approaches. The main reason is that aggregate production operations correspond to whole production phases, thus requiring planning, scheduling, and procurement activities to be considered at the same decision level. This makes project scheduling approaches particularly suitable for this context. However, the pervasive use of human resources (most operations are executed manually) poses other problems related to the definition of activity durations. In fact, the duration of an activity cannot be a priori defined because it is related to the amount of allotted resources, which in turn depends on the number of products processed at the same time in the shop floor and on the number of workers involved, which can also vary over time. This impacts also on the possibility of correctly modelling the precedence relations between aggregate activities. In this chapter we propose a way to tackle such problems, using a project scheduling approach with a variable intensity formulation and feeding precedence relations and show its application to a real industrial case.

Keywords Aggregate production planning • Feeding precedence relations • Make-to-order • Production scheduling • Project scheduling

56.1 Introduction

One of the core activities in the management of production systems is production planning. It deals with the decisions on how and when the production has to be made. These decisions must take into account the customers' orders and the availability of materials and resources (machines, tools, etc.) with the aim to

A. Alfieri (✉)
Department of Management and Production Engineering, Politecnico di Torino, Torino, Italy
e-mail: arianna.alfieri@polito.it

M. Urgo
Department of Mechanical Engineering, Politecnico di Milano, Milano, Italy
e-mail: marcello.urgo@polimi.it

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_26

minimise production time and cost, to use resources efficiently, and to maximise the overall efficiency of the production system. Furthermore, financial, marketing and technological constraints might be present, thus increasing the complexity of the decisions.

Production planning typically addresses tactical decisions (Anthony 1965). The production volumes in the different periods, the size of the workforce, the amount of overtime and subcontracting work are set over a medium-term time horizon, and are typically revised with a frequency not smaller than 6–12 months and not larger than 2–3 years. Considering the time horizon of these decisions, it is clear that they need to be based on aggregate information. In fact, detailed information is not available and/or would increase the complexity of the problem. Hence, similar products are combined into aggregate product families that can be planned together, production resources, such as distinct machines or human workers, are combined into an aggregate machine or labor resource, and time periods are usually defined on a monthly basis. The existence of an aggregation structure for products, resources and time periods is a necessary requirement for aggregate planning.

Once the aggregate plan has been devised, the availability of raw materials and components must be assured. In fact, finished products are usually composed of many components and sub-assemblies that must be available in the production system before the production of end items starts. Otherwise, the production plan cannot be executed. This availability is assured by the so-called material requirements planning (MRP) that works over a shorter time horizon, disaggregating the aggregate demand and considering the bill-of-material structure in details. The MRP provides the supply plan for the dependent-demand items (i.e., components and sub-assemblies) in a coordinated and systematic way (Vollman et al. 1992). Aggregate production planning and material requirement planning heavily depend on each other and their interactions have a strong impact on the production performance (Harris et al. 2002).

The decision phase following the MRP entails the definition of a detailed plan for the production activities at the operational level, e.g., tool loading, job scheduling or dispatching. This phase explicitly considers a higher level of details and addresses the assignment of activities to production resources, the precedence relations among them, etc.

This sequence of approaches resembles the so-called hierarchical production planning and control framework (Hax and Meal 1975; Bitran and Tirupati 1993; Hopp and Spearman 2000). It is characterised by the idea of separating the decisions at the different levels of detail and uses linking mechanisms to transfer the results from higher levels (those with less detailed information) to lower ones (those with more detailed information). The hierarchical approach fits very well mass production systems where repetitive operations (due to the presence of large batches of identical products that require the same operations) do not need to be scheduled in detail when dealing with tactical decisions.

On the contrary, in make-to-order (MTO) systems that produce items with high complexity like instrumental goods, aircrafts, power generation devices, the effectiveness of the hierarchical approach is somewhat decreased. The production,

in these cases, is the so-called one-of-a-kind production, much more similar to the execution of a project rather than to the production of goods. Hence, exactly as in project management, planning, scheduling and material procurement tend to work on similar time horizons, as it will be discussed in Sect. 56.2, and project scheduling approaches are a suitable tool to address the production planning problem (Márkus et al. 2003).

This chapter addresses the application of project scheduling approaches to the production planning problem in MTO systems producing high-complexity items. Section 56.2 contains the industrial motivation while Sect. 56.3 revises the application of project scheduling approaches to MTO systems. The mathematical formulation of the production planning problem analysed under a project scheduling framework is described in Sect. 56.4. Finally, Sect. 56.5 presents an application of the proposed approach to an industrial case of machining centre production.

56.2 Production Planning in MTO

The production of highly complex and customised items, such as production lines, special production equipment, specifically designed civil or military aircrafts or helicopters, can be considered as a one-of-a-kind process. In fact, not only the system must be a MTO system, but each product has its own characteristics, tailored for the specific customer, that need specific/dedicated design, production and delivery activities and specific requirements in terms of work content, number and kind of components.

As anticipated in Sect. 56.1, the hierarchical production planning and control framework does not fit very well one-of-a-kind MTO production systems. However, the concept of aggregation can be used also in this case, to make the planning problem more easy to study. The aggregation can be performed by grouping distinct production operations into aggregate activities and single machines and workers into groups of production resources. The main difference from the mass production case is that, in one-of-a-kind MTO systems, aggregate activities often represent whole production phases whose duration could be within the range of weeks or either months. Also, although aggregate activities are an aggregation of production operations, they refer to a single product (or to a very small batch of products) whose completion has to meet possible due dates negotiated with the customer. Hence, even at the aggregate level, precedence constraints between activities cannot be ignored and have to be considered because of their impact on the resource load. This makes project scheduling approaches more suitable for one-of-a-kind MTO systems compared to production and rough cut capacity planning approaches traditionally adopted in mass production environments.

As in project scheduling, when considering aggregate activities, the classical finish-to-start precedence relations, representing technological constraints between single manufacturing operations, might not correctly represent the real production process. A common approach to more accurately model the precedence relations is

the use of *generalised precedence relations* (GPRs) (Elmaghraby 1977; Elmaghraby and Kamburowski 1992) that allow a certain amount of overlap among activities and have been extensively considered in the literature on project scheduling to model complex precedence structures in activity networks (Demeulemeester and Herroelen 1997; Neumann and Schwindt 1998; De Reyck and Herroelen 1999; Klein 2000).

In addition, in one-of-a-kind MTO systems, production operations basically refer to the production or the acquisition of a set of components that are assembled together. In other words, each production operation models the assembling, mounting or wiring of a single sub-assembly and many of these operations are often performed manually. This characteristic makes it impossible to precisely define, schedule and control every single activity because its execution, if not constrained to a specific sequence, is autonomously managed at the shop floor level by the workers. Their decisions could depend on the immediate availability of components, material or equipment, the accessibility of the operation area (it could be already occupied by other workers still operating there) or on other factors whose influence is not visible at the planning level. In this context, a detailed planning is difficult or even impossible and aggregate planning is the only viable choice.

As previously stated, single resources are grouped into aggregate production resources. In case of human workers, these aggregate production resources consist of teams of workers in charge of executing a set of tasks (an aggregate operation). In a team, a worker can be assigned to different short activities in the same time period and/or more workers can be assigned to the same activity. Hence, the concepts of unary resource and activity duration need to be reassessed since either the resource used in each time period or the duration of the activity are not univocally defined. This makes the traditional project scheduling approaches no longer suitable and claims for an approach able to consider a variable use of resources during the execution of a production activity, such as the variable intensity formulation of the resource constrained project scheduling problem (Leachman et al. 1990; Kis 2005).

However, combining generalised precedence relations with variable resource intensity is a very critical task since the variable resource effort translates into an infinite number of possible execution modes for the activities. The execution mode of an activity influences its progress in time that is no longer a priori fixed. This leads to the inability of GPRs to exhaustively describe the overlapping among activities (Kis 2006; Tolio and Urgo 2007) and requires the development of a planning and scheduling approach able to consider the execution of the activities both from the temporal and the work content point of view, in order to guarantee that the precedence relations correctly represent the characteristics and the constraints of the real production problem (Alfieri et al. 2012b).

56.3 Project Scheduling Approaches for MTO

As described in Sect. 56.2, the production of a high-complexity customised product can be modelled as the execution of a project and the simultaneously production of different products can be rephrased as different projects being executed at the same

time and competing for the same production resources (machines, workers, etc.). Moreover, the variable intensity of resources has to be consider to correctly model the presence of manually executed operations.

The use of a project scheduling approach for aggregate production planning in MTO systems has been described by Neumann and Schwindt (1998), Hans (2001), Neumann et al. (2003), Márkus et al. (2003), Alfieri et al. (2011, 2012a). Working at an aggregate level, the planner can devise a production plan, constrained by the capacity requirements, over a time horizon ranging from several weeks to several months. This plan serves as a tool to manage customer orders, and their associated due dates, as well as material and resource availability (Alfieri et al. 2012b).

To address the variable resource utilisation and its relation with the execution of the activities, the variable intensity formulation of the resource constrained project scheduling problem has been proposed in the literature. This formulation is based on the introduction of an *intensity variable* used to define the effort dedicated to process each activity in each time period (Leachman et al. 1990; Kis 2005). Resources are considered continuously divisible and are used to process the activities in amounts that can vary over time. In this framework, the execution of activity i is described by a set of continuous variables x_{it} representing the percentage of activity i executed in time period t. The amount of work performed in each period t is not a priori given but depends on the amount of resources committed to the activity, as it is typical for activities executed by human workers.

As discussed in Sect. 56.2, the variable intensity formulation allows an infinite number of execution modes since the time to process an activity is not a priori defined. Specifically, the execution time is strictly related to the value of the intensity execution variables x_{it} and ranges between a minimal and a maximal duration. These minimal and maximal durations are related to the minimal and maximal amount of resources that can be allocated to each activity in each time period. Since the durations of the activities are not a priori defined, the percentage executed in a given time interval does not completely depend on the length of the time interval and, if preemption is allowed, the maximum duration, in terms of number of time units from activity starting and ending instant, is also not constrained.

The presence of an infinite number of execution modes for each activity causes GPRs no longer to be suitable for modeling overlapping between activities in terms of their percentage execution. To overcome this difficulty, the concept of *feeding precedence relation* was introduced by Kis (2005), for the completion-to-start precedence, through binary variables that define an *execution mask*. Each of these masks cannot be assigned the value zero if the associated activity has not been executed (completely or at least for a given percentage). Feeding precedence relations have been further extended in Alfieri et al. (2011) to model precedences different from the completion-to-start one. They are needed to represent the execution of the activity according to the values of the intensity variables and four cases can be defined:

- *%Completed-to-Start (CtS) precedence:* the successor activity j can start its processing only when, in time period t, the percentage of the predecessor activity i that has been processed is greater than or equal to q_{ij} (Fig. 56.1a).

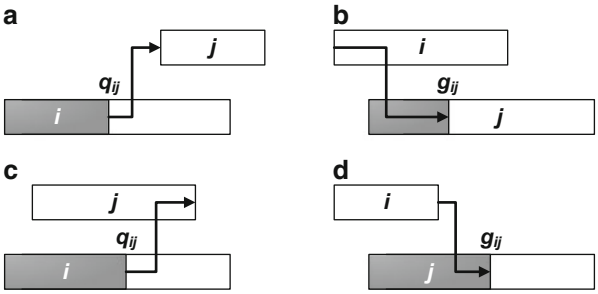


Fig. 56.1 Feeding precedence relations

- *%Completed-to-Finish (CtF) precedence:* the successor activity j can be completed only when, in time period t, the percentage of the predecessor activity i that has been processed is greater than or equal to q_{ij} (Fig. 56.1c).
- *Start-to-%Completed (StC) precedence:* the percentage execution of the successor activity j, in time period t, can be greater than g_{ij} only if the execution of the predecessor activity i has already started (Fig. 56.1b).
- *Finish-to-%Completed (FtC) precedence:* the percentage execution of the successor activity j, in time period t, can be greater than g_{ij} only if the execution of the predecessor activity i has been completed (Fig. 56.1d).

Carefully analysing the above described cases, it is clear that feeding precedences provide a different perspective on the role of precedence relations between pairs of activities by considering both their start and finish time and the progression of their execution. Feeding precedence relations or similar concepts have also been addressed in Bianco and Caramia (2012) and Schwindt and Haselmann (2012).

56.3.1 Aggregate Activity Definition

Aggregate activities are defined applying an aggregation criterion to the detailed set of operations to be scheduled. Several aggregation criteria can be used such as the required resource, the component on which the operations have to operate or the type of tasks to be executed (Fig. 56.2).

Given the detailed network of production operations and their precedence relations (Fig. 56.2a), if only Finish-to-Start precedence relations are considered, then the aggregation causes a single precedence relation between two original operations to enforce a precedence relation between two aggregate activities (Fig. 56.2b). The feeding precedence relations are able to properly represent the relations between aggregate activities, matching the real technological constraints. In fact, as illustrated in Fig. 56.3, there exists a set of operations (belonging to the aggregate activity j) that can be executed even if the predecessor aggregate activity i has not yet been completed. The amount of resources required to process the two sets of operations needs to be computed by considering the work content of each

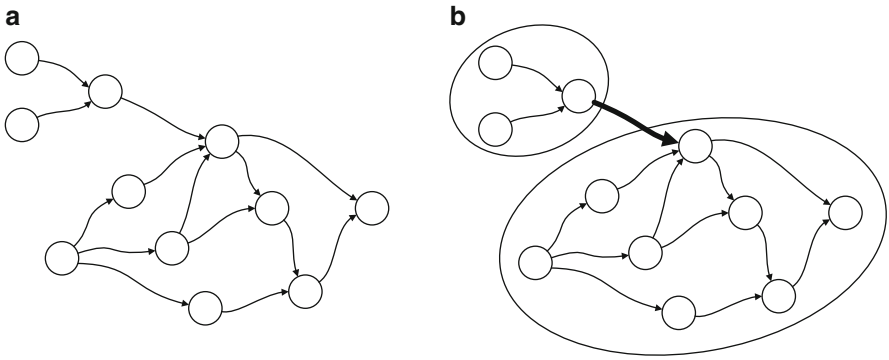


Fig. 56.2 Aggregation of activities

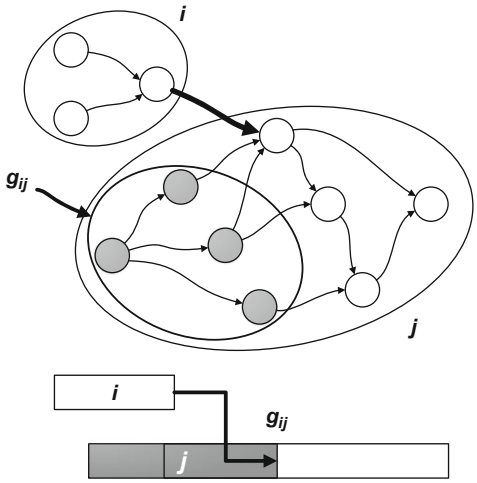


Fig. 56.3 Feeding precedence on aggregate activities

single operation. This allows to estimate the percentage of j that can be executed even if i has not been completed (i.e., g_{ij}).

An overlapping between the execution of the two aggregate activities i and j is therefore allowed. This overlapping is not defined on a temporal basis but it refers to a certain percentage of the predecessor or successor activity that has been completed.

56.3.2 Aggregate Activity Disaggregation

The aggregate production plan provides start and finish times for the aggregate activities but obviously no information about the execution of each single operation

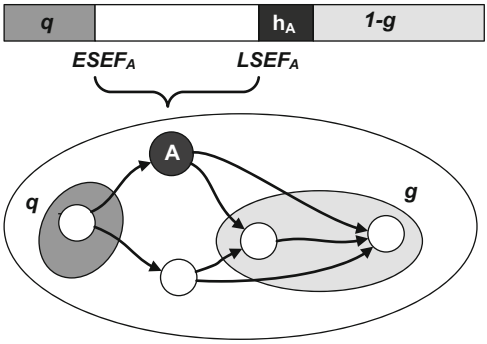


Fig. 56.4 Earliest and latest start execution fraction

within the aggregate activities. However, such information is very important to properly plan the procurement of materials, since the necessary components must be available before the single operation starts. Requiring all the components needed by the aggregate activity to be available before the start of the aggregate activity itself would be ineffective from the point of view of system WIP and might also constrain material procurement too much.

A possible way of dealing with this problem is to combine the information about the start and finish times of each aggregate activity, the information on the single manufacturing operations that are part of the aggregate activity, and the detailed precedence relations among them (Alfieri et al. 2012b). The aggregate planning, working on a medium/long time horizon, gives the exact time intervals for the execution of each aggregate activity. This time interval and the knowledge about the single operations (and their precedences) inside the aggregate activity allow to define a range for the start time and the finish time of each operation. The length of these ranges mostly relies on the structure of aggregate activities, as shown in the example in Fig. 56.4. This approach is referred to as *activity disaggregation* and is detailed in the following.

Given an aggregate activity and a manufacturing operation A within it, the information on the other operations in the aggregate activity is used to provide additional constraints on the start time of A. Considering the structure of the precedence relations, it is possible to identify a set of operations (highlighted in dark gray in Fig. 56.4) that must be executed before operation A can start. The fraction of these operations, with respect to the whole aggregate activity, represents the fraction q of the aggregate activity that must be processed before operation A can start, i.e., the Earliest Start Execution Fraction for operation A ($ESEF_A$). Similarly, it is possible to find a set of operations (highlighted in light gray in Fig. 56.4) that can be executed only after A has been completed, corresponding to a fraction g of the aggregate activity. Let h_A be the fraction of the aggregate activity devoted to the execution of A, then $1 - g - h_A$ is the maximum percentage of the aggregate activity that can be executed even if operation A has not started, i.e., the Latest Start Execution Fraction ($LSEF_A$) for operation A.

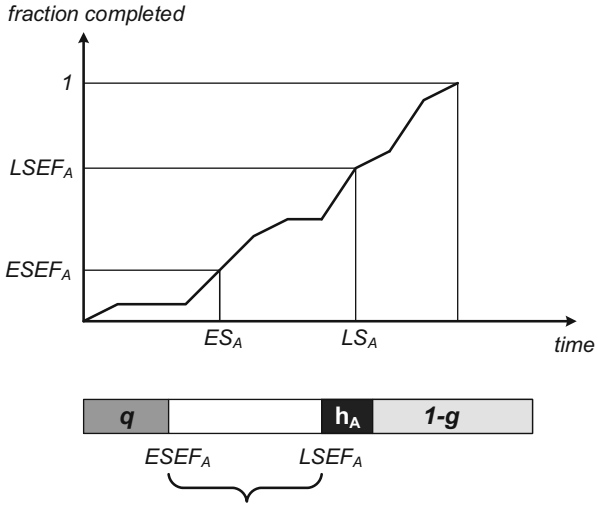


Fig. 56.5 Earliest and latest start time

Both the *ESEF* and *LSEF* are based on the percentage execution of the aggregate activity. Thus, the percentage and the temporal execution of each aggregate activity must be matched to have a correct estimation in terms of the earliest and latest start time for operation *A*. This can be done using the aggregate production plan. Specifically, given the resource effort for the execution of the aggregate activity over time (Fig. 56.5), *ESEF* and *LSEF* provide the Earliest Start Time (*ES*) and the Latest Start Time (*LS*) for the considered operation.

Given the aggregate production plan, the time interval between *ES* and *LS* represents the range for the actual start time of operation *A*. Because there is a group of materials (components) associated with each operation, according to the bill of materials of the final product, this range also provides the earliest and latest due dates for the materials and components needed for the execution of the manufacturing operation. If these components are available before the earliest due date (*ES*), the operation can start at any time within the range. On the contrary, if the components are available only after the latest due date *LS*, the manufacturing operation will have to be delayed and hence also the aggregate activity may be delayed with respect to the planned completion time. Finally, if the components are available at some time between the earliest and latest due dates, the plan is considered feasible by the time analysis but it cannot be assured that no delay will occur. In fact, since the aggregate production plan does not provide the detailed resource utilisation, the joint utilisation of the production resources could constrain the execution of the single operations (if a resource is used for operations *A*, is not available for other operations that might need it at the same time). However, although not providing a complete description of the feasibility for the material procurement phase, the information obtained through the disaggregation process can play a significant role in the definition of the Material Requirement Plan.

56.4 Problem Formulation

The aggregate production planning for the MTO systems previously described is modelled using the mathematical formulation proposed in Alfieri et al. (2011) and reported in the following.

Let $V = \{0, \ldots, n+1\}$ be the set of activities to be scheduled over $T = \{1, \ldots, \overline{d}\}$ time periods and $\mathscr{R} = \{1, \ldots, K\}$ be the set of available resources. Let $R_k(t)$ be the total amount of resource $k \in \mathscr{R}$ available in time period $t \in T$ and $\mathscr{T}$ the set of precedence relations. This set is partitioned into four subsets that refer to the different types of feeding precedence relations:

$\mathscr{T}_1$: subset of precedence relations of type %Completed-to-Start
$\mathscr{T}_2$: subset of precedence relations of type Start-to-%Completed
$\mathscr{T}_3$: subset of precedence relations of type %Completed-to-Finish
$\mathscr{T}_4$: subset of precedence relations of type Finish-to-%Completed

Each precedence relations $p \in \mathscr{T}$ is characterised by a predecessor activity $i_p \in V$ and a successor activity $j_p \in V$. In addition, precedence relations in $\mathscr{T}_1$ and $\mathscr{T}_3$, i.e., %Completed-to-Start and %Completed-to-Finish relations, are characterised by a value q_p representing the minimal fraction of activity i_p that must be processed before activity j_p can start or finish. Analogously, precedence relations in $\mathscr{T}_2$ and $\mathscr{T}_4$, i.e., Start-to-%Completed and Finish-to-%Completed relations, are characterised by a value g_p that represents the maximal fraction of activity j_p that can be processed before activity i_p has started or finished.

The following parameters are associated with each activity $j \in V$ and completely characterise it:

B_j: maximum percentage of work that can be done in a single time period
b_j: minimum percentage of work that can be done in a single time period
r_j: release date
d_j: due date
Q_{jk}: amount of resource k needed to completely process activity j

Differently from parameters, variables are not a priori known and represent the decisions that have to be taken to define the production plan. In the problem we study, the following variables can be defined:

C_{max}: maximum completion time (makespan)
x_{jt}: continuous positive variable representing the percentage of work done on activity j during time bucket t
η_{jt}: binary variable whose value is 1 if activity j is processed in time bucket t, 0 otherwise

In addition, an *execution mask* z_{jt} is defined for each activity j. Activity j can be processed in a time period t only if the value of the execution mask z_{jt} is 1. Each z_{jt} is assigned value 1 at $t = 0$ and its behaviour is constrained to be non-increasing,

assuming value 0 only after activity j has been completed. An execution mask z_{pt} is also associated to each feeding precedence relation $p \in \mathscr{T}$, in particular:

- *%Completed-to-Start* and *%Completed-to-Finish* precedences: the execution mask z_{pt} associated to these relations has value 1 as long as the percentage of the predecessor activity i_p is smaller than q_p. If the completed percentage of the predecessor activity i_p becomes greater than or equal to q_p in time period t, then the value of the execution mask z_{pt} must be 0 for each $t \geq t + 1$.
- *Start-to-%Completed* and *Finish-to-%Completed* precedences: the execution mask z_{pt} associated to these relations has value 1 as long as the processing percentage of the successor activity j_p is smaller than g_p. When, in time period t, this percentage becomes greater than or equal to g_p, then the value of the execution mask z_{pt} must be 0 for each $t \geq t + 1$.

Execution masks z_{jt} and z_{pt} are represented by binary variables.

Using the variables and parameters previously defined, the production planning problem can be formulated as follows:

$$\text{Min. } C_{max} \tag{56.1}$$

$$\text{s.t.} \quad C_{max} \geq t \cdot z_{jt} \quad (j \in V;\ t \in T) \tag{56.2}$$

$$\sum_{t=r_j}^{d_j} x_{jt} = 1 \quad (j \in V) \tag{56.3}$$

$$x_{jt} \leq B_j \eta_{jt} \quad (j \in V;\ t \in T) \tag{56.4}$$

$$x_{jt} \geq b_j \eta_{jt} \quad (j \in V;\ t \in T) \tag{56.5}$$

$$x_{jt} \leq B_j z_{jt} \quad (j \in V;\ t \in T) \tag{56.6}$$

$$z_{j,t-1} \geq z_{jt} \quad (j \in V;\ t \in T) \tag{56.7}$$

$$\sum_{j \in V} Q_{jk} x_{jt} \leq R_k(t) \quad (k \in \mathscr{R};\ t \in T) \tag{56.8}$$

$$z_{p,t-1} \geq z_{pt} \quad (p \in \mathscr{T};\ t \in T) \tag{56.9}$$

$$x_{jt} \leq B_j (1 - z_{pt}) \quad (p \in \mathscr{T}_1;\ j = j_p \in V;\ t \in T) \tag{56.10}$$

$$\sum_{h=1}^{t-1} x_{ih} \geq b_i - z_{pt} \quad (p \in \mathscr{T}_2;\ i = i_p \in V;\ t \in T) \tag{56.11}$$

$$(1 - \sum_{h=1}^{t} x_{jh}) \geq b_j z_{pt} \quad (p \in \mathscr{T}_3;\ j = j_p \in V;\ t \in T) \tag{56.12}$$

$$x_{it} \leq B_i z_{pt} \quad (p \in \mathscr{T}_4;\ i = i_p \in V;\ t \in T) \tag{56.13}$$

$$\sum_{h=1}^{t-1} x_{ih} \geq q_p(1 - z_{pt}) \quad (p \in (\mathscr{T}_1 \cup \mathscr{T}_3);\ i = i_p \in V;\ t \in T) \tag{56.14}$$

$$(1 - \sum_{h=1}^{t} x_{jh}) \geq (1 - g_p)z_{pt} \quad (p \in (\mathscr{T}_2 \cup \mathscr{T}_4);\ j = j_p \in V;\ t \in T) \tag{56.15}$$

The objective function simply states that the performance measure to be minimized is the makespan. It is a classical objective function in the theory of scheduling but it also has an industrial relevance since the makespan minimization is linked to the maximization of the utilization rate, i.e., the more efficient use of production resources.

Equation (56.2) defines the makespan as the finishing time of the last activity that finishes. Equations (56.3)–(56.8) model the execution of the activities from the point of view of single activity parameters while Eqs. (56.10)–(56.15) manage the precedence relations. The non-increasing behavior of the execution masks z_{pt} is forced by Eq. (56.9).

56.5 Industrial Application

A machining centre is a CNC (Computer Numerical Controlled) machine integrated with an automatic tool changer and equipment for pallet or part handling. It is typically made of a multi-axis computer controlled milling machine plus a set of additional equipment providing different functionalities (e.g., devices to automatically change the tools with, a tools storage, devices for automatically change the machined pallets with new ones to be processed, a pallets storage, equipment providing cooling and lubrication, a device for the disposal of metal chips, automatic controllers and computers to manage the high degree of automation, etc.). Although machining centre producers provide standard configurations for their products, customers often ask for modifications tailored to their specific needs. This is a common practice for European (and in particular Italian) machining centre producers.

As a matter of facts, the production of a machining centre is a complex one-of-a-kind process typically addressed by project scheduling approaches. Upon the design of the customised characteristics, the production of a significant set of components is assigned to external suppliers, while only high precision manufacturing activities for critical components are internally executed. Hence, the production process begins with the assembling of the kernel structure of the machining centre together with the main components, e.g., the spindle and the working table, that are internally manufactured (Fig. 56.6). This is a critical step since the capability of the machining centre (in terms of accuracy, repeatability and performance) strictly depends on the quality of these components and on how they are assembled together.

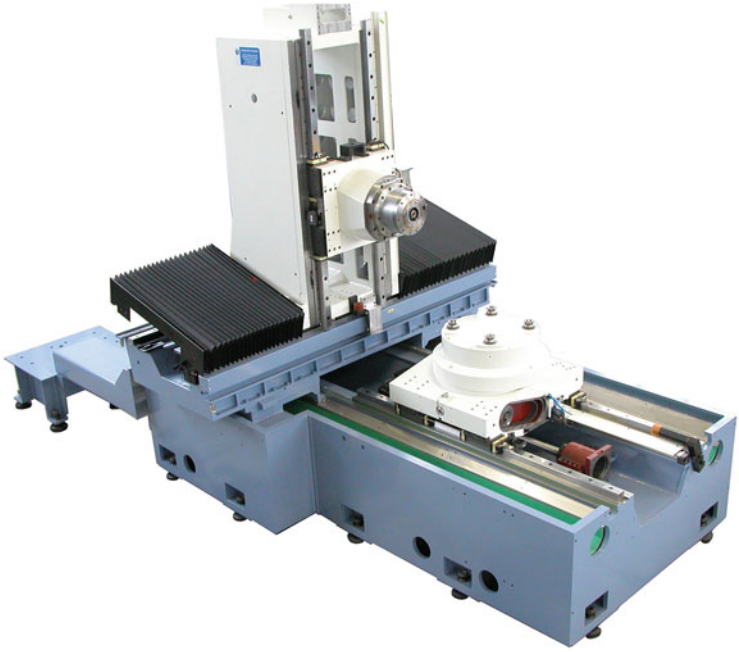


Fig. 56.6 Machining centre structure with preassembled components

Fig. 56.7 Complete machining centre

All the other components and parts are assembled around the main structure and wired together. The final assembly (Fig. 56.7) is tested and then partially disassembled to be delivered to the customer.

For modelling purposes, the detailed production process has been considered taking into consideration about two hundreds of production activities, each addressing the assembling, wiring or testing of a specific set of components. The definition of aggregate activities has been carried out based on the bill of materials of the machining centre. Components have been grouped into functional units and an aggregate manufacturing or assembling activity has been defined for each group.

Once the definition of aggregate activities have been performed, the production of a machining centre entails eight main phases:

- A01: Structure Preparation. The machine centre structure is prepared for the assembling phase. Scraping operations are performed to provide a proper finishing level where needed.
- A02: Pallet Preparation. The pallets are prepared for the assembling phase. Scraping operations are performed to provide a proper finishing level where needed.
- A03: Structure Painting. The machine centre structure is painted.
- A04: Autonomous Components Assembling. Autonomous components (e.g., spindle head, machine table, electrical cabinet), to be installed onto the machining centre, are separately assembled.
- A05: Assembling. The machine centre structure is placed in the assembling area and all the components are installed.
- A06: Wiring. Electrical connection is provided for all the installed components and for the control system.
- A07: Testing. The main functionalities are tested according to the main regulations and internal standards. The machine centre accuracy is tested against its declared capabilities and the customer's specifications.
- A08: Disassembling and Delivery. The machining centre is partly disassembled and delivered to the customer.

Feeding precedence relations have been used to correctly model the production process. The need for feeding precedence relations is motivated by the fact that finish-to-start precedence relations among aggregate activities would impose unnecessary constraints with respect to the real manufacturing process. Assembling phase is made of a large number of sub-phases devoted to the separate assembling of single autonomous components, i.e., the electrical cabinet, the spindle head, the working table. These autonomous components need to be installed onto the machining centre structure but, as a matter of fact, they need not be completely processed at the time the Assembling phase starts. On the contrary, considering the detailed production process, they can be mounted onto the machining centre's structure only after a certain set of other assembling operations have been completed. At the same time, they must be completed at latest before the machining centre is ready to have them installed onto.

For such cases, a *Finish-to-%Completed* precedence constraint can be used to allow the assembly of different autonomous components to be completed at the latest after a certain percentage of the machining centre assembling has been executed. This percentage represents the percentage of the assembling activity that

can be carried out even if the considered subassembly is not yet ready to be installed onto the machining centre.

An analogous consideration can be done referring to the relations between the Assembling and the Wiring phase. The last should not wait for the completion of the whole Assembling phase to start but wiring can start as soon as components that need to be wired together are installed. In this case, the wiring activity must be allowed to start at the earliest after a certain percentage of the assembling activity has been completed. Hence a %Completed-to-Start precedence constraint can be used to allow the wiring phase to start as soon as the components that need to be cabled together are installed onto the machining centre.

These phases are mainly processed by workers. Workers are grouped into seven different types according to their particular skills and each production phase requires only one type of skilled workers. The workers in a team can operate on different production operations belonging to the same aggregate activity as well as some of them can be moved to different teams working on different machining centres that are produced at the same time. Their behaviour can be correctly modelled using the variable intensity formulation that allows a variable resource utilisation. The resource availability is considered constant even if, in the real industrial environment, it depends on the requirements of the other orders that might be simultaneously in production (Fig. 56.8).

The obtained schedule is represented in Fig. 56.9 that reports the execution of the production activities in terms of their intensity. The schedule clearly shows that the activity $A05$, which models the assembling of the machining centre, is the longest one, starting on day 10 and finishing on day 27. As discussed before, the execution

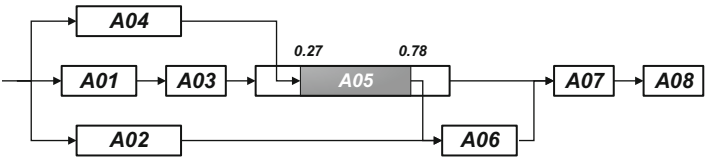


Fig. 56.8 Aggregate activities network

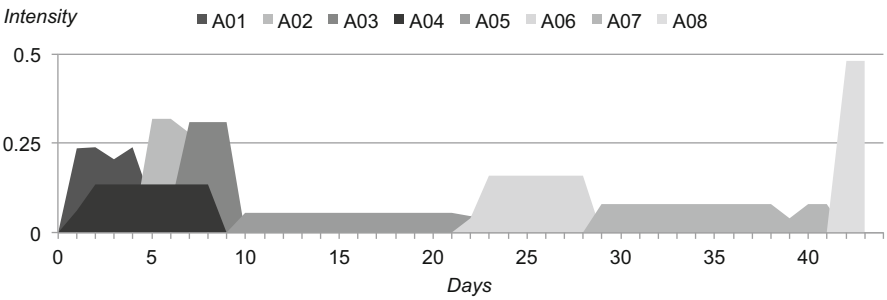


Fig. 56.9 Activity execution profile

Table 56.1 Results

Sub-activity	ESEF	LSEF	ES	LS	TF
$A05_1$	0.00	0.17	10	12	2
$A05_2$	0.17	0.34	13	15	2
$A05_3$	0.17	1.00	13	26	13
$A05_4$	0.22	0.48	13	17	4
$A05_5$	0.22	0.48	13	17	4
$A05_6$	0.17	0.48	13	17	4
$A05_7$	0.43	0.55	17	18	1
$A05_8$	0.50	1.00	18	26	8
$A05_9$	0.50	1.00	18	26	8
$A05_{10}$	0.50	1.00	18	26	8
$A05_{11}$	0.50	1.00	18	26	8
$A05_{12}$	0.50	1.00	18	26	8
$A05_{13}$	0.50	1.00	18	26	8
$A05_{14}$	0.53	0.97	19	26	8
$A05_{15}$	0.54	0.97	19	26	8

of activity $A05$ is associated to the availability of all the set of components to be assembled together. However, requiring all the components to be available at the beginning of the activity could be over-constraining.

To address this problem, the activity could be disaggregated to estimate time intervals in which the different components should be required. The (aggregate) assembling activity is decomposed into 15 sub-activities, each associated to the assembling of a specific set of components. Thus, given the precedence structure among these sub-activities, the *ESEF* and *LSEF* can be calculated and, considering the execution of activity $A05$ in the schedule, the Earliest Start Time (*ES*) and Latest Start Time (*LS*) can be computed for each sub-activity. These values actually provide the Earliest and Latest Due Dates for the availability of the components associated with each sub-activity (Table 56.1). The results also reports the values of the total float (*TF*), i.e., the range between *ES* and *LS*.

The results show that several manufacturing operations, i.e., $AO5_1$, $AO5_2$, $AO5_4$, $AO5_5$, $AO5_6$ and $AO5_7$, have a range between EST and LST between 1 and 4 days. In such cases, the accuracy in the estimation of the start time of the sub-activity can be considered good. Other assembling operations, i.e., $A05_3$, $A05_8$, $A05_9$, $A05_{10}$, $A05_{11}$, $A05_{12}$, $A05_{13}$, $A05_{14}$ and $A05_{15}$, show a bigger range, between 8 and 13 days, i.e., between 1 and 2 weeks. Such a wide range, however, is mostly due to the fact that these assembling operations can be processed in parallel with other assembling operations that are on a critical path. Hence, a shift of only one of these sub-activities within the provided range, due to a late component supply, will not cause a delay of the whole assembling phase. However, when more than a single group of components is supplied later than the *ES*, only a detailed scheduling phase can verify the effective occurrence of a delay. Sub-activity $AO5_3$, having the widest range, represents the installation of the hydraulic system, an external component

that can be installed at any time after the axes and actuators have been assembled onto the machining centre.

Even if the range could be considered not accurate, compared to what can be obtained through a detailed scheduling and material requirement planning approach, these results are based on an aggregate plan and exploit the available information, i.e., the detailed structure of the aggregate activities, Hence, they allow to achieve a fair compromise between the anticipation of procurement and the avoidance of a computationally intensive detailed scheduling phase.

56.6 Conclusions

In this chapter an application of project scheduling to the production planning problem of MTO manufacturing systems producing highly complex and customized items has been addressed. A mathematical model based on a variable intensity formulation and feeding precedence relations was proposed. The applicability of the suggested approach has been shown through the application to a real industrial case.

Acknowledgements The authors thanks M.C.M. S.p.A. for their support in the definition of the industrial case.

References

Alfieri A, Tolio T, Urgo M (2011) A project scheduling approach to production planning with feeding precedence relations. Int J Prod Res 49(4):995–1020

Alfieri A, Tolio T, Urgo M (2012a) A two-stage stochastic programming project scheduling approach to production planning. Int J Adv Manuf Tech 62(1–4):279–290

Alfieri A, Tolio T, Urgo M (2012b) A project scheduling approach to production and material requirement planning in manufacturing-to-order environments. J Intell Manuf 23(3):575–585

Anthony, RN (1965) Planning and control systems: a framework for analysis. Harvard University, Boston

Bianco L, Caramia M (2012) Minimizing the completion time of a project under resource constraints and feeding precedence relations: an exact algorithm. 4OR-Q J Oper Res 10(4): 361–377

Bitran GR, Tirupati D (1993) Hierarchical production planning. In: Graves SC, Rinnooy Kan AHG, Zipkin PH (eds) Logistics of production and inventory. Handbooks in operations research and management science, vol 4. Elsevier, Amsterdam, pp 523–568

Demeulemeester EL, Herroelen WS (1997) A branch-and-bound procedure for the generalized resource-constrained project scheduling problem. Oper Res 45(2):201–212

De Reyck B, Herroelen W (1999) The multi-mode resource-constrained project scheduling problem with generalized precedence relations. Eur J Oper Res 119(2):538–556

Elmaghraby SEE (1977) Activity networks. Wiley, New York

Elmaghraby ES, Kamburowski J (1992) The analysis of activity networks under generalized precedence relations (gprs). Manag Sci 38(9):1245–1263

Hans E (2001) Resource loading by branch-and-price techniques. Ph.D. dissertation, University of Twente

Harris B, Lewis F, Cook DJ (2002) A matrix formulation for integrating assembly trees and manufacturing resource planning with capacity constraints. J Intelli Manuf 13(4):239–252

Hax AC, Meal HC (1975) Hierarchical integration of production planning and scheduling. In: Logistics. Studies in the management sciences, vol 1. North-Holland, Amsterdam

Hopp WJ, Spearman ML (2000) Factory physics: foundations of manufacturing management. McGraw-Hill, Irwin

Kis T (2005) A branch-and-cut algorithm for scheduling of projects with variable-intensity activities. Math Program 103(3):515–539

Kis T (2006) RCPS with variable intensity activities and feeding precedence constraint. In: Józefowska J, Węglarz J (eds) Perspectives in modern project scheduling. Springer, New York, pp 105–129

Klein R (2000) Project scheduling with time-varying resource constraints. Int J Prod Res 38(16):3937–3952

Leachman RC, Dtncerler A, Kim S (1990) Resource-constrained scheduling of projects with variable-intensity activities. IIE Trans 22(1):31–40

Márkus A, Váncza J, Kis T, Kovács A (2003) Project scheduling approach for production planning. CIRP Ann-Manuf Techn 52(1):359–362

Neumann K, Schwindt C (1998) A capacitated hierarchical approach to make-to-order production. Eur J Automat 32(1):397–413

Neumann K, Schwindt C, Zimmermann J (2003) Project scheduling with time windows and scarce resources: temporal resource-constrained project scheduling with regular and nonregular objective functions. Springer, Berlin

Schwindt C, Haselmann T (2012) The preemptive project scheduling problem with generalized precedence relationships. In: Proceedings of the 13th international conference on project management and scheduling, Leuven, pp 278–281

Tolio T, Urgo M (2007) A rolling horizon approach to plan outsourcing in manufacturing-to-order environments affected by uncertainty. CIRP Ann-Manuf Techn 56(1):487–490

Vollman TE, Berry WL, Whybark DC (1992) Manufacturing planning and control systems. Richard D. Irwin, Homewood

Chapter 57
Pharmaceutical R&D Pipeline Planning

Matthew Colvin and Christos T. Maravelias

Abstract The development of new drugs is a long and expensive process with multiple sources of uncertainty and complex trade-offs. In this chapter, we discuss how multi-stage stochastic programming (SP) methods can be used to develop tools for decision-making in this sector. First, we present a basic model for the stochastic resource constraint project scheduling problem we consider in this chapter, we discuss extensions for features such as outlicensing and outsourcing, and present modeling methods for risk management. Second, we review theoretical results that allow us to formulate tractable SP models that account for endogenous observation of uncertainty. Finally, we discuss solution methods that allow us to address realistic instances and present two examples.

Keywords Endogenous uncertainty • Mixed-integer programming • Nonanticipativity • Project scheduling • Risk management • Stochastic programming

57.1 Introduction

The development of a new drug includes multiple stages: (1) discovery, (2) preclinical testing, (3) clinical testing, and (4) regulatory approval. Once a compound is found to be successful in a simple system (e.g., tube or individual cells) it moves on to preclinical testing, where it is determined what happens when the compound is metabolized and whether it is safe to test on people. Clinical testing typically includes three phases: (1) phase I (PI): 20–100 healthy volunteers are used to determine safety and dosage; (2) phase II (PII): 100–500 patient volunteers are used to determine efficacy and side effects; and (3) phase III (PIII): 500–1,500

M. Colvin
ASPEN Tech, Houston, TX, USA
e-mail: matthew.colvin@aspentech.com

C.T. Maravelias (✉)
Department of Chemical and Biological Engineering, University of Wisconsin, Madison, WI, USA
e-mail: christos@engr.wisc.edu

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_27

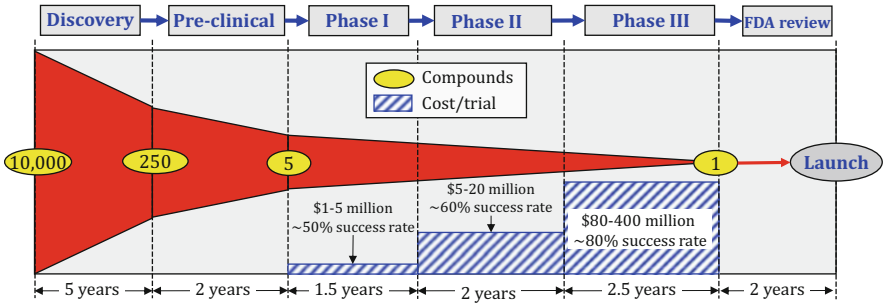


Fig. 57.1 Pharmaceutical research and development activities

patient volunteers are used to monitor adverse reactions to long-term use. Upon successful completion of all clinical trials, a compound can gain approval from regulatory organizations. However, if a compound fails a trial, then its development is discontinued. For each drug entering the market place, 5–10 drugs have to be typically tested (see Fig. 57.1) and the total investment can exceed $1 billion. Furthermore, a pharmaceutical company has limited resources for the development of new drugs, which means that given a portfolio of compounds, projects have to be prioritized, then resources be allocated to competing projects, and finally R&D activities be sequenced.

Accordingly, the goal of this chapter is to present a stochastic programming framework for the planning of research and development (R&D) activities in pharmaceutical pipelines, including modeling approaches, theoretical results, and solution methods (Colvin and Maravelias 2008, 2009, 2010, 2011).

The present chapter is structured as follows. In Sect. 57.2, we present background information. In Sect. 57.3, we present the basic mixed-integer programming (MIP) multi-stage stochastic programming model and a number of modeling extensions. In Sect. 57.4, we present theoretical results that allow us to formulate tractable models, and in Sect. 57.5 we present three solution methods. Finally, in Sect. 57.6 we present two examples.

57.2 Background

In this section, we present the formal statement of the problem we consider, present a literature review, and discuss stochastic programming and endogenous observation of uncertainty.

57.2.1 Problem Statement

The problem we consider can be stated as follows. Given are:

1. A set of drugs, $i \in I$, at different stages of development. It is assumed that all drugs in the pipeline have an approved patent just before PI clinical trial starts, which means that the revenues from the successful lunching of a drug decrease with development delays (since these lead to a shorter active patent live). Also, it is assumed that due to competition, the market share of a drug decreases with time, so the revenues also decrease with the launch time, regardless of the duration of the active patent life.
2. A set of trials, $j \in J = \{\text{PI}, \text{PII}, \text{PIII}\}$. The trials have to be performed sequentially, PI $\rightarrow$ PII $\rightarrow$ PIII. At a given time, a drug has to undergo a subset of trials J_i. For simplicity, we will refer to trial j of drug i as trial (i, j). The cost and duration of trial (i, j) is c_{ij} and p_{ij}, respectively; the probability of trial (i, j) being successful is π_{ij}.
3. A set of resources, $k \in \mathscr{R}$. The availability of resource k is R_k; the requirement of trial (i, j) for resource k is denoted by r_{ijk}.

The objective function is to maximize the expected net present value (ENPV) of the pipeline. We assume that uncertainty occurs only in the outcome of clinical trials, with no advantage gained, either in knowledge or revenue from a failed trial. We assume that all uncertainty is independent, though this can be relaxed. We also assume that resources are renewable and have fixed availability. Both of these assumptions can also be relaxed. Finally, our approach can be extended to account for uncertainties in cost, duration, and resource requirements, but this would lead to prohibitively large formulations.

57.2.2 Literature Review

The planning of R&D activities is similar to a stochastic version of the resource-constrained project scheduling problem (RCPSP) since each product can be viewed as a project consisting of a number of tasks (trials) with given processing times and resource requirements, subject to precedence and resource constraints. However, traditional methods for RCPSP cannot readily address the problem at hand for three reasons. First, the focus in RCPSP has been on the development of methods for instances with a large number of tasks. The number of tasks in this problem is relatively small—what makes this problem hard is the stochastic nature of the process and the fact that the decision-maker can alter uncertainty observation. Second, most stochastic approaches to RCPSP consider uncertainty in the duration and/or resource requirements of tasks. The major uncertainty in this application however is in the outcome of a task. Third, RCPSP formulations often assume the resource level is known and fixed at the beginning of the planning horizon. With development projects spanning multiple years, the ability to adjust resource levels can be as important as the timing of development tasks.

The *static* selection of R&D projects has been the topic of extensive research (Souder and Mandakovic 1986; Steele 1988). Heuristic methods for the RCPSP can be generally categorized as genetic algorithms (Hartmann 1998), local search methods (Mika et al. 2005; Bouleimen and Lecocq 2003), ant colony optimization (Merkle et al. 2002), and forward-backward improvement (Tormos and Lova 2001; Valls et al. 2005). An overview of computational results for a number of algorithms (Kolisch and Hartmann 2006) and a general overview of RCPSP (Brucker et al. 1999) are available. The RCPSP has also been extended to include ideas such as uncertain duration (Herroelen and Leus 2005), partially renewable resources (Bottcher et al. 1999), and maximizing net present value rather than minimizing makespan (Neumann and Zimmermann 2000). As far as R&D planning is concerned, the problems of portfolio selection in the pre-clinical trials section (Charalambous and Gittins 2008), and the planning of R&D activities with technical failure without resource constraints (De Reyck and Leus 2008) have been addressed. Recently, an anticipatory algorithm for the stochastic RCPSP (Mercier and Van Hentenryck 2008) and a MIP model for portfolio optimization (Solak et al. 2010) were proposed.

A number of approaches have also been presented in the process systems engineering literature (Shah 2004), including deterministic MIP models (Schmidt and Grossmann 1996; Jain and Grossmann 1999; Maravelias and Grossmann 2004); a simulation-optimization framework (Subramanian et al. 2001, 2003); a dynamic programming approach (Choi et al. 2004); and a real-options strategy (Rogers et al. 2002). Finally, researchers have proposed methods for the related problem of capacity planning (Gatica et al. 2003; Levis and Papageorgiou 2004). The interested reader is also referred to papers offering a general discussion of new product development (Stonebraker 2002; DiMasi and Grabowski 2007).

57.2.3 Stochastic Programming

In a two-stage SP problem the decision maker makes a set of first-stage decisions x before the realization of uncertainty, and then makes second-stage decisions y (i.e., takes recourse action) upon uncertainty realization (Kall and Wallace 1994; Birge and Louveaux 1997):

$$\min \left\{ c^T x + E_{\xi \in \Xi} \left[Q\left(x, \xi\right) \right] : x \in X \right\} \tag{57.1}$$

with

$$Q(x, \xi) = \min \left\{ f(\xi)^T y : T(\xi)x + Wy = h(\xi), y \in Y \right\} \tag{57.2}$$

where $X \subseteq \mathbb{R}^{n1}, Y \subseteq \mathbb{R}^{n2}$, are polyhedral sets, ξ is a random vector from an induced probability space (Ξ, F, P) with $\Xi \subseteq \mathbb{R}^L, f : \Xi \to \mathbb{R}^{n2}, h : \Xi \to \mathbb{R}^{m2}, W : \Xi \to \mathbb{R}^{m2 \times n2}$, and $T : \Xi \to \mathbb{R}^{m2 \times n1}$. Problem (57.1) is the first-stage problem

with decisions x and problem (57.2) is the second-stage (recourse) problem with decisions y.

If the probability density function for uncertain parameter ξ_l, $l = 1, \ldots, L$ is represented (approximated) via a discrete density function with sample space $\Xi_l = \{\xi_l^m, m = 1, 2, \ldots, M^l\}$, then each combination of realizations for vector $\xi = (\xi_1, \xi_2, \ldots, \xi_L)$ corresponds to a scenario $\sigma \in \Sigma$. The total number of scenarios is $|\sum| = \prod_l M^l$. If the probability of outcome m for parameter ξ_l is $\pi\left(\xi_l = \xi_l^m\right) = \pi_l^m$ and $\xi_l = \xi_l^{m=m(\sigma)}$ in scenario σ, then the probability of scenario σ is $\pi_\sigma = \prod_l \pi_l^{m(\sigma)}$. In this case, the two-stage problem can be written as:

$$\min \left\{c^T x + \sum\nolimits_{\sigma \in \sum} \pi_\sigma q^T y_\sigma : x \in X; T(\xi^\sigma)x + W y_\sigma = h(\xi^\sigma), \sigma \in \sum; y_\sigma \geq 0\right\} \tag{57.3}$$

where ξ^σ is the vector in scenario σ, q is the second-stage cost, y_σ is the solution of the second-stage problem in scenario σ, and the optimal solution to (57.3) consists of a set of single first-stage decisions x and a collection of recourse (second-stage) decisions y_σ.

Two-stage problems can naturally be extended to multi-stage problems where uncertainty unfolds at different stages. Stages typically (but not always) correspond to time periods, i.e., the planning horizon is divided into multiple periods (stages) $t \in \mathscr{T} = \{1, 2, \ldots, T\}$. At the beginning of the horizon, no uncertainty is known, while at the end of each stage t the realizations of a subset of uncertain parameters are observed. At $t = 1$ the decision-maker has to make a unique set of first-stage decisions because all scenarios are *indistinguishable*. As the random variables are observed, scenarios become distinguishable and the decision-maker can take different actions to react to different realizations of uncertainty. At $t = 2$, the decision-maker takes *recourse* action (*second-stage* decisions) to compensate for the effect of the uncertain parameters that were realized earlier, and the process is repeated at $t = 3$.

If we know the stage $t^{\sigma,\sigma'}$ at which scenarios σ and σ' become distinguishable, and for simplicity we assume that the same decisions y are made at each stage t, then the decisions for scenarios σ and σ' at stage t must be identical if $t < t^{\sigma,\sigma'}$,

$$y_{t\sigma} = y_{t\sigma'}((t; \sigma; \sigma') : t < t^{\sigma,\sigma'}, \sigma < \sigma') \tag{57.4}$$

where $y_{t\sigma}$ is the vector of optimization decisions at stage t in scenario σ. Alternatively, if $\sum_t^n$ is a maximal subset of indistinguishable scenarios at stage t; i.e., $\sum_t^n = \left\{\sigma : t^{\sigma,\sigma'} > t, \sigma' \in \sum_t^n\right\}$, then the above equation can be re-written as:

$$y_{t\sigma} \sum\nolimits_{\sigma' \in \sum_t^n} \pi_{\sigma'} - \sum\nolimits_{\sigma' \in \sum_t^n} \pi_{\sigma'} y_{t\sigma'} = 0 \ (t; n \in N_t; \sigma \in \sum\nolimits_t^n) \tag{57.5}$$

where N_t is the family of maximal subsets $\sum_t^n \subseteq \sum$.

The equalities in Eq. (57.5) express the fact that we cannot *anticipate* possible outcomes, thus our decisions have to be *nonanticipative* of future outcomes (Birge and Louveaux 1997). Hence, Eq. (57.5) enforces *nonanticipativity*, and the equalities in Eq. (57.5) are called *nonanticipativity* constraints (NACs).

57.2.4 Endogenous Observation of Uncertainty

Stochastic programming methods have been for the most part used to address problems under purely *exogenous* uncertainty, that is, problems where the underlying stochastic process does not depend on the optimization decisions, which means that uncertainty distributions and the way uncertain parameters are observed are *fixed*.

However, there are problems where the decision maker alters the underlying stochastic process by either changing the distribution of random parameters or by changing the time at which uncertainty is observed. In the first case, we have problems under *endogenous* uncertainty, while in the second case we have problems under *endogenous observation* of uncertainty.

As explained in the previous section, nonanticipativity in the presence of exogenous uncertainty is enforced via Eq. (57.5), which leads to variable elimination. In the presence of endogenous uncertainty observation, however, Eq. (57.5) cannot be used because the stage $t^{\sigma,\sigma'}$ at which scenarios σ and σ' become distinguishable is not known prior to optimization, which means that nonanticipativity is equivalent to the following logic condition: *If* $t < t^{\sigma,\sigma'}$ *then* $y_{t\sigma} = y_{t\sigma}$, which can be rewritten as:

$$\{t < t^{\sigma,\sigma'}\} \Rightarrow \{y_{t\sigma} = y_{t\sigma'}\} \tag{57.6}$$

To develop a mathematical programming formulation, we need to convert the logic condition in Eq. (57.6) into algebraic constraints. To do so, we introduce binary variable $z_{t\sigma\sigma'}$ which is 1 if scenarios σ and σ' are distinguishable at stage t, i.e., $z_{t\sigma\sigma'} = 1$ if $t \geq t^{\sigma,\sigma'}$. If all decision variables are binary, then nonanticipativity is enforced by,

$$-z_{t\sigma\sigma'}\mathbf{1} \leq y_{t\sigma} - y_{t\sigma'} \leq z_{t\sigma\sigma'}\mathbf{1}, \;\; (t;\sigma;\sigma' > \sigma) \tag{57.7}$$

where $\mathbf{1}$ is vector of appropriate dimension whose elements are all equal to 1 (in the general case, vector $\mathbf{1}$ should be replaced by the vector of upper bounds on variables $y_{t\sigma}$). Furthermore, constraints that *activate* variables $z_{t\sigma\sigma'}$, which are typically linked with variables $y_{t\sigma}$, are needed.

An interesting implication of the *variability* of the NACs in Eq. (57.7) is that the scenario tree is *dynamic*. A simplified instance of the stochastic-RCPSP problem that illustrates this concept is shown in Fig. 57.2. Given are two tasks, T1 and T2, that can either pass or fail; each task incurs a cost, but if it passes leads to revenue. Each trial lasts 1 month and due to resource constraints trials cannot be carried out

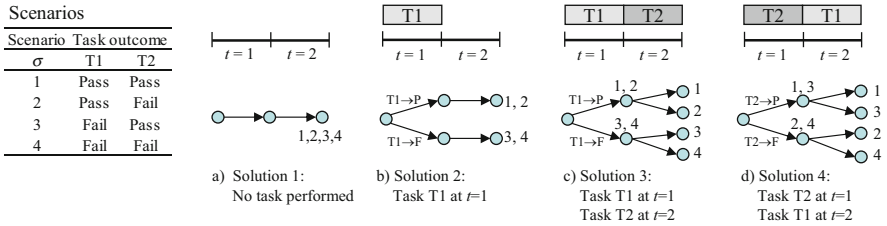


Fig. 57.2 Stochastic-RCPSP: example of dynamic scenario tree. Scenarios: (1) T1 and T2 pass; (2) T1 passes, T2 fails; (3) T1 fails, T2 passes; (4) T1 and T2 fail. Four different solutions lead to four different scenario trees. In solution 1, no tasks are carried out and thus no uncertainty is observed; all scenarios remain indistinguishable. In solution 2, only T1 is executed resulting into a tree with a branching after the first period; scenarios $\{1, 2\}$ and $\{3, 4\}$ remain indistinguishable. In solution 3, T1 is executed at $t = 1$ and T2 at $t = 2$ which means that all scenarios become distinguishable; note that the first branching corresponds to the outcome of T1. In solution 4, all scenarios become distinguishable (as in solution 3) but the first branching corresponds to the outcome of task T2, leading to a different tree

simultaneously. Given a planning horizon of 2 months divided into two 1-month periods, the goal is to determine whether and when to carry out the trials. As shown in Fig. 57.2, different solutions lead to different scenario trees.

57.3 Mathematical Formulation

The planning horizon is divided into time periods, $t \in \mathscr{T} = \{1, 2, \ldots, T\}$, which also correspond to the stages of our SP model. The uncertainty considered in this work is discrete: a drug either passes (P) or fails (F) a clinical trial. In other words, the outcome of clinical trial (i, j), can be viewed as a discrete random variable with sample space $\Xi_{ij} = \{\mathrm{P}, \mathrm{F}\}$. However, since a drug will not undergo a trial if it has failed a previous trial, it is possible to aggregate these outcomes into four events per drug based upon when the drug first fails a trial or successfully navigates all trials. Thus, uncertainty can be represented via a single uncertain parameter, ξ_i, per drug $i \in I$ with sample space $\Xi_i = \{\text{PI-F, PII-F, PIII-F, PIII-P}\}$.

57.3.1 Basic Model

For a given stage t and scenario σ, only a subset $IJ_{t\sigma} \subset I \times J$ of clinical trials can be performed. A trial is not included in this subset because: (1) it was completed prior to the start of the planning horizon, i.e., drug is ready to undergo PII or PIII trials at $t = 1$; (2) a prerequisite trial fails in this scenario; and (3) there have been insufficient stages to perform all prerequisite trials. We denote the subset of drugs

that successfully complete all clinical trials in a given scenario by I_σ, and the first clinical trial that must be performed on a drug by $f(i)$.

The major variable in this problem is binary $x_{ijt\sigma}$—it indicates whether clinical trial (i, j) begins testing at stage t in scenario σ. We define two additional variables through $x_{ijt\sigma}$. Variable $y_{ijt\sigma}$ indicates if trial (i, j) has been completed by stage t:

$$y_{ijt\sigma} = y_{ij,t-1,\sigma} + x_{ij,t-p_{ij},\sigma} = y_{ij0} + \sum\nolimits_{t'<t-p_{ij}} x_{ijt'\sigma} \quad (t;\sigma;(i,j) \in IJ_{t-p_{ij},\sigma}) \tag{57.8}$$

where y_{ij0} is 1 if drug i has already passed trial j, i.e., $y_{ij0} = 1$ if $j < f(i)$.

Variable $z_{ijt\sigma}$ indicates if all prerequisites to trial (i, j) have been completed by stage t in scenario σ, but (i, j) has not been initiated:

$$z_{ijt\sigma} = z_{ij0} + \sum\nolimits_{t'\le t-p_{i,j-1}} x_{i,j-1,t'\sigma} - \sum\nolimits_{t'\le t} x_{ijt'\sigma} \quad (t;\sigma;(i,j) \in IJ_{t\sigma}) \tag{57.9}$$

where z_{ij0} is a parameter indicating the status of trial (i, j) at the beginning of the horizon; e.g., if drug i has passed preclinical testing and is ready to undergo PI trials, then $z_{i,\mathrm{PI},0} = 1$ and $z_{i,\mathrm{PII},0} = z_{i,\mathrm{PIII},0} = 0$.

A clinical trial is not allowed to start until the preceding trial has been completed:

$$\sum\nolimits_{t'\le t} x_{ijt'\sigma} \le y_{i,j-1,t\sigma} \quad ((t,\sigma);(i,j) \in IJ_{t\sigma} : j \ne f(i)) \tag{57.10}$$

A resource limit is enforced over all trials being carried out at a given stage:

$$\sum\nolimits_{(i,j)\in IJ_{t\sigma}} \sum\nolimits_{t'=t-p_{ij}+1}^{t'=t} r_{ijk} x_{ijt'\sigma} \le R_k \quad (k;t;\sigma) \tag{57.11}$$

The time value of money is considered via a discount factor β_t. The cost, CST_σ, associated with a scenario can then be calculated by summing the discounted costs of all clinical trials performed in that scenario:

$$CST_\sigma = \sum\nolimits_{(i,j)\in IJ_{t\sigma},t} \beta_t c_{ij} x_{ijt\sigma} \quad (\sigma) \tag{57.12}$$

Revenue, RV_σ, is modeled as a decreasing function of completion time and developmental delays. The term rv_i^{max} represents the revenue received if a drug were to have successfully completed all trials at the start of the time horizon. To indicate the opportunity cost, both in terms of lost market share and alternate investment possibilities, we estimate the fraction of revenue lost with parameter γ_{it}. To model shorter effective patent life, we introduce parameter Δ_i which denotes how much revenue is lost for each stage development is delayed once started:

$$RV_\sigma = \sum\nolimits_{i\in I_\sigma,t} \left[rv_i^{max}(1-\gamma_{i,t+p_{i,PIII}}) x_{i,PIII,t\sigma} - \delta_i (z_{i,PII,t\sigma} + z_{i,PIII,t\sigma}) \right] (\sigma) \tag{57.13}$$

Because positive cash flow occurs at the completion of PIII clinical trial and the time horizon is finite, the optimization will lead to solutions with empty pipelines towards the end of the horizon. At any stage, there are four *states* a drug can be in: successfully completed, failed, idle, or undergoing a clinical trial. To simulate an infinite horizon, it is necessary to approximate the revenue achieved beyond the end of the time horizon, for which only the last two states are relevant. To do this, we introduce three parameters, pr_{ijt}^{run}, pr_{ij}^{open} and ε_{ij}, defined in Sect. 57.5.1. The first two are approximations of the revenue that will be received from the successful completion of drugs beyond the end of the planning horizon, while the last one is a discount factor to favor the completion of drugs within the planning horizon. The anticipated future revenues, FR_σ, can then be calculated by:

$$FR_\sigma = \sum_{i \in I_\sigma} \left[\sum_j \varepsilon_{ij} pr_{ij}^{open} z_{ijT\sigma} + \sum_{j \neq PIII, t > T - p_{ij}} \varepsilon_{ij} pr_{ijt}^{run} x_{ijt\sigma} \right] \quad (\sigma) \tag{57.14}$$

where the first (second) term represents revenues from drugs that are idle (undergoing trials) at the end of the time horizon.

The objective function, the expected net present value (ENPV) of the R&D pipeline, can then be calculated as follows:

$$ENVP = \sum_\sigma \pi_\sigma NPV_\sigma = \sum_\sigma \pi_\sigma \left(RV_\sigma + FR_\sigma - CST_\sigma \right) \tag{57.15}$$

with π_σ being the probability of scenario σ occurring.

The multi-stage SP model for the scheduling of clinical trials consists of Eqs. (57.8)–(57.15) plus nonanticipativity constraints. In Sect. 57.4 we show how the structure of the problem can be exploited to reduce the number of equations necessary to enforce nonanticipativity.

Finally, we note that variables $y_{ijt\sigma}$ and $z_{ijt\sigma}$ can be eliminated. The former were introduced to express precedence relationships (Eq. (57.10)) and the latter to calculate revenues (Eqs. (57.13) and (57.14)). Since both of them are defined by equalities in terms of $x_{ijt\sigma}$, we can remove them. To do this however we have to introduce Eq. (57.16) to forbid profit generating trials to be run multiple times:

$$\sum_t x_{ijt\sigma} \leq 1 \quad (i; j; \sigma) \tag{57.16}$$

57.3.2 Extensions

In pharmaceutical R&D, a company can outlicense products that have successfully passed clinical trials. Outlicensing allows the allocation of scarce resources to a subset of compounds while securing some revenue and it offers a way to reduce

risk. If $w_{ijt\sigma}$ denotes the outlicensing of i at time t after it successfully passed trial $(i, j-1)$ in scenario σ, then variable $z_{ijt\sigma}$ is calculated as follows:

$$z_{ijt\sigma} = z_{ij0} + \sum\nolimits_{t' \le t - p_{i,j-1}} x_{i,j-1,t'\sigma} - \sum\nolimits_{t' \le t} \left(x_{ijt'\sigma} + w_{ijt'\sigma} \right) \quad (i; j; t; \sigma) \tag{57.17}$$

and a revenue term $\beta_t rv_{ij}^{out} w_{ijt'\sigma}$ is added in Eq. (57.13) where rv_{ij}^{out} is the revenue from outlicensing.

Another feature that differentiates the planning of R&D activities from the traditional RCPSP is the flexibility to consider resource planning and project scheduling simultaneously since pharmaceutical companies can plan the acquisition of resources during the course of testing. Furthermore, outsourcing can be selectively used during periods of high resource demand. To account for resource planning, we introduce variable $R_{kt\sigma}$ to denote the availability of resource k and variable $R_{kt\sigma}^E$ to denote the level of expansion of resource k (reductions can be modeled similarly). The availability of resource k is then given by

$$R_{kt\sigma} = R_{k,t-1,\sigma} + R_{k,t-1,\sigma}^E \quad (k; t; \sigma) \tag{57.18}$$

If variable $R_{kt\sigma}^O$ denotes the level of outsourcing, then the resource constraint becomes:

$$\sum\nolimits_{(i,j) \in IJ_{t\sigma}} \sum\nolimits_{t'=t-p_{ij}+1}^{t'=t} r_{ijk} x_{ijt'\sigma} \le R_{kt\sigma} + R_{kt\sigma}^O \quad (k; t; \sigma) \tag{57.19}$$

The cost of resource planning, CSR_σ, is calculated as

$$CSR_\sigma = \sum\nolimits_t \sum\nolimits_k \beta_t \left(c_k^E R_{kt\sigma}^E + c_k^O R_{kt\sigma}^O \right) \quad (\sigma) \tag{57.20}$$

where c_k^E and c_k^O is the unit cost for resource expansion and outsourcing. It is then added to the calculation of the expected NPV of the pipeline

The basic model can also be extended to account for the following features (Colvin and Maravelias 2010, 2011):

1. *General activities*. In addition to clinical trials, a manufacturing process has to be developed and validated prior to the introduction of a new drug into the market. Also, in some cases, new manufacturing capacity has to be installed. This means that activities other than clinical trials have to also be taken into account. The outcome of these activities is typically deterministic, but they have resource requirements and are subject to precedence constraints with respect to clinical and non-clinical activities.
2. *Shared activities*. In pharmaceutical manufacturing it is common to produce many different drugs in the same facility. Thus, if the installation of new manufacturing capacity is considered (e.g., via expansion or retrofit of an

existing facility), then the same *CapacityInstallation* activity belongs to the sets of activities of many compounds, which makes the precedence constraints of different compounds interconnected.

3. *Uncertainty interdependence*. First, the observation of uncertainty in one task can merely change the probability of an outcome of another task because the outcomes of the two are correlated. This type of interdependence arises when compounds target similar conditions or their testing is similar. In this case, the interdependence is used to calculate the probability of each scenario, which can be performed offline, and the system can be modeled using basically the same SP model. However, if the outcome of a task can lead to changes in the way we perform tasks later (e.g., we can use our findings to redesign trials for other compounds), then the decision-maker changes the probability distributions resulting in a system under endogenous uncertainty, which is harder to address using SP methods.
4. *Operational interdependency*. This type of interdependence relates to resource use or revenue. An example would be a group of scientists and engineers able to develop processes for two similar compounds at the same time allowing for lower resource use than if the two processes were developed separately. Similarly, products can be competitive (or complementary) in the marketplace, allowing for lower (or higher) revenues if both are launched.

57.3.3 Risk Management

Risk considerations are often as important as the maximization of ENPV. In this section, we present some methods for incorporating risk into our framework.

The first approach is to limit the probability that negative revenue will be realized. To accomplish this, we introduce binary variable u_σ, which is 1 if scenario σ has an NPV below threshold γ (Eq. (57.21)), and then bound the probability of having an NPV below γ to be less than α (Eq. (57.22)), where M is a sufficiently large number:

$$NPV_\sigma \geq \gamma - u_\sigma M \quad (\sigma) \tag{57.21}$$

$$\sum_\sigma \pi_\sigma u_\sigma \leq \alpha \tag{57.22}$$

An alternate approach is to use the probability weighted sum of NPV below a threshold level γ as the downside risk (Eppen et al. 1989). To do this, we calculate (Eq. (57.23)) the shortfall, $NNPV_\sigma \geq 0$, and then constrain the weighted sum (Eq. (57.24)):

$$NNPV_\sigma \geq \gamma - NPV_\sigma \quad (\sigma) \tag{57.23}$$

$$\sum_\sigma \pi_\sigma NNPV_\sigma \leq \beta \tag{57.24}$$

Finally, we can use the concept of conditional value at risk (CVaR) which is a *coherent measure of risk*, in which the expected value of scenarios representing the worst α percent is used as the risk measure (Rockafellar and Uryasev 2002; Andersson et al. 2001). The standard CVaR methods cannot be readily used in this problem because the scenarios cannot be ordered, from the worst to the best, prior to optimization, but we can develop an approach building upon previously presented concepts (Schultz and Tiedemann 2006). We first find the variable risk threshold, *VRT*, above which at least $1-\alpha$ scenarios reside (Eq. (57.25)) and then the deviation from this threshold is calculated by Eq. (57.26):

$$NPV_\sigma \geq VRT - u_\sigma M \quad (\sigma) \tag{57.25}$$

$$NNPV_\sigma \geq VRT - NPV_\sigma \quad (\sigma) \tag{57.26}$$

Variable *VRT* may not match the value at risk (VaR), but the term $VRT - \frac{1}{\alpha}\sum_\sigma \pi_\sigma NNPV_\sigma$ will be equal to the *CVaR* in an optimal solution, so we can augment the objective function to include the risk term with weight ω

$$\max \sum\nolimits_\sigma \pi_\sigma NPV_\sigma + \omega \left(VRT - \tfrac{1}{\alpha}\textstyle\sum_\sigma \pi_\sigma NNPV_\sigma\right) \tag{57.27}$$

The above CVaR approach is based upon the fact that variable *VRT* appears in the objective function, which means that it cannot be used in instances where risk metrics are needed but their calculation cannot be enforced through the optimality of the solution; e.g., when a risk threshold is added as constraint. To calculate CVaR in this case, we introduce a second indicator variable $\hat{u}_\sigma$, with $VRU = VRT$,

$$NPV_\sigma \geq VRU - \hat{u}_\sigma M \quad (\sigma) \tag{57.28}$$

$$\sum\nolimits_\sigma \pi_\sigma \hat{u}_\sigma \geq \alpha \tag{57.29}$$

which should satisfy:

$$u_\sigma \leq \hat{u}_\sigma \quad (\sigma) \tag{57.30}$$

$$\sum\nolimits_\sigma \hat{u}_\sigma - \sum\nolimits_\sigma u_\sigma = 1 \tag{57.31}$$

57.4 Theoretical Properties

The decision-maker in this problem affects the underlying process because decisions $x_{ijt\sigma}$ determine when stochastic parameters (trial outcome) are observed, which means that nonanticipativity should be enforced via constraints in the form of Eq. (57.7). However, the structure of the problem allows us to develop properties that reduce the number of necessary NACs. Let the outcome of drug i, ξ_i, in scenario σ

be denoted by ξ_i^σ. Then we can show the following (Colvin and Maravelias 2010, 2011):

Proposition 1. *It is sufficient to express NACs only for pairs of scenarios* (σ, σ') *that differ in the outcome of a single drug; i.e.,* (σ, σ') *does not need to be pairwise constrained if* $\xi_i^\sigma \neq \xi_i^{\sigma'}$ *and* $\xi_{i'}^\sigma \neq \xi_{i'}^{\sigma'}$ *for any* $i \neq i'$. *For a constrained scenario pair* (σ, σ') *we will call the drug in which they differ the critical drug and denote it by* $i^{\sigma,\sigma'}$.

Proposition 2. *It is sufficient to express NACs only for pairs of scenarios* (σ, σ') *that differ in the outcome of a single trial; i.e.,* (σ, σ') *does not need to be pairwise constrained if* ξ_i^σ *and* $\xi_i^{\sigma'}$ *are not consecutive elements in ordered set* $\Xi_i = \{$*PI-F, PII-F, PIII-F, PIII-S*$\}$. *For the constrained scenario pair* (σ, σ') *we will call the trial in which they differ the critical trial and denote it by* $(i^{\sigma,\sigma'}, j^{\sigma,\sigma'})$. *The reduced set of scenario pairs that must be constrained is denoted by* Ψ.

Proposition 3. *For scenario pair* $(\sigma, \sigma') \in \Psi$, *it is possible to express Eq. (57.7) using decision variables present in the problem; i.e., it is not necessary to introduce binary variable* $z_{t\sigma\sigma'}$. *Specifically, we can use variable* $y_{ijt\sigma}$ *for the critical trial* $(i^{\sigma,\sigma'}, j^{\sigma,\sigma'})$.

Proposition 4. *For a given scenario pair* $(\sigma, \sigma') \in \Psi$, *uncertainty can be treated as exogenous until the earliest time,* $t_{min}^{\sigma,\sigma'}$, *these two scenarios could become distinguishable, i.e., for* $t < t_{min}^{\sigma,\sigma'} = \tau_{i^{\sigma,\sigma'},f(i)} + \ldots + \tau_{i^{\sigma,\sigma'},j^{\sigma,\sigma'}} + 1$.

Lemma 1. *Given a scenario* σ *and stage* t, *NACs between* σ *and all* $\sigma' : t < t_{min}^{\sigma,\sigma'}$ *can be expressed using NACs in the form of Eq. (57.4).*

Proposition 5. *Decision variables* $x_{ijt\sigma}$ *for trials* $(i^{\sigma,\sigma'}, f(i^{\sigma,\sigma'})), \ldots, (i^{\sigma,\sigma'}, j^{\sigma,\sigma'})$ *in scenarios* σ *and* σ' *are identical. Variables for trial* $(i^{\sigma,\sigma'}, j^{\sigma,\sigma'} + 1)$ *should not be subject to NACs.*

Proposition 6. *Let* $\sum_t^n$, $n \in N_t$ *be a maximal subset of scenarios that are indistinguishable at time* t *(i.e.,* $t < t_{min}^{\sigma,\sigma'}$ *if* $\sigma \in \sum_t^n$ *and* $\sigma' \in \sum_t^n$ *for some* n*). NACs among scenarios in* $\sum_t^n$ *can be enforced using an equality in the form of Eq. (57.5).*

When resource planning decisions are considered, NACs should also be enforced for decisions $R_{kt\sigma}$, $R_{kt\sigma}^O$, and $R_{kt\sigma}^E$. However, we can reduce the number of NACs using the following results.

Proposition 7. *If variables* $x_{ijt\sigma}$ *and* $R_{kt\sigma}$ *satisfy nonanticipativity, then outsourcing* $R_{rt\sigma}^O$ *and expansion* $R_{rt\sigma}^E$ *variables satisfy nonanticipativity in an optimal solution.*

Proposition 1 says that it is sufficient to express NACs only for pairs of scenarios that differ in the outcome of a single uncertain parameter (Colvin and Maravelias 2008). It was first proposed in the context of offshore gas field planning (Goel and Grossmann 2004, 2006). It is applicable to all SP problems, but leads to significant reductions when applied to problems under endogenous uncertainty observation.

Proposition 2 is applicable to problems where a subset of decisions has to be made in a predefined sequence and each decision in this subset can lead to the observation of a single outcome of a single uncertain parameter ξ_i. In the problem we discuss in this chapter, the clinical trials of a single drug have to be carried out in sequence which means that parameter ξ_i is observed *sequentially*.

Proposition 3 is applicable to problems where the stage at which scenarios s and σ' become distinguishable is unknown but it depends on the timing of a single decision. In the problem addressed in this chapter, the outcome of a trial can be observed only upon completion of this trial. Thus, the decision to perform a trial determines if and when a subset of scenarios will become distinguishable. Furthermore, for a given pair of scenarios in Ψ we determine that the trial that leads to their differentiation is the critical trial $(i^{\sigma,\sigma'}, j^{\sigma,\sigma'})$ which means that the main NAC can be written as follows:

$$-y_{i^{\sigma,\sigma'}j^{\sigma,\sigma'}t\sigma} \leq x_{ijt\sigma} - x_{ijt\sigma'} \leq y_{i^{\sigma,\sigma'}j^{\sigma,\sigma'}t\sigma} \quad ((\sigma,\sigma') \in \Psi; t > 1; (i,j) \in IJ_{t\sigma}) \tag{57.32}$$

Proposition 4 is applicable to problems where we can calculate the earliest stage at which scenarios can become distinguishable. It is more general than Property 2 because it is not necessary to have an one-to-one association between decisions and uncertain parameter observation. Proposition 4 implies (Lemma 1) that some of the double inequalities in Eq. (57.32) can be replaced by equalities:

$$x_{ijt\sigma} = x_{ijt\sigma}, \quad ((\sigma,\sigma') \in \Psi; 1 < t < t_{min}^{\sigma,\sigma'}; (i,j) \in IJ_{t\sigma}) \tag{57.33}$$

Proposition 5 is also applicable to problems where subsets of decisions have to be made sequentially. It implies that the inequality NACs for the critical drug of pair $(\sigma,\sigma') \in \Psi$ can be replaced with the following equality:

$$x_{i^{\sigma,\sigma'}jt\sigma} = x_{i^{\sigma,\sigma'}jt\sigma}, \quad ((\sigma,\sigma') \in \Psi; t > 1; (i^{\sigma,\sigma'}, j) \in IJ_{t\sigma}, j \leq j^{\sigma,\sigma'}) \tag{57.34}$$

Property 6 is the most general one; it is applicable to all SP problems with binary decision variables. It means that instead of expressing pairwise NACs, a single NAC is sufficient for all scenarios in $\sum_t^n$.

Finally, it can be shown that Proposition 7 is valid in any feasible solution where NACs are satisfied for variables $x_{ijt\sigma}$ and $R_{kt\sigma}$, and the remaining variables are obtained by solving an LP. Thus, if variables $x_{ijt\sigma}$ and $R_{kt\sigma}$ satisfy nonanticipativity in an integer feasible solution, then a feasible solution where all variables satisfy nonanticipativity can be obtained by simply fixing variables $x_{ijt\sigma}$ and $R_{kt\sigma}$ and solving an LP, so no NACs need to be expressed for $R_{kt\sigma}^O$ and $R_{kt\sigma}^E$.

Propositions 1–6 lead to a dramatic reduction in the number of NACs. For example, the number of NACs for an instance with six drugs is reduced from $\sim 1.2 \cdot 10^9$ to $\sim 1.5 \cdot 10^6$. Furthermore, Propositions 1–6 result in an almost linear, instead of quadratic, growth of NACs in the number of scenarios.

57.5 Solution Methods

Despite the reductions based on the theoretical properties discussed in the previous section, the size of the models that have to be generated to address realistic instances with more than ten drugs in various stages of development remain prohibitively large, so additional solution methods are required. In this section, we outline three such methods.

57.5.1 Infinite Horizon Approximations

In general, the planning horizon should be sufficiently long so that the testing of all drugs can be completed. If a drug cannot be completed within the horizon, then its testing will never be started since this would lead to costs but no revenues. However, using a long planning horizon increases the number of stages and thus the size of the formulation. To address this challenge, we formulate our model over a *medium* horizon but augment it with an approximation of the *future* effect of our decisions. We achieve this by approximating the costs that will incur and the revenues that will be generated beyond the horizon due to our decisions within the horizon.

First, we calculate an approximation of the profit pr_{ij}^{open} that would be generated by a drug that has passed trial (i, j) but is currently idle:

$$pr_{ij}^{open} = rv_i^{max}\left[1 - \gamma_{i,T+\sum_{j'\geq j} p_{ij'}}\right] - \sum\nolimits_{j'\geq j} \beta_T c_{ij}$$

The two terms inside the brackets approximate the revenues from the successful development of drug i, assuming it will be finished at $T + \sum_{j'\geq j} p_{ij'}$, i.e., there will be no delay in its development beyond the time horizon [delays within the horizon are accounted for in Eq. (57.13)]. The summation in the right-hand side is an approximation of the fixed costs for the remaining trials discounted at $t = T$.

Second, we approximate the profit pr_{ijt}^{run} that will be generated by a drug that is undergoing PI or PII clinical testing at the end of the horizon. If the trial (i, j) started at stage $t > T - p_{ij}$ then it will finish at $t + p_{ij} > T$, and PIII will be completed at $t + \sum_{j'\geq j} p_{ij'}$ assuming again no delay.

$$pr_{ijt}^{run} = rv_i^{max}\left[1 - \gamma_{i,t+\sum_{j'\geq j} p_{ij'}}\right] - \sum\nolimits_{j'>j} \beta_T c_{ij'}$$

Finally, to avoid solutions where drugs are not developed despite resource availability because their approximated potential profit is comparable to the actual profit that would have been achieved if they were developed within the planning horizon, we multiply pr_{ij}^{open} and pr_{ijt}^{run} with $\varepsilon_j \in [0.8, 0.9]$ in Eq. (57.14).

57.5.2 Rolling Horizon Using Relaxed Model

In general, accounting for uncertainties which will be realized in the future and modeling our recourse actions is beneficial because it can lead to better *early* decisions. The solution of our model therefore has to be feasible and (near) optimal over a few *early* stages, the decisions for which will be implemented. Based on this observation, we formulate a relaxed model (RM) consisting of Eqs. (57.8)–(57.15); all equality NACs, Eqs. (57.33)–(57.34); and inequality NACs, Eq. (57.32) only for the first t^* stages. The solution of (RM) is feasible for the first t^* stages because all NACs are expressed for $t \in \{1, \ldots, t^*\}$, and it yields (near) optimal solutions. Therefore, we use model (RM) in an iterative solution method that typically yields the optimal solution: solve model (RM) for $\mathscr{T} = \{1, 2, \ldots, T\}$; implement the solution for $t = 1$; taking into account uncertainty observation at $t = 1$ reformulate (RM) for $\mathscr{T} = \{1, 2, \ldots, T + 1\}$ and resolve; repeat as necessary.

57.5.3 Branch-and-Cut Algorithm

Even though none of the NACs is redundant, only a small fraction of these constraints are active in any feasible solution. Specifically, it can be shown that at most 12.5 % of inequality NACs can be violated in any solution, and in practice less than 4 % of these constraints were violated (Colvin and Maravelias 2010). This observation allows us to develop a branch and cut (B&C) algorithm in which the initial formulation includes only equality NACs and inequality NACs are added only if violated. Note that unlike typical B&C algorithms, we remove *essential*, not tightening constraints, which allows for integer solutions that are infeasible to the full model to be found feasible. To handle this, a number of modifications to the standard B&C algorithm are made. First, heuristics are turned off to prevent infeasible solutions to be found and used as lower bound for pruning. Second, bound updating is modified so the lower bound is not updated immediately upon finding an integer solution, as the solution must first be checked for NACs feasibility. Finally, if an integer solution is found to violate removed NACs, it is resolved after the addition of violated NACs (possibly leading to a fractional solution), and subsequently partitioned using standard branching.

The algorithm can be further improved by developing specialized node selection rules. Note that the advantage of local search of having the basis of the previous node is diminished because hundreds or thousands of violated NACs are added, especially in early nodes. In later nodes where fewer NACs are added and a lower bound is available, the advantages of local search are important. We found that using a best first search for a fixed number of nodes, N^{max}, before using local search provided for the fasted solution times.

Related to the node selection rules is the decision to test for violated NACs at all nodes or at integer nodes only. Checking for infeasibilities at all nodes increases the

computational cost per node substantially, but reduces the total cut additions because a cut added at an early fractional node is carried to all descendent nodes. Testing for infeasibilities at every node outperformed testing only at integer solutions; however, when used in conjunction with the hybrid node selection rule, adding cuts at all early nodes, but only at integer nodes after a fixed depth, D^{max}, performed better than either pure approach.

The proposed B&C algorithm allows us to solve problems with up to ten drugs. It yields solutions faster and, most importantly, allows us to generate and solve models for problems that were intractable because the corresponding models could not be generated.

57.6 Examples

We present two examples to show how the proposed framework can be applied. Problem data can be found in Colvin and Maravelias (2011).

57.6.1 Resource and Revenue Interdependence

We consider an example with *deterministic* tasks performed in parallel with *stochastic* tasks and interdependencies (modified from De Reyck and Leus 2008). We develop two products, D1 and D2, which require two resources: one consumable and one renewable. Both drugs have the precedence graph shown in Fig. 57.3. Each product has to pass six tasks with stochastic outcome leading to 12 total outcomes for each drug and 144 total scenarios. Additionally, if task Tox II is performed simultaneously with task Other II, then the combined resource requirements are reduced (resource interdependence); and additional revenue is gained if both drugs are launched. The model has 30 3-month stages (periods) and consists of 272,391 constraints, and 7,344 continuous and 63,504 binary variables. It was solved to 0.1 % optimality in 3,461 CPU seconds and 823 nodes.

Figure 57.4 shows the Gantt charts of two representative scenarios. The first scenario represents the case in which D1 can be successfully launched while D2

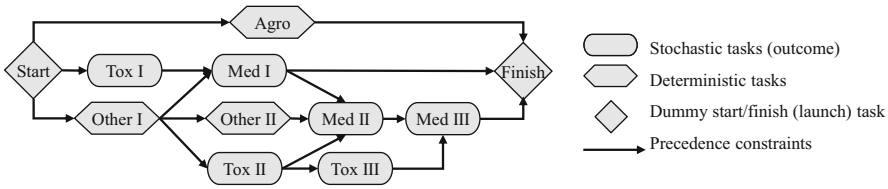


Fig. 57.3 Activity-on-node precedence graph for example in sect. 57.6.1

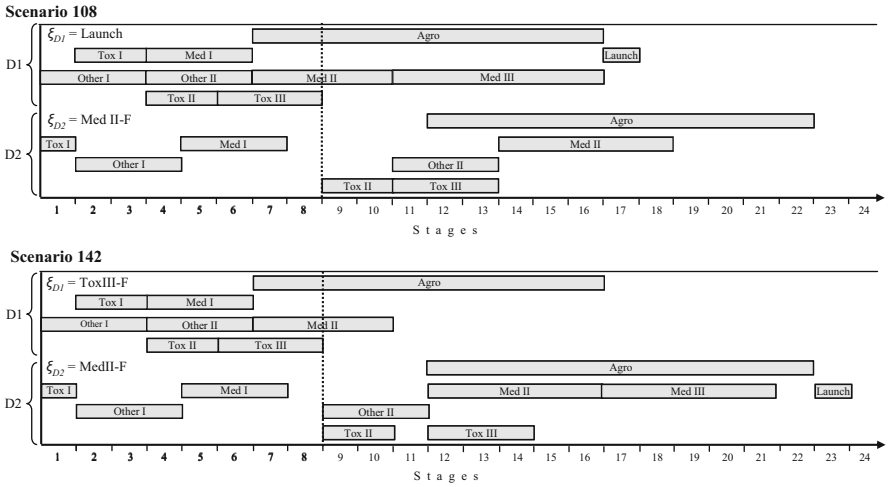


Fig. 57.4 Gantt chart of representative scenarios for Example 1 (*dashed line* represents the point at which the two scenarios become distinguishable)

fails Med II. The second scenario represents the case in which D1 fails Tox III, but D2 reaches the market. As shown in Fig. 57.4, the expensive and long Agro task is delayed until after much of the uncertainty is resolved without delaying the launch of the drug. Also, Tox II and Other II start simultaneously in both scenarios, exploiting the interdependency in resource requirements. Additionally, D1 is developed as quickly as possible while D2 is delayed due to resource constraints.

57.6.2 Risk Management

In this example, we examine how the different risk management approaches affect the solution. We consider a problem with four drugs having to undergo three clinical trials resulting in 256 scenarios, and a 48-month time horizon divided into eight 6-month stages. All drugs can be outlicensed after successful completion of PI or PII clinical trials. We studied four approaches: (1) no risk management; (2) the probability of incurring a loss is limited to 20 % using Eqs. (57.21)–(57.22); (3) the downside risk is limited to under \$35M using Eqs. (57.23)–(57.24); and (4) the CVaR formulation with $\alpha = 5\,\%$ using Eqs. (57.22), (57.25)–(57.27). Figure 57.5 shows the breakdown of revenue coming from outlicensing, completed drugs, and anticipated revenue from drugs under development at the end of the horizon.

If risk is not considered, no drug was outlicensed, leading to over 40 % of scenarios having a negative NPV, but the expected NPV is \$1,835M. The expectation of scenarios with negative NPV was found to be −\$50M.

In the probabilistic approach, the probability of realizing a negative NPV was limited to 20 %, approximately the probability of all drugs failing. In this case,

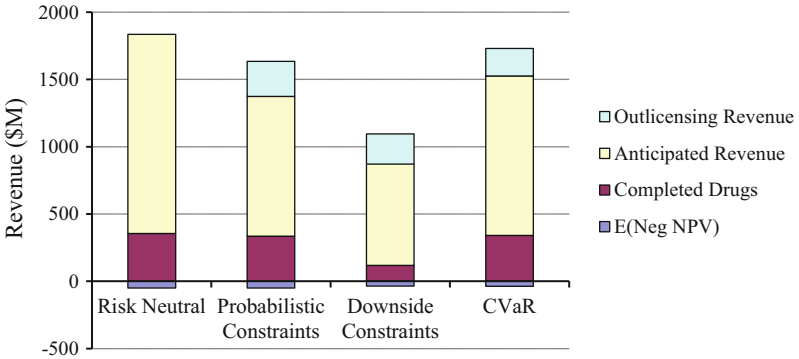


Fig. 57.5 Breakdown of revenue sources for different risk management solutions

not all drugs are developed and some outlicensing occurs. The development of drugs completed within the time horizon is similar to the instance with no risk management, but drugs that were anticipated to be completed in the future were outlicensed for a secure revenue source lowering the expected NPV to $1,615M.

In the third case, the downside risk was limited to roughly the expected cost of developing a single drug. This approach yielded the greatest changes. PI trials were performed for multiple drugs initially. If any drugs successfully reached PII clinical trials, at least one of them was outlicensed or the development was stopped, which reduced the number of trials performed and thus the revenue from completed drugs and anticipated future revenues. Also, lower outlicensing revenue was realized due to selling the drugs earlier (after completion of PI instead of PII trials). As a result, the expected NPV dropped to $1,095M with a 50 % probability of realizing a negative NPV.

Finally, using CVaR (with risk level of 5 %) led to a solution between the risk neutral and the probabilistically constrained approach with an expected NPV of $1,740M and a CVaR of −$101M. Similar to the probabilistically constrained model, future *uncertain* revenues were traded for smaller but *secure* outlicensing revenues, though this occurred more often when multiple drugs reached PIII testing. Roughly 25 % of scenarios had a negative NPV with probability-weighted loss of $50M.

57.7 Conclusions

In this chapter, we presented a multi-stage stochastic programming framework for the planning of R&D activities in the pharmaceutical industry. Also, we discussed several extensions, including risk management approaches. Finally, we presented theoretical properties that allow us to formulate tractable models, and solution methods that allow us to address realistic instances.

References

Andersson F, Mausser H, Rosen D, Uryasev S (2001) Credit risk optimization with conditional value-at-risk criterion. Math Program 89(2):273–291

Birge JR, Louveaux F (1997) Introduction to stochastic programming. Springer, New York

Bottcher J, Drexl A, Kolisch R, Salewski F (1999) Project scheduling under partially renewable resource constraints. Manag Sci 45(4):543–559

Bouleimen K, Lecocq H (2003) A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. Eur J Oper Res 149(2):268–281

Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999) Resource-constrained project scheduling: notation, classification, models, and methods. Eur J Oper Res 112(1):3–41

Charalambous C, Gittins JC (2008) Optimal selection policies for a sequence of candidate drugs. Adv Appl Probab 40(2):359–376

Choi J, Realff MJ, Lee JH (2004) Dynamic programming in a heuristically confined state space: a stochastic resource-constrained project scheduling application. Comput Chem Eng 28(6–7): 1039–1058

Colvin M, Maravelias CT (2008) A stochastic programming approach for clinical trial planning in new drug development. Comput Chem Eng 32(11):2626–2642

Colvin M, Maravelias CT (2009) Scheduling of testing tasks and resource planning in new product development using stochastic programming. Comput Chem Eng 33(5):964–976

Colvin M, Maravelias CT (2010) Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. Eur J Oper Res 203(1):205–215

Colvin M, Maravelias CT (2011) R&D pipeline management: task interdependencies and risk management. Eur J Oper Res 215(3):616–628

De Reyck B, Leus R (2008) R&D project scheduling when activities may fail. IIE Trans 40(4):367–384

DiMasi JA, Grabowski HG (2007) The cost of biopharmaceutical R&D: is biotech different? Manag Decis Econ 28(4–5):469–479

Eppen GD, Martin RK, Schrage L (1989) A scenario approach to capacity planning. Oper Res 37(4):517–527

Gatica G, Papageorgiou LG, Shah N (2003) Capacity planning under uncertainty for the pharmaceutical industry. Chem Eng Res Des 81(A6):665–678

Goel V, Grossmann IE (2004) A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. Comput Chem Eng 28(8):1409–1429

Goel V, Grossmann IE (2006) A class of stochastic programs with decision dependent uncertainty. Math Program 108(2–3):355–394

Hartmann S (1998) A competitive genetic algorithm for resource-constrained project scheduling. Nav Res Logist 45(7):733–750

Herroelen W, Leus R (2005) Project scheduling under uncertainty: survey and research potentials. Eur J Oper Res 165(2):289–306

Jain V, Grossmann IE (1999) Resource-constrained scheduling of tests in new product development. Ind Eng Chem Res 38(8):3013–3026

Kall P, Wallace SW (1994) Stochastic programming. Wiley, Chichester

Kolisch R, Hartmann S (2006) Experimental investigation of heuristics for resource-constrained project scheduling: an update. Eur J Oper Res 174(1):23–37

Levis AA, Papageorgiou LG (2004) A hierarchical solution approach for multi-site capacity planning under uncertainty in the pharmaceutical industry. Comput Chem Eng 28(5):707–725

Maravelias CT, Grossmann IE (2004) Optimal resource investment and scheduling of tests for new product development. Comput Chem Eng 28(6–7):1021–1038

Mercier L, Van Hentenryck P (2008) Amsaa: a multistep anticipatory algorithm for online stochastic combinatorial optimization. In: Proceedings of the 5th international conference on integration of AI and OR techniques in constraint programming for combinatorial optimization problems, Paris. LNCS, vol 5015, Springer, New York, pp 173–187

Merkle D, Middendorf M, Schmeck H (2002) Ant colony optimization for resource-constrained project scheduling. IEEE Trans Evol Comput 6(4):333–346

Mika M, Waligóra G, Węglarz J (2005) Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models. Eur J Oper Res 164(3):639–668

Neumann K, Zimmermann J (2000) Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. Eur J Oper Res 127(2):425–443

Rockafellar RT, Uryasev S (2002) Conditional value-at-risk for general loss distributions. J Bank Finance 26(7):1443–1471

Rogers MJ, Gupta A, Maranas CD (2002) Real options based analysis of optimal pharmaceutical research and development portfolios. Ind Eng Chem Res 41(25):6607–6620

Schmidt CW, Grossmann IE (1996) Optimization models for the scheduling of testing tasks in new product development. Ind Eng Chem Res 35(10):3498–3510

Schultz R, Tiedemann S (2006) Conditional value-at-risk in stochastic programs with mixed-integer recourse. Math Program 105(2–3):365–386

Shah N (2004) Pharmaceutical supply chains: key issues and strategies for optimisation. Comput Chem Eng 28(6–7):929–941

Solak S, Clarke JPB, Johnson EL, Barnes ER (2010) Optimization of R&D project portfolios under endogenous uncertainty. Eur J Oper Res 207(1):420–433

Souder WE, Mandakovic T (1986) Research-and-development project selection models. Res Manag 29(4):36–42

Steele LW (1988) What we've learned: selecting R&D programs and objectives. Res Technol Manag 31(2):17–36

Stonebraker JS (2002) How Bayer makes decisions to develop new drugs. Interfaces 32(6):77–90

Subramanian D, Pekny JF, Reklaitis GV (2001) A simulation-optimization framework for research and development pipeline management. AlChE J 47(10):2226–2242

Subramanian D, Pekny JF, Reklaitis GV, Blau GE (2003) Simulation-optimization framework for stochastic optimization of R&D pipeline management. AlChE J 49(1):96–112

Tormos P, Lova A (2001) A competitive heuristic solution technique for resource-constrained project scheduling. Ann Oper Res 102:65–81

Valls V, Ballestín F, Quintanilla S (2005) Justification and RCPSP: a technique that pays. Eur J Oper Res 165(2):375–386

Part XVIII
Case Studies in Project Scheduling

Chapter 58
Robust Multi-Criteria Project Scheduling in Plant Engineering and Construction

Maurizio Bevilacqua, Filippo E. Ciarapica, Giovanni Mazzuto, and Claudia Paciarotti

Abstract In this chapter we consider a real-life problem which consists in scheduling the activities of a project subject to precedence and resource constraints so as to optimize several conflicting goals. The durations of the activities cannot be specified precisely in advance. Rather, we assume that based on the experience with previous projects, the means, the standard deviations, and certain percentiles of the respective probability distributions can be reliably estimated. The algorithm applied to solve this problem relies on goal programming techniques in conjunction with Goldratt's Critical Chain Project Management (CCPM) method. The algorithm was applied to a case study dealing with the construction of an accommodation module for an oil rig. Goal programming is a multi-objective programming technique which attempts to minimize the deviations to a set of target values for the given objectives in such a way that all operational restrictions of the problem are satisfied. Several solutions can be obtained, and the best solution will depend on the priority associated to each goal. In this work we considered the minimization of the project makespan and the levelling of the project resources as the objectives to be pursued. The results obtained using the proposed algorithm have been compared with classical project management techniques (PERT/CPM) that the company involved in the case study used in many projects.

Keywords Case study • Critical chain • Goal programming • Robust project scheduling

M. Bevilacqua (✉) • F.E. Ciarapica • G. Mazzuto • C. Paciarotti
Department of Industrial Engineering and Mathematical Science,
Marche Polytechnic University, Ancona, Italy
e-mail: m.bevilacqua@univpm.it; f.ciarapica@univpm.it; g.mazzuto@univpm.it; c.paciarotti@univpm.it

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_28

58.1 Introduction

Critical Chain Project Management (CCPM) considers both the stochastic approach and resource constraints (Goldratt 1964). This method, in our opinion, is still a topic that needs to be further researched. One of the problems to be solved is the definition of activity priorities. By changing the priority of activities the project duration will change. The problem of CCPM scheduling has been highlighted recently by Long and Ohsato (2008). They developed a fuzzy Critical Chain method trying to incorporate some beneficial features of CCPM for project scheduling under resource constraints and uncertainty. Bevilacqua et al. (2009) proposed a new method based on risk assessment in order to minimize the risks related to accidents in the refinery plants.

The procedure proposed in this chapter uses the goal programming (GP) technique to tackle this problem. The proposed method for project scheduling was applied to a case study considering the construction of an accommodation module for an oil rig.

Goal programming is a well-known technique for decision-makers (DMs) to solve multi-objective decision making (MODM) problems in finding a set of satisfying solutions. It was first introduced by Charnes and Cooper (1961) and further developed by Lee (1972), Ignizio (1976), and Tamiz et al. (1998). The purpose of GP is to minimize the deviations between the achievement of goals and their aspiration levels. The possibility of defining multiple goals from the mathematical point of view can help project managers in their work.

Some GP applications deal with project selection problems. An example is the work of Kim and Emery (2000), who use GP for project selection and the resources levelling. Their GP model has been developed to determine which programs to pursue in an effort to maximize profit over a 4-year period, develop machine procurement plans, and estimate personnel requirements.

Azaron and Moghaddam (2006) develop a multi-objective model for the resource allocation problem in a dynamic PERT network, where the activity durations are exponentially distributed random variables and the new projects are generated according to a Poisson process (see also Chap. 38 of this book). However, they note that the limitation of this model is that the state space can grow exponentially with the size of the network. Azaron et al. (2006) consider the case of Erlang distributed durations, leading to the same problem.

In the present chapter a new approach of GP has been developed for CCPM, defining an algorithm for the activities priority in order to minimize the project makespan (i.e., the project duration) and to level the resource usage over time (i.e., resource levelling).

The remainder of this chapter is organized as follows. Section 58.2 gives a short introduction to basic CCPM and GP concepts. The procedure applied in this paper is described in Sect. 58.3. In Sect. 58.4 we present the case study. The results obtained with the approach devised in this chapter are compared with schedules generated by other techniques to assess the validity of the new procedure. This comparison is discussed in Sect. 58.5. Finally, conclusions are drawn in Sect. 58.6.

58.2 Methodological Background

In order to explain the procedure developed in this chapter for defining activity priorities and scheduling the project, we provide some basic concepts of Critical Chain Project Management and goal programming.

58.2.1 Critical Chain Project Management

Our procedure for scheduling a project relies on the Critical Chain Project Management (CCPM) method. CCPM is based on algorithms derived from the Theory of Constraints (TOC), which has been devised by Goldratt (1997). The name TOC refers to the hypothesis that any manageable system is limited in achieving its goals by a very small number of constraints and that there always exists at least one limiting constraint. Specifically, a constraint is anything that prevents the system from improving the achievement of one or several goals.

Shou and Yao (2000) underlined that the Critical Chain planning method was developed as a result of the chronic problems that existing methods present. Goldratt (1997) argues that safety time and its use is of crucial importance in project management. PERT/CPM is criticized as dealing with uncertainty in the same way for all activities, even if they are not on the critical path. In the PERT/CPM approach a safety time is added at the end of each activity, while in the Critical Chain approach the safety times, which are viewed as "padding", are aggregated, adjusted, and relocated in strategic positions to protect the overall critical chain. Time estimates may be reduced, but a project buffer is added at the end of the project. This has the effect of reducing the length of the critical path and hence the overall project duration (Watson et al. 2007).

Goldratt (1997) proposed to concentrate the uncertainty for each activity at the end of the project in a buffer. His method is based on the property that the variance of the sum of samples from a number of independent distributions is the sum of variances for the populations from which samples come. The variance is the square of standard deviation. The standard deviation is proportional to the amount of variation in a single activity. In other words, the uncertainty in the amount of activity in the critical chain is the square root of the sum of the squares of individual variations.

To determine the critical activities of a project, the Critical Chain method requires the priority definition of activities. Considering the example in Fig. 58.1, the problem consists of defining a criterion for the optimal scheduling of the activities (the set of critical activities immediately results from the generated schedule). We assume that resource Y cannot process both activities B and D in parallel. Using backward scheduling, when the activity E can start, it is important to determine the activity that is completed immediately before. We show that the sequencing of activities B and D significantly affects the duration of the critical chain. If activity B

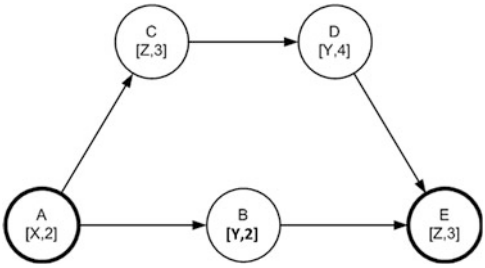


Fig. 58.1 Example (nodes are labelled with activity name [type of resource, duration])

has a lower priority than D, D will start before B and the critical chain is composed by all activities, leading to a project duration of 14 time units (activities sequence: ACDBE). Conversely, if D has a lower priority than B, the critical chain is composed of the activities A, C, D, and E with a total duration of 12 time units. This simple example (with only five activities) shows that it is necessary to develop a reliable tool that can help the project manager during the scheduling process.

58.2.2 Goal Programming

Goal programming (GP) is a pragmatic and flexible methodology capable of addressing complex decision problems where several objectives as well as many variables and constraints are involved (Tamiz et al. 1998). The approach of GP relies on Simon's (1955) concept of satisfying objectives.

According to Chap. 20 in the first volume of this handbook, in its basic form, the GP methodology uses the same structure as linear programming (LP), but it aims at optimizing a set of objective functions. To put it differently, it optimizes the following function defined as the vector of the ν objective functions:

$$f\ (x) = \begin{cases} f_1(x_1, x_2, \ldots, x_n) \\ \qquad \ldots \\ f_\nu(x_1, x_2, \ldots, x_n) \end{cases} \tag{58.1}$$

Extending the hypothesis of linearity (derived from LP) to the objective function vector, the value z_μ of the μ-th objective function can be written as

$$z_\mu = \sum_{i=1}^{n} c_{\mu i} x_i \quad (\mu = 1, \ldots, \nu) \tag{58.2}$$

where $c_{\mu i}$ is the coefficient of the objective function that represents the marginal contribution in reaching the μ-th target by the i-th decision variable x_i. μ denotes the index of objective functions and n is the number of decision variables. Each objective function in (58.1) is associated with a target value z_μ that indicates a realistic value to be reached for the μ-th objective. By combining an objective

with a target value we obtain a goal (Romero and Rehman 1989). Goals can be considered "soft constraints" that can be violated without producing infeasible solutions. The amount of deviation from the given target value expresses the under/overachievement of the target. That is why the deviation, δ_μ, can be positive (overrunning z_μ) or negative (failing z_μ). Hence, the deviation is usually represented as the difference of two nonnegative deviational variables: one variable for the case of overachievement, δ_μ^+, and one variable for the case of underachievement, δ_μ^-.

The μ-th goal of can now be expressed as follows:

$$z_\mu = \sum_{i=1}^{n} c_{\mu i} x_i + \delta_\mu^- - \delta_\mu^+ \quad (\mu = 1, \ldots, \nu) \tag{58.3}$$

The most widely used GP formulation of the problem is given by the minimization of Eq. (58.4) subject to the constraints (58.3) and (58.5)–(58.7):

$$z_{GP} = \sum_{\mu=1}^{\nu} (\delta_\mu^- + \delta_\mu^+) \tag{58.4}$$

$$\sum_{i=1}^{n} a_{ik} x_i \leq R_k \quad (k \in \mathscr{R}) \tag{58.5}$$

$$x_i^{min} \leq x_i \leq x_i^{max} \quad (i = 1, \ldots, n) \tag{58.6}$$

$$\delta_\mu^- \geq 0, \quad \delta_\mu^+ \geq 0 \quad (\mu = 1, \ldots, \nu) \tag{58.7}$$

In a production-theoretical context,the parameters of the problem can be interpreted as follows: a_{ik} is the production coefficient, which represents the quantity of resource $k \in \mathscr{R}$ used by the production process i; R_k denotes the limited availability of resource k; x_i^{min} and x_i^{max} are the minimum and the maximum values, respectively, of decision variable x_i; n designates the number of original decision variables.

Deviational variables and goals have the same relative importance in the GP model, whereas decision makers usually face the need to assign them different priorities. This consideration is very important considering that GP models can be classified into three major subsets:

1. Weighted Goal Programming (WGP), see Ballarin et al. (2011);
2. Minimax Goal Programming (MINMAX GP), see Inuiguchi and Sakawa (1995), Yang (2000), or Romero and Rehman (1989);
3. Lexicographic Goal Programming(LGP).

In our study the LGP approach is used. The rationale behind LGP is based on the observation that in some decision making systems some goals seem to prevail. Preemptive weights are attached to the goals, which are classified in different priorities. The procedure begins with comparing all the alternatives with respect to the highest priority goals and continues with the next priority until only one alternative is left. In other words, the fulfilment of goals that are rated with a certain priority is immeasurably preferable to the achievement of any other set of goals rated with a lower priority (Romero and Rehman 1989).

58.3 Research Approach

In this section we explain how we have approached the real-life project planning problem described in the case study of Sect. 58.4. First we briefly sketch the general project planning procedure and then provide a goal programming formulation of the bi-objective scheduling problem.

58.3.1 Overview of the Project Planning Approach

The proposed procedure is shown in Fig. 58.2.

Firstly, it is necessary to decompose the project into a set of individual activities. Subsequently, all constraints with respect to resources, activity durations, and precedence relationships among the activities have to be analyzed.

Having identified the constraints, it is then possible to formalize the planning goals, giving rise to the objective functions of the project scheduling problem. In this work we consider two conflicting criteria: the minimization of the project makespan and the levelling of resources. These objective functions will be explained

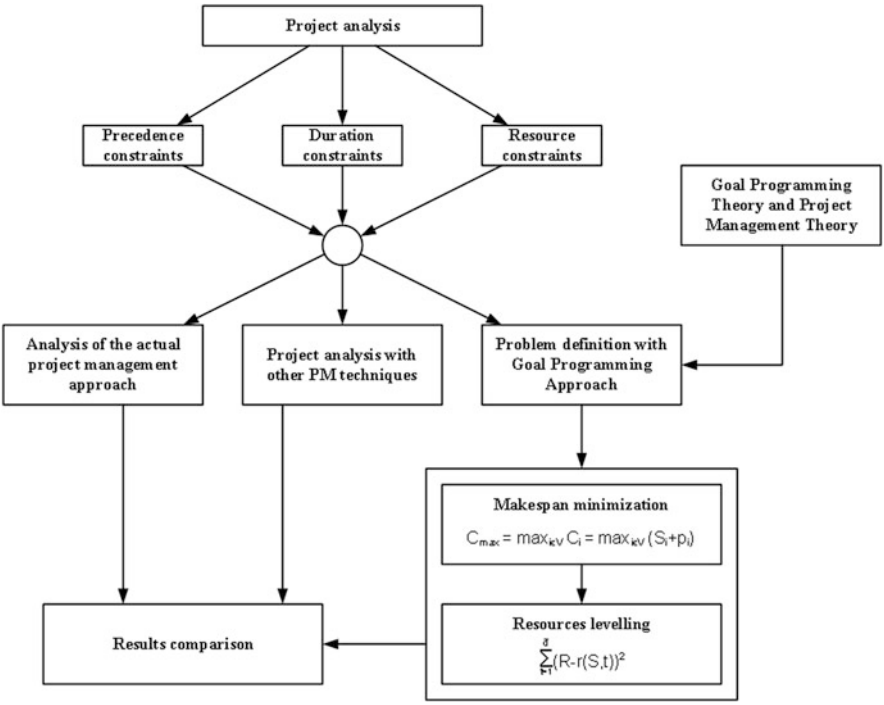


Fig. 58.2 Proposed approach

in more detail in Sect. 58.3.2, where we propose a goal programming formulation for resource-constrained project scheduling.

58.3.2 Goal Programming Formulation

Having analyzed the project with all its constraints regarding precedencies, durations, and resources, it is possible to formulate the GP problem. As mentioned before, we define two goals for the GP problem:

1. Minimization of the project makespan;
2. Levelling of resources;

These two goals have been chosen because of their importance in terms of efficiency. The project manager of the case study presented in Sect. 58.4 decided that the first goal should have higher priority than the second one.

Regarding the first goal and denoting the project duration by C_{max}, the objective function can be written as follows:

$$C_{max} = \max_{i \in V} C_i = \max_{i \in V}(S_i + p_i) \tag{58.8}$$

where V is the set of all activities and S_i, C_i, and p_i respectively designate the starting and completion times of activity i and its duration. Duration p_i may vary in interval $\left[\, p_i^{min},\ p_i^{max} \,\right]$ with lower limit p_i^{min} and upper limit p_i^{max}; the corresponding constraint is formulated in Eq. (58.9):

$$p_i^{min} \leq p_i \leq p_i^{max} \qquad (i \in V) \tag{58.9}$$

Moreover, all the technological constraints must be taken into account. Considering the starting time S_j of activity j and the starting times S_i of all activities i that are predecessors of j, it is possible to write the precedence constraints as in Eq. (58.10):

$$S_j \geq S_i + p_i \qquad (j \in V;\ i \in Pred(j)) \tag{58.10}$$

where $Pred(j)$ stands for the set of all predecessors of activity j.

The last constraints are given considering a finite set of resources $k \in \mathscr{R}$. In fact, if $r_k(S,t)$ is the total demand of the resource k at day t and $R_k(t)$ is the maximum availability of k at t, then the constraints may be written as Eq. (58.11):

$$r_k(S,t) \leq R_k(t) \qquad (k \in \mathscr{R};\ t = 1, \ldots, \bar{d}) \tag{58.11}$$

where $\bar{d}$ is some deadline for the project termination and

$$r_k(S,t) = \sum_{i \in \mathscr{A}(S,t)} r_{ik} = \sum_{i \in \mathscr{A}(S,t)} \frac{wl_{ik}}{p_i} \tag{58.12}$$

In Eq. (58.12) r_{ik} denotes the requirement of activity i for resource k and $\mathscr{A}(S,t)$ is the set of activities being in progress at time t. Since we assume that the durations of activities i can be varied in the given interval, r_{ik} is expressed as the ratio of the given workload $wl_{ik} = p_i \cdot r_{ik}$ of resource k incurred by activity i and the (variable) duration of activity i.

Obviously, the starting time of each activity i must be chosen within the time window between its earliest and latest starting times ES_i and LS_i arising from deadline $\bar{d}$ for the completion of the project and the classical temporal scheduling computations in the project network:

$$ES_i \leq S_i \leq LS_i \quad (i \in V) \tag{58.13}$$

Regarding the second goal, the resource levelling techniques aim at minimizing the variations in the resource loads over time by shifting activities according to their time windows (see, e.g., Chap. 17 in the first volume of this handbook). The model determines the starting times of non-critical activities subject to the given deadline of the project by optimizing its objective function. A well-suited objective function to be minimized for this aim is given in Eq. (58.14):

$$\sum\nolimits_{t=1}^{\bar{d}} (R - r(S,t))^2 \tag{58.14}$$

where R and $r(S,t)$ are the availability of man-hours for each considered period and the amount of man-hours that is required at time t. This problem is subject to the following constraints:

$$S_i + p_i \leq \bar{d} \quad (i \in V) \tag{58.15}$$

$$S_j \geq S_i + p_i \quad (j \in V; i \in Pred(j)) \tag{58.16}$$

$$S_i \geq 0 \quad (i \in V) \tag{58.17}$$

This model and the steps of our scheduling procedure have been implemented using the Matlab R2011 software.

58.4 Case Study: The Halfdan Northeast Field Project

The case study examined in this chapter deals with the timing and scheduling of activities for the construction of a module for an oil rig project located in Denmark.

The site consists of three plants: Center Dan, Halfdan and Tyra West. The plant is shown in Fig. 58.3, where the ellipse highlights the Halfdan structure that is analyzed in this study.

The Halfdan structure is composed of three main modules: a jacket type monopalo, a top side used for production, and a topside accommodation module. Across the riser inside of the jacket, deck hydrocarbons arrive at HCA. Here the

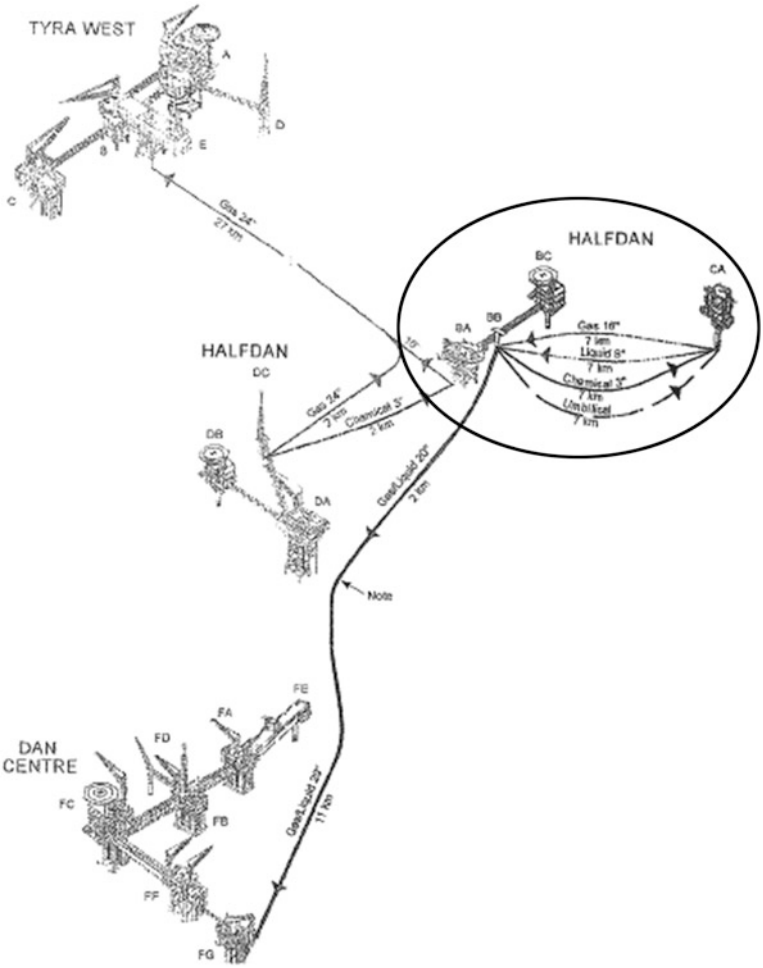


Fig. 58.3 Extraction field plant

activities associated with the separation of gases by water are performed, and then, using two subsea pipelines of 7 km length each, the gas and the liquid are sent to the module and then to the platform HBB Dan Centre. The accommodation module houses the management and maintenance personnel of the extraction Halfdan Field.

58.4.1 Project Analysis

The realization of the project can be subdivided into three sub-projects, one for each module. However, the three modules have to be realized in parallel because the

Table 58.1 Phases and activities of the Halfdan project

Activity	No.	Activity	No.	Activity	No.
Engineering phase		**Construction phase**		**Construction phase**	
Start project	1	Structure	11	Insulation	21
Structure	2	Painting in shed	12	Plumbinginstalling	22
Architecture	3	Assembling	13	Fire and safety installing	23
Electronics	4	Machinery installing	14	Commissioning	24
Piping	5	Piping installing	15	Load out	25
HVAC	6	Architecture	16	Ready to sail away	26
Equipment on board	7	HVAC installing	17		
Management		Electronics installing	18		
Strategydefinition	8	Equipment installing	19		
Supplying	9	Painting on site	20		
Subcontracting	10				

customer provides a means of on-site towing, a single pontoon. This is one of the most important constraints for the whole project. The activities for each sub-module are shown in Table 58.1.

They have been grouped into three sections: activities for the engineering phase, for the management phase and concluding, for the construction phase. The first six activities are concerned with several areas of engineering. This phase is essential for managing all operations to be performed on the construction site. The following phase is related to the procurement. The construction phase is the most interesting part of the project with respect to the planning problem and for this reason it is used for validating the procedure proposed in this chapter.

In the construction phase, the accommodation module has been identified as the critical element among the three modules, and this is due to a number of reasons:

1. There is less time available for this module than what is considered to be optimum for the fulfilment of that part of the order;
2. It requires high development efforts, both on basic and detailed engineering for all the facilities to be installed on the module;
3. It is equipped with complex and technologically advanced systems;
4. An extensive control and management of engineering, procurement, and construction activities is required.

The accommodation module must be designed completing the basic engineering and developing the detailed engineering. To have an overall understanding of the problem it is necessary to analyze the precedence and resource constraints for each activity of the project.

In Table 58.2 the precedence constraints for the construction phase are shown. Regarding the resource constraints, considering that the module construction starts when all the materials are ready to be used in the construction site, the only resource to be considered is the manpower. Table 58.2 shows the necessary resource k for

Table 58.2 Activities of the construction phase

Activity no.	Specification	Description	Resources	Demand (workers)	Successor(s)
11	Structure	External structure envelope construction	Structural engineers	305	12
12	Painting in shed	Painting of the various structural blocks inside the shed	Painters	129	13
13	Assembling	Welding of all the building blocks realized	Structural engineers	305	14, 15, 21
14	Machinery installing	Assembly of all machines for extracting and storing	Piping technicians, electricians	338	15
15	Piping installing	Installation of the piping for the extraction plant	Piping technicians	264	16, 17, 20, 22
16	Architecture	Interior design of the module	Architects	99	18, 19
17	HVAC installing	Installation of heating and conditioning systems	HVAC installers	84	18, 19
18	Electronics installing	Installation of electrical panels	Electricians	74	23
19	Equipment installing	Installation devices for the module habitableness	Instrument fitters	122	23
20	Painting on site	Further painting	Painters	129	24
21	Insulation	Installation of insulated panels to thermally isolate the interior of the building	Insulation engineers	149	24
22	Plumbing installing	Installation of the piping for housing	Plumbers	96	24
23	Fire and safety installing	Assembly and installation of the safety and firefighting systems	Plumbers, electricians	170	24
24	Commissioning	Commissioning of various equipment and systems	Commissioning	51	25
25	Load out	Module loading above of the pontoon	Commissioning	51	26
26	Ready to sail away	Order fulfilment	Commissioning	51	–

Table 58.3 Activity durations in days

Activity	Mean value	Standard deviation	50 % percentile	90 % percentile
1. Start	0.00	0.00	0	0
2. Structure	132.17	10.83	135	546
3. Painting in shed	116.92	9.58	118	402
4. Assembling	116.92	9.58	117	406
5. Machinery installing	147.42	12.08	148	718
6. Piping installing	167.75	13.75	169	979
7. Architecture	162.67	13.33	158	883
8. HVAC installing	147.42	12.08	148	698
9. Electronics installing	122.00	10.00	121	437
10. Equipment installing	101.67	8.33	101	294
11. Painting on site	61.00	5.00	61	116
12. Insulation	71.17	5.83	71	151
13. Plumbing installing	132.17	10.83	130	520
14. Fire and safety installing	96.58	7.92	97	263
15. Commissioning	50.83	4.17	51	87
16. Load out	10.17	0.83	10	11
17. Ready to sail away	5.08	0.42	5	5
18. Finish	0.00	0.00	0	0

each project activity i. Specifically, the demand r_{ik} for each resource has been given in number of required workers. The maximum availability for the number of workers is 338.

58.4.2 Critical Chain/Goal Programming Approach

The average duration and standard deviation of each activity has been estimated using company historical data stemming from 15 projects previously carried out. We assumed that all activity durations have a log-normal distribution. In Table 58.3 the 50 and 90 % percentiles of the duration distributions are reported, where the activities of the construction phase have been renumbered from 1 to 18.

The 50 % duration percentiles have been used for project planning. According to Goldratt's (1997) theory, project buffer and feeding buffers have been determined considering the 50 % percentile of the critical chain duration and all critical activities. Figure 58.4 shows the result obtained using the CCPM method and scheduling the activities according to the goal programming model.

The critical activities (the activities that compose the critical chain) are represented as red rectangles, the black rectangles corresponding to no-critical activities. Yellow rectangles are feeding buffers, while the (right-most) green rectangle shows the project buffer.

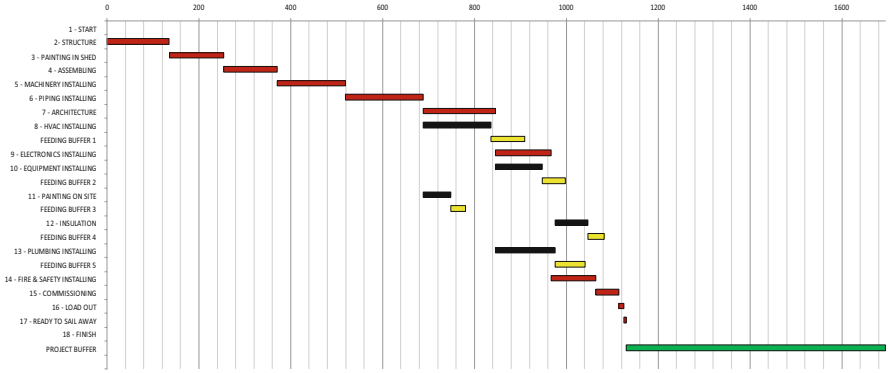


Fig. 58.4 Critical Chain program

58.5 Discussion

In order to analyze advantages and disadvantages of different techniques, the proposed method, based on CCPM in combination with GP techniques, has been compared with PERT/CPM and traditional CCPM. In doing so, the results yielded by the new approach have been compared with those the company obtained using traditional approaches. Fifteen similar projects have been compared.

An initial comparison can be made by considering the adherence to the project deadlines. As illustrated in Table 58.4 the new method allowed for a project duration prediction which was closer to the real duration. Since activity times vary depending on the availability of materials, workers, tools, and in some cases the weather, it is natural for the estimator to integrate some margin of error into the estimate. Therefore, in previous projects, it was not uncommon to have estimates reflecting 90–95 % confidence that the activity would be performed within the estimated time. Furthermore, the practice of this type of scheduling prevents project managers from taking advantage of the buffers built into the single activities. For this reason, should a critical activity exceed its estimate for completion, the entire project will be delayed. In this way variation in individual activity completion time accumulates and on-time delivery is compromised. Critical Chain Planning allows activities to be carried out according to their 50 % duration percentile, yielding considerably shorter activity durations; moreover, by using the buffers, the reliability of the completion dates has been increased and the need for frequent schedule changes has been reduced.

Moreover, a comparison was made with respect to the resource utilization (i.e., the ratio worked hours/total paid hours). The improvement shown in Table 58.4 can be explained considering that in traditional project management any delay in a critical activity will delay the project, but the opportunity to accelerate a critical activity will generally not be seized:

Table 58.4 Results obtained

	Planned duration [days] (standard deviation)	Project duration [days] (standard deviation)	Resource saturation (standard deviation)
PERT/CPM	1,978 (157)	2,286 (203)	49 % (12 %)
CCPM	1,764 (103)	1,811 (109)	62 % (9 %)
CCPM/GP	1,695	1,720	71 %

- a person working on a project might prefer to review his or her work rather than to report that is has been completed;
- in a CPM program, if a predecessor activity is completed earlier than planned, the resource required to perform a successor activity might not be ready to start at an earlier date.

58.6 Conclusions

In this chapter we proposed a model, which is able to support the scheduling of activities for the realization of a project of considerable complexity. In particular, the presented case study involved the construction of the accommodation module for the Halfdan Northeast field of oil extraction.

The combined application of the Critical Chain Project Management (CCPM) method and techniques of goal programming (GP) provided a feasible schedule of good quality within the constraints of limited resources, considering two important criteria: the minimization of the project duration and the levelling of the resources of the project in question. In this way it has been possible to comply with all technical constraints and priorities of the practical resource-constrained project scheduling problem (RCPSP).

In comparison to the traditional RCPSP methods applied to CPM/PERT networks, the method proposed in this chapter uses the CCPM scheduling logic in order to concentrate the uncertainty for each activity in a project buffer at the end of the project. This allowed the project manager to reduce project duration. This happens because the traditional CPM methods aim at bringing individual activities on time. In contrast, the proposed method is designed to produce schedules that complete the entire projects on time. The method provides a tool to proactively manage projects in such a way that harm caused by variations in activity completion times is mitigated. In the control phase of the Critical Chain method, the penetration of buffers was monitored. When the percentage of buffer penetration exceeded the relative progress of the project, project crashing was carried out until the relative progress again exceeded the proportion of the consumed buffer.

The applied method also demonstrates the potential of goal programming. In fact, like the GP of linear programming, our model is able to take into account multiple objectives simultaneously instead of one only, as well as a large number

of variables and constraints. The capability of hedging against uncertainty and to consider several goals simultaneously represent the most important advantages of this combined CCPM/GP approach. In particular, the presented approach serves to implement projects with limited resource availabilities that are completed in a short amount of time with a levelled utilization of resources, without having to suppress all time buffers that could absorb unforeseen delays.

References

Azaron A, Tavakkoli-Moghaddam R (2006) A multi-objective resource allocation problem in dynamic PERT networks. Appl Math Comput 181(1):163–174

Azaron A, Katagiri H, Sakawa M, Kato K, Memariani A (2006) A multi-objective resource allocation problem in PERT networks. Eur J Oper Res 172(3):838–854

Ballarin A, Vecchiato D, Tempesta T, Marangon F, Troiano S (2011) Biomass energy production in agriculture: a weighted goal programming analysis. Energ Policy 39(3):1123–1131

Bevilacqua M, Ciarapica FE, Giacchetta G (2009) Critical chain and risk analysis applied to high-risk industry maintenance: a case study. Int J Proj Manag 27(4):419–432

Charnes A, Cooper WW (1961) Management model and industrial application of linear programming, vol 1. Wiley, New York

Goldratt EM (1964) Handbook of mathematical functions with formulas, graphs, and mathematical tables. Dover Printing, Mineola

Goldratt EM (1997) Critical chain. North River Press, Great Barrington

Ignizio JP (1976) Goal programming and extensions. Lexington Books, Lexington

Inuiguchi M, Sakawa M (1995) Minimax regret solution to linear programming problems with an interval objective function. Eur J Oper Res 86(3):526–536

Kim GC, Emery J (2000) An application of zero–one goal programming in project selection and resource planning: a case study from the Woodward Governor Company. Comput Oper Res 27(14):1389–1408

Lee SM (1972) Goal programming for decision analysis. Auerbach, Philadelphia

Long LD, Ohsato A (2008) Fuzzy critical chain method for project scheduling under resource constraints and uncertainty. Int J Proj Manag 26(6):688–698, doi:10.1016/j.ijproman.2007.09.012

Romero C, Rehman T (1989) Multiple criteria analysis for agricultural decisions. Elsevier, Amsterdam

Shou Y, Yao KT (2000) Estimation of project buffers in critical chain project management. In: Proceedings of the IEEE international conference on management of innovation and technology 2000 (ICMIT 2000), pp 162–167

Simon HA (1955) A behavioral model of rational choice. Q J Econ 69(1):99–118

Tamiz M, Jones D, Romero C (1998) Goal programming for decision making: an overview of the current state-of-the-art. Eur J Oper Res 111(3):569–581

Watson KJ, Blackstone JH, Gardiner SC (2007) The evolution of a management philosophy: the theory of constraints. J Oper Manag 25(2):387–402

Yang JB (2000) Minimax reference point approach and its application for multiobjective optimisation. Eur J Oper Res 126(3):541–556

Chapter 59
Multi-Criteria Multi-Modal Fuzzy Project Scheduling in Construction Industry

Jiuping Xu and Ziqiang Zeng

Abstract The construction project managers often face the challenges to compromise among different conflicting aspects of a project, especially for the criteria of time, cost, quality, and environment in a multi-modal project. This leads to a multi-criteria multi-modal project scheduling decision making problem. The complexity in the project scheduling is concentrated on the search for an ideal level of balance among all the conflicting criteria. In fact, the tradeoff among these criteria is determined by the modal selection and the duration reduction applied within the selected mode in a multi-modal project. Therefore, it is important to improve the optimality and effectiveness of modal selection that significantly contributes to the success of a project. In this chapter, the introduction and statement of a discrete time-cost-environment tradeoff problem (DTCETP) for large-scale construction systems with multiple modes under fuzzy uncertainty are presented. The modelling process of DTCETP is explained in detail. Since the DTCETP belongs to the class of non-deterministic polynomial-time hard problems, a fuzzy-based adaptive-hybrid genetic algorithm is developed to efficiently find feasible solutions. Finally, the case study of Jinping-II Hydroelectric Project is employed as a practical application example.

Keywords Construction industry • Fuzzy sets • Multi-criteria decision-making • Multi mode • Project scheduling

59.1 Introduction

Time and project cost are crucial criteria of construction projects and have received significant attention for several years (Akkan et al. 2005; Leu et al. 2001). As stated in Chaps. 29 and 30 in the first volume of this handbook, the discrete time-cost tradeoff problem (DTCTP), which was introduced by Harvey and Patterson (1979)

J. Xu (✉) • Z. Zeng
Uncertainty Decision-Making Laboratory, Sichuan University, Chengdu, People's Republic of China
e-mail: xujiuping@scu.edu.cn; zengziqiangscu@gmail.com

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_29

and Hindelang and Muth (1979), is an important subject in multi-criteria project scheduling theory and applications (Peng and Wang 2009).

Nowadays, the quantity and scale of construction projects world wide have developed rapidly. Construction project managers often face the challenges to compromise among different conflicting criteria of a project (Liang 2010; Xu and Zeng 2011). This leads to a multi-criteria project scheduling decision-making problem. In fact, multi-criteria decision-making or multiple criteria decision analysis is a sub-discipline of operations research that explicitly considers multiple criteria in decision-making environments.

Generally, the project management decisions focus on the minimization of project completion time, and/or the minimization of total project cost through crashing or shortening the duration of particular activities (Akkan et al. 2005; Eshtehardian et al. 2009; Leu et al. 2001). Thus, a project decision maker may be able to shorten project completion time, realizing savings on indirect costs, by increasing direct expenses to accelerate the project (Liang 2010). Additionally, although various project management decision techniques have been developed to minimize project duration and/or total project cost, most do not minimize the environmental impact (Ammar 2011; Wang et al. 2010). Therefore, this chapter considers that construction managers need to develop a project management methodology for directing and controlling not only the total project duration and project cost, but also the environmental impact to achieve management objectives. This leads to a discrete time-cost-environment tradeoff problem (DTCETP), an extension of DTCTP. The criteria of the project management decision is to find a starting time and a crashing time (Klerides and Hadjiconstantinou 2010) for each activity such that the makespan is minimized and the schedule is feasible with respect to some constraints, such as precedence (Al-Fawzana and Haouari 2005), crashing time, total budget, and duration (Long and Ohsato 2008). In this chapter, four criteria (objectives) for the DTCETP are considered: (1) the minimization of the total project cost; (2) the minimization of the total project duration; (3) the minimization of the total crashing cost; (4) the minimization of the environmental impact.

In project scheduling, every activity can be executed in the crashing way in which the project direct costs are used to shorten the activity duration. According to the extent of the crashed activity duration, the execution of every activity can be classified into different execution modes. So the decision maker should decide which execution mode is selected while executing every activity. The crashed duration of activities was introduced to project scheduling in Ahn and Erenguc (1998), where the duration/cost of an activity is determined by the modal selection and the duration reduction (crashing) applied within the selected mode. This leads to a multi-mode project scheduling decision making problem (see Chap. 21 in the first volume of this handbook).

Due to incomplete or unavailable relevant information over the project planning horizon, the model inputs for the project management decisions are normally imprecise in practice (see Chap. 42). In non-routine projects (e.g., new construction projects) (Long and Ohsato 2008), the duration of each activity and completion time

may be uncertain, and the project manager has to handle multiple conflicting goals in an uncertain environment with information that may be incomplete or unavailable. Uncertainty in the activity durations can be modeled by two groups of methods such as probability-based methods and fuzzy set-based methods (Demeulemeester and Herroelen 2002; Węglarz 1999), the use of which depends on the situation and the project manager's preference. In a new construction project it is difficult for a project manager to characterize these random variables correctly, because activities tend to be unique, and therefore lack historical data. For this reason, the fuzzy method is considered an effective approach for such situations. First proposed by Zadeh (1965), and consequently developed by researchers such as Dubois and Prade (1988), and Nahmias (1978), fuzzy theory has been a useful tool in dealing with ambiguous information.

In this chapter, a multi-criteria multi-modal fuzzy project scheduling problem is discussed. The remainder of this chapter is organized as follows: Sect. 59.2 describes the statement of DTCETP. In Sect. 59.3, the modelling process of DTCETP is explained in detail. In Sect. 59.4, a fuzzy-based adaptive hybrid genetic algorithm ((f)a-hGA) is developed to solve the problem in construction projects. Section 59.5 involves a case study regarding the works of a construction system for a large-scale hydroelectric project, a sensitivity analysis and a comparison of the results of the (f)a-hGA with other heuristic algorithms are also provided. Finally, concluding remarks and future research are outlined in Sect. 59.6.

59.2 Statement of DTCETP

This section, based on the analysis of construction systems, presents a discrete time-cost-environment tradeoff problem (DTCETP) of multi-criteria multi-modal fuzzy project scheduling in construction industry.

59.2.1 Problem Description

In a construction project, the construction planners have to compromise between different aspects of projects, especially time, cost, and quality. In recent years, there is increasing pressure on decision makers to search for a plan that minimizes not only construction cost and time, but also environmental impact. In fact, in a multi-modal project scheduling, the construction time, cost and environmental impact can be determined by the mode selection and the duration reduction (crashing) applied within the selected mode. The execution mode can be defined by the crashing way in which the project direct costs are used to shorten the activity duration. According to the extent of the crashed activity duration, the execution of every activity can be classified into different execution modes. Therefore the problem for the decision maker consists in how to optimally select the execution mode to obtain an ideal tradeoff level of balance among time, cost, and environment for a construction

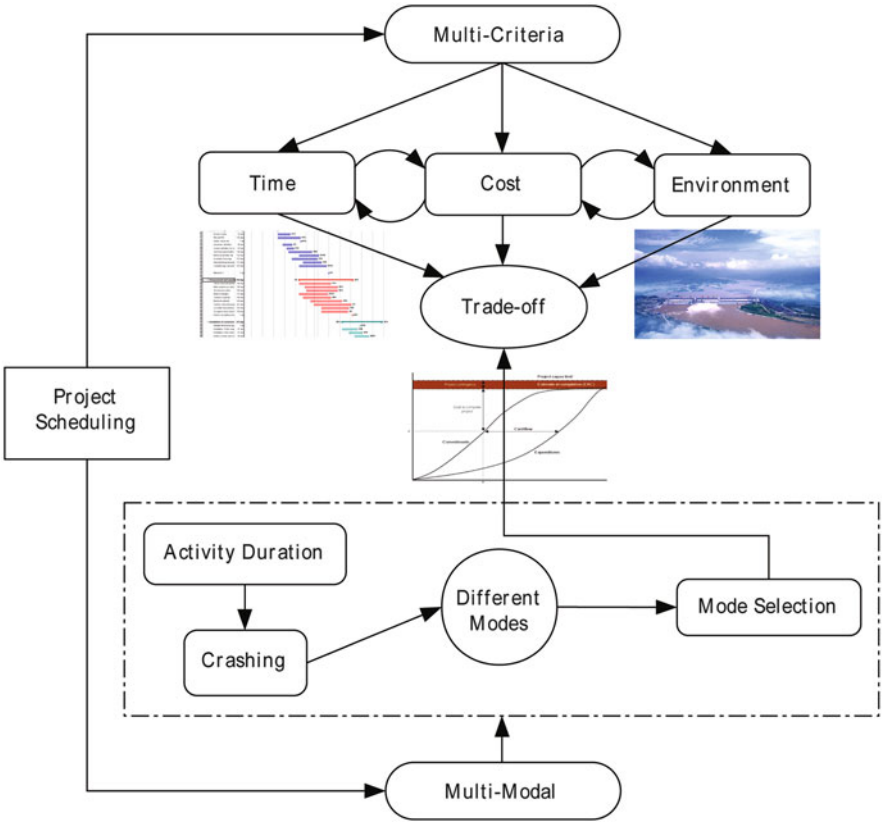


Fig. 59.1 Relationship of elements in a multi-criteria multi-modal project scheduling problem

project. Figure 59.1 shows a relationship of elements in a multi-criteria multi-modal project scheduling problem.

59.2.2 Environmental Impact on Project

Recently, the construction industry has been accused of causing environmental problems ranging from excessive consumption of global resources both in terms of construction and operation to the polluting of the surrounding environment (Ding 2008). In particular, hydroelectric projects significantly contribute to changes in river environments (Chen et al. 2011), in which eco-environmental impact may arise during all project phases (Brismar 2004). In this chapter, it is suggested that the environment of a construction project may be affected by mode selection. As such, when a hydroelectric project is being planned, its environmental impact should be taken into consideration along with the time and cost tradeoffs.

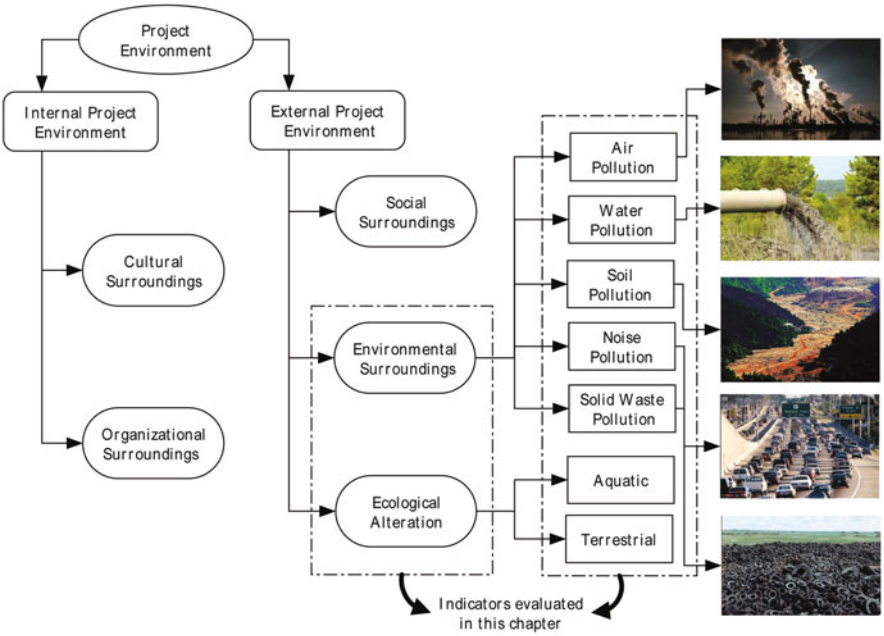


Fig. 59.2 Indicators of project environment

Project environment means that project teams are comfortable with, and sympathetic towards, their cultural, organizational, social, environmental, and ecological surroundings (Reschke and Schelle 1990; Xu and Li 2012). While the cultural and organizational surroundings are part of the internal project environment (Pheng and Chuan 2006), this chapter focuses on the external project environment (i.e., environmental and ecological surroundings), which includes air, water, soil, noise, and solid waste pollution, and ecological alterations (Liu and Lai 2009). The indicators of the project environment that have been evaluated by this study have been highlighted in Fig. 59.2. The main objective is to analyze the cause-and-effect relationship between the project environment and the environmental impact. The environmental impact is a quantifiable measurement of the project environment.

59.2.3 Motivation for Employing Fuzzy Variables in DTCETP

The need to address uncertainty in construction project scheduling is widely recognized, as uncertainties exist in a variety of system components. As a result, the inherent complexity and uncertainty existing in real-world project scheduling problems has essentially placed them beyond conventional deterministic optimization methods. Although probability theory has been proved to be a useful tool for dealing with uncertainty in project scheduling problems in hospital management,

sometimes it may not be suitable for new construction projects due to the lack of historical data. While it may be easy to estimate the probability distributions for the duration of an operation for surgery since there are sufficient historical data, it is usually difficult to do this for the duration of an activity in a new construction project, especially in the early construction stages. In this case, fuzzy theory is more suitable to deal with such situations encountered in the DTCETP.

The duration for each activity is a typical uncertain variable, which can fluctuate because of many factors, such as, the weather, equipment properties, labor efficiency, execution errors of decision makers, supply conditions of materials, coordination problems among stakeholders and other uncertain factors. Let p_i be the normal duration for activity i, $\hat{p}_i$ the crashed duration for activity i, and Y_i the crashing time for activity i. In practice, the decision makers may give a statement such as "it is possible that the normal duration for activity i is within an optimistic and pessimistic range, where the optimistic margin is 15 months, and the pessimistic margin is 23 months, and the most likely value of the normal duration is 19 months", which can be translated into a triangular fuzzy number $p_i = (15, 19, 23)$. If the crashing time is 3 months, then the crashed duration for activity i can be calculated as a triangular fuzzy number $\hat{p}_i = (12, 16, 20)$ as shown in Fig. 59.3.

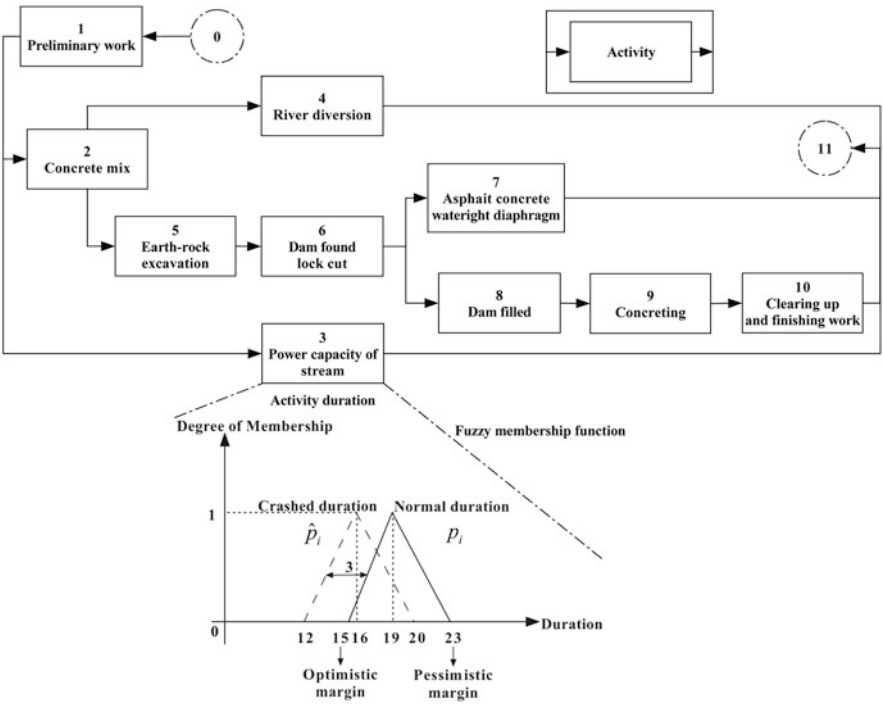


Fig. 59.3 The membership function of duration for each activity

59.3 Modelling Process of DTCETP

To model the multi-criteria multi-modal fuzzy project scheduling problem in this chapter, the assumptions dealing with the fuzzy variables, the model formulation, and the model analysis are presented subsequently.

59.3.1 Assumptions

To model the DTCETP of construction systems for large-scale project scheduling under fuzzy uncertainty, the following assumptions are made:

1. A single project consists of a number of activities, the duration of each activity is considered as a triangular fuzzy number, and the minimum crashed duration is known (Leu et al. 2001).
2. The fixed cost, unit variable cost, and the unit crashing cost of each activity are known.
3. The capital used by all activities does not exceed limited quantities in any time period, and the total project budget is within a deterministic limit.
4. The starting time of each activity is dependent upon the completion of its predecessor.
5. Variable cost increases linearly with the duration of each activity which is reduced from its normal duration to its crashed duration.
6. The total project cost can be divided into fixed cost and variable cost and crashing cost. The unit variable cost keeps the same value during the whole activity duration.
7. When an activity begins, it cannot be interrupted.
8. Managerial criteria are to minimize the total project cost, project duration, crashing cost, and the environmental impact.
9. The decision maker takes a compromise attitude to risk.

Based upon the assumptions above, a multi-criteria model of DTCETP for construction systems under fuzzy uncertainty is proposed.

59.3.2 Dealing with Fuzzy Variables

In order to transform these fuzzy numbers into crisp values, the expected value operator for the fuzzy measure *Me* (Xu and Zhou 2011) is introduced to deal with the uncertainty in the DTCETP. The expected value of a triangular fuzzy number is presented as follows,

$$E^{Me}[\xi] = \begin{cases} \frac{\eta}{2}r_1 + \frac{r_2}{2} + \frac{1-\eta}{2}r_3, & if \quad r_3 \le 0 \\ \frac{\eta}{2}(r_1 + r_2) + \frac{\eta r_3^2 - (1-\eta)r_2^2}{2(r_3 - r_2)}, & if \quad r_2 \le 0 \le r_3 \\ \frac{\eta}{2}(r_3 + r_2) + \frac{(1-\eta)r_2^2 - \eta r_1^2}{2(r_2 - r_1)}, & if \quad r_1 \le 0 \le r_2 \\ \frac{(1-\eta)r_1 + r_2 + \eta r_3}{2}, & if \quad 0 \le r_1 \end{cases} \tag{59.1}$$

where η is the optimistic-pessimistic index to determine the combined attitudes of decision makers.

As all fuzzy variables are non-negative triangular fuzzy numbers (Fig. 59.3 shows an example) in this chapter, the DTCETP belongs to the case $0 \le r_1$, i.e., $E^{Me}[\xi] = \frac{(1-\eta)r_1 + r_2 + \eta r_3}{2}$ (Xu and Zhou 2011).

For example, $\hat{p}_i \longrightarrow E[\hat{p}_i] = \frac{(1-\eta)\hat{p}_{i1} + \hat{p}_{i2} + \eta\hat{p}_{i3}}{2}$, since all the fuzzy variables in the problem are triangular fuzzy numbers, the transformations are presented as follows,

$$p_i \to E[p_i] = \frac{(1-\eta)p_{i1} + p_{i2} + \eta p_{i3}}{2},$$

$$ES_i \to E[ES_i] = \frac{(1-\eta)ES_{i1} + ES_{i2} + \eta ES_{i3}}{2}$$

$$S_0 \to E[S_0] = \frac{(1-\eta)S_{01} + S_{02} + \eta S_{03}}{2},$$

$$S_{n+1} \to E[S_{n+1}] = \frac{(1-\eta)S_{n+1,1} + S_{n+1,2} + \eta S_{n+1,3}}{2}$$

$$C_{max} \to E[C_{max}] = \frac{(1-\eta)C_{max,1} + C_{max,2} + \eta C_{max,3}}{2},$$

$$\overline{d} \to E[\overline{d}] = \frac{(1-\eta)\overline{d}_1 + \overline{d}_2 + \eta\overline{d}_3}{2}$$

where ES_i is the earliest starting time of activity i; S_0 is the project starting time; S_{n+1} is the project completion time; C_{max} is the project completion time under normal conditions; $\overline{d}$ is the specified project completion time.

59.3.3 Model Formulation

The problem is represented on an activity-on-node (AoN) network with a single starting and a single ending node, both corresponding to dummy activities. Based on the decision maker's criteria for the project, a fuzzy optimization model is proposed. The following subsections in this chapter explain the uncertain multi-criteria model

in detail with criterion functions and constraints discussed separately to illustrate the model more clearly.

59.3.3.1 Criterion Functions

The presented optimization model is formulated to minimize project time and cost, while also minimizing its environmental impact. To this end, the model incorporates four major criterion functions, shown in the following four equations, to enable the evaluation of the performance in project time, cost, and environment, respectively.

Project Cost: Usually total project cost changes because of many factors, such as fixed cost, duration, and the crashing time mode of each activity. Therefore, decision makers aim to achieve the best option for the execution of the process, where the total project cost is minimal. The total project cost is the sum of the fixed cost, variable cost, and the crashing cost of each activity. The sum of "fixed cost of activity i + expected duration of activity i in a certain executed crashing mode × unit variable cost of activity i + unit crashing cost of activity i × crashing time of activity i" is shown in the Eq. (59.2). The first criterion is to minimize the expected total project cost, which is the minimization of the sum of the expected completion cost for all activities.

$$\text{Min. } z_1 = \sum_{i \in V} (c_i^{fix} + E[\hat{p}_i] c_i^{var}) + \sum_{i \in V} c_i^{cr} Y_i \tag{59.2}$$

where z_1 is the total project cost; c_i^{fix} is the fixed cost of activity i; c_i^{var} is the unit variable cost of activity i; c_i^{cr} is the unit crashing cost of activity i; and Y_i is the crashing time of activity i; $V = \{1, 2, \cdots, n\}$ is the set of (real) activities.

Project Duration: The second criterion seeks to minimize the expected total project duration, which is the expected makespan between the project starting time and project completion time. It is also the minimization of the expected duration for all activities.

$$\text{Min. } z_2 = E[S_{n+1}] - E[S_0] \tag{59.3}$$

where z_2 is the total project duration.

Crashing Cost: The third criterion is to minimize the total crashing cost as follows. It indicates that the crashing cost for each activity should be minimized as much as possible. Generally, decision makers attempt to confirm the expected duration for each activity beforehand. However, in practice, if an activity finishes too early, it may cause additional costs, an unexpected change to the environment or lead to the payment of penalties. Thus, crashing cost should be minimized individually while keeping in mind the first criterion (Liang 2010).

$$\text{Min. } z_3 = \sum_{i \in V} c_i^{cr} Y_i \tag{59.4}$$

where z_3 is the total crashing cost.

Environmental Impact: The fourth criterion is to minimize environmental impact that is measured and quantified. It enables the aggregation of the estimated environmental impact for all the considered activities to provide an overall environmental impact on the project level using a simple weighted approach.

$$\text{Min. } z_4 = \sum_{i \in V} w_i \sum_{p \in \mathscr{P}} w_{ip}^m \times Q_{ip}^m \tag{59.5}$$

where z_4 is the total environmental impact; w_i is the weight of activity i compared to other activities in the project; w_{ip}^m is the weight of score p for environmental impact on activity i using the m-th crashing mode; and Q_{ip}^m is the score p for environmental impact on activity i using the m-th crashing mode; $\mathscr{P}$ is the set of scores for the environmental impact.

Estimating and quantifying the environmental impact of a given crashing time option on the environment of an activity and the entire project is a more challenging task than predicting its impact on project cost and duration. This can be attributed to two major challenges: (1) the difficulty in measuring and quantifying the impact of each crashing time option on the environment of the activity being considered; and (2) the complexity of aggregating environmental impact at the activity level to provide an overall environmental impact at the project level. In order to overcome these two major challenges, the present model incorporates a new criterion function for optimizing the environmental impact. The following two paragraphs provide a brief description of how this newly formulated function is designed to overcome the above two major challenges in quantifying and considering environmental impact in the optimization process.

In order to facilitate the measurement and quantification of the environmental impact, the incorporated environmental impact criterion function enables the consideration of a number of measurable environmental indicators for each activity in the project. These indicators have been investigated and identified in studies aimed at developing environment-based systems (Chiang and Lai 2002). The identified environmental indicators were derived from the long-term performance of each activity in the performance-based models. Environmental indicators should be selected in a way that allows a practical and objective measurement of the performance in each activity. For each crashing time option, the results of these environmental impact tests can be easily obtained and stored. In fact, many data are currently gathered and stored from ongoing projects using a variety of forms. This collected and stored data can be statistically analyzed in order to estimate environmental impact.

It should be noted that test results for the selected environmental indicators (i.e., air, water, soil, noise, and solid waste pollution, and ecological alteration), are often expressed in different units of measurement (Chen et al. 2011; Liu and Lai 2009; Li et al. 2010). As such, they need to be transformed into a unified measurement system that can be consistently used to evaluate the performance of the different environment indicators. In this model, the results of the different environmental test

indicators are transformed to a common score ranging from 0 to 100 to represent the degree of severity of environmental impact in each activity.

The formulated environmental criterion function enables the aggregation of the estimated environmental impact for all considered activities to provide an overall environmental impact at the project level using a simple weighted approach (Ding 2008; Peche and Rodríguez 2009). For each activity being evaluated for environmental impact, this method requires planners to identify two types of weights: (1) weight of activity i (i.e., w_i) that represents the importance and contribution of the environmental impact of this activity to the overall environmental impact of the project; and (2) weight of the score p of the environmental impact of activity i using the m-th crashing mode (i.e., w_{ip}^m) that indicates the relative importance of this score to the others. These two types of weights are used to estimate the overall environmental impact at the project level. The illustrated method can be applied to additional construction activities using other measurable environmental indicators (Gangolells et al. 2009; Li et al. 2010).

59.3.3.2 Constraints

Precedence Constraint: In a project, precedence is an important basic term ensuring the rationality of the arrangement. Under this term, successive activities must be and can only be started with a certain crashing time option when all the predecessors have already been completed. Therefore, this constraint is used for activity i considering its immediate predecessor h, one by one, here, the index $h \in Pred(i)$, where $Pred(i)$ is the set of the immediate predecessors of activity i. The relationship among the starting time and duration of predecessor h and the starting time of activity i is *the starting time of predecessor* h + *the duration of predecessor* h − *the starting time of activity* $i \leq 0$. It is important that none of the precedence constraints are violated for all predecessors of activity i as shown in Eq. (59.6).

$$E[ES_h] + E[\hat{p}_h] - E[ES_i] \leq 0 \quad (i \in V;\ h \in Pred(i)) \tag{59.6}$$

where $Pred(i)$ is the set of the immediate predecessors of activity i.

Crashing Time Constraint: The crashed duration is equal to the normal duration of activity i minus its crashing time. This is shown in Eq. (59.7).

$$E[\hat{p}_i] = E[p_i] - Y_i \quad (i \in V) \tag{59.7}$$

On the other hand, the crashing time cannot exceed the difference between the normal duration and the minimum crashed duration of activity i.

$$Y_i \leq E[p_i] - \hat{p}_i^{min} \quad (i \in V) \tag{59.8}$$

where $\hat{p}_i^{min}$ is the minimum crashed duration of activity i.

Total Budget Constraint: It is also important for projects to limit the total capital within a deterministic limit (i.e., b). The range of total project cost is between its normal cost and its crashed total cost during the project execution. Eq. (59.9) can be used to describe these requirements.

$$z_1 \leq b \tag{59.9}$$

where b is the available total budget.

Logical Constraints: In order to describe non-negative variables and 0–1 variables in the model, the constraints in Eqs. (59.10)–(59.11) are presented.

$$E[p_i], E[\hat{p}_i], Y_i, E[ES_i], E[\overline{d}] \geq 0 \quad (i \in V) \tag{59.10}$$

$$x_{imt} \in \{0,\ 1\} \quad (i \in V;\ m \in \mathscr{M}_i;\ t \in H) \tag{59.11}$$

where $\mathscr{M}_i = \{1, 2, \ldots, M_i\}$ is the set of crashing modes of activity i and $H = \{1, 2, \ldots, \lceil E[ES_{n+1}] \rceil\}$ is the set of periods in the planning horizon, $x_{imt} = 1$ means that activity i in mode m is scheduled in time period t.

Cash Flow Constraint: The sum of total fixed cost, variable cost, and crashing cost of the activities that are scheduled in time period t cannot exceed the capital limit per time period, as shown in Fig. 59.4 (Xu et al. 2012). The sum of the capital of the activities which are scheduled in a certain time period during the whole project duration, as well as in a certain crashing time option is shown in Eq. (59.12).

$$\sum_{i \in V} \sum_{m \in \mathscr{M}_i} \frac{c_i^{fix} + E[\hat{p}_i]c_i^{var} + c_i^{cr} Y_i}{E[\hat{p}_i]} x_{imt} < lim \quad (t \in H) \tag{59.12}$$

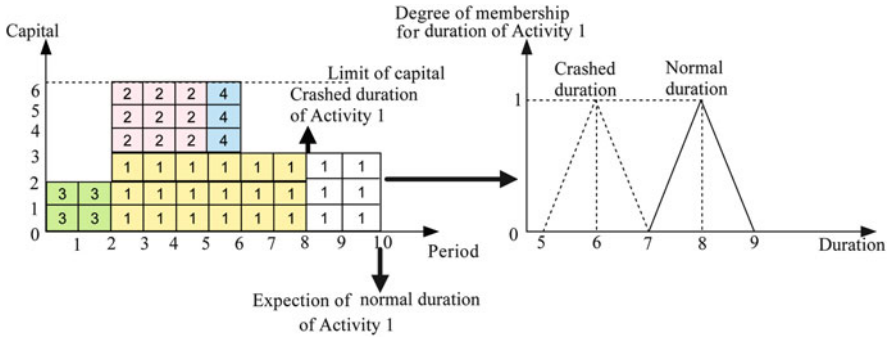


Fig. 59.4 Capital limit in a period for the DTCETP example (Xu et al. 2012)

Duration Constraint: Generally, the decision makers need to confirm an expected project duration beforehand to allow for the coordination of parallel projects or other resources. Certainly, the expected duration is determined by the decision makers based on their accumulated work experience regarding to the appropriate finishing time of activities. So it cannot exceed the expected project completion time, which is shown in Eq. (59.13).

$$E[S_{n+1}] \leq E[\overline{d}] \tag{59.13}$$

Based on the above discussion, by integrating Eqs. (59.2)–(59.13), the mathematical model of DTCETP for multi-criteria multi-modal fuzzy project scheduling problem can be stated as follows:

$$\begin{aligned}
&\text{Min.} \quad \{f_1, f_2, f_3, f_4\} \\
&\text{s.t.} \quad E[ES_h] + E[\hat{p}_h] - E[ES_i] \leq 0 \quad (i \in V;\ h \in Pred(i)) \\
&\qquad E[\hat{p}_i] = E[p_i] - Y_i \quad (i \in V) \\
&\qquad Y_i \leq E[p_i] - \hat{p}_i^{min} \quad (i \in V) \\
&\qquad z_1 \leq b \\
&\qquad E[p_i], E[\hat{p}_i], Y_i, E[ES_i], E[\overline{d}] \geq 0 \quad (i \in V) \\
&\qquad x_{imt} \in \{0,\ 1\} \quad (i \in V;\ m \in \mathscr{M}_i;\ t \in H) \\
&\qquad E[S_{n+1}] \leq E[\overline{d}] \\
&\qquad \sum_{i \in V} \sum_{m \in \mathscr{M}_i} \frac{c_i^{fix} + E[\hat{p}_i]c_i^{var} + c_i^{cr} Y_i}{E[\hat{p}_i]} x_{imt} < lim \quad (t \in H)
\end{aligned} \tag{59.14}$$

where f_μ ($\mu = 1, \ldots, 4$) are the objective functions $z_1, \ldots, z_4$ in the mathematical model of DTCETP.

59.3.4 Model Analysis

From the above model, we can see if $c_i^{cr} < c_i^{var}$, then we have

$$\begin{aligned}
C_{\hat{p}_i} &= c_i^{fix} + c_i^{var}\hat{p}_i + c_i^{cr}(p_i - \hat{p}_i) = c_i^{fix} + c_i^{var}(p_i - Y_i) + c_i^{cr} Y_i \\
&< c_i^{fix} + c_i^{var}(p_i - Y_i) + c_i^{var} Y_i = c_i^{fix} + c_i^{var} p_i = C_{p_i}
\end{aligned} \tag{59.15}$$

It proves that if $c_i^{cr} < c_i^{var}$, then $C_{\hat{p}_i} < C_{p_i}$ and if $c_i^{cr} > c_i^{var}$, then $C_{\hat{p}_i} > C_{p_i}$. So the time-cost tradeoff problem can be optimized by comparing c_i^{cr} and c_i^{var}. If $c_i^{cr} < c_i^{var}$, the crashing activity duration can be easily calculated to a minimum value which can minimize project duration and cost simultaneously. If $c_i^{cr} > c_i^{var}$, a decision is made according to the practical situation or the decision maker's preference. That is also an easy problem to solve. However, in this study, the DTCETP with total capital constraint and cash flow constraint in any time period under fuzzy uncertainty needs to be considered. To obtain an optimal crashing time

decision for all activities is an $\mathscr{NP}$-hard problem and cannot be determined simply by determining whether $c_i^{cr} > c_i^{var}$ or $c_i^{cr} < c_i^{var}$.

59.4 Fuzzy-Based Adaptive Hybrid Genetic Algorithm for DTCETP

Existing techniques for DTCTP can be categorized into two areas: namely, exact and heuristic. Exact algorithms, including linear programming (Liu et al. 1995; Pagnoni 1990), dynamic programming (De et al. 1995), enumeration algorithms (Harvey and Patterson 1979), or branch-and-bound algorithms (Demeulemeester et al. 1996), have been extensively employed to solve DTCTPs (Eshtehardian et al. 2009). However, none of the exact algorithms are able to solve large and hard instances measured in terms of, say, the number of activities. In terms of what current state-of-art algorithms can do, and considering the structure of the project networks as well as the number of modes per activity, instances with a large number of activities cannot be solved optimally in a reasonable amount of time (Tareghian and Taheri 2007). De et al. (1997) have shown that the DTCTP is an $\mathscr{NP}$-hard problem and difficult to solve (Deineko and Woeginger 2001). Consequently, the DTCETP, which is an extension of the classical DTCTP, is also $\mathscr{NP}$-hard. It follows that the search for exact algorithms which are also formally efficient is all but futile and that one should instead search for effective heuristic algorithms to solve a general DTCTP. In this case, the use of heuristic solution procedures is justified. Among these algorithms, GA has produced outstanding results in optimization problems (Montoya-Torres et al. 2010). In order to cope with the larger and more complex instances of the DTCETP and to improve the computation efficiency, this section presents a fuzzy-based adaptive hybrid genetic algorithm ((f)a-hGA).

59.4.1 General Concept of (f)a-hGA

The GA, which was first introduced by Holland (1992), uses the Darwinian concept to solve problems. GA belongs to the class of meta-heuristic optimization techniques, and is very useful when there is a large search space with little knowledge. A balance between exploitation and exploration in the search space is one of the important factors when using GA. To provide this balance, the determination of the design strategy for GA parameters, such as population size, cross over probability, and maximum generation and mutation probability is one of the critical issues. In contrast to prior studies, the (f)a-hGA proposed here can deal with a multi-criteria objective function (i.e., total project cost, project duration, cashing cost, and environmental impact) and constraints of the DTCETP simultaneously under fuzzy uncertainty much more appropriately and effectively. The algorithm proposed

here has two modes for linear and non-linear situations, respectively. In the linear case, the fuzzy expected value model (EVM) is embedded in the a-hGA to deal with the fuzzy variables that appear in linear functions, while in the non-linear case, the fuzzy simulation is combined with a-hGA for handling the nonlinearity of the fuzziness. For the multi-criteria model formulated above, it is possible to find several Pareto optimal solutions for the problem. However, in construction project practice, only one exact optimized solution could conduct the decision-making in a pressing situation. Thus, the weighting method is applied to transform the multi-criteria model into a single-criterion one.

In the method of this section, firstly, the fuzzy EVM technique is used to defuzzify the fuzzy variable. Secondly, the weight-sum procedure is used to concentrate multiple criteria in management practice while reflecting the importance of each criterion in the decision maker's mind. Thirdly, priority-based encoding, and multistage-based encoding are introduced for activity priority, and activity crashing time for GA encoding, respectively, with corresponding GA decoding and GA evaluation proposed by Gen et al. (2008). Additionally, the one-point crossover operator for the activity precedence and a repairing strategy for the mutation operator for activity crashing time is adopted. Finally, an iterative hill climbing routine and an adaptive regulation mechanism are introduced to carry out searches around a convergence solution in the GA loop and to achieve faster algorithm convergence.

59.4.2 Overall Procedure of the Proposed Method

According to the DTCETP under fuzzy uncertainty, the method consists of five subsystems (see Fig. 59.5): (1) the input subsystem; (2) the activity duration generation subsystem; (3) the project duration determination subsystem; (4) the time-cost-environment tradeoff subsystem; (5) and the output subsystem. Each subsystem has its own purpose.

The activity duration generation subsystem is created to manipulate the generation of all possible activity durations. A pool of chromosomes represents the possible activity durations. Each gene in a chromosome represents the duration of a corresponding activity, whose value can be obtained based upon fuzzy EVM or fuzzy simulation (Xu and Zhou 2011).

Next, for the project duration determination subsystem, the normal duration and minimum crashed duration are determined, based upon the aforementioned activity duration and activity precedence relationships (defined in Eq. (59.6)).

Third, based upon a selected project duration within the normal duration and the minimum crashed duration, the minimum project total cost, crashing cost, and environmental impact are searched for in the time-cost-environment tradeoff subsystem. The subsystem uses the one-point crossover and repairing strategy for

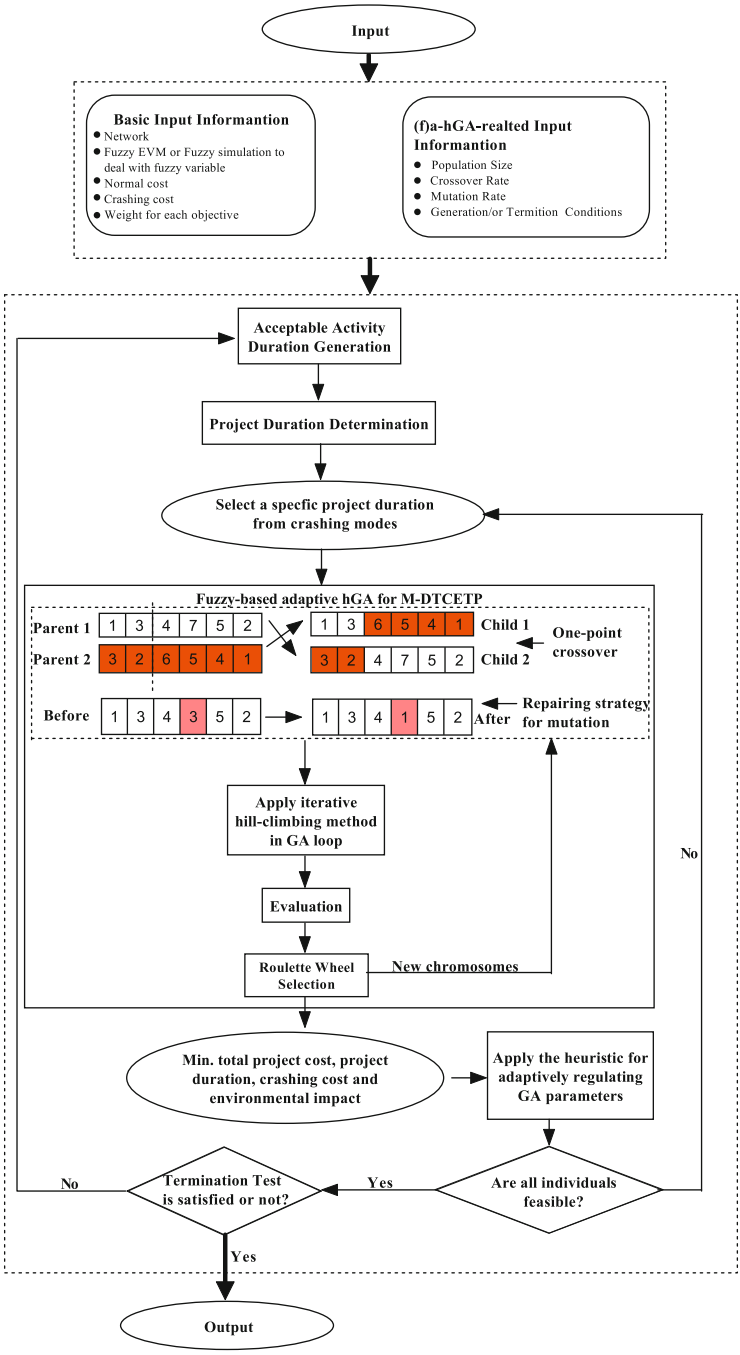


Fig. 59.5 Fuzzy adaptive hybrid genetic algorithm for the DTCETP (Xu et al. 2012)

the mutation operators to generate feasible child chromosomes. The concepts of the one-point crossover and mutation operators (see Fig. 59.5) are introduced as follows. The calculated fitness values in terms of the genetic algorithm are used to select the surviving chromosomes for the next generation, according to the criterion functions in Eqs. (59.2)–(59.5) and the weight-sum procedure. The surviving chromosomes for the next generation are selected at the selection subsystem according to the roulette wheel principle. The probability of variable selection is proportional to its fitness value in the population, according to the formula given by Holland (1992).

$$\pi(I_\mu) = \frac{F(I_\mu)}{\sum_{\nu=1}^{\sigma_{pop}} F(I_\nu)} \tag{59.16}$$

where $F(I_\mu)$ represents the fitness value of the μ-th chromosome, and σ_{pop} is the population size. The iterative hill climbing method can guarantee the properties of local search techniques for hybridization. The iterative hill climbing method suggested by Michalewicz (1996) is applied in the GA loop. In the final step of the time-cost-environment tradeoff subsystem, the optimal solution is exported to the output subsystem.

The process from Subsystem 2 to Subsystem 4 will be repeated until all the values for the project duration are within the possible ranges, and individuals have passed the feasible test.

At the final output subsystem, the total project cost, project duration, crashing cost, and environmental impact are brought together for further plotting and analysis of the results.

59.4.3 Weight-Sum Procedure

In this chapter, the weight-sum procedure is adopted to deal with the multi-criteria model. The aggregated criterion in the form of a weighted sum makes it possible to find the Pareto optimal solutions only when the solution set is convex (Gen et al. 2008). However, this term can be satisfied, because the criterion functions and constraints of the model are convex. This means that the mathematical model of DTCETP is a convex program. It is not hard to prove that the solution set of this model is convex. Thus, this form of aggregated criterion is suitable for the problem.

To ensure the validity of conformity in multi-criteria models, dividing out the dimensions and unifying the order of magnitude must be performed before the weight-sum procedure. The estimated maximal value is used to divide out the dimensions and unify the orders of magnitude. Here, the last four criteria are:

$$\begin{cases} z_1^{'} = \frac{z_1}{z_1^{max}} \\ z_2^{'} = \frac{z_2}{z_2^{max}} \\ z_3^{'} = \frac{z_3}{z_3^{max}} \\ z_4^{'} = \frac{z_4}{z_4^{max}} \end{cases} \tag{59.17}$$

z_1^{max}, z_2^{max}, z_3^{max} and z_4^{max} is the maximal value of z_1, z_2, z_3 and z_4.

For a given individual, the weighted-sum criterion function is given by the following equation:

$$eval = \min(w_1 z_1^{'} + w_2 z_2^{'} + w_3 z_3^{'} + w_4 z_4^{'}) \tag{59.18}$$

Wherein, the weight w_1 for the total project cost, w_2 for the total project duration, w_3 for the total crashing cost, and w_4 for the environmental impact are given by decision makers, and reflect the importance of each criterion in the decision makers' mind. Obtaining the weight of the criteria from the decision makers is the most direct and convenient way to reflect their requirements. The weights should satisfy the equation below.

$$w_1 + w_2 + w_3 + w_4 = 1 \tag{59.19}$$

59.4.4 Fuzzy Adaptive Hybrid GA Operator

In this subsection, the crossover operator and the mutation operator used in the (f)a-hGA approach are explained as follows.

59.4.4.1 Crossover Operators

The goal of the crossover is to exchange information between two parent chromosomes in order to produce two new offspring for the next population. There is a limit on the crashing time for each activity in the DTCETP. That means there is a limited number of different modes to be executed for different activities. For example, if activity 1 can be crashed 4 units of time at most, then it has 5 modes to be executed, and if activity 2 can be crashed 2 units of time at most, then it has 3 modes to be executed. So the one-point crossover operator proposed by Goldberg (1989) is used in this chapter. The one-point crossover operator randomly selects one point at the parent strings and exchanges the right parts of two parent strings to generate offspring strings (see Fig. 59.5). It changes one or more genes in corresponding activities only if it guarantees the feasibility of the solution.

59.4.4.2 Repairing Strategy for Mutation Operators

Uniform mutation is altering one or more genes in a chromosome, depending on a predefined mutation rate. Each genotype has the probability of a mutation. A mutation operator is a random process where one genotype is replaced by another to generate a new chromosome. In our problem, there is a limited number of different modes to be executed for different activities. If the altered genes go beyond the limited modes, they should be repaired. Repairing a chromosome means taking an infeasible chromosome and generating a feasible one through some repairing procedure. For each particular problem, a specific repairing algorithm is designed. A safe value is set for each gene according to the executed modes limit for each activity. If the mutated gene value exceeds the corresponding limit, then the gene will adopt the maximum value.

59.4.5 Regulation of GA Parameters by an Adaptive Method

In this section, the crossover and mutation ratios are adaptively regulated during the genetic search process. Here, the crossover and mutation operator rates are auto-tuning and evaluated repeatedly for all the generations during the genetic search process. The occurring crossover rates and mutation operators are adaptively regulated according to the results of the proposed procedure (Gen et al. 2008).

59.5 Case Study: DTCETP for the Jinping-II Hydroelectric Project

This section presents a practical application for the DTCETP in a large-scale hydroelectric construction project. The project contains eleven activities and two dummy activities (start and end activity). Each activity has a certain maximal crashing time limit.

59.5.1 Presentation of Case Problem

In recent years, as China has experienced rapid growth in both the economy and society, the need for energy has also grown exponentially. New and renewable sources of energy have become more important and consequently hydropower resources have also become more important. Hydropower resources play an important role in China, especially in the Sichuan Province. The Chinese government has emphasized renewable energy development particularly in the areas of water

conservancy and hydropower. The Ertan Hydropower Development Company, Ltd. (EHDC) has been appointed to supply clean and renewable energy to support the economic and social development of the Sichuan-Chongqing region through the development of hydropower resources on the Yalong River in Sichuan. The Jinping-II Hydroelectric Project is one of EHDC's projects under construction. This chapter focuses on a discrete time-cost-environment tradeoff problem (DTCETP) in the Jinping-II Hydroelectric Project for minimizing the total project cost, project duration, crashing cost, and environmental impact.

All data for the Jinping-II Hydroelectric Project are obtained from the Ertan Hydropower Development Company. The multi-criteria DTCETP model for this project is established based on the model formulation method, which can also be used in similar projects.

The Jinping-II Hydroelectric Project is located on the large Jinping River Bend, and is the second of the five cascade projects on the river section from Kala down to the estuary. It is designed to cut the 150 km river bend with a group of power tunnels to use the natural drop created by the bend. The project primarily consists of a headwork sluice dam, spillway structures, power tunnels and a powerhouse complex. The dam is located 7.5 km downstream of the Jinping-I dam. The catchment area upstream of the dam measures 103,000 km^2, and the multi-year average inflow at the dam site is 1,220 m^3/s. The Jinping-II reservoir itself only has a capacity of daily regulation, but when jointly operated with the Jinping-I, it also has the capacity of yearly regulation. The four power tunnels have an average length of 16.6 km and an excavated diameter of 13 m, which makes them among the world's largest and longest hydraulic tunnels. The powerhouse complex is located underground on the other side of the river bend. The project has a total installed capacity of 4,800 MW (8×600 MW), which gives a multi-year average annual generation of 24.23 TWh (Fig. 59.6).

The decision maker needs to optimize the work on the construction project but is faced with an uncertain situation for the duration of each activity. To optimize

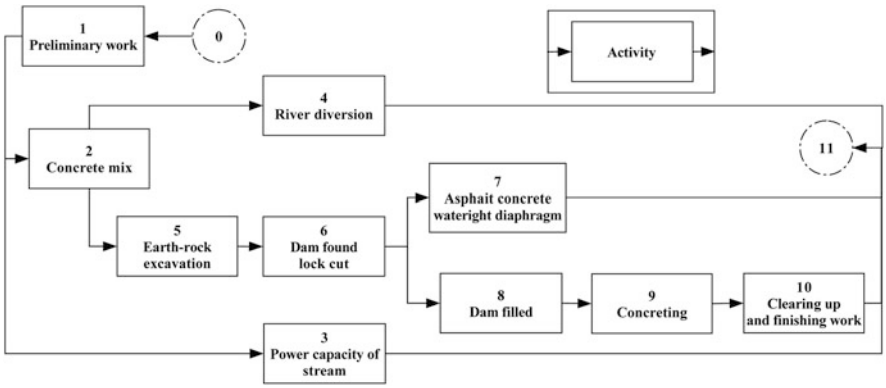


Fig. 59.6 Configuration of the construction project (Xu et al. 2012)

all aspects of the project, the decision makers want to pursue their management objectives through a better arrangement of the activity sequences and the selection of crashing time options. The proposed model and method are used to assist the decision maker in scheduling the construction activities optimally. The project has ten activities from preliminary work to clearing up and finishing work. Each of these has certain predecessors, successors, and a normal completion time. The company traditionally defines one month as a time unit (i.e., 1 month per unit). Two dummy activities were set up to help for the convenience of the model. The detailed corresponding data for each activity are shown in Tables 59.1 and 59.2.

Based on the representation of the case problem, the proposed method can be used to establish the mathematical model [i.e., Eq. (59.14)] for the DTCETP in the Jinping-II Hydroelectric Project.

Other relevant data are as follows: the total budget is 265, the maximum amount of capital is 12 units for each time period, the total project duration under normal condition is 43 months, and the decision maker expects that the total project duration will be below 40 months. The duration of each activity is given as follows:

$$p_0 = (0, 0, 0),\ p_1 = (10, 13, 16),\ p_2 = (1, 2, 3),\ p_3 = (1, 4, 7)$$
$$p_4 = (7, 9, 11),\ p_5 = (9, 13, 17),\ p_6 = (3, 5, 7),\ p_7 = (2, 4, 6)$$
$$p_8 = (2, 3, 4),\ p_9 = (1, 3, 5),\ p_{10} = (2, 4, 6),\ p_{11} = (0, 0, 0)$$

59.5.2 Result of the Case Problem

Based on the above model, the proposed (f)a-hGA is programmed using the Visual C++ language and run on a Core i3, 3.20 GHz clock pulse with 4 GB memory. The performance of this method is then compared with other heuristic algorithms.

The evolutionary parameters for the problem are set as follows: the population size is 20, the rate of crossover and mutation are 0.6 and 0.1, respectively, maximal generation is 200, the optimistic-pessimistic index is $\eta = 0.5$.

The results are shown in Table 59.3, which are obtained based on the parameter values $w_1 = 0.1$, $w_2 = 0.5$, $w_3 = 0.1$, $w_4 = 0.3$. It should be noted that the results are obtained based on the following optimistic-pessimistic index, i.e., $\eta = 0.5$. Using the chromosome illustrated above, Table 59.4 is obtained.

The detailed results are shown in Table 59.3, with the dummy activities not included. When the weight-sum procedures are used to deal with the multiple criteria, an equivalent treatment is proposed to obtain the fitness of each chromosome.

The above strategy is offered for the project, that is: arrange the activities in the order as proposed in Table 59.3, and choose the corresponding crashing time which fulfils the decision maker's requirements. It should be noted that some non-critical activities are flexible with respect to their execution time such as activity 4, which could be finished before the 23rd month and activity 7, which can be executed between the 31st and the 38th month as shown in Fig. 59.7. Therefore, the project manager can schedule these activities according to the situation which

Table 59.1 Detailed information of each activity in Jinping-I Hydroelectric Project-1

Activity number	Expected normal duration ($E[p_i]$) (month)	Minimum crashed time (p_i^{min}) (month)	Normal cost C_{p_i} (billion)	Fixed cost (billion)	Unit variable cost (billion/month)	Incremental crashing cost (billion)	Weight of activity (wt_i) (%)
0	Dummy activity						
1	13	9	25.88	13.53	0.95	0.8	17
2	2	2	11.64	10.82	0.41	∞	14
3	4	2	9.45	9.01	0.11	1	2
4	9	7	65.84	15.26	5.62	2	4
5	13	8	40.79	9.98	2.37	2.5	13
6	5	3	25.46	13.76	2.34	2.5	15
7	4	2	22.73	8.93	3.45	1	2
8	3	3	30.6	14.46	5.38	∞	11
9	3	3	15.22	10.24	1.66	∞	11
10	4	2	15.79	10.23	1.39	2.8	11
11	Dummy activity						

Table 59.2 Detailed information of each activity in Jinping-II Hydroelectric Project-2

Activity	Crashing time option	Q_{ip}^m	w_{ip}^m	Activity	Crashing time option	Q_{ip}^m	w_{ip}^m	Activity	Crashing time option	Q_{ip}^m	w_{ip}^m
1	$Y = 4$	99	0.3	4	$Y = 2$	98	0.7	6	$Y = 2$	98	0.7
		98	0.4			97	0.3			97	0.3
		97	0.3		$Y = 1$	95	0.5		$Y = 1$	97	0.5
	$Y = 3$	98	0.6			94	0.4			96	0.5
		96	0.4			92	0.1		$Y = 0$	95	0.6
	$Y = 2$	97	0.5		$Y = 0$	92	0.3			90	0.4
		96	0.5			90	0.6	7	$Y = 2$	98	0.5
	$Y = 1$	94	0.3			89	0.1			96	0.5
		90	0.7	5	$Y = 5$	99	0.1		$Y = 1$	96	0.7
	$Y = 0$	92	0.3			98	0.9			92	0.3
		89	0.7		$Y = 4$	98	0.1		$Y = 0$	92	0.6
2	$Y = 0$	99	0.1			97	0.9			89	0.4
		97	0.8		$Y = 3$	97	0.9	8	$Y = 0$	99	0.7
		96	0.1			93	0.1			92	0.3
3	$Y = 2$	99	0.5		$Y = 2$	94	0.1	9	$Y = 0$	97	0.5
		98	0.5			93	0.9			93	0.5
	$Y = 1$	97	0.3		$Y = 1$	93	0.7	10	$Y = 2$	98	0.5
		92	0.6			92	0.2			97	0.5
		91	0.1			91	0.1		$Y = 1$	94	0.9
	$Y = 0$	92	0.1		$Y = 0$	92	0.6			91	0.1
		91	0.2			90	0.3		$Y = 0$	92	0.7
		90	0.7			89	0.1			89	0.3

Table 59.3 Optimal solution for $\eta = 0.5$, $w_1 = 0.1$, $w_2 = 0.5$, $w_3 = 0.1$, $w_4 = 0.3$

$z_1 = 262.93$	$Y_1 = 4, Y_2 = 0, Y_3 = 0, Y_4 = 0, Y_5 = 1$
$z_2 = 38.0$	$Y_6 = 0, Y_7 = 0, Y_8 = 0, Y_9 = 0, Y_{10} = 0$
$z_3 = 5.70$	$t_1 = 9, t_2 = 2, t_3 = 4, t_4 = 9, t_5 = 12$
$z_4 = 94.62$	$t_6 = 5, t_7 = 4, t_8 = 3, t_9 = 3, t_{10} = 4$

Table 59.4 Schedule for the DTCETP in Jinping-II Hydroelectric Project

$a_0(0) : 0 - 0$	$a_1(4) : 0 - 9$	$a_2(0) : 9 - 11$	$a_3(0) : 9 - 13$	$a_5(1) : 11 - 23$	$a_4(0) : 13 - 22$
$a_6(0) : 23 - 28$	$a_8(0) : 28 - 31$	$a_9(0) : 31 - 34$	$a_7(0) : 31 - 35$	$a_{10}(0) : 34 - 38$	$a_{11}(0) : 38 - 38$

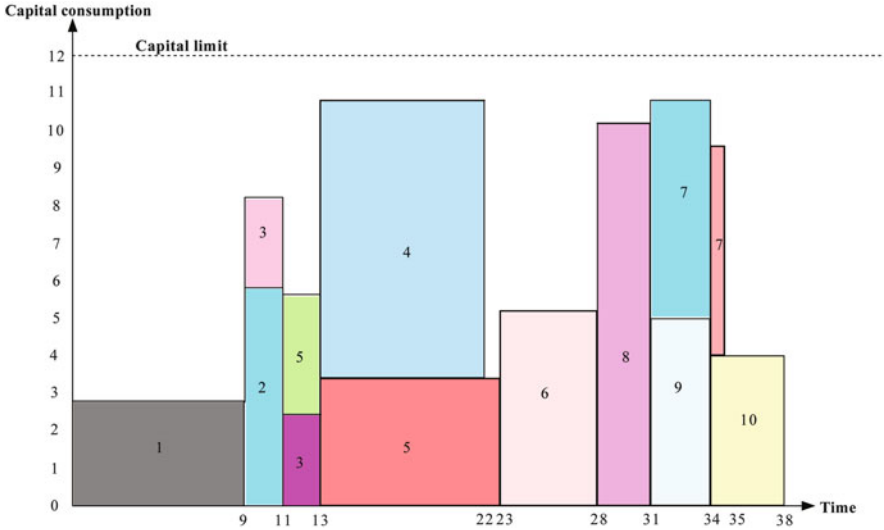


Fig. 59.7 Gantt chart for the DTCETP in Jinping-II Hydroelectric Project (Xu et al. 2012)

can be impacted by available manpower, equipment and holidays, and the need to harmonize with other parallel projects or activities.

Since our mathematical model is formulated with some assumptions, there may be some possible modeling errors. Thus, the results obtained above cannot 100 % surely represent an optimal time-cost-environment tradeoff solution. However, to some extent, our results can be used to provide decision makers with a theoretical optimal schedule for guiding current practice.

59.5.3 Sensitivity Analysis

A sensitivity analysis is performed: the weights of these four criteria are adjusted, and the results of the analysis are shown in Table 59.5. It shows that the solutions are not significantly influenced by the change in weight, and the result is satisfactory to the decision makers of this company. It can be concluded from Combination 1 that if the balance of the tradeoff is a trend for the total project duration instead of environmental impact, then the duration of the critical activity (i.e., activity 5) will be crashed to a lower level, as shown in Table 59.5. On the contrary, from Combination 3 it follows that, if the weights of the total project cost and crashing cost increase, then the critical activity (i.e., activity 5) will not be crashed. It should be noted that the results are obtained based on the following optimistic-pessimistic index, i.e., $\eta = 0.5$.

Table 59.5 Sensitivity analysis for the weights of criteria

Combination	Weights of criterion				Fitness value	The optimal order and mode of crashing time									
	w_1	w_2	w_3	w_4											
Combination 1	0.15	0.6	0.15	0.1	0.627	2	3	4	6	5	7	9	10	8	11
						9	2	4	10	9	5	3	3	4	4
Combination 2	0.1	0.5	0.1	0.3	0.570	2	3	4	6	5	7	9	10	8	11
						9	2	4	12	9	5	3	3	4	4
Combination 3	0.2	0.4	0.2	0.2	0.648	2	3	4	6	5	7	9	10	8	11
						9	2	4	13	9	5	3	3	4	4

Table 59.6 Sensitivity analysis for optimistic-pessimistic index of decision maker

Optimistic-pessimistic index	z_1	z_2	z_3	z_4	Crashing time for each activity
$\eta = 0.1$	248.51	31.2	13.94	95.59	$Y_1 = 2.8, Y_5 = 3.4,$ $Y_6 = 1.2, Y_2 = Y_3 = Y_4 = Y_7$ $= Y_8 = Y_9 = Y_{10} = 0$
$\eta = 0.2$	252.07	32.4	12.98	95.53	$Y_1 = 3.1, Y_5 = 2.8,$ $Y_6 = 1.4, Y_2 = Y_3 = Y_4 = Y_7$ $= Y_8 = Y_9 = Y_{10} = 0$
$\eta = 0.3$	255.16	33.6	12.22	95.07	$Y_1 = 3.4, Y_5 = 2.2,$ $Y_6 = 1.6, Y_2 = Y_3 = Y_4 = Y_7$ $= Y_8 = Y_9 = Y_{10} = 0$
$\eta = 0.4$	259.25	36.6	11.56	94.68	$Y_1 = 3.7, Y_5 = 1.6, Y_2 = Y_3$ $= Y_4 = Y_6 = Y_7 = Y_8 = Y_9$ $= Y_{10} = 0$
$\eta = 0.5$	262.93	38.0	5.70	94.62	$Y_1 = 4, Y_5 = 1, Y_2 = Y_3$ $= Y_4 = Y_6 = Y_7 = Y_8 = Y_9$ $= Y_{10} = 0$
$\eta = 0.6$	$267.50 > 262$				Infeasible

The comparison, in Table 59.6, shows the sensitivity analysis for the optimistic-pessimistic index of the decision maker. Analytical results obtained from the Jinping-II Hydroelectric Project indicate that the optimistic-pessimistic index of the decision maker has a significant impact on the decision. Since this chapter focuses on discussing the environmental impact of the hydroelectric project, Combination 2 (i.e., $w_1 = 0.1$, $w_2 = 0.5$, $w_3 = 0.1$, $w_4 = 0.3$), which has the highest weight for environmental impact compared with the two others, is selected for this comparison. It should be noted that the optimistic-pessimistic index (i.e., η) is used to determine the combined attitude of a decision maker, which is relevant to the total project duration. For the DTCETP, the second criterion is to minimize the total project

duration, where $\eta = 1$ is the pessimistic extreme, and $\eta = 0$, to the opposite, is the optimistic extreme. Based on the results in Table 59.6, it can be concluded that a more optimistic attitude of the decision maker will lead to an optimization trend for the total project cost and total project duration. On the contrary, a more pessimistic attitude of the decision maker will lead to an optimization trend for the total crashing cost and the environmental impact. As shown in Table 59.6, if the decision maker becomes more and more optimistic (i.e., η decreases), then the total project duration (i.e., z_2) will gradually reduce. Accordingly, the total project cost will reduce as well since variable cost will largely decrease along with the project duration. As a result, the optimization of the total crashing cost and the environmental impact will be weakened in this tradeoff. These results are quite useful and may serve as a reference for decision makers in guiding current practice.

59.5.4 Comparison with Other Algorithms

Based on the preceding project examples, the (f)a-hGA is compared with some heuristic algorithms, i.e., the fuzzy-based genetic algorithm ((f)GA), and the fuzzy-based hybrid genetic algorithm ((f)hGA). In order to carry out comparisons under a similar environment, the three algorithms are programmed using the same Visual C++ language, and are run on Core i3 with the same GA parameters in which the rate of crossover and mutation are 0.6 and 0.1, respectively. The performance of the iterative process for each algorithm is shown in Fig. 59.8.

It can be concluded that the (f)a-hGA obviously performs better compared to the other two for the practical problem. It is demonstrated that the (f)a-hGA for the DTCETP can perform scheduling better than the (f)GA and (f)hGA which may lead to a local search. The convergence histories (Fig. 59.8) are based on the average

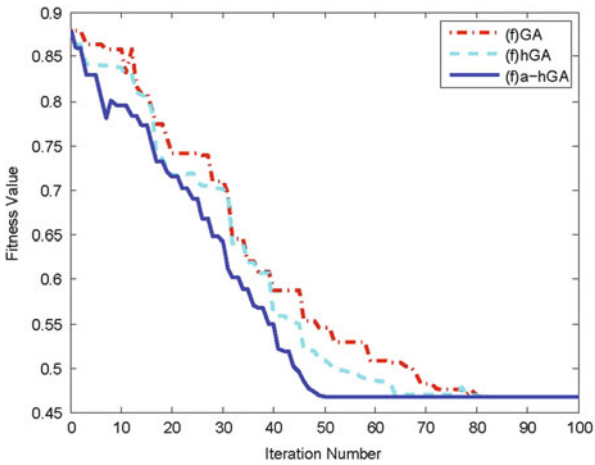


Fig. 59.8 The iterative process of application by (f)GA, (f)hGA and (f)a-hGA (Xu et al. 2012)

of the optimal results from 50 runs of the experiments (for (f)a-hGA, (f)GA, and (f)hGA, respectively) that are able to obtain the minimal fitness value, excluding the locally trapped ones.

The convergence histories in Fig. 59.8 indicate that: (1) (f)a-hGA converges a little faster, that is, requires fewer iterations than (f)GA and (f)hGA to find optimum solutions; and (2) (f)a-hGA has a more stable tendency than (f)GA and (f)hGA while searching for the optimum. Due to the one-point crossover and repairing strategy for mutations which are designed to avoid infeasible solutions, the proposed (f)a-hGA shows an improved search performance when compared to (f)GA and (f)hGA under similar conditions.

59.6 Conclusions

In this chapter, a multi-criteria DTCETP for a construction project is proposed for dealing with multi-mode project scheduling while minimizing the total project cost, project duration, crashing cost, and the environmental impact under fuzzy uncertainty. The main advantage of the proposed method is that it provides a systematic workable method for the decision-making process, enabling decision makers to control the schedule according to their optimistic-pessimistic index. The application of fuzzy variables makes the proposed multi-criteria model more suitable for describing vague situations in the real world. In order to solve the problem, an (f)a-hGA is developed to enhance the optimization quality and stability. The one-point crossover and repairing strategy for mutation are designed to avoid infeasible solutions. Finally, the Jinping-II Hydroelectric Project is used as a practical example to demonstrate the practicality and efficiency of the model. The results and a sensitivity analysis are presented to highlight the performance of the optimization method, which is very effective and efficient as compared to other algorithms. Future research will be concerned with three aspects: firstly, investigate other uncertainties, such as fuzzy random or bi-fuzzy systems to handle the model more reasonably and effectively. Secondly, more complex practical problems such as multi-project scheduling problems and more dimensions should be considered. Thirdly, more efficient heuristic methods to solve these $\mathcal{NP}$-hard problems with more constraints could be developed. Each of these areas is very important and equally worthy of attention. A detailed analysis and further research are necessary to reveal more properties for solving these problems.

Acknowledgements This research was supported by the Key Program of NSFC (Grant No. 70831005), the Research Foundation of the Ministry of Education for the Doctoral Program of Higher Education of China (Grant No. 20130181110063), and the China Scholarship Council (CSC).

References

Ahn T, Erenguc SS (1998) The resource constrained project scheduling problem with multiple crashable modes: a heuristic procedure. Eur J Oper Res 107:250–259

Akkan C, Drexl A, Kimms A (2005) Network decomposition-based benchmark results for the discrete time-cost tradeoff problem. Eur J Oper Res 165:339–358

Al-Fawzana MA, Haouari M (2005) A bi-objective model for robust resource-constrained project scheduling. Int J Prod Econ 96:175–187

Ammar MA (2011) Optimization of project time-cost trade-off problem with discounted cash flows. J Constr Eng Manag ASCE 137:65–71

Brismar A (2004) Attention to impact pathways in EISs of large dam projects. Environ Impact Assess Rev 24:59–87

Chen S, Chen B, Su M (2011) The cumulative effects of dam project on river ecosystem based on multi-scale ecological network analysis. Procedia Environ Sci 5:12–17

Chiang CM, Lai CM (2002) A study on the comprehensive indicator of indoor environment assessment for occupants' health in Taiwan. Build Environ 37:387–392

De P, James DE, Ghosh JB, Wells CE (1995) The discrete time-cost tradeoff problem revisited. Eur J Oper Res 81:225–238

De P, James DE, Ghosh JB, Wells CE (1997) Complexity of the discrete time-cost tradeoff problem for project networks. Oper Res 45:302–306

Deineko VG, Woeginger GJ (2001) Hardness of approximation of the discrete time-cost tradeoff problem. Oper Res Lett 29:207–210

Demeulemeester EL, Herroelen WS (2002) Project scheduling: a research handbook. Kluwer Academic, Dordrecht

Demeulemeester EL, Herroelen WS, Elmaghraby SE (1996) Optimal procedures for the discrete time/cost trade-off problem in project networks. Eur J Oper Res 88:50–68

Ding GKC (2008) Sustainable construction: the role of environmental assessment tools. J Environ Manag 86:451–464

Dubois D, Prade H (1988) Possibility theory: an approach to computerized processing of uncertainty. Plenum, New York

Eshtehardian E, Afshar A, Abbasnia R (2009) Fuzzy-based MOGA approach to stochastic time-cost trade-off problem. Autom Constr 18:692–701

Gangolells M, Casals M, Gasso S, Forcada N, Roca X, Fuertes A (2009) A methodology for predicting the severity of environmental impacts related to the construction process of residential buildings. Build Environ 44:558–571

Gen M, Cheng R, Lin L (2008) Network models and optimization: multiobjective genetic algorithm approach. Springer, Berlin

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Professional, Boston

Harvey RT, Patterson JH (1979) An implicit enumeration algorithm for the time/cost tradeoff problem in project network analysis. Found Control Eng 4:107–117

Hindelang TJ, Muth JF (1979) A dynamic programming algorithm for decision CPM networks. Oper Res 27:225–241

Holland JH (1992) Adaptation in natural and artificial system. MIT Press, Cambridge

Klerides E, Hadjiconstantinou E (2010) A decomposition-based stochastic programming approach for the project scheduling problem under time/cost trade-off settings and uncertain durations. Comput Oper Res 37:2131–2140

Leu SS, Chen AT, Yang CH (2001) A GA-based fuzzy optimal model for construction time-cost trade-off. Int J Proj Manag 19:47–58

Li X, Zhu Y, Zhang Z (2010) An LCA-based environmental impact assessment model for construction processes. Build Environ 45:766–775

Liang TF (2010) Applying fuzzy goal programming to project management decisions with multiple goals in uncertain environments. Expert Syst Appl 37:8499–8507

Liu KFR, Lai JH (2009) Decision-support for environmental impact assessment: a hybrid approach using fuzzy logic and fuzzy analytic network process. Expert Syst Appl 36:5119–5136

Liu L, Burns SA, Feng CW (1995) Construction time-cost trade-off analysis using LP/IP hybrid method. J Constr Eng Manag ASCE 121:446–454

Long LD, Ohsato A (2008) Fuzzy critical chain method for project scheduling under resource constraints and uncertainty. Int J Proj Manag 26:688–698

Michalewicz Z (1996) Genetic algorithm + data structure = evolution programs. Springer, Berlin

Montoya-Torres JR, Gutierrez-Franco E, Pirachican-Mayorga C (2010) Project scheduling with limited resources using a genetic algorithm. Int J Proj Manag 28:619–628

Nahmias S (1978) Fuzzy variables. Fuzzy Set Syst 1:97–110

Pagnoni A (1990) Project engineering: computer oriented planning and operational decision making. Springer, Berlin

Peche R, Rodríguez E (2009) Environmental impact assessment procedure: a new approach based on fuzzy logic. Environ Impact Assess Rev 29:275–283

Peng W, Wang C (2009) A multi-mode resource-constrained discrete time-cost tradeoff problem and its genetic algorithm based solution. Int J Proj Manag 27:600–609

Pheng LS, Chuan QT (2006) Environmental factors and work performance of project managers in the construction industry. Int J Proj Manag 24:24–37

Reschke H, Schelle H (1990) Demensions of project management. Springer, Berlin

Tareghian HR, Taheri SH (2007) A solution procedure for the discrete time, cost and quality tradeoff problem using electromagnetic scatter search. Appl Math Comput 190:1136–1145

Wang YS, Liang CY, Ju YZ (2010) Model and resolution approach for problem of multi-project time-cost trade-off. Comput Eng Appl 46:237–240

Węglarz J (1999) Project scheduling: recent models, algorithms and applications. Kluwer Academic Publishers, Dordrecht

Xu JP, Li ZM (2012) A review on ecological engineering based engineering management. Omega-Int J Manag S 40:368–378

Xu JP, Zeng ZQ (2011) A discrete time optimal control model with uncertainty for dynamic machine allocation problem and its application to manufacturing and construction industries. Appl Math Model 36:3513–3544

Xu JP, Zhou XY (2011) Fuzzy-like multiple objective decision making. Springer, Berlin

Xu JP, Zheng H, Zeng ZQ, Wu SY, Shen MB (2012) Discrete time-cost-environment trade-off problem for large-scale construction systems with multiple modes under fuzzy uncertainty and its application to Jinping-II Hydroelectric Project. Int J Proj Manag 30:950–966

Zadeh LA (1965) Fuzzy sets. Inform Control 8:338–353

Part XIX
Project Management Information Systems

Chapter 60
Impact of Project Management Information Systems on Project Performance

Louis Raymond and François Bergeron

Abstract Project management information systems (PMIS) usually acquired by organizations as software packages are meant to provide managers with the decision-making support needed in planning, organizing and controlling projects. However, the actual contribution of PMIS to project success or performance is still unknown. The purpose of this study is to empirically assess the quality of the PMIS presently used in organizations and to examine their impact on project managers and project performance, based on a PMIS success model. This model is composed of five constructs: the quality of the PMIS, the quality of the PMIS information output, the use of the PMIS, the individual impacts of the PMIS, and the impacts of the PMIS on project success. Analysis of questionnaire data obtained from 39 project managers confirms the significant contribution of PMIS to successful project management. Improvements in effectiveness and efficiency in managerial tasks were observed here in terms of better project planning, scheduling, monitoring, and control. Improvements were also observed in terms of timelier decision-making.

Keywords Information systems success • Project management • Project management information systems • Project manager • Project success

This chapter draws heavily on Raymond and Bergeron (2008).

L. Raymond (✉)
Institut de recherche sur les PME, Université du Québec à Trois-Rivières, Trois-Rivières, QC, Canada
e-mail: louis.raymond@uqtr.ca

F. Bergeron
Ecole des sciences de l'administration, TELUQ, Montreal, QC, Canada
e-mail: francois.bergeron@teluq.ca

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_30

60.1 Introduction

Globalization and the internationalization of markets have increased competitive pressures on business enterprises. This has led companies to engage in projects that are critical to their performance, if not their survival. These projects, common in industries such as engineering services, information technology, construction, and pharmaceutical have one thing in common: they need to be managed, that is, they need to be planned, staffed, organized, monitored, controlled and evaluated (Liberatore and Pollack-Johnson 2003). In order to succeed, companies must deliver projects on time and within budget, and meet specifications while managing project risk.

While large amounts of time and resources are dedicated to selecting and designing projects, it remains of paramount importance that projects be adequately managed in organizations if they are to achieve their performance objectives. In this regard, what are we to think of the management of the Athens Olympic Games, first estimated at a cost of 3 billion €, that finally ended costing 12 billion € (Scotsman.com 2005)? Of the Canadian Arms Registry, an information system first estimated at "no more than 2 million dollars a year" in 1995, that finally cost close to one billion $ (CAN) 10 years later (CTV.ca News Staff 2006)? Or of the 275 % cost overrun in Boston's Big Dig (Central Artery/Tunnel Project), totaling 11 billion $ (US) as of 2006 (Wikipedia 2007)? Even smaller projects face cost overruns: the elaborate $18 million clubhouse reconstruction at the Belmont Country Club in Massachusetts, initiated in 2011, has grown to a nearly $30 million fiasco in only 2 years, costing each member at least $28,000 (Borchers 2013). Thus, "project management *still* remains a highly problematical endeavour" (White and Fortune 2002).

In the information technology (IT) industry, Gartner Research estimates that 75 % of large IT projects managed with the support of a *project management information system* (PMIS) will succeed, while 75 % of projects without such support will fail (Light et al. 2005). Using PMIS to manage projects, while not sufficient to insure project success, has thus become a necessity (Raymond and Bergeron 2008). Project management, which has long been considered an important characteristic of successful companies (Peters and Waterman 1982), is more than ever necessary to efficiently and effectively manage these projects and to support project managers in their decision-making. As powerful project management software has been developed and diffused in all types of organizations (Trautmann, Chap. 62 of this handbook), be they large or small, private or public, extended or not (Braglia and Frosolini 2013), they are meant to make a significant contribution to project management.

Similar to other information systems (IS), a successful PMIS should have individual impacts in terms of satisfied users and effective use (Caniëls and Baken, Chap. 61 of this handbook). But a successful PMIS should also have organizational impacts, that is, impacts on project success in terms of respecting budget, schedule and specifications. While PMIS are increasingly used by project managers in all

types of industry, not much is known on the characteristics of these systems that contribute to project success. Thus the purpose of this study is first, to empirically assess the quality of the PMIS presently used in organizations and second, to examine their impact on project managers and project performance.

Having defined the research question, the remainder of the chapter is outlined as follows: The research background of the study is first presented, leading to the elaboration of a research model. The survey research methodology designed to test this model is then described. There follows a presentation and discussion of the study's results. The chapter ends with the limitations and conclusions of this research on project management information systems.

60.2 Research Background and Model

In the project management literature, IT-based information systems were deemed early on to be essential to project managers in support of their planning, organizing, control, reporting and decision making tasks. As defined by Cleland and King (1983), the basic function of a PMIS was to provide managers with "essential information on the cost-time performance parameters of a project and on the interrelationship of these parameters". The nature and role of a PMIS within a project management system, as presented in Fig. 60.1, have been characterized as fundamentally "subservient to the attainment of project goals and the implementation of project strategies" (Raymond 1987).

Notwithstanding the theoretical and practical importance of PMIS to the project management field, there have been as of yet few studies on the actual use and impacts of these systems, thus highlighting the need to extend project management theory in relation to the developing practice in this regard (Winter et al. 2006). Empirical studies of PMIS have been mostly limited to describing the demographics of project management software usage (Liberatore and Pollack-Johnson 2003) and to evaluating specific applications of these systems or software modules to support project management tasks such as planning (Amami et al. 1993), communicating and reporting (Brackett and Isbell 1989), managing risks (Jaafari 1996), scheduling (Herroelen 2005), estimating and controlling costs (Mahaney and Lederer 2010), and managing documents (Amami and Beghini 2000). Project management software usage has also been found to have many drawbacks and limitations, both in theory when compared to an ideal PMIS by researchers (Jaafari and Manivong 1998) and in practice as perceived by project managers (White and Fortune 2002).

An IS-based conceptualisation and definition of project management software facilitates the import of knowledge from the IS field or discipline, knowledge that can provide a deeper understanding of the PMIS usage phenomenon and help in answering questions on the factors that explain the use and non use of PMIS, and on the actual impacts of these systems on project managers and project performance. This study will thus be founded on the recurrent constructs of antecedents and consequences of IS use developed in DeLone and McLone's (1992) IS success

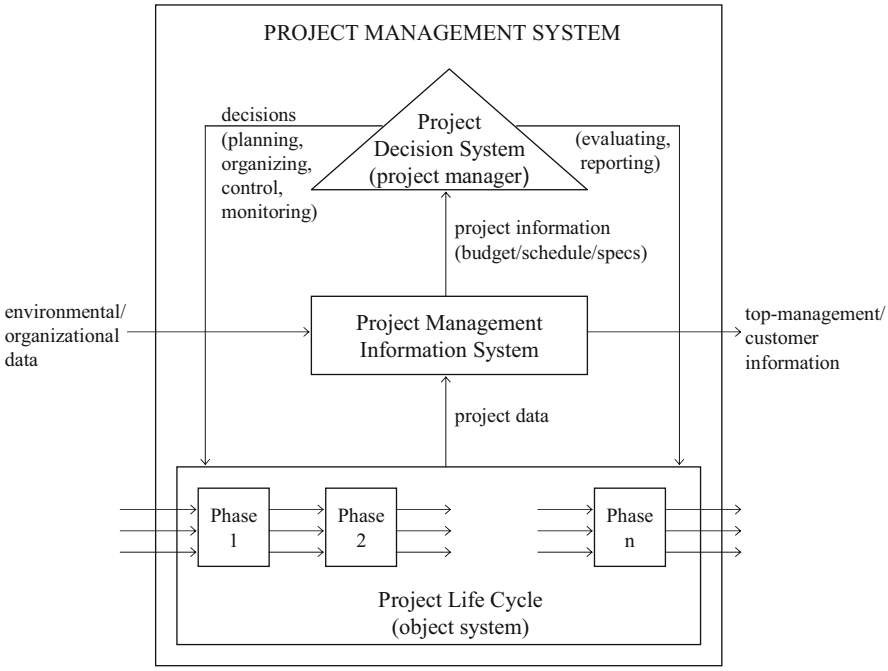


Fig. 60.1 The PMIS within the project management system (*source*: Raymond 1987)

model (ISSM), later updated (DeLone and McLean 2003), and in Davis' (1989) technology acceptance model (TAM) (Davis et al. 1989). These models stand out by the continuance of their constructs, after a review of theories and models of IS use that focused on their chronological examination and their cross-influences and convergences. The ISSM incorporates information quality and system quality as antecedents of IS use, leading to individual IS impacts, that is, on users and their work (e.g., in regard to their effectiveness), and in turn to organizational impacts (e.g., in regard to business strategy and performance). While the TAM explains IS use in a similar manner by the system's perceived usefulness and perceived ease of use. Both the ISSM and the TAM offer widely accepted and validated representations and explanations of the IS use phenomenon (Larsen 2003; Lee et al. 2003; Rai 2002).

Our objective is thus to improve our understanding of the impacts of PMIS on project managers and on project performance. More specifically, one intends to ascertain the success of these systems, i.e., their level of use by project managers, as determined by the quality of PMIS and of the information they provide. One will also ascertain to what extent PMIS contribute to the successful completion of projects through their individual and organizational impacts. Indeed, one aims to verify if the use of a PMIS is related to efficiency, productivity and effectiveness of a project manager, and to the performance of the project itself. Thus, the following

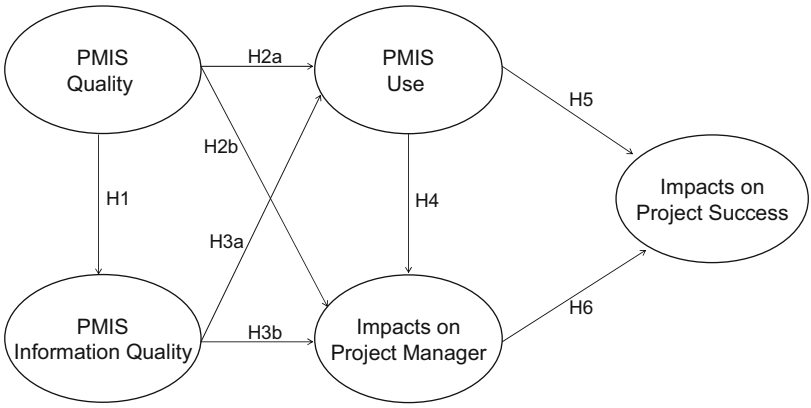


Fig. 60.2 Research model on project management IS success

research questions: What are the main determinants of the success of the PMIS currently used? Does the use of PMIS increase the efficiency, productivity, and effectiveness of project managers? And what is the contribution of the PMIS to project success?

Given the research questions, an adaptation to—and specification for—project management of the ISSM and the TAM was deemed to be most appropriate. As presented in Fig. 60.2, the model as adapted and specified is composed of five constructs, namely the quality of the PMIS, the quality of the PMIS information output, the use of the PMIS, the individual impacts of the PMIS, and the impacts of the PMIS on project success, linked through research hypotheses that follow.

Hypothesis 1: Greater PMIS quality is associated to greater quality of information output by the system. This first hypothesis is based on empirical evidence linking the technical and service aspects of an information system (e.g., ease of use, response time) to the user's satisfaction with the information output by the system (e.g., perceived usefulness, timeliness of the information) (Igbaria et al. 1995).

Hypothesis 2: Greater PMIS quality is associated to greater system use (H2a) and greater system impacts on the project manager (H2b). In applying IS theory and results to project management, one finds that previous empirical tests of the ISSM and the TAM have shown system quality to positively influence system use and positively affect individual user performance in terms of job effectiveness, quality of work and decision-making (Bergeron et al. 1995; Igbaria et al. 1997; Taylor and Todd 1995; Weill and Vitale 1999).

Hypothesis 3: Greater quality of the information output by the PMIS is associated to greater system use (H3a) and greater system impacts on the project manager (H3b). The third hypothesis extends to project management the notion that the managers' use of IT-based information systems and their performance are dependent upon the quality of information provided to them by these systems (Bergeron et al. 1995; Etezadi-Amoli and Farhoomand 1996; Teng and Calhoun 1996; Wixom and Watson 2001).

Hypothesis 4: Greater use of the PMIS is associated to greater system impacts on the project manager. A number of IS studies have demonstrated that the depth and breadth of IS use (e.g., usage dependency, pattern and frequency), if voluntary and appropriate to the task, has positive impacts on users in terms of job performance and decision-making performance (Igbaria and Tan 1997; Seddon and Kiew 1994; Torkzadeh and Doll 1999; Yuthas and Young 1998).

Hypothesis 5: Greater use of the PMIS is associated to greater impacts of the PMIS on project success. A number of IS researchers believe that the quality and intensity of information system use, and the "full functionality" of this use in particular, are essential to the achievement of desired organizational results or to the realization of anticipated organizational benefits (DeLone and McLean 2003; van der Meijden et al. 2003).

Hypothesis 6: Greater impacts of the PMIS on the project manager are associated to greater impacts of the PMIS on project success. This last hypothesis is based on IS theory and evidence that the organizational impacts results not only from IS use but also from the individual impacts of the system (Jurison 1996; Teo and Wong 1998), i.e., that projects led by more efficient and effective managers, due to their use of a PMIS, tend to be more successful in terms of meeting project schedules, budgets, and specifications.

60.3 Research Methodology

To test the research model, a survey of 224 project managers and project management consultants was conducted, identified from a list of participants to a project management national conference held in Canada. The questionnaire was sent by e-mail, as electronic surveys allow the transmission of more information, they support a better interaction between the researchers and the respondents, and they contribute to a better quality of information, to a faster response cycle, and to a reduction in research costs (Tse 1998; Klassen and Jacobs 2001). Forty five questionnaires were received, out of which 39 were considered valid, thus a 17.4 % final response rate. Information on the respondent organizations and on the respondents' demographics is presented in Table 60.1.

The information quality, system quality, and system use constructs were measured by adapting to the specific PMIS context instruments previously developed and validated in a general IS context (Bergeron et al. 1995; Wang and Strong 1996). The quality of the PMIS was measured with eight items: accessibility, response time, flexibility, ease of use, querying ease, learning ease, systems integration, and multi-project capability. Each of these items was measured on a five-point scale varying from 1 (low quality) to 5 (high quality). The quality of information was measured with six items: availability, relevance, reliability, precision, comprehensiveness, and security. Each of these items was measured on a five-point scale varying from 1 (low quality) to 5 (high quality).

Table 60.1 Characteristics of the sample

Characterization of the respondents ($n = 39$)	% of sample
Sector of respondents' organization	
Services	74 %
Manufacturing	13 %
Public sector	8 %
Construction	5 %
Function	
Project manager/director/coordinator	51 %
Project manager consultant/senior advisor	23 %
Top-manager (general manager, president, vice-president)	13 %
Project engineer/analyst	13 %
Membership in a professional association	
Yes (Project Management Institute mostly)	85 %
No	15 %
Education level	
Master's degree	43 %
Bachelor's degree	41 %
College degree	16 %
Gender	
Male	79 %
Female	21 %
Project management experience	
More than 30 years	25 %
20–30 years	41 %
10–20 years	31 %
Less than 10 years	3 %
PMIS software used (38 % use more than 1)	
MS Project	90 %
Work Bench	15 %
Primavera	10 %

The use of the PMIS was measured by ascertaining the extent to which various system functions and their associated tools were actually used by project managers. The PMIS functions were divided into five categories. The planning function tools aim at preparing the overall project plan; they include work breakdown structure, resource estimation, overall schedule, Gantt, PERT, and CPM. The monitoring function tools are used to regularly assess project progress; they are used for progress reports and curves, and to update operational reports such as completed tasks, percent project completed, effective schedule, remaining tasks, and remaining days to complete. The controlling function tools are used to make specific changes to the project; they allow the project manager to fine-tune forecasts, modify tasks, reassign resources to lower the costs, cancel tasks, and modify the cost of resources.

The evaluating function tools are targeted toward project auditing; these tools allow the identification of cost and schedule variations, and tracking the use of resources. The reporting function tools give information on the most basic aspects of the project; they include an overview of the project as well as reports on work-in-progress, budget overruns, and task and schedule slippages. A score for each category was obtained by averaging the project managers' use of specific tools. The five categories and their specific number of tools are: planning (6), monitoring (7), controlling (6), evaluating (2), and reporting (9). Five-point scales were employed: 1 (never used), 2 (rarely used), 3 (occasionally used), 4 (often used), and 5 (very often used).

Impacts on the project managers were measured by the perceived effect of the PMIS on the following ten items: improvement of productivity at work, increase in the quality of decisions, reduction of the time required for decision-making, reduction of the time required to complete a task, improved control of activity costs, better management of budgets, improved planning of activities, better monitoring of activities, more efficient resource allocation, and better monitoring of the project schedule (Lalonde and St-Pierre 2000). A five-point Likert scale was used, varying from 1 (completely disagree) to 5 (completely agree). The impacts of the PMIS on project success was based on the perceived contribution of the PMIS with regard to three performance criteria: respecting deadlines, respecting budgets, and respecting quality specifications (Shenhar et al. 1997), using a five-point scale varying from 1 (null contribution) to 5 (very high contribution).

60.4 Results and Discussion

Descriptive results on the antecedents, consequences and nature of PMIS use by the respondents are presented in Table 60.2.

60.4.1 Test of the Measurement Model

To test the multivariate relationships hypothesised by the research model, structural equation modelling was used. The partial-least-squares (PLS) method was chosen for its robustness as it does not require a large sample or normally distributed multivariate data in comparison to covariance structure methods such as LISREL and EQS (Fornell and Larcker 1981). Figure 60.3 summarizes the results obtained. The PLS method simultaneously assesses the theoretical propositions and the properties of the underlying measurement model. Note that PLS does not provide goodness-of-fit indices; model fit is rather assessed by the reliability of each construct, the significance of the path coefficients, and the percentage of variance explained (R^2) for each dependent construct (Gefen et al. 2000).

Table 60.2 Characteristics of the respondents' use of PMIS

Respondents' characterization of PMIS ($n = 39$)	% of sample
Experience in the use of PMIS more than 6 years	36 %
3–6 years	54 %
1–3 years	8 %
Less than 1 year	2 %
Most important indicator of PMIS Quality	
Ease of use	33 %
Flexibility	23 %
Accessibility	23 %
Satisfaction with PMIS Quality	
Very high	13 %
High	48 %
Project Manager work indicator most impacted by PMIS	
Better monitoring of activities	46 %
Better planning of activities	41 %
Increase in productivity at work	39 %
Satisfaction with Information Quality	
Very high	18 %
High	48 %
Impact of PMIS on project manager's work	
Very high	13 %
High	51 %
Project Success indicator most impacted by PMIS	
Meeting deadlines	59 %
Respecting budgets	41 %
Meeting project specifications	10 %

Internal consistency of measures, i.e., their unidimensionality and their reliability must be verified first. The observable variables measuring a non-observable construct (or latent variable) must be unidimensional to be considered unique values. Unidimensionality is usually satisfied by retaining variables whose loadings (λ) are above 0.5, indicating that they share sufficient variance with their related construct. The unidimensionality criteria are thus met. Reliability can be verified by considering the value of the rho (ρ) coefficient, defined as the ratio between the square of the sum of the loadings plus the sum of the errors due to construct variance. A ρ greater than 0.7 indicates that the variance of a given construct explains at least 70 % of the variance of the corresponding measure, as is the case in Table 60.3 for all constructs in the research model.

There is also evidence in Table 60.3 of the convergent validity of the constructs, as their average variance extracted ranges from 0.72 to 0.83 in value. The last property to be verified is discriminant validity. It shows the extent to which each

Table 60.3 Reliability and discriminant validity of the research constructs

Variable	ρ[a]	1.	2.	3.	4.	5.
1. PMIS quality	0.96	0.74[b]				
2. PMIS information quality	0.97	0.69	0.83			
3. PMIS use	0.95	0.37	0.49	0.77		
4. Impacts on project manager	0.96	0.71	0.72	0.66	0.72	
5. Impacts on project success	0.92	0.46	0.41	0.48	0.71	0.79

[a]Reliability coefficient $= (\Sigma\lambda_i)^2/((\Sigma\lambda_i)^2 + \Sigma(1\text{-}\lambda_i{}^2))$
[b]Diagonal: average variance extracted $= (\Sigma\lambda_i{}^2/n)$
Sub-diagonals: shared variance = (correlation)^2

construct in the research model is unique and different from the others. The shared variance between a construct and other constructs (i.e., the squared correlation between two constructs) must be less than the average variance extracted (i.e., the average variance shared between a construct and its measures). Table 60.3 shows this to be the case for all constructs.

60.4.2 Test of the Theoretical Model

The research hypotheses are tested by analyzing the direction, the value and the level of significance of the path coefficients (betas) estimated by the PLS method, as presented in Fig. 60.3. The high percentage of variance explained in each dependent construct, varying from 0.49 to 0.83, is indicative of model fit.

H1—A positive and highly significant path coefficient ($\beta = 0.83$) confirms that the quality of information output by a PMIS is strongly associated to the technical and service aspects of the system, that is, to system quality. From the project manager's point of view, the PMIS cannot be considered simply as a "black box" but must be evaluated for its level of sophistication and support provided by the organization's IS function and by the system providers, be they inside or outside the organization.

H2—The second hypothesis could not be confirmed as PMIS quality was not found to directly influence the use of the system ($\beta = 0.09$), nor its impacts on the project manager ($\beta = 0.20$). There are however a significant indirect effect of system quality on system use (equal to 0.83×0.62) and on impacts on the project manager (equal to 0.83×0.40), that is, through the mediating influence of information quality.

H3—The third hypothesis, presuming a positive influence of the quality of information provided by the PMIS upon the use of the system and its impacts on the project manager is confirmed. Indeed, the quality of information output is significantly related to the use of the PMIS by project managers (H3a, $\beta = 0.62$). Path analysis also confirms the existence of a significant relation between the

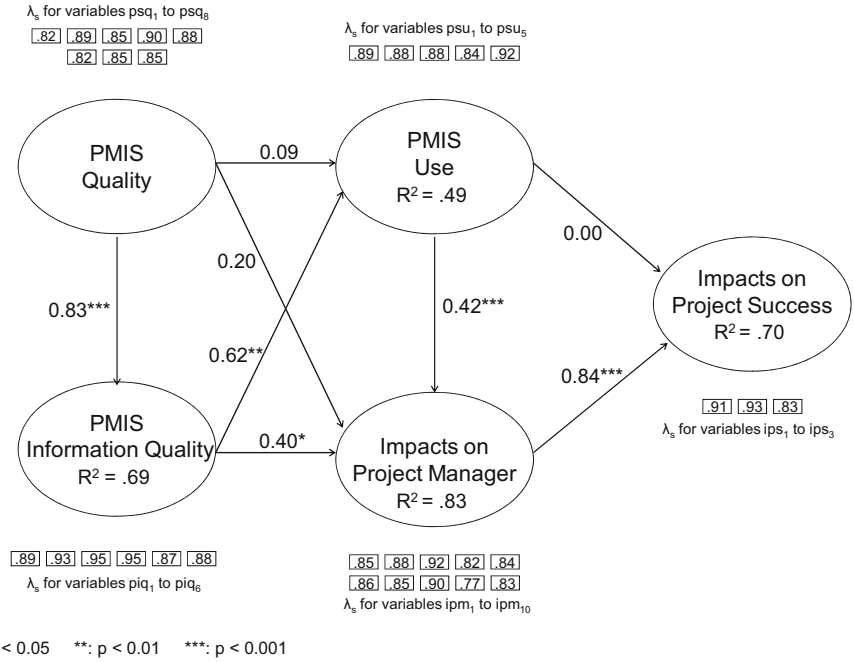


Fig. 60.3 Results of evaluating the research model with PLS

quality of information output and the system's impacts on project managers (H3b, $\beta = 0.40$). Hence a PMIS must provide information on project costs, resources, and milestones that is perceived to be relevant, reliable and accurate by project managers if they are to use these systems in their planning, controlling, monitoring, and reporting tasks and if they are to be more efficient and effective in accomplishing these tasks.

H4—Testing the fourth hypothesis confirmed that the use of a PMIS is positively related to its impacts on the project manager ($\beta = 0.42$). In other words, the use of a PMIS by project managers increases their productivity, effectiveness and efficiency in decision-making due to the quality of the information output by the PMIS. Therefore, using project management software tools that enhance their capacity to plan, control, monitor, audit, and report provides tangible benefits to project managers and improves the quality of their work.

H5—The fifth hypothesis could not be confirmed as no direct relationship was found between PMIS use and the system's impacts on project success ($\beta = 0.00$). Significant improvements in project performance in terms of meeting deadlines, respecting budgets, and meeting specifications can be obtained indirectly however, through the system's impacts on project managers.

H6—Results confirmed the positive association between the impact of PMIS on the project manager and the impact of PMIS on project success ($\beta = 0.84$). Hence, the more project managers perceive their task to be positively impacted by their use of project management software, greater is their belief in the positive contribution of this software to the attainment of their projects' performance objectives.

60.4.3 Discussion

The objective of this research is to have a better understanding of the elements that contribute to the impact of a PMIS on project success. The study results are discussed in terms of direct and indirect effects on PMIS project success. To ease the discussion, the elements are grouped in three dimensions: technical (PMIS quality and quality of information), managerial (PMIS use and impact on project manager), and organizational (PMIS impact on project success).

At the technical level, the first element indirectly influencing the impact of a PMIS on project success is PMIS quality. The system's ease of use, flexibility, response time, learning ease and system integration play an important role in producing quality information, as perceived by the project manager. Indeed, PMIS quality is a strong predictor of the quality of information to be obtained from the system. In the case of a higher-quality PMIS, the information output is more available, reliable, precise, comprehensive, and secure. Conversely, a PMIS that produces information of poor quality would be a system that is more difficult to use, less flexible, and less integrated to other organizational information systems used by the project manager and other managers or employees. This means that project information quality requires sophisticated, well-serviced information systems.

The quality of information is directly and strongly related to PMIS use and to the system's impacts on the project manager. Information quality is not an end by itself however, as it leads only indirectly to project success. At the managerial level, it is only through the actual use of the PMIS by—and the system's impacts on—the project manager that the quality of information can influence project success. Better quality of information output increases the opportunity of the PMIS being used, which in turn allows the system to have a positive impact on the project manager. As such, the quality of information output by the PMIS leverages the project manager's work as a professional. The latter will feel more professional at work if he or she has access to project information of high quality and uses the system more intensively and more extensively for the planning, control, monitoring, and reporting activities. This combination of quality information and extensive use of the system allows the project manager to feel more productive at work and provides improved support for decision-making.

This leads us to the final relationship, at the organizational level, specifically the impacts of the PMIS on project success. First, the PMIS itself has no direct influence upon project success; it is only through higher-quality information, extensive use of

the system, and individual impacts on the project manager that the system has an effect on project success. While a positive impact on managerial work is essential to project success, greater use of a PMIS does not lead per se to greater impacts on project performance. It is only indirectly, through its contribution to managerial work that this use contributes to project success. In summary, if it is to make a significant contribution to the attainment of project objectives, i.e., to make an impact in terms of project budget, schedule, and specifications, a PMIS must first be sufficiently sophisticated and serviced and produce information of sufficient quality. It must then be used with sufficient depth and breadth by project managers and it must have a sufficiently beneficial impact on their work.

It is also worth noting that among the managers who participated in the study, a number indicated strong impacts of the PMIS upon the successful completion of their projects, while others did not. The results indicate that, in general, the latter depended upon a PMIS of lower quality that produced lower quality information; hence they used their system less and were less supported in their project management task. Whereas generally speaking, the former were those for whom the sufficient conditions were met, that is, PMIS quality, information output quality, PMIS use, and positive impacts on managerial work.

Additional comments can be made in explaining these relationships. First, it is worth noting that a reverse or "feedback" relationship is possible between individual impacts of a PMIS and its use (DeLone and McLean 2003). As project managers perceive the PMIS to be beneficial to them, it is likely that they will increase their use of the system. Second, other dimensions of project management, related to the organizational environment, evidently play a role in explaining project performance; thus the managers' authority on project activities, their involvement in project design, and their accountability in meeting project objectives are potential success factors other than the PMIS (Bergeron 1986). Third, another interesting aspect to consider is the possible reluctance of project managers to report "bad news" on a project, and the subsequent effect it could have on the accuracy of project reports and on the assessment of project success (Keil et al. 2007). Finally, as suggested by Shenhar et al. (1997), future studies of PMIS success could evaluate project success or performance from the client's perspective, that is, evaluate if the impacts of the PMIS on project outcomes provide an adequate solution to the client's problem, bring true advantages to the organization in terms of quality of product/services offered, greater output volume, quicker delivery and better strategic positioning, and provide tangible benefits such as increased sales and revenues.

60.4.4 Limitations

This research has limitations however. First, the convenient rather than random nature of the sample and its small size impose care in the generalizing the results of this study to all project managers. Second, an electronic questionnaire limits the number of questions and variables that can be addressed and, being

self-administered, is subject to respondent bias. While the items measuring PMIS quality, information quality, PMIS use, PMIS impacts on the project manager, and PMIS impacts on project success were placed in separate parts of the questionnaire to mitigate autocorrelation effects, other sources of common method or mono-method biases may yet remain in the survey instrument (Podsakoff et al. 2003). Third, to lessen this bias, one could have used additional objective rather than perceptual measures of the impact of the PMIS on project success; this would have been particularly interesting for the productivity measures. Finally, as the nature of the study is cross-sectional rather than longitudinal, causality cannot be inferred.

60.5 Conclusions

The research aim of this study was to determine the actual impacts of IT-based project management information systems upon project managers and project performance. More specifically, one objective was to identify the main determinants of PMIS and determine the extent to which these systems assist project managers in terms of increased efficiency, productivity, and efficiency. Another objective was to get a better understanding of the contribution of these systems to the success of projects.

Following the conclusions of previous research that PMIS success models should continue to be validated and challenged, the results of this research show that the use of a project management information system is in fact advantageous to project managers. Improvements in effectiveness and efficiency in managerial tasks were observed here in terms of better project planning, scheduling, monitoring, and control. Improvements in productivity were also observed in terms of timelier decision-making. Advantages obtained from PMIS use are not limited to individual performance but also include project performance. These systems were found to have direct impacts on project success, as they contribute to improving budget control and meeting project deadlines as well as fulfilling technical specifications. One can therefore conclude that PMIS make a significant contribution to project success and should continue to be the object of project management research.

References

Amami M, Beghini G (2000) Project management and communication of product development through electronic document management. Proj Manag J 31(2):6–19

Amami M, Beghini G, La Manna M (1993) Use of project management information system for planning information systems development projects. Int J Proj Manag 11(1):21–28

Bergeron F (1986) Factors influencing the use of DP chargeback information. MIS Q 10(3):225–238

Bergeron F, Raymond L, Rivard S, Gara M-F (1995) Determinants of EIS use: testing a behavioral model. Decis Support Syst 14:131–146

Borchers C (2013) Clubby atmosphere vanishes as costs escalate in Belmont. The Boston Globe, May 21

Brackett SW, Isbell A-M (1989) PMIS: an integrated approach for the management and distribution of project information. Proj Manag J 20(3):5–10

Braglia M, Frosolini M (2013) An integrated approach to implement project management information systems within the extended enterprise. Int J Proj Manag. Available online 16 January. http://dx.doi.org/10.1016/j.ijproman.2012.12.003

Cleland DJ, King WR (1983) Systems analysis and project management. McGraw-Hill, New York

CTV.ca News Staff (2006) Conservatives move to disarm long-run registry. June 19. http://www.ctv.ca

Davis FD, Bagozzi RP, Warshaw PR (1989) User acceptance of computer technology: a comparison of two theoretical models. Manag Sci 35(8):982–1003

DeLone WH, McLean ER (1992) Information systems success: the quest for the dependent variable. Inform Syst Res 3(1):60–95

DeLone WH, McLean ER (2003) The DeLone and McLean model of information systems success: a ten-year update. J Manag Inform Syst 19(4):9–30

Etezadi-Amoli J, Farhoomand AF (1996) A structural model of end user computing satisfaction and user performance. Inform Manag 30(2):65–73

Fornell CR, Larcker DF (1981) Evaluating structural equation models: LISREL and PLS applied to consumer exit-voice theory. J Mark Res 18:39–50

Gefen D, Straub DW, Boudreau M-C (2000) Structural equation modeling and regression: guidelines for research practice. Commun AIS 4(7):1–76

Herroelen W (2005) Project scheduling - theory and practice. Prod Oper Manag 14(4):413–432

Igbaria M, Tan M (1997) The consequences of information technology acceptance on subsequent individual performance. Inform Manag 32(3):113–121

Igbaria M, Iivari J, Maragahh H (1995) Why do individuals use computer technology? A Finnish case study. Inform Manag 29(5):227–238

Igbaria M, Zinatelli N, Cragg P, Cavaye A (1997) Personal computing acceptance factors on small firms: a structural equation model. MIS Q 21(3):279–302

Jaafari A (1996) Time and priority allocation scheduling technique for projects. Int J Proj Manag 14(5):289–299

Jaafari A, Manivong K (1998) Toward a smart project management information system. Int J Proj Manag 16(4):249–265

Jurison J (1996) The temporal nature of IS benefits: a longitudinal study. Inform Manag 30(2):75–79

Keil M, Im GP, Mähring M (2007) Reporting bad news on software projects: the effects of culturally constituted views of face-saving. Inform Syst J 17(1):59–87

Klassen RD, Jacobs J (2001) Experimental comparison of Web, electronic and mail survey technologies in operations management. J Oper Manag 19(6):713–728

Lalonde B, St-Pierre A (2000) Microsoft Project 2000: La gestion informatisée de projet. Editions Vermette, Montréal

Larsen KRT (2003) A taxonomy of antecedents of information systems success: variable analysis studies. J Manag Inform Syst 20(2):169–246

Lee Y, Kozar KA, Larsen KRT (2003) The technology acceptance model: past, present, and future. Commun AIS 12:752–780

Liberatore MJ, Pollack-Johnson B (2003) Factors influencing the usage and selection of project management software. IEEE Trans Eng Manag 50(2):164–174

Light M, Rosser B, Hayward S (2005) Realizing the benefits of projects and portfolio management. Gartner, Research ID G00125673, pp 1–31 (January 5)

Mahaney RC, Lederer AL (2010) The role of monitoring and shirking in information systems project management. Int J Proj Manag 28(1):14–25

Peters T, Waterman DL (1982) In search of excellence. Warners Books, New York

Podsakoff PM, MacKenzie SB, Lee J-Y, Podsakoff NP (2003) Common method biases in behavioral research: a critical review of the literature and recommended remedies. J Appl Psychol 88(5):879–903

Rai A, Lang SS, Welker RB (2002) Assessing the validity of IS success models: an empirical test and theoretical analysis. Inform Syst Res 13(1):50–69

Raymond L (1987) Information systems design for project management: a data modeling approach. Proj Manag J 18(4):94–99

Raymond L, Bergeron F (2008) Project management information systems: an empirical study of their impact on project managers and project success. Int J Proj Manag 26(2):213–220

Scotsman.com (2005) Athens counting cost of the Olympics. The Scotsman, August 4. http://news.scotsman.com/topics.cfm?tid=1137&id=1726242005

Seddon PB, Kiew M-Y (1994) A partial test and development of the DeLone and McLean model of IS success. In: Proceedings of the fifteenth international conference on information systems, Atlanta, pp 99–110

Shenhar AJ, Levy O, Dvir D (1997) Mapping the dimensions of project success. Proj Manag J 28(2):5–13

Taylor S, Todd P (1995) Understanding information technology usage: a test of competing models. Inform Syst Res 6(2):144–176

Teng J, Calhoun K (1996) Organizational computing as a facilitator of operational and managerial decision making: an exploratory study of managers' perceptions. Decis Sci 27(4):673–710

Teo TSH, Wong PK (1998) An empirical study of the performance impact of computerization in the retail industry. Omega 26(5):611–621

Torkzadeh G, Doll WJ (1999) The development of a tool for measuring the perceived impact of information technology on work. Omega 27(3):327–339

Tse A (1998) Comparing the response rate, response speed and response quality of two methods of sending questionnaires: E-mail vs. mail. J Mark Res Soc 40(1):353–361

van der Meijden MJ, Tange HJ, Troost J, Hasman A (2003) Determinants of success of inpatient clinical information systems: a literature review. J Am Med Inform Assn 10(3):235–243

Wang RY, Strong DM (1996) Beyond accuracy: what data quality means to data consumers. J Manag Inform Syst 12(4):5–34

Weill P, Vitale M (1999) Assessing the health of an information system portfolio: an example from process engineering. MIS Q 23(4):601–624

White D, Fortune J (2002) Current practice in project management: an empirical study. Int J Proj Manag 20:1–11

Wikipedia (2007) Cost Overrun. http://en.wikipedia.org/wiki/Cost_Overrun. Accessed 23 Feb 2007

Winter M, Smith C, Morris P (2006) Directions for future research in project management: the main findings of a UK government-funded research network. Int J Proj Manag 24:638–649

Wixom BH, Watson HJ (2001) An empirical investigation of the factors affecting data warehousing success. MIS Q 25(1):17–41

Yuthas K, Young ST (1998) Material matters: assessing the effectiveness of materials management IS. Inform Manag 33(3):115–124

Chapter 61
Project Management Information Systems in a Multi-Project Environment

Marjolein C.J. Caniëls and Ralph J.J.M. Bakens

Abstract Project management information systems (PMIS) should provide project managers with decision making support for planning, organizing, and controlling projects. Most project managers are dissatisfied with the information produced by PMIS. Based on a survey among 101 project managers the interactions between six factors related to PMIS information quality and usage and their effect on decision making are examined in a multi-project environment. Using structural equation modeling, new insights were gained in these complex relationships. Results indicate that the use of a project management information system is advantageous to project managers, while no adverse effects were observed due to project and information overload. PMIS information quality is positively related to quality of the decisions, satisfaction of project managers with PMIS and use of PMIS information. Simultaneous handling of multiple projects causes project managers to extend conclusions about the information quality for one project to all projects at hand.

Keywords Information quality • Multi-project environment • Project management • Project management information systems • Structural equations modelling

61.1 Introduction

The current business environment is complex. Managers need to make fast decisions, allocate scarce resources efficiently, and have a clear focus. In organizations that are engaged in many projects simultaneously, management is faced with

This chapter draws heavily on Caniëls and Bakens (2012). Copyrights for reuse were cleared.

M.C.J. Caniëls (✉) • R.J.J.M. Bakens
Faculty of Management, Science and Technology, Open University of the Netherlands, DL Heerlen, The Netherlands
e-mail: marjolein.caniels@ou.nl; fam.bakens@upcmail.nl

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_31

multiple challenges (Elonen and Artto 2003). Project managers handling different projects with different scopes, complexities and timelines face particular problems. These can be related to resource conflicts and throughput times (Maylor et al. 2006; Platje and Seidel 1993). Inadequate balancing of scarce resources often results in additional pressure on the organization, which leads to poor quality of information and longer lead times of projects (Elonen and Artto 2003). Interdependencies and interactions between projects (Patanakul and Milosevic 2008b) and information and project overload (Engwall and Jerbrant 2003; Zika-Viktorsson et al. 2006) present specific challenges as well. Managers may become overwhelmed by the amount of information that is available for decision making, and therefore they may lose sight of relevant information or be unaware of inaccuracies.

In general, poor information quality leads to poor decision making (Blichfeldt and Eskerod 2008; Elonen and Artto 2003; Engwall and Jerbrant 2003). The use of Project Management Information Systems (PMIS) is considered advantageous to project managers because of the alleged contribution regarding timelier decision making and project success (Raymond and Bergeron 2008). The implementation of PMIS in a multi-project environment may help to accomplish a realistic project assignment, which is an effective strategy when managing multiple projects (Patanakul and Milosevic 2008a).

Studies on the use of PMIS have predominantly focused on single projects with high complexity, and PMIS are considered advantageous in such environments (Raymond and Bergeron 2008). Project managers who deal with single projects that are less complex may not be willing to use PMIS, because the time they have to invest in keeping the system up to date may exceed the benefits gained from utilizing the system (Ali and Money 2005; Bendoly and Swink 2007). However, little research has been done to find out whether project managers handling multiple but less complex projects benefit from PMIS. The objective of our study is to gain better understanding of the elements of PMIS that contribute to adequate decision making in a multi-project environment, and to provide insights in the relationship between PMIS information quality and the project manager's satisfaction with PMIS.

In this study we define a multi-project environment as a setting in which project managers are in charge of several (more than one) projects on the operational level at the same time (see also Zika-Viktorsson et al. 2006 for characteristics of a multi-project setting). Hence, a project manager simultaneously supervises several teams performing product development work according to a project specific delivery plan. Multi-project managers allocate resources to various projects on a short term basis in an attempt to achieve maximum progress for each project. Multi-project management differs from project portfolio management. Whereas portfolio managers have projects that are strategically related, the projects of a multi-project manager may be related on a strategic level, but projects may also be independent strategically, and only share scarce time and resources with other projects (Dye and Pennypacker 2000).

Concrete, this study is of an empirical nature and aims to identify and quantify the effects of PMIS information use on decision making in a multi-project environment, as perceived by project managers. PMIS information use is seen as a function of PMIS satisfaction and the quality of PMIS information. On the basis of a survey among 101 project managers in a multinational pharmaceutical company this study will provide insights in the problems that project managers encounter in a multi-project environment, namely:

1. The extent to which PMIS information quality is perceived by project managers to contribute to enhanced decision making in a multi-project environment. PMIS information quality reflects whether the information generated by the PMIS is perceived to be readily at one's disposal (available); sound and dependable (reliable); closely connected or appropriate to the matter in hand (relevant); correct in all details (accurate) and understandable (comprehensible) (Zmud 1978; O'Reilly 1980).
2. The extent to which project overload and information overload is perceived by project managers to influence the quality of PMIS information.

The organization of this chapter is as follows. The next section will review the literature about project management, PMIS, and the factors that influence decision making in a multi-project environment. This section will also introduce the research model. Subsequently, the research methodology will be presented. Then, the results are reported, followed by the discussion and conclusion, and limitations and issues for further research.

61.2 Literature Review

61.2.1 (Multi) Project Management

Project management "covers all project management processes that are related to planning, controlling, and coordinating projects" (Ahlemann 2009, pp. 19–20). Project management is an intricate task regarding the complexity, uncertainties, and large number of activities involved, even in a single-project environment (Mota et al. 2009). In a multi-project environment it is common that one project manager leads multiple concurrent projects at the same time (Patanakul and Milosevic 2008a).

Issues related to (multi) project management are addressed in many studies, see Table 61.1 for an overview. Empirical studies regarding (multi) project management have largely focused on resource allocation issues (Blichfeldt and Eskerod 2008; Hendriks et al. 1999; Laslo and Goldberg 2008; Payne 1995; Yaghootkar and Gil 2012; Yang and Fu 2014), managerial problems in the form of delayed projects, stress and lack of overview (Blichfeldt and Eskerod 2008; Patanakul 2013), differences between single and multi-project environment (Aritua et al. 2009), projectification and programmification (Maylor et al. 2006), and planning

Table 61.1 Overview of studies on project management and PMIS

References	Studied areas (1) Single-project management, (2) Multi-project management, (3) PMIS, (4) Project overload, (5) Information overload, (6) Information quality, (7) Satisfaction with IS, (8) IS use, (9) Decision making								
	1	2	3	4	5	6	7	8	9
Ahlemann (2009)			x						
Ali and Money (2005)	x					x	x	x	
Ali et al. (2008)	x					x	x	x	
Aritua et al. (2009)		x							
Atkinson (1999)	x								
Blichfeldt and Eskerod (2008)		x							
Canonico and Söderlund (2010)	x								
Cooper et al. (2001)		x				x			x
DeLone and McLean (2003)						x	x	x	
Dietrich and Lehtonen (2005)	x	x				x			x
Dvir et al. (2003)	x								
Engwall and Jerbrant (2003)		x		x					
Hendriks et al. (1999)	x								x
Laslo and Goldberg (2008)	x	x							
Martinsuo and Lehtonen (2007)		x							x
Maylor et al. (2006)		x							
Mota et al. (2009)	x	x							x
O'Reilly (1980)					x				
Patanakul (2013)		x							x
Patanakul and Milosevic (2008a)		x							
Patanakul and Milosevic (2008b)		x							
Payne (1995)		x							
Platje and Seidel (1993)		x							
Platje et al. (1994)		x							
Raymond (1987)	x		x						x
Raymond and Bergeron (2008)	x		x			x		x	x
Saeed and Abdinnour-Helm (2008)			x			x			x
Seddon and Kiew (1994)						x	x	x	
Turner and Speiser (1992)		x							
Yaghootkar and Gil (2012)		x							
Yang and Fu (2014)		x							
Zika-Viktorsson et al. (2006)		x		x					

and control (Canonico and Söderlund 2010; Dvir et al. 2003; Platje and Seidel 1993; Platje et al. 1994; Turner and Speiser 1992). All these studies have in common that they focus on organization design and the management of projects. However, no study has examined the use of PMIS for multi-project management.

In a multi-project environment, project managers make use of several pools of mostly limited resources that they must share with other project managers. This simultaneous management of the throughput times and resource allocations of projects is a complex process in which the often-conflicting interests of multiple participants have to be weighed and assessed (Maylor et al. 2006; Platje and Seidel 1993). Sharing pools of limited resources for multiple projects makes it possible for organizations to use these resources efficiently (Zika-Viktorsson et al. 2006). Pooling resources reduces idle time, and allows sharing of expertise. However, in the case of shared resources it is likely that disturbances to one project affect other projects. Since the prerequisites for valid planning and control in such situations are impaired, there is a need to make the situation as a whole more predictable by systematic planning and control (Zika-Viktorsson et al. 2006). When it comes to multiple projects, a project manager has to manage interdependencies and interactions among projects, in addition to managing each individual project. Project managers can do so by integrating the activities of planning/scheduling, monitoring/control, and resource management of different projects in order to manage them simultaneously. Project managers have few tools and techniques available to help them oversee the whole picture of all interdependencies and interactions (Patanakul and Milosevic 2008b).

Project overload is also common in a multi-project environment. Project overload is associated with over-commitment, i.e., too many projects in relation to the existing level of resources (Engwall and Jerbrant 2003). Zika-Viktorsson et al. (2006) found that the number of simultaneous projects in which a project manager is engaged predicts project overload and that project overload results in a negative impact on project performance measured in terms of adherence to time schedules and quality of work. In order to prevent project overload it is essential to achieve balance between project demand and available human resources (Zika-Viktorsson et al. 2006). A PMIS is considered valuable in providing the information needed to manage multiple project simultaneously (Patanakul and Milosevic 2008a). In this study we aim to advance upon the current knowledge on the use of PMIS in the decision making in a multi-project environment.

61.2.2 Project Management Information Systems (PMIS)

PMIS have become "comprehensive systems that support the entire life-cycle of projects, project programs, and project portfolios" (Ahlemann 2009, p. 19). They can support project managers in their planning, organizing, control, reporting, and decision making tasks, while evaluating and reporting at the same time (Raymond and Bergeron 2008).

Studies have shown that there are several important factors that encourage project managers to use PMIS (Ali and Money 2005; Dietrich and Lehtonen 2005; Raymond and Bergeron 2008). First, whether or not project managers will use PMIS strongly depends on the quality of the information generated by the PMIS (Ali and

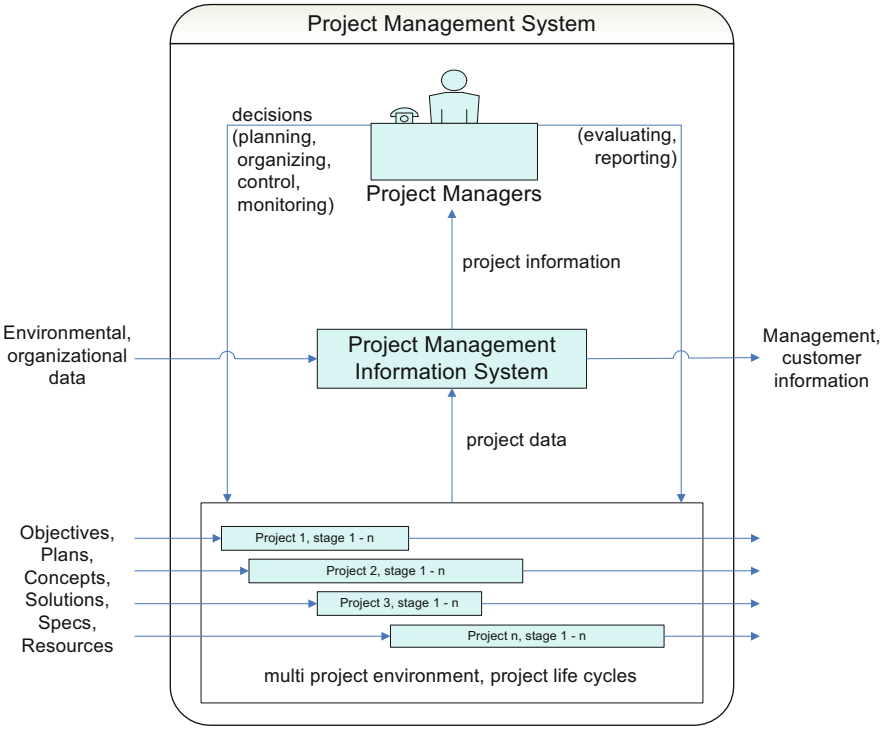


Fig. 61.1 The PMIS within the project management system (adapted from Raymond 1987 to reflect multi-project setting)

Money 2005; Dietrich and Lehtonen 2005; Raymond and Bergeron 2008; Gelbard et al. 2002; Raz and Globerson 1998). Second, project managers are more eager to use an information system if it provides them with the appropriate level of detail in relation to their needs (Ali and Money 2005; Raymond and Bergeron 2008). Third, it is important that the information generated is free of complexity, easy to understand, and easy for project managers to share with the project team's members (Ali and Money 2005). Fourth, PMIS facilitates continuous monitoring of progress (Ali and Money 2005).

Figure 61.1 shows the role of PMIS in a multi-project environment. The project management system consists of three parts: the project managers, the PMIS itself, and the project life cycles. The project life cycles consist of various evolving stages; objectives, plans, concepts, solutions, specifications and resources, and contains all information needed to support project managers in their planning, organizing, control, reporting, and decision making tasks. The role of the PMIS can be seen as the link between the multi-project environment and the project managers. This role includes capturing, storing, and processing project data to assist the project managers in their decision making tasks (Raymond 1987).

61.3 Research Model and Hypotheses

Our research model links PMIS information quality to decision making quality. Project and information overload are considered to influence PMIS information quality, while satisfaction with and use of PMIS, together with PMIS information quality, influence the quality of decision making.

61.3.1 Project Overload

There is a limit as to how many projects one project manager can handle simultaneously, based on available resource capacity. Routines and procedures can be helpful in that if project processes are standardized, project workers know what to do and how the work has to be carried out. However, too many or too few routines can easily become a burden for project workers when effort and pay-off are not balanced (Patanakul 2013). Too many procedures shift attention from the actual project management tasks to procedural activities, while too few routines create uncertainties about what to do next. Other issues are the interdependencies and interactions between projects and managing lead times (Engwall and Jerbrant 2003). Since schedules of different projects in a multi-project environment (partly) depend on each other, knowing the available time and resources at every moment in time is crucial for project progress. The limited amount of time available has to be spread over simultaneously running projects, which may result in time pressures and few opportunities for recuperation (Zika-Viktorsson et al. 2006; Patanakul 2013). Project teams acknowledge that it is very important to evaluate projects. However, in practice, due to time pressures project members are involved in the next project before having time to evaluate what went wrong and what went right in the previous project and draw lessons from this experience (Zika-Viktorsson et al. 2006). This suggests that in situations of project overload there may be too little time available for project managers to feed a PMIS with high quality information at the end of the project as well as during the project itself. Hence, we hypothesize,

Hypothesis 1a: Project overload has a negative impact on PMIS information quality in a multi-project environment

61.3.2 Information Overload

According to O'Reilly (1980) there is a relation between information overload and reduced project performance. Beyond some optimal point more information can lead to decreased decision making performance. Too much information may cause problems in selecting relevant information, due to difficulties in identifying relevant information from the total set available and distractions that reduce the available time for information processing (O'Reilly 1980). In a multi-project environment

the information available to the project manager is multiplied by the number of projects carried out simultaneously. When project information is abundant for each single project, it becomes problematic in a multi-project environment. A multi-project environment is characterized by a lack of transparency in project information and quality of project information (Elonen and Artto 2003). Increased complexity leads to confusion which makes that project workers are uncertain about what information should be delivered to whom, when it should be delivered, and in what format (Elonen and Artto 2003). In such settings project managers may have trouble seeking out quality information. Therefore, we hypothesize,

Hypothesis 1b: Information overload has a negative impact on the PMIS information quality in a multi-project environment

61.3.3 PMIS Information Quality

With regard to PMIS information quality we found empirical evidence that it directly as well as indirectly relates to timelier decision making and therefore project success (Martinsuo and Lehtonen 2007; Raymond and Bergeron 2008).

Dietrich and Lehtonen (2005) found a strong statistical correlation between the availability, topicality and validity of information and project success as well as adequate decision making. This indicates the importance of high quality information as an enabler for organizations to successful project management. Cooper et al. (2001) state that many of the go versus kill decisions of managers are made in the absence of solid information and therefore are questionable. Having the right—relevant, accurate, and reliable—information quickly available, allows project managers to make deliberate decisions. However, the focus of these studies was on project management in general and not explicitly on the use of PMIS as the source of information.

Saeed and Abdinnour-Helm (2008) explicitly study information systems. In particular, they explore the effects of characteristics of the information system on its perceived usefulness. They find that the availability of high-quality information in an information system is essential, because it assists a user in making sound decisions and thereby improves a project manager's work performance. In contrast, information systems that provide users with unreliable and inaccurate information have an adverse impact on its usefulness. Gelbard et al. (2002) show that reliability of estimations regarding time and effort are crucial for successful project management.

Research on project risk management pointed out that firms widely use tools to analyze, track, and control project risks. Raz and Michael (2001) identified several tools that have a great potential for contribution to successful risk management. These tools, like for example risks impact assessment and risk classification and ranking, are typically present in PMIS software packages like Primavera and Microsoft Project and are expected to support and ameliorate decision making.

On the basis of extant literature we expect that PMIS information quality is positively associated with adequate decision making in a multi-project environment. Thus,

Hypothesis 2: Greater PMIS information quality is associated with more adequate decision making in a multi-project environment

61.3.4 Project Manager Satisfaction with PMIS

User satisfaction is generally defined as fulfillment of one's wishes, expectations, or needs, or the pleasure derived from this (Seddon and Kiew 1994). Ali and Money (2005) reviewed several studies that relate relevance, accuracy, availability, reliability, consistency, and timeliness of information to user satisfaction with an information system. They conclude that the information quality has a crucial effect on the use of project management software. Project managers appear more eager to accept PMIS when the quality of the information output is high (Raymond and Bergeron 2008), and willing to use software that provides them with data that has an appropriate level of details, fits their work needs, is free of complexity, and is easy to understand and share with project team members. In a study about Departmental Accounting Systems, Seddon and Kiew (1994) found evidence that the level of information quality generated by an information system is an important determinant of user satisfaction with the system. In addition, Raymond and Bergeron (2008) find that PMIS information quality has a positive impact on the self-image of the project manager. Access to high quality project information stimulates the use of PMIS.

A multi-project environment increases the need for high quality information being readily available, since project managers have little time to check the accuracy and reliability of the information. Hence, we hypothesize,

Hypothesis 3: Greater PMIS information quality is associated with greater satisfaction of the project manager with PMIS in a multi-project environment

61.3.5 PMIS Information Use

Many authors have employed the term 'use' as an objective measure of system success. Note that, use and user satisfaction are strongly interrelated because a user can only be satisfied when he has first used the system. Positive experiences during the use of the system will automatically cause greater user satisfaction, which then in turn lead to an increased intention to use, and thus use (DeLone and McLean 2002). A multi-project environment generates repeated encounters of the project manager with the PMIS. If the project manager is not satisfied with the accuracy or depth of the information generated by the PMIS, he will not solicit PMIS for the next project (Raymond and Bergeron 2008). Conversely, if the information provided by the PMIS is in accordance with or even exceeds the project manager's expectations

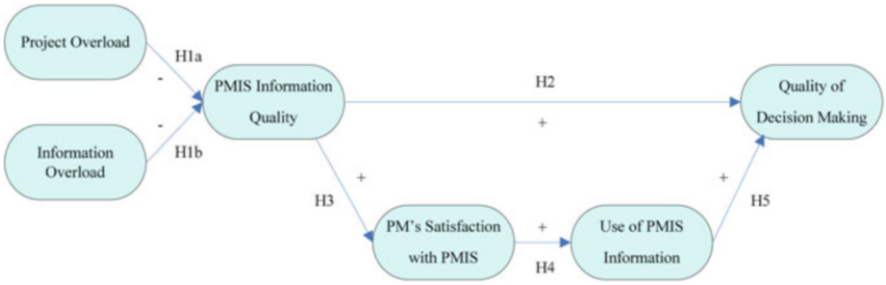


Fig. 61.2 Research model

and hence the satisfaction with PMIS is high, then the project manager is likely to use PMIS information. This is in line with DeLone and McLean's (2003) finding that increased user satisfaction will lead to increased intention to use, and in turn increased use. Thus,

Hypothesis 4: Greater satisfaction of the project manager with PMIS is associated with intensified use of PMIS information in a multi-project environment

61.3.6 Quality of Decision Making

Raymond and Bergeron (2008) examined the effect of PMIS use on project success, but they did not find support for a direct relationship. However, they did find an indirect relationship between PMIS use via project manager performance to timelier decision making. To our knowledge, no literature explicitly examines a direct relationship between the use of PMIS information and the quality of decision making. It is reasonable to assume that the use of PMIS information will lead to better decision making, especially when we take into account the hypothesis that PMIS information will only be used in a multi-project setting if this information has proved to be satisfactory in past projects. We hypothesize:

Hypothesis 5: Intensified use of PMIS information has a positive impact on the quality of decision making in a multi-project environment

The resulting research model is shown in Fig. 61.2.

61.4 Methodology

61.4.1 Sample and Data Collection

The target respondents for this questionnaire were project managers with at least two simultaneously active projects. We solicited the support of a large Dutch

pharmaceutical company for identifying project managers with multiple projects. This company develops and produces prescription drugs through pharmaceutical research. All respondents belonged to project oriented departments operating within a rather complex multi-project environment, e.g., process development, engineering, clinical trials, R&D, and quality control. Several PMIS are in use to support the project managers in managing their projects. Project managers are free to choose a PMIS, since the company does not have a central project management organization or a specific PMIS policy.

Data for this study was collected using a survey of 142 project managers, identified from a list of project managers managing at least two simultaneously active projects. The total number of project managers in the company is about 200. Respondents were screened and questionnaires were handed out personally. Completed questionnaires could be returned anonymously using a white envelope. A total of 110 responses were received. The answers were reviewed by two researchers independent from each other. Afterwards the independent judgements were compared and proved to be identical. An answer was judged ambiguous when more than one answer option was circled for one question (two cases), an answer was judged missing as no answer option was circled at all (six cases). Respondents that indicated that they were handling only one project at a time, were removed from the database as well (one case). Removing all responses containing incomplete or ambiguous answers resulted in 101 valid responses (71 % response rate). The respondents' demographics are presented in Table 61.2. Note that the

Table 61.2 Characteristics of the sample

Characterization of the respondents (N = 101)	% of sample
Project management experience	
More than 20 years	7
15–19 years	13
10–14 years	23
5–9 years	38
0–4 years	20
Gender	
Male	88
Female	12
Age in years	
60–69	3
50–59	17
40–49	48
30–39	31
20–29	2
PMIS software used	
Primavera	69
MS Project 90	22
Other (Excel, Access)	10

majority of respondents used Primavera as PMIS software. Primavera is a project-management software package that enables users to track and analyze performance. It is a multiuser, multiproject system with scheduling and resource control capabilities. It supports control and monitoring of costs and project budgets. Resources representing labor, materials and equipment are used to track time, and costs for projects. Slippage of projects' activities are updated resulting in the adjustment of time-related Gantt bars. Primavera supports multi-tiered project hierarchies and it includes tools for risk management (Primavera P6 Project Management Reference Manual Version 6.2, p. ix). Microsoft Project has comparable features to Primavera, however, Primavera is specifically developed to handle large complex projects, whereas MS Project is somewhat limited in in-depth analysis of large data sets and multiple projects. Hence, in a multi-project environment Primavera is best suited and therefore used most often.

Even though the pharmaceutical company openly endorsed the study the data were collected and analyzed without company involvement. Since the company has no specific PMIS policy forcing project managers to use a certain PMIS, social desirability bias is reduced. The questionnaire was accompanied by a cover letter stating the purpose of the study and an assurance of confidentially and anonymity. Prior to the distribution of the questionnaire, three subject-matter experts were asked to provide comments and suggestions on the clarity and readability of the questionnaire's items. Based on their feedback, the content of the cover letter and the design of the questionnaire were adapted to improve clarity and readability. These procedures also reduce social desirability (Podsakoff et al. 2003). To encourage submission of the questionnaire, each respondent was given a chance to win a gift worth € 20.

As both the predictor variables and the criterion variable were measured with self-reports, correlations between constructs may be inflated as a result of using a monomethod design (Podsakoff et al. 2003). Spector (2006) however argues and shows that the threat of common method bias is generally exaggerated. Still, we believe a discussion of this threat to validity is warranted. To minimize common method bias the following procedural remedies were undertaken in designing and administering the questionnaire. First, the respondents' anonymity was protected, respondents were assured that there are no right or wrong answers, and they were urged to answer questions as honestly as possible (Podsakoff et al. 2003). Second, several questions were reverse coded, reducing the threat of respondent "guessing", which is one possible source of common method variance, together with social desirability (Malhotra et al. 2006). In this way respondents cannot easily combine related items and produce the correlation needed to produce common method variance biased pattern of responses (Chang et al. 2010; Murray et al. 2005). Third, the research model (Fig. 61.2) is quite complex, hence it is not likely that the hypothesized relationships are part of the respondents cognitive map (Harrison et al. 1996; Chang et al. 2010). Fourth, our questionnaire contained only 35 items. Therefore, it was short enough to avoid boredom and fatigue, which might shift the

cognitive effort of respondents away from response accuracy to response speed (Yu and Cooper 1983). This would make the last items of the questionnaire vulnerable to biases in the direction of consistency with previous responses, and stereotypical responding, such as all midrange responses or all extreme responses (Lindell and Whitney 2001).

We examined the potential for common method variance via Harman's one-factor test recommended by Podsakoff and Organ (1986). Specifically, we performed an unrotated, principal components factor analysis with all manifest variables, extracting five factors with eigenvalues larger than 1, and the first factor accounting for only 36.7 % of variance. If common method variance existed, a single factor would have emerged in the analysis, or one general factor would have accounted for most of the covariance in the independent and criterion variables. Taken together, the threat of common method variance in the data is considered to be low.

Another potential threat to validity is non-response bias. Non-response bias threatens the validity of the findings if there is reason to suspect that non-respondents may exhibit different traits than respondents (Armstrong and Overton 1977). Groves and Peytcheva (2008) indicate that non-response bias is lower in case the respondent has some involvement with the sponsor, the questionnaire is self-administered rather than interviewer-administered, and the survey population is specific rather than general. These factors all are in favor of our study. At the same time however, Groves and Peytcheva (2008) suggest that surveys with questions about attitude (like ours) show higher non-response biases. We tested the extent of non-response bias in our sample using the procedure recommended by Armstrong and Overton (1977). T-tests indicated that no statistical significant differences existed with respect to any of the demographic variables, nor on the manifest variables or latent constructs between first respondents and late respondents. Hence, the threat of non-response bias in the data is believed to be low.

61.4.2 Measures

Multiple-item scales, closely following previous studies, were used to measure each construct. The items that were used to assess the construct variables as well as their internal consistency are reported in Appendix 1. All items were measured on 5-point Likert scales. We provided verbal labels for the midpoint of scales and avoided using bipolar numerical scale values (e.g., −2 to +2) in order to reduce acquiescence bias (Tourangeau et al. 2000). Wherever possible, existing measures of the constructs were adapted and used. The survey items assessing project overload are based on Hochdorfer and Bjarnason (2007). Information overload items are taken from O'Reilly (1980). Items for project manager's satisfaction with PMIS, PMIS information quality, the use of PMIS information and the quality of decision making are adopted from Raymond and Bergeron (2008). Table 61.3 presents an overview of the main construct variables with definitions and item sources.

Table 61.3 Constructs with definitions and item sources

Construct	Definition	Items adapted from
Project overload (PO)	Project overload is defined as having not enough capacity to deal with the amount of given projects and their unique schedules, tasks, and deadlines at the same time. The assessment of project overload is a subjective appraisal.	Hochdorfer and Bjarnason (2007, p. 28)
Information overload (IO)	The information overload construct measures the extent in which respondents feel that their processing capabilities differ with the information load encountered. The assessment of information overload is a subjective appraisal.	O'Reilly (1980)
PMIS information quality (IQ)	PMIS information quality is measured by assessing the degree in which information from the PMIS is (1) available, that is whether the PMIS information is readily at one's disposal; (2) reliable, that is whether the PMIS information is sound and dependable; (3) relevant, that is whether the PMIS information is closely connected or appropriate to the matter in hand; (4) accurate, that is whether the PMIS information is correct in all details; and (5) comprehensible, that is whether the PMIS information is understandable.	Raymond and Bergeron (2008)
Project managers satisfaction with PMIS (SAT)	Project managers satisfaction represents the affective attitude towards using the PMIS. An example of an item is "The PMIS is very useful in managing projects". The construct evaluates the PMIS' perceived adequacy, effectiveness, and efficiency.	Raymond and Bergeron (2008)
Use of PMIS information (USE)	The use of PMIS information measures the perceived use of the PMIS for different project management tasks, including using overview reports, project summary reports, project budget reports, resource usage reports, and tasks in progress reports.	Raymond and Bergeron (2008)
Quality of decision making (DM)	The quality of the decision making construct is composed of items such as: a perceived increase in the quality of decisions and reduction of the time required for decision making.	Raymond and Bergeron (2008)

In addition, the following demographical and control variables were included in the survey: age, gender, years of project management experience, and name of the used PMIS. In order to avoid including "impotent control variables" (Becker 2005) and reducing power of our analyses unnecessarily, we checked whether control variables were correlated with the core variables of interest. For none of the control variables we found significant correlations with the core constructs of our model. Hence, we excluded the control variables in the analysis of our model.

61.5 Results

A component based structural equations modeling (SEM) method, more specifically Partial Least Squares (PLS), was used to test the hypotheses. SEM was chosen because it allows the analyses of systems of independent and dependent variables at the same time, whereas multiple regression analysis does not. We found component based SEM, and in particular PLS, more adequate for our purposes than covariance based SEM methods such as LISREL and EQS (Fornell and Larcker 1981), as PLS is robust with respect to multicollinearity (Cassel et al. 2000), small sample sizes (Haenlein and Kaplan 2004), complex modeling including models with hierarchical constructs, mediating and moderating effects (Chin et al. 2003; Wetzels et al. 2009), and even violations of the normality distribution assumption (Haenlein and Kaplan 2004; Cassel et al. 1999). For an overview of conditions under which PLS might be more appropriate than covariance based SEM, see Wetzels et al. (2009).

To carry out PLS we used SmartPLS software (Ringle et al. 2005). PLS examines the significance of the relationships and their resulting R^2 (Gefen et al. 2000). Path coefficients in PLS indicate the strength of the relationship between constructs and can be interpreted as regression coefficients between standardized variables. Appendix 2 shows the correlation classification used in this study. The sample size requirement for PLS analysis was met (Gefen et al. 2000). A power analysis was performed using G*Power 3.1.2 (downloaded from http://www.psycho.uni-duesseldorf.de/abteilungen/aap/gpower3) and showed that our sample size was suitable (Erdfelder et al. 1996; Faul et al. 2009). Tenenhaus et al. (2005) suggest a goodness of fit (GoF) measure for PLS path modeling that is defined as the geometric mean of the average communality and average R^2 for endogenous constructs $GoF = \sqrt{\overline{AVE} \cdot \overline{R}^2}$. Wetzels et al. (2009) have derived the following GoF criteria for small, medium, and large effect sizes of R^2. $GoF_{small} = 0.1$, $GoF_{medium} = 0.25$, and $GoF_{large} = 0.36$. For our model GoF was 0.38, exceeding the cut-off value of 0.36 for large effect sizes of R^2. Hence we conclude that our model performs well compared to the baseline values as defined by Wetzels et al. (2009).

61.5.1 Reliability and Validity

Reliability was assessed by evaluating the unidimensionality of items through their factor loadings and by noting composite reliability as calculated in the PLS analysis. Unidimensionality is usually satisfied by retaining the items whose loadings (λ) are above 0.7, indicating that they share sufficient variance with their related construct (Ringle et al. 2005). A few items were excluded from the constructs in order to fulfill unidimensionality of each construct. See Appendix 1 for all items and their respective loadings.

Following Kaiser and Ahlemann (2010), we determined the composite reliability of all the constructs to ensure that the items of the measurement models were consistent internally. Composite reliability scores for each construct exceeded the 0.7 value recommended by Hock and Ringle (2010), and are shown in Table 61.4. A composite reliability score greater than 0.7 indicates that the variance of a given construct explains at least 70 % of the variance of the corresponding measure, as is the case for all constructs in our research model. Since composite reliability is above 0.7 for all constructs, the measures are reliable (Lewis et al. 2005).

Convergent and discriminant validity were assessed by examining the average variance extracted (AVE) and the item construct correlations as generated by PLS. Convergent validity tests whether the measures of constructs that should be related, are related (Trochim 2010). AVE is the percentage of the total variance of a measure represented or extracted by the variance due to the construct and ranges from 0 to 1. It should be 0.50 or above to exhibit convergent validity (Fadel and Brown 2010; Hock and Ringle 2010). Table 61.4 shows the AVE values for each construct. Except for 'project overload' all constructs meet the criteria for convergent validity. Retaining the minimum of three items per construct (Ringle et al. 2005), resulted in an AVE of 0.460 for project overload. Hence, strictly speaking project overload does not meet the criterion for convergent validity, but we feel that it's AVE value is close enough to 0.50 to be able to maintain this construct into our analysis.

Discriminant validity tests whether believed unrelated measures of constructs are, in fact, unrelated (Trochim 2010). Adequate discriminant validity at the construct level is established if the square root of AVE values (on the diagonal of Table 61.5) is larger than the off-diagonal correlations. The criteria for this test are met for all constructs. Cross-loadings are another test of discriminant validity,

Table 61.4 Means, standard deviations, PLS composite reliabilities

Construct	No. of items	Mean	SD	Composite reliability
Quality of decision making	4	3.45	0.67	0.84
Information overload	3	3.32	0.50	0.76
PMIS information quality	4	3.25	0.52	0.84
PMs satisfaction with PMIS	3	3.09	0.55	0.80
Project overload	3	3.47	0.42	0.71
Use of PMIS information	3	2.94	0.82	0.76

Table 61.5 Construct AVE's and inter-construct correlations

#	Construct	AVE	1	2	3	4	5	6
1	Quality of decision making	0.561	0.749					
2	Information overload	0.525	0.067	0.725				
3	PMIS information quality	0.567	0.574	0.174	0.753			
4	PMs satisfaction with PMIS	0.567	0.565	0.080	0.577	0.753		
5	Project overload	0.460	0.078	−0.018	0.215	0.117	0.678	
6	Use of PMIS information	0.518	0.622	0.015	0.459	0.360	0.238	0.720

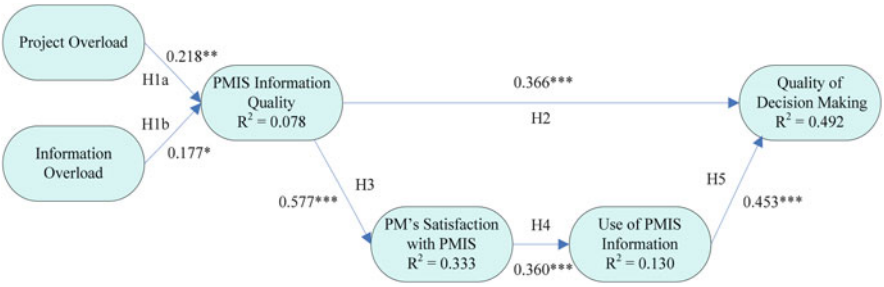


Fig. 61.3 Results of evaluating the research model with SmartPLS ($n = 101$) significance level of path coefficients: $^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$

the item-construct cross-loadings are shown in Appendix 3. Each block of items should load higher for its respective construct than for the block of items of the other constructs. The criteria for this test is also met for all constructs, hence both tests indicate adequate discriminant validity.

61.5.2 Structural Model

The structural model represents the relationships between constructs that were hypothesized in the research model. For PLS there are no well-established overall fit measures. Path coefficients (statistical and practical significance) and coefficients of determination (R^2) together indicate how well the model performed. The R^2 are measures of the variance in endogenous constructs accounted by other constructs that were hypothesized to have an effect on them. Therefore, they can be interpreted as R^2 in regression analysis (Gil-Garcia 2005). The hypotheses are tested by analyzing the direction, the value and level of significance of the path coefficients (gammas) estimated by the PLS method. A bootstrapping resampling procedure (200 samples) was used to test the significance of path coefficients. The results of the analysis are shown in Fig. 61.3.

The hypothesis that project overload has a negative impact on the quality of the PMIS information output (H1a) is not supported. The hypothesis that information overload has a negative impact on the quality of the PMIS information quality (H1b) is not supported as well. The significant path coefficients ($\gamma = 0.218$, $p < 0.01$, and $\gamma = 0.177$, $p < 0.05$, respectively) indicate that there is a weak association of project overload as well as information overload with PMIS information quality. Instead of the expected negative associations, we found positive associations of project overload as well as information overload with PMIS information quality.

The second hypothesis (H2) is supported and indicates that a greater quality of the PMIS information output is significantly and positively associated with decision making by project managers in a multi-project environment ($\gamma = 0.366$; $p < 0.001$). Hence, a significant improvement in decision making in terms of improved quality of the decisions, reduced time in making decisions, better allocation of resources and better monitoring activities can be obtained directly by improving the quality of the PMIS information output. In addition we found evidence for an indirect effect of PMIS information quality on decision making (equal to $0.577 \times 0.360 \times 0.453$). The indirect effect works via the mediating influence of the project manager's satisfaction with PMIS and the use of PMIS information. However, the indirect effect ($\gamma = 0.094$) is much less than the direct effect ($\gamma = 0.366$).

Path analysis also confirms the existence of a significant relationship between the quality of the PMIS information output and the satisfaction of the project manager with PMIS ($\gamma = 0.577$; $p < 0.001$), hypothesis 3 (H3). A higher quality of the PMIS information output is associated with higher levels of satisfaction of project managers with PMIS in terms of having faith in the reports generated by the PMIS, easy interaction with the PMIS and increased use of the PMIS.

The fourth hypothesis (H4) concerns the positive relation between the satisfaction of the project manager with PMIS to intensified use of PMIS information. This hypothesis is supported ($\gamma = 0.360$; $p < 0.001$). Indeed, the use of PMIS information in the form of overview reports, resource usage reports, and task in progress reports is positively influenced by the project manager's satisfaction with the PMIS.

The fifth hypothesis (H5) suggests a positive association between intensified use of PMIS information and the quality of decision making. This hypothesis is supported ($\gamma = 0.453$; $p < 0.001$). In other words, using reports generated by the PMIS increases the overall quality of decision making by enhancing the quality of decisions, shorten the time to come to a decision, better allocating resources, and better monitoring activities.

About 49 % of the variance with regard to the quality of decision making is accounted for by its explanatory constructs. Similarly, the model explains about 33 % of the variance in project manager's satisfaction with PMIS, 13 % of the variance in the use of PMIS information, and 8 % of the variance in PMIS information quality. The average explanatory power of the endogenous constructs in the model is about 26 % ($R^2 = 0.258$).

61.6 Conclusions

61.6.1 Summary and Relevance

The aim of this study was to gain a better understanding of the elements of PMIS that contribute to adequate decision making in a multi-project environment, and to provide insights in the relationship between PMIS information quality and the project manager's satisfaction with PMIS. Most of the findings of this study are in line with prior studies regarding PMIS and studies about single complex projects, however, a few deviations were found.

Two factors were expected to have a negative relationship with PMIS information quality, namely project overload and information overload. The findings of this study are not in line with what was expected beforehand. We found that project overload as well as information overload are positively, albeit weakly, related to PMIS information quality. An explanation for this seemingly paradoxical effect is as follows. Previous research has indicated that the hours worked per week are positively related to the total output of a project worker with an maximum of 60 h per week for a full time project worker. When working more than 60 h per week, output drops, not only per hour but in total as well (Hochdorfer and Bjarnason 2007). Hence, if the project overload experienced by the respondents in our study is below the maximum of 60 h per week per full time employee, there will not actually be a situation of overall overload, although the project worker perceives it as such. A similar reasoning can be given with respect to information overload. It may also be true for information overload that only beyond some optimal point too much information can lead to a decrease in the PMIS information quality (O'Reilly 1980). Below this optimal point a respondent can still perceive information overload, but it might not result in actual problems for output, i.e., PMIS information quality. In fact, this might also give an explanation for the weak positive relationship we found between information overload and PMIS information quality. One can imagine that up to the presumed optimal point, extra information, although being excessive in the eyes of the project manager, can lead to increased PMIS information quality.

We found that in a multi-project environment the availability of higher quality information in the PMIS is associated with project managers that are more satisfied with PMIS. These findings are in line with prior research in the field of accounting systems (Seddon and Kiew 1994), that indicate that the level of information quality generated by an information system is an important determinant of user satisfaction with the system. In addition, evidence from single-project environments points in a similar direction (Ali and Money 2005). Apparently, a multi-project environment generates a high need for high quality information, since project managers are under extreme time pressures and will not often investigate whether the information is accurate and reliable.

The project manager's satisfaction with PMIS was expected to be indirectly related to the quality of decision making via the use of PMIS information. In our study we found a positive effect between these constructs. These findings are in line

with prior research (Ali and Money 2005), that showed that information quality has a significant effect on the use of PMIS and that project managers are more likely to use PMIS information that is free of complexity and is easy to understand. This may indicate that the more satisfied a project manager is with the PMIS, the more he will use the information generated by the PMIS, which in turn has a positive impact on the quality of his decision making. With respect to the project manager's satisfaction with PMIS it is interesting to note that among the project managers who participated in our study, only 37 % indicated to be more than averagely satisfied with the quality of the information provided by the PMIS they use. Even 90 % of the participants reported that they were particularly dissatisfied with the reliability of the information. These results indicate that broadly speaking, project managers who are dependent upon a PMIS that produces low quality information, are less satisfied and as a consequence do not use the generated information in simultaneously running projects. In turn, they are to a lesser extent supported in their decision making and the quality of their decision making is negatively affected. The opposite may be true for project managers who can rely upon a PMIS that produces high quality information. In the PMIS literature this relationship is recognized as a "feedback" relationship (DeLone and McLean 2003). As project managers perceive the PMIS information to be beneficial to them, it is likely that they will increase their use of the PMIS information. In a multi-project setting this effect is enhanced, because project leaders will draw conclusions about the information quality for one project and extend this conclusion to their other simultaneously running projects. When the PMIS generates low quality information for one of their projects, project managers are likely to draw negative conclusions about the quality of information for all their simultaneously running projects, without checking whether the PMIS for these projects might actually generate high quality information.

In this study, two factors directly influence the quality of decision making. First, we found that the quality of the information produced by the PMIS is directly related to the quality of decision making. This finding is consistent with Saeed and Abdinnour-Helm (2008) who studied the effects of information system characteristics and perceived usefulness on post adoption usage of information systems. They found that high quality information helps project managers in making sound decisions and improving their performance. In addition to the quality of decision making, PMIS information quality also directly influences satisfaction with the PMIS of multi-project managers. This supports the DeLone and McLean (1992) model of information system success, in which information quality explained 33 % of the variance in the project manager's satisfaction with PMIS. Hence, we conclude that reliability, relevance, accuracy as well as comprehensiveness of the PMIS information play an important role in the quality of decision making, especially in a multi-project environment. A PMIS that produces poor quality information will not be used by project managers for their simultaneously running projects. The use of PMIS information is a second factor that directly impinges on the quality of decision making. We found that the use of PMIS information is significantly and quite strongly related to the quality of decision making.

The theoretical contribution of this research lies primarily in the fact that the study sheds light on factors that are important for the quality of decision making, specifically in a multi-project environment. Our study suggests the presence of spillover effects in the opinion of the project manager about PMIS information from one project to another, simply because these are managed by the same person. Whereas project managers always are in need of high quality information from a PMIS, this need is even larger in a multi-project environment. Extreme time pressures leave no time to multi-project managers to investigate whether PMIS information is accurate and reliable. In a multi-project environment, the perceived quality of PMIS information has an oil spotting effect. The perception of PMIS information being trustworthy or not affects the opinion, and therefore the behavior, of project managers in all of their simultaneously running projects at hand. As project managers perceive the PMIS information to be beneficial to them for one project, they extend this conclusion to their other projects, without checking whether the PMIS for these projects indeed generate high quality information.

The findings from our study also have managerial relevance. Multi-project environments generate specific challenges that find their origin in increased complexity. Linkages and interdependencies between simultaneously running projects are at the root of this increased complexity. It can be concluded from this study that project managers running several projects at the same time benefit from using a PMIS. Not all companies with a substantial part of activities organized in projects adopt a central PMIS. This study suggests that the management of such firms might want to design policy on the use of project management information systems. Furthermore, companies that do have a PMIS policy should assess whether project managers are satisfied with its information. Especially in a multi-project environment, companies should adapt their PMIS or switch to another one much sooner as compared to companies that mainly work with single projects, because the perception of untrustworthy information in one project immediately spills over to parallel running projects and hence the PMIS loses its function. Another option for companies could be to appoint an assistant to the project manager, who has the particular task of checking PMIS information quality, in order to ensure that inadequate conclusions about information do not multiply and spill over to other projects. Moreover, companies should invest in PMIS and devote time to certify that high quality information is generated by the PMIS. Since, high quality PMIS information will lead to high quality decision making.

In addition, our research suggests that up to a certain threshold no adverse effects are to be expected from project and information overload, even when project managers themselves perceive to be burdened by excess information. Management should use this finding cautiously, because further research is needed on where this threshold might lie. It would be unwise to jeopardize the well-being of project managers because this will certainly affect the quality of work.

61.6.2 Limitations and Issues for Further Research

The results of this study should be interpreted cautiously. The model explains nearly half of the variance on the quality of decision making as perceived by the project manager. The quality of decision making seems to be affected by the quality of the PMIS information and the actual use of this information. However, the quality of decision making is unexplained for the other half of the variance which may indicate that there are other technical and managerial factors, beside PMIS information quality and the use of PMIS information, that affect the quality of decision making. This also holds for the constructs of PMIS information quality and the use of PMIS information quality. The variance in the quality of the PMIS information is explained for only 7.8 % by project and information overload. The variance in the use of PMIS information is explained for 13.0 % by the project manager's satisfaction with the PMIS. The variance of the latter is, in turn, explained for 33.3 % by the quality of the PMIS information. The unexplained parts of the variance in these constructs may indicate that there are other factors that affect these constructs. Hence, future research should take into account a larger set of factors and develop a better explanation of, especially, the "PMIS Information" and "Use of PMIS Information" constructs.

Another interesting avenue for further research is the counterintuitive finding regarding the effect of project and information overload on the quality of the PMIS information. Future studies should focus on the extent to which project overload as well as information overload strengthens PMIS information quality. An additional interesting aspect for further research regarding information overload might be the possible positive effect of the substantial amount of graphical reports generated by PMIS to reduce the reverse effects of information overload (Chan 2001).

In this study, the sample consisted of the multi-project managers of a multinational firm. The set of respondents is certainly not a random sample of multi-project managers worldwide and across all industries. Hence, the findings of this study can only be generalized with caution. Further research should show whether our findings can be generalized across industries and countries.

Finally, since the majority of our respondents indicated to be unsatisfied with the quality of their PMIS a suggestion for further research is to investigate what factors are important, in the perception of project managers, to generate high quality information with respect to availability, accuracy, relevance, comprehensiveness, and particularly, reliability. Factors like effective sizing and content definition of work packages might play a crucial role in this (Raz and Globerson 1998) and should be object of further study.

For the objectives of our study we focused on PMIS and whether and under what conditions PMIS can lead to better quality of decision making for project managers in a multi-project environment. From the literature on strategic Decision Support Systems we know that various computer based information systems exist that specifically are designed for supporting strategic business decision making activities (e.g., Reich and Kapeliuk 2005). Decision Support Systems serve management,

operations, and planning departments of an organization and help them to make decisions. It might be worthwhile for further research to explore whether project Decision Support Systems and knowledge based systems can provide project managers with accurate predictions, help them design the desired project trajectory, and validate process changes (Donzelli 2006), and save them from having to go through large information systems that can generate overload.

Appendix 1: Constructs and Measures

Construct	Abbreviation	Item	PLS factor loading[c]
Project overload (PO)	PO-1	On how many projects do you usually work at the same time?	0.56
	PO-2	How often do you switch between your projects?	0.81**
	PO-3	How often do you have to do the job of other people?	0.64*
	PO-4[a]	How often do you change the priorities in your work?	(0.09)
	PO-5[a]	How often do you have the feeling that you are wasting time on a task?	(0.26)
Information overload (IO)	IO-1[b]	On some occasions you might have too little information that you could consistently handle for making the best possible work-related decisions. In a typical work week, approximately how often does this situation happen?	0.83***
	IO-2[b]	Sometimes at work you may receive more information than you can efficiently use. At other times, however, you may feel that you are not receiving all the information you need. How often during a week would you say that this lack of information arises?	0.55*

(continued)

Construct	Abbreviation	Item	PLS factor loading[c]
	IO-3	Is the total amount of information you receive in a typical work week enough to meet the information requirements for your job?	0.76***
PMIS information quality (IQ)	IQ-1	Availability	(0.31)*
	IQ-2	Reliability	0.75***
	IQ-3	Relevance	0.76***
	IQ-4	Accuracy	0.81***
	IQ-5	Comprehensiveness	0.68***
Project managers satisfaction with PMIS (SAT)	SAT-1	The PMIS is very useful in managing projects	(0.57)***
	SAT-2	I really trust the reports from the PMIS	0.77***
	SAT-3	The interaction with the PMIS is fairly easy	0.62***
	SAT-4	The understanding of the PMIS is not difficult	(0.49)***
	SAT-5	My satisfaction with the PMIS makes me use it more	0.78***
Use of PMIS information (USE)	USE-1	Overview reports	0.72***
	USE-2	Project summary reports	(0.60)***
	USE-3	Project budget reports	(0.55)***
	USE-4	Resource usage reports	0.70***
	USE-5	Task in progress reports	0.64***
Quality of decision making (DM)	DM-1	The PMIS improves the quality of my decisions	0.81***
	DM-2	The PMIS reduces the time of my decision making	0.76***
	DM-3	The PMIS helps me to better manage the budget for activities	(0.56)***
	DM-4	The PMIS helps me to better allocate resources	0.66***
	DM-5	The PMIS helps me to better monitor activities	0.66***

Significance level of PLS factor loading: * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

[a] Reverse-coded for a correct calculation of the composite reliability (Ringle et al. 2005)

[b] Reverse-coded

[c] After removing the items PO-4 and 5, IQ-1, SAT-1 and 4, USE-2 and 3, and DM-3

Appendix 2: Correlation Classification

Strength (Rubin 2009)	Correlation coefficient	
Perfect	−1	1
Strong	−0.999 to −0.500	0.500 to 0.999
Moderate	−0.499 to −0.300	0.300 to 0.499
Weak	−0.299 to −0.100	0.100 to 0.299
No correlation	−0.099 to 0.000	0.000 to 0.099

Appendix 3: Item-Construct Cross-Loadings

	Quality of decision making (DM)	Information overload (IO)	PMIS information quality (IQ)	Project overload (PO)	Project managers satisfaction with PMIS (SAT)	Use of PMIS information (USE)
DM-1	0.819	0.082	0.482	0.089	0.479	0.537
DM-2	0.802	0.004	0.464	0.004	0.458	0.445
DM-4	0.679	−0.004	0.457	0.063	0.387	0.474
DM-5	0.686	0.141	0.278	0.082	0.353	0.384
IO-1	0.033	0.832	0.152	−0.005	0.076	−0.016
IO-2	0.108	0.552	0.070	−0.046	0.051	0.133
IO-3	0.040	0.760	0.136	−0.007	0.046	−0.021
IQ-2	0.350	0.230	0.762	0.108	0.554	0.270
IQ-3	0.544	0.105	0.764	0.197	0.408	0.515
IQ-4	0.429	0.182	0.805	0.173	0.412	0.320
IQ-5	0.399	−0.021	0.673	0.173	0.349	0.253
PO-1	0.018	0.148	0.127	0.558	0.043	0.065
PO-2	0.075	−0.099	0.186	0.811	0.109	0.238
PO-3	0.060	−0.050	0.111	0.640	0.078	0.155
SAT-2	0.419	0.228	0.529	0.098	0.770	0.275
SAT-3	0.289	0.047	0.369	0.007	0.683	0.142
SAT-5	0.544	−0.121	0.384	0.140	0.801	0.369
USE-1	0.440	−0.011	0.370	0.327	0.265	0.686
USE-4	0.440	0.084	0.407	0.162	0.254	0.748
USE-5	0.461	−0.038	0.218	0.029	0.259	0.724

References

Ahlemann F (2009) Towards a conceptual reference model for project management information systems. Int J Proj Manag 27(1):19–30

Ali ASB, Money WH (2005) A study of project management system acceptance. In: Proceedings of the 38th Hawaii international conference of systems science, HI, USA

Ali ASB, Anbari FT, Money WH (2008) Impact of organizational and project factors on acceptance and usage of project management software and perceived project success. Proj Manag J 39(2):5–33

Aritua B, Smith NJ, Bower D (2009) Construction client multi-projects: a complex adaptive systems perspective. Int J Proj Manag 27(1):72–79

Armstrong JS, Overton TS (1977) Estimating nonresponse bias in mail surveys. J Mark Res 14(3):396–402

Atkinson R (1999) Project management: cost, time and quality, two best guesses and a phenomenon, it's time to accept other success criteria. Int J Proj Manag 17(6):337–342

Becker TE (2005) Potential problems in statistical control of variables in organizational research: a qualitative analysis with recommendations. Organ Res Methods 8:274–289

Bendoly E, Swink M (2007) Moderating effects of information access on project management behavior, performance and perceptions. J Oper Manag 25(3):604–622

Blichfeldt BS, Eskerod P (2008) Project portfolio management: there's more to it than what management enacts. Int J Proj Manag 26(4):357–365

Caniëls MCJ, Bakens RJJM (2012) The effects of project management information systems on decision making in a multi project environment. Int J Proj Manag 30(2):162–175

Canonico P, Söderlund J (2010) Getting control of multi-project organizations: combining contingent control mechanisms. Int J Proj Manag 28:796–806

Cassel CM, Hackl P, Westlund AH (1999) Robustness of partial least-squares method for estimating latent variable quality structures. J Appl Stat 26(4):435–446

Cassel CM, Hackel P, Westlund AH (2000) On measurement of intangible assets: a study of robustness of partial least squares. Total Qual Manag 11(7):897–907

Chan S-Y (2001) The use of graphs as decision aids in relation to information overload and managerial decision quality. J Inf Sci 27(6):417–425

Chang S-J, Van Witteloostuijn A, Eden L (2010) Common method variance in international business research. J Int Bus Stud 41:178–184

Chin WW, Marcolin BL, Newsted PR (2003) A partial least squares latent variable modeling approach for measuring interaction effects: results from a Monte Carlo simulation study and an electronic-mail emotion/adoption study. Inf Syst Res 14(2):189–217

Cooper RG, Scott J, Kleinschmidt EJ (2001) New problems, new solutions: making portfolio management more effective. J Prod Innov Manag 18(1):52–53

DeLone WH, McLean ER (1992) Information systems success: the quest for the dependent variable. Inf Syst Res 3(1):60–95

DeLone WH, McLean ER (2002) Information systems success revisited. In: Proceedings of the 35th Hawaii international conference on system sciences, HI, USA

DeLone WH, McLean ER (2003) The DeLone and McLean model of information systems success: a ten-year update. J Manag Inf Syst 19(4):9–30

Dietrich P, Lehtonen P (2005) Successful management of strategic intentions through multiple projects: reflections from empirical study. Int J Proj Manag 23(5):386–391

Donzelli P (2006) A decision support system for software project management. IEEE Softw 23(4):67–75

Dvir D, Raz T, Shenhar AJ (2003) An empirical analysis of the relationship between project planning and project success. Int J Proj Manag 21(2):89–95

Dye LD, Pennypacker JS (2000) Project portfolio management and managing multiple projects: two sides of the same Coin? In: Proceedings of the project management institute annual seminars & symposium, Houston, TX, USA

Elonen S, Artto KA (2003) Problems in managing internal development projects in multi-project environments. Int J Proj Manag 21(6):395–402

Engwall M, Jerbrant A (2003) The resource allocation syndrome: the prime challenge of multi-project management? Int J Proj Manag 21(6):403–409

Erdfelder E, Faul F, Buchner A (1996) GPOWER: a general power analysis program. Behav Res Methods Instrum Comput 28(1):1–11

Fadel KJ, Brown SA (2010) Information systems appraisal and coping: the role of user perceptions. Commun Assoc Inf Syst 26(6):107–126

Faul F, Erdfelder E, Buchner A, Lang A-G (2009) Statistical power analyses using G*Power 31: tests for correlation and regression analyses. Behav Res Methods 41:1149–1160

Fornell C, Larcker DF (1981) Evaluating structural equation models with unobservable variables and measurement error. J Mark Res 18(1):39–50

Gefen D, Straub D, Boudreau M (2000) Structural equation modeling techniques and regression: guidelines for research practice. Commun Assoc Inf Syst 7(7):1–78

Gelbard R, Pliskin N, Spiegler I (2002) Integrating systems analysis and project management tools. Int J Proj Manag 20:461–468

Gil-Garcia JR (2005) Exploring the success factors of state website functionality: an empirical investigation. In: Proceedings of the 2005 national conference on digital government research, San Diego, CA, USA

Groves RM, Peytcheva E (2008) The impact of nonresponse rates on nonresponse bias: a meta-analysis. Public Opin Q 72(2):167–189

Haenlein M, Kaplan AM (2004) A beginner's guide to partial least square analysis. Underst Stat 3(4):283–297

Harrison DA, McLaughlin ME, Coalter TM (1996) Context, cognition, and common method variance: psychometric and verbal protocol evidence. Organ Behav Hum Decis Process 68(3):246–261

Hendriks MHA, Voeten B, Kroep L (1999) Human resource allocation in a multi-project RandD environment: resource capacity allocation and project portfolio planning in practice. Int J Proj Manag 17(3):181–188

Hochdorfer T, Bjarnason G (2007) Project overload in project based organizations: causes, symptoms and effects: a study of project members and their projects. Master thesis, University of Kalmar, Sweden

Hock M, Ringle C (2010) Local strategic networks in the sofware industry: an empirical analysis of the value continuum. Int J Knowl Manag Stud 4(2):132–151

Kaiser MG, Ahlemann F (2010) Measuring project management information systems success. Paper presented at the 18th European conference on information systems, Pretoria, South Africa

Laslo Z, Goldberg AI (2008) Resource allocation under uncertainty in a multi-project matrix environment: is organizational conflict inevitable? Int J Proj Manag 26(8):773–788

Lewis BR, Templeton GF, Byrd TA (2005) A methodology for construct development in MIS research. Eur J Inf Syst 14(4):388–400

Lindell MK, Whitney DJ (2001) Accounting for common method variance in cross-sectional research designs. J Appl Psychol 86(1):114–121

Malhotra NK, Kim SS, Patil A (2006) Common method variance in IS research: a comparison of alternative approaches and a reanalysis of past research. Manag Sci 52(12):1865–1883

Martinsuo M, Lehtonen P (2007) Role of single-project management in achieving portfolio management efficiency. Int J Proj Manag 25(1):56–65

Maylor H, Brady T, Cooke-Davies T, Hodgson D (2006) From projectification to programmification. Int J Proj Manag 24(8):663–674

Mota, CMDM, De Almeida AT, Alencar LH (2009) A multiple criteria decision model for assigning priorities to activities in project management. Int J Proj Manag 27(2):175–181

Murray JY, Kotabe M, Zhou JN (2005) Strategic alliance based sourcing and market performance: evidence from foreign firms operating in China. J Int Bus Stud 36(2):187–208

O'Reilly CA (1980) Individuals and information overload in organizations: is more necessarily better? Acad Manag J 23(4):684–696

Patanakul P (2013) Key drivers of effectiveness in managing a group of multiple projects. IEEE Trans Eng Manag 60(1):4–17

Patanakul P, Milosevic D (2008) The effectiveness in managing a group of multiple projects: factors of influence and measurement criteria. Int J Proj Manag 27:216–233

Patanakul P, Milosevic D (2008) A competency model for effectiveness in managing multiple projects. J High Tech Manag Res 18(2):118–131

Payne JH (1995) Management of multiple simultaneous projects: a state-of-the-art review. Int J Proj Manag 13(3):163–168

Platje A, Seidel H (1993) Breakthrough in multiproject management: how to escape the vicious circle of planning and control. Int J Proj Manag 11(4):209–213

Platje A, Seidel H, Wadman S (1994) Project and portfolio planning cycle: project-based management for the multiproject challenge. Int J Proj Manag 12(2):100–106

Podsakoff PM, Organ DW (1986) Self-reports in organizational research: problems and prospects. J Manag 12(4):531–544

Podsakoff PM, MacKenzie SB, Lee JY, Podsakoff NP (2003) Common method biases in behavioral research: a critical review of the literature and recommended remedies. J Appl Psychol 88(5):879–903

Raymond L (1987) Information systems design for project management: a data modeling approach. Proj Manag J 18(4):94–99

Raymond L, Bergeron F (2008) Project management information systems: an empirical study of their impact on project managers and project success. Int J Proj Manag 26(2):213–220

Raz T, Globerson S (1998) Effective sizing and content definition of work packages. Proj Manag J 98:17–23

Raz T, Michael E (2001) Use and benefits of tools for project risk management. Int J Proj Manag 19(1):9–17

Reich A, Kapeliuk A (2005) A framework for organizing the space of decision problems with application to solving subjective, context-dependent problems. Decis Support Syst 41(1):1–19

Ringle C, Wende S, Will A (2005) SmartPLS (Version 20M3). Available from: http://smartplsde/forum

Rubin A (2009) Statistics for evidence-based practice and evaluation, 2nd edn. Thomas Brooks/Cole, Belmont

Saeed KA, Abdinnour-Helm S (2008) Examining the effects of information system characteristics and perceived usefulness on post adoption usage of information systems. Inf Manag 45(6):376–386

Seddon P, Kiew M-Y (1994) A partial test and development of the DeLone and McLean model of IS success. Australas J Inf Syst 4(1):90–104

Spector PE (2006) Method variance in organizational research: truth or urban legend? Organ Res Methods 9:221–232

Tenenhaus M, Vinzi VE, Chatelin Y-M, Lauro C (2005) PLS path modeling. Comput Stat Data Anal 48(1):159–205

Tourangeau R, Rips LJ, Rasinski K (2000) The psychology of survey response. Cambridge University Press, Cambridge

Trochim WMK (2010) Research methods knowledge base. Available from: http://www.socialresearchmethods.net/kb/convdisc.php

Turner JR, Speiser A (1992) Programme management and its information systems requirements. Int J Proj Manag 10(4):196–206

Wetzels M, Odekerken-Schroder G, van Oppen C (2009) Using PLS path modeling for assessing hierarchical construct models: guidelines and empirical illustration. MIS Q 33(1):177–195

Yaghootkar K, Gil N (2012) The effects of schedule-driven project management in multi-project environments. Int J Proj Manag 30:127–140

Yang S, Fu L (2014) Critical chain and evidence reasoning applied to multi-project resource schedule in automobile R&D process. Int J Proj Manag 32(1):166–177

Yu J, Cooper H (1983) A quantitative review of research design effects on response rates to questionnaires. Public Opin Q 20:36–44

Zika-Viktorsson A, Sundström P, Engwall M (2006) Project overload: an exploratory study of work and management in multi-project settings. Int J Proj Manag 24(5):385–394

Zmud RK (1978) An empirical investigation of the dimensionality of the concept of in-formation. Decis Sci 9(2):187–195

Chapter 62
Resource-Constrained Project Scheduling with Project Management Information Systems

Philipp Baumann and Norbert Trautmann

Abstract Most commercial project management software packages include planning methods to devise schedules for resource-constrained projects. As it is proprietary information of the software vendors which planning methods are implemented, the question arises how the software packages differ in quality with respect to their resource-allocation capabilities. We experimentally evaluate the resource-allocation capabilities of eight recent software packages by using 1,560 instances with 30, 60, and 120 activities of the well-known PSPLIB library. In some of the analyzed packages, the user may influence the resource allocation by means of multi-level priority rules, whereas in other packages, only few options can be chosen. We study the impact of various complexity parameters and priority rules on the project duration obtained by the software packages. The results indicate that the resource-allocation capabilities of these packages differ significantly. In general, the relative gap between the packages gets larger with increasing resource scarcity and with increasing number of activities. Moreover, the selection of the priority rule has a considerable impact on the project duration. Surprisingly, when selecting a priority rule in the packages where it is possible, both the mean and the variance of the project duration are in general worse than for the packages which do not offer the selection of a priority rule.

Keywords Experimental performance analysis • Project management • Project management information systems • Project scheduling

62.1 Introduction

A project is a temporary endeavor that can be divided into a series of activities which are interrelated by precedence constraints and require time and resources for their execution. The planning of a project includes the computation of the earliest and latest start times and the slack times of the activities (temporal

P. Baumann (✉) • N. Trautmann
Department of Business Administration, University of Bern, Bern, Switzerland
e-mail: philipp.baumann@pqm.unibe.ch; norbert.trautmann@pqm.unibe.ch

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0_32

scheduling), and the allocation of scarce resources over time to the execution of the activities (resource allocation). In practice, project managers use project management software for temporal scheduling and resource allocation (cf., e.g., White and Fortune 2002; Liberatore and Pollack-Johnson 2003; Herroelen 2005). While the temporal-scheduling problem can be solved efficiently by longest path length calculations (cf., e.g., Lawler 1976), the resource-allocation problem is in general difficult to solve. Therefore, project management software packages use heuristic solution procedures for allocating resources to activities.

In this chapter, we analyze the resource-allocation capabilities of recent project management software packages by means of an experimental performance analysis. Following previous papers on this subject (cf., e.g., Johnson 1992; Kolisch 1999; Mellentien and Trautmann 2001; Trautmann and Baumann 2009), we investigate instances of the resource-constrained project scheduling problem RCPSP. This problem consists in determining a start time for each activity subject to finish-start precedence relationships and constraints on the resource capacities such that the project duration is minimized. Our analysis refers to the software packages Acos Plus.1, Adept Tracker Professional, CS Project Professional, Microsoft Project 2010, Microsoft Project 2013, Primavera P6, Sciforma 5.0, and Turbo Project Professional. For the experimental performance analysis we use the 1,560 PSPLIB instances with 30, 60, and 120 activities and the software packages. Results for (rather few) other instances have also been presented by Maroto and Tormos (1994), Farid and Manoharan (1996), Khattab and Søyland (1996), Lova and Tormos (2001), and Kastor and Sirakoulis (2009).

The outline of the present chapter is as follows. In Sect. 62.2, we describe the resource-allocation features of the different software packages; for the packages which offer the selection of a priority rule, we provide a short description of the rules. In Sect. 62.3, we report the results of our experimental performance analysis. In Sect. 62.4, we give some concluding remarks.

62.2 Project Management Information Systems

We restricted our analysis to commercial software packages that comprise a resource-allocation procedure and can be controlled by Visual Basic for Applications (VBA); the latter is required to perform the experimental analysis automatically. Table 62.1 lists for each package the name, the developer, and the number of the release used.

62.2.1 Acos Plus.1

In Acos Plus.1 (cf. Fig. 62.1) the user may choose between 12 different priority rules. A description of each rule is given in Table 62.2.

Table 62.1 Analyzed software packages

Name	Developer	Release (Built)
Acos Plus.1	ACOS Projektmanagement GmbH	9.4b (755)
Adept Tracker Professional	WangTuo Software	3.13 (10953)
CS Project Professional	CREST Software	3.8 (.06)
Microsoft Project 2010	Microsoft Corporation	14 (6129.5000)
Microsoft Project 2013	Microsoft Corporation	15 (4420.1017)
Primavera P6	Oracle Corporation	8.2 (1926)
Sciforma 5	Sciforma Corporation	5.0c (2994)
Turbo Project Professional	OfficeWork Software	4 (221.5)

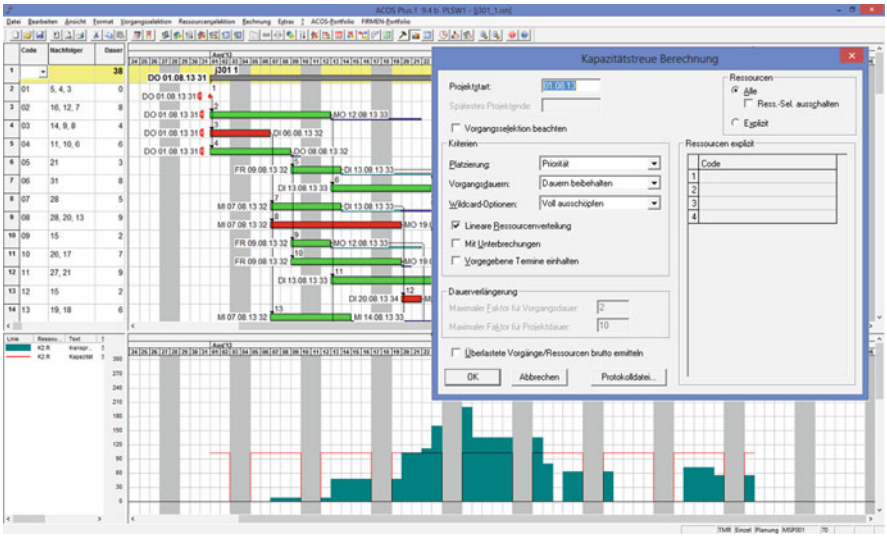


Fig. 62.1 Acos Plus.1 (Instance j301_1)

The user may specify priority values for individual activities from the interval [0,999]. The default priority value for each activity is 0. In the resource-allocation dialog of Acos Plus.1, the user has the possibility to (a) limit resource allocation to specific resources or selected activities, (b) chose a priority rule, and (c) allow the procedure to change durations of tasks or to interrupt activities. By default, priority rule A1 is selected, even if the user has not specified any priority values at all.

In our analysis we tested priority rules A1, A2, A4, A5, A6, A7 and A8. When priority rule A3 was selected, Acos Plus.1 created infeasible schedules for some instances. Therefore, we excluded priority rule A3 from our tests. We did not test rules A10, A11 and A12 because we did neither perform a specific sorting of activities nor defined any main activities.

Table 62.2 Priority rules of Acos Plus.1

Name	Id	Description
Priority	A1	Higher priorities for activities with a lower priority value
Total float	A2	Higher priority for activities with less total float
Critical path	A3	Higher priority for activities that constitute the critical path
Longest duration	A4	Higher priority for activities with a longer duration
Free float	A5	Higher priority for activities with less free float
Total number of predecessors	A6	Higher priority for activities with more direct predecessors
Total number of successors	A7	Higher priority for activities with more direct successors
Total number of predecessors and successors	A8	Higher priority for activities with a larger total number of predecessors and successors
Shortest duration	A9	Higher priority for activities with a shorter duration
Sorting	A10	Higher priority for activities with a lower position in the current sorted list of activities
Priority of main-activity	A11	Higher priority for activities whose main activity has a lower priority value
Sorting of main-activity	A12	Higher priority for activities whose main activity has a lower position in the current sorted list of activities

62.2.2 Adept Tracker Professional

In Adept Tracker Professional (ATP) (cf. Fig. 62.2), five different resource-allocation methods are available. According to the help file, all methods aim at minimizing the project duration when resolving resource over-allocations. The methods differ in the degree of optimization used for resource allocation. Methods with a high degree of optimization may adjust the sequence and the individual start times of all activities without limitation, whereas methods with a low degree of optimization try to modify the current schedule as little as possible when resolving resource conflicts. For example, the method "stable order" is not allowed to change the current sequence of activities. The user may define priority values of activities manually. The larger this value, the higher is the priority. By default, a value of 500 is set for each activity. There are no priority rules to choose from. In the resource allocation dialog, the user can (a) specify a start date for the

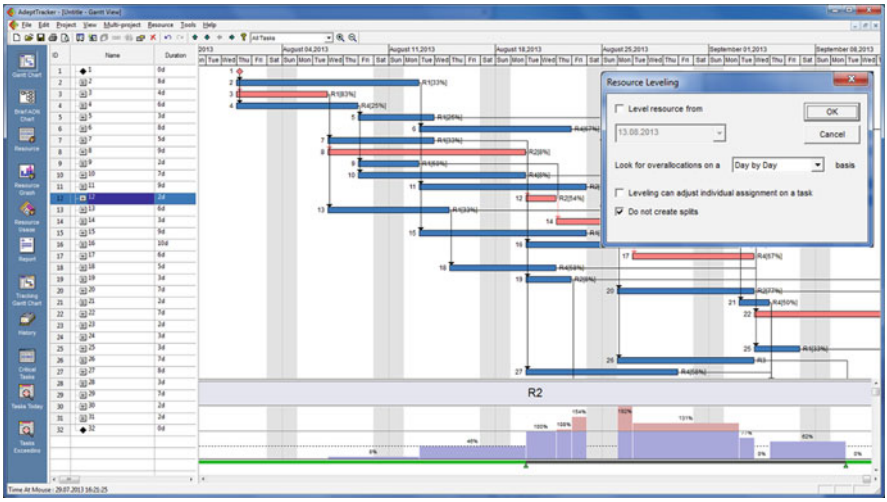


Fig. 62.2 Adept Tracker Professional (Instance j301_1)

resource allocation procedure, (b) allow activity interruptions, and (c) adjustments of resource requirements of activities.

For our analysis we investigated the method "level resources manually" which has the highest degree of optimization. We did not specify any priority values for activities.

62.2.3 CS Project Professional

CS Project Professional (cf. Fig. 62.3) offers eight rules to compute priority values for activities. A description of each rule can be found in Table 62.3.

The user may specify priority values for individual activities from the interval [1,64000]. The default priority value for each activity is 50. In the resource-allocation dialog, the user can define multi-level priority rules using the rules listed in Table 62.3 in ascending (a) or descending (d) order (cf. Fig. 62.3). Thereby up to four levels can be used. By default, the combination C3a-C1a-C5a-C2a is selected. Instead of manually defining multi-level priority rules, the user can apply the CARLO ("cost and resource levelling optimisation") algorithm which automatically tries different combinations and returns, according to the help file, the best schedule found. However, it is not indicated which criterion is used to select the best schedule.

In our analysis we tested all combinations of rules C1, C2, C4, C5, and C8 (ascending and descending) for the first three levels plus the selection of no rule for the second and the third level (570 combinations in total). In addition, we tested the default rule and the CARLO algorithm. We did not test rules C3, C6 and C7 because we did not specify activity-specific priority values or baseline schedules.

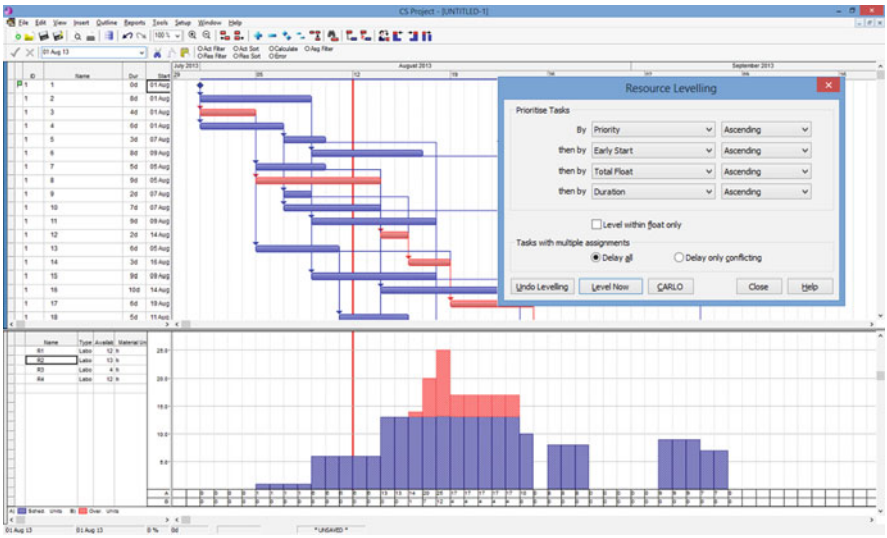


Fig. 62.3 CS Project Professional (Instance j301_1)

Table 62.3 Priority rules of CS Project Professional

Name	Id	Ascending	Descending
Total float	C1	Higher priority for activities with less total float	Higher priority for activities with more total float
Free float	C2	Higher priority for activities with less free float	Higher priority for activities with more free float
Priority	C3	Higher priority for activities with a lower priority value	Higher priority for activities with a higher priority value
Duration	C4	Higher priority for activities with a shorter duration	Higher priority for activities with a longer duration
Early start	C5	Higher priority for activities with an earlier early start date	Higher priority for activities with a later early start date
Start-baseline	C6	Higher priority for activities with an earlier start date in the baseline schedule	Higher priority for activities with a later start date in the baseline schedule
Finish-baseline	C7	Higher priority for activities with an earlier finish date in the baseline schedule	Higher priority for activities with a later finish date in the baseline schedule
Late finish	C8	Higher priority for activities with an earlier late finish date	Higher priority for activities with a later late finish date

62.2.4 Microsoft Project 2010

In Microsoft Project 2010 (cf. Fig. 62.4), the user must choose a period length (e.g., minute, day, week) for resource allocation. A resource is considered as

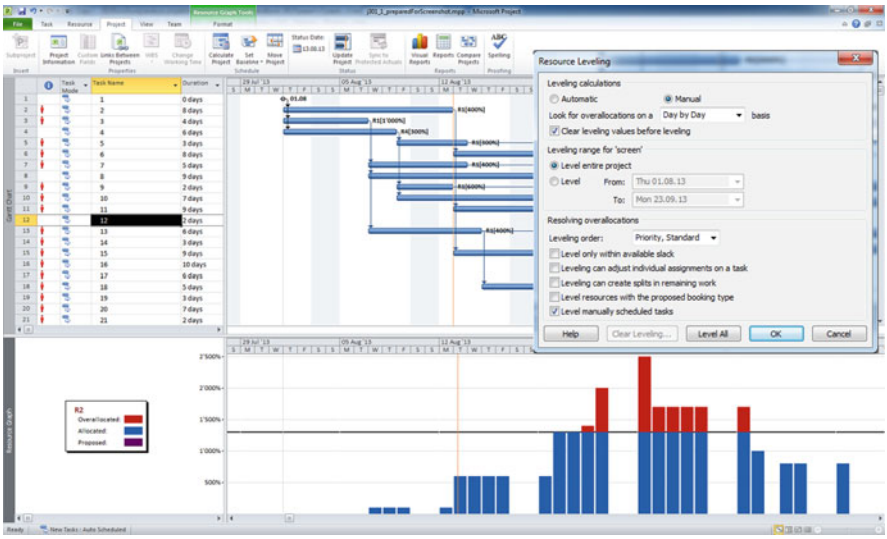


Fig. 62.4 Microsoft Project 2010 (Instance j301_1)

over-allocated only when the total requirement within the period exceeds the total capacity. Individual time points within the period are not checked for over-allocations.

The user may specify priority values for individual activities from the interval [0,1000]; a higher value indicates a higher priority. Activities with a priority value of 1,000 cannot be moved by the resource-allocation procedure. The default priority value for each activity is 500. Alternatively, the priority value for an activity can be set to the activity ID or to a predefined value which is computed based on precedence relationships and float times.

In our analysis we used the predefined priority values for resource allocation.

62.2.5 Microsoft Project 2013

Microsoft Project 2013 (cf. Fig. 62.5) does not offer any additional options for resource allocation. However, the computation of the predefined priority values for activities differs from the 2010 version.

62.2.6 Primavera P6

In Primavera P6 (cf. Fig. 62.6) 13 different priority rules are implemented (cf. Table 62.4).

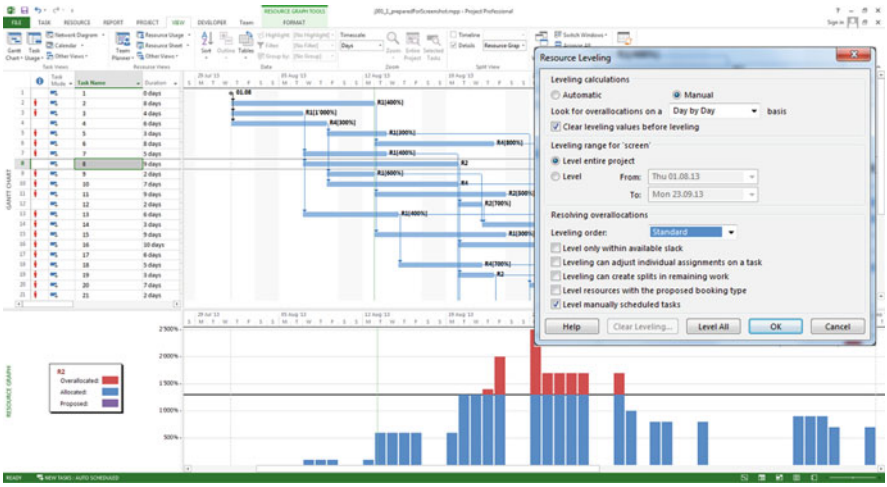


Fig. 62.5 Microsoft Project 2013 (Instance j301_1)

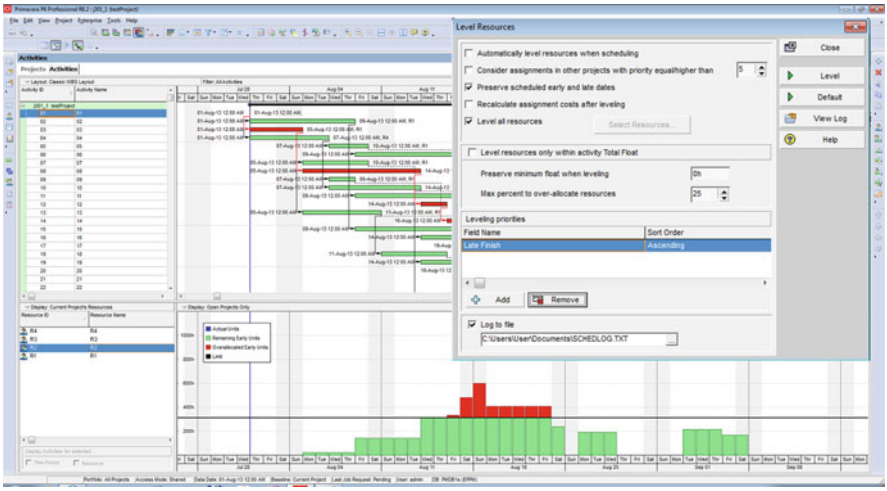


Fig. 62.6 Primavera P6 (Instance j301_1)

Similar to CS Project Professional, each of these rules can be combined in ascending or descending order in a multi-level hierarchy. In Primavera P6, no limit on the number of levels is imposed. By default, P2 is selected as a single-level rule. If the user wants to specify individual priority values for activities, he must choose between five different values (1,. . . ,5). The default priority value for each activity is 3.

Table 62.4 Priority rules of Primavera P6

Name	Id	Ascending	Descending
Activity ID	P1	Higher priority for activities with a lower ID	Higher priority for activities with a higher ID
Activity leveling priority	P2	Higher priority for activities with a low leveling priority	Higher priority for activities with a high leveling priority
Early finish	P3	Higher priority for activities with an earlier early finish date	Higher priority for activities with a later early finish date
Early start	P4	Higher priority for activities with an earlier early start date	Higher priority for activities with a later early start date
Free float	P5	Higher priority for activities with less free float	Higher priority for activities with more free float
Late finish	P6	Higher priority for activities with an earlier late finish date	Higher priority for activities with a later late finish date
Late start	P7	Higher priority for activities with an earlier late start date	Higher priority for activities with a later late start date
Planned duration	P8	Higher priority for activities with a shorter planned duration	Higher priority for activities with a longer planned duration
Planned finish	P9	Higher priority for activities with an earlier planned finish date	Higher priority for activities with a later planned finish date
Planned start	P10	Higher priority for activities with an earlier planned start date	Higher priority for activities with a later planned start date
Project leveling priority	P11	Higher priority for activities that belong to a lower priority project	Higher priority for activities that belong to a higher priority project
Remaining duration	P12	Higher priority for activities with a shorter remaining duration	Higher priority for activities with a longer remaining duration
Total float	P13	Higher priority for activities with less total float	Higher priority for activities with more total float

In our analysis we tested all combinations of rules P3, P4, P5, P6, P7, P8 and P13 (ascending and descending) for the first two levels (196 combinations in total). In addition, we tested the default rule of Primavera P6. We did not test rules P1, P9, P10, P11 and P12 because we considered only single project instances for which the planned start and finish dates coincide with the early start and early finish dates, and for which the remaining durations of activities coincide with the respective planned durations.

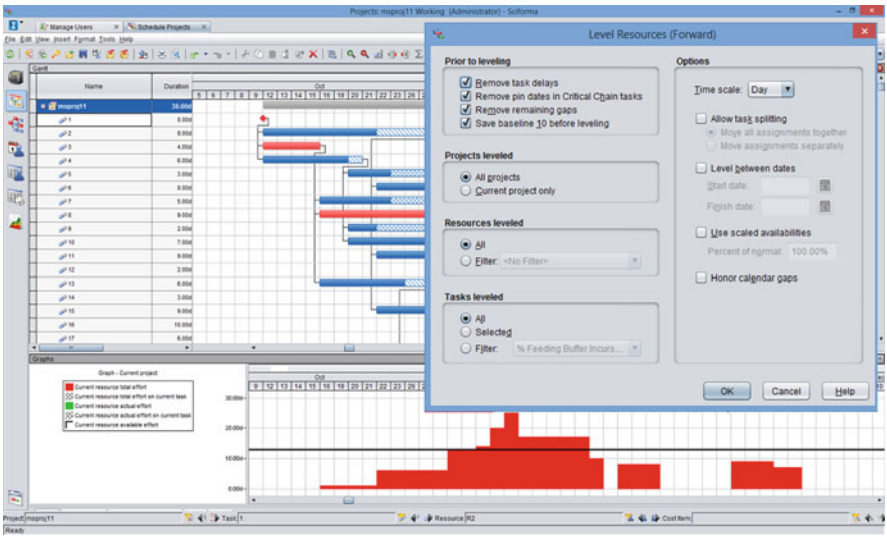


Fig. 62.7 Sciforma 5.0 (Instance j301_1)

62.2.7 Sciforma 5.0

In Sciforma 5.0 (cf. Fig. 62.7) the user has to choose a period length for resource allocation, similar to Microsoft Project 2010 or 2013. Resource allocation can be performed for the complete project or for selected resources and tasks only. The time horizon within over-allocations are resolved can also be specified explicitly. Furthermore it is possible to allow a certain exceedance of the resource capacities. The user may set individual priority values, but cannot choose between different priority rules.

62.2.8 Turbo Project Professional

Turbo Project Professional (cf. Fig. 62.8) also offers the possibility to limit resource allocation to certain resources, tasks and a specific time horizon. The activity priorities can be set to a user-defined value or to a predefined value. The package does not offer different priority rules.

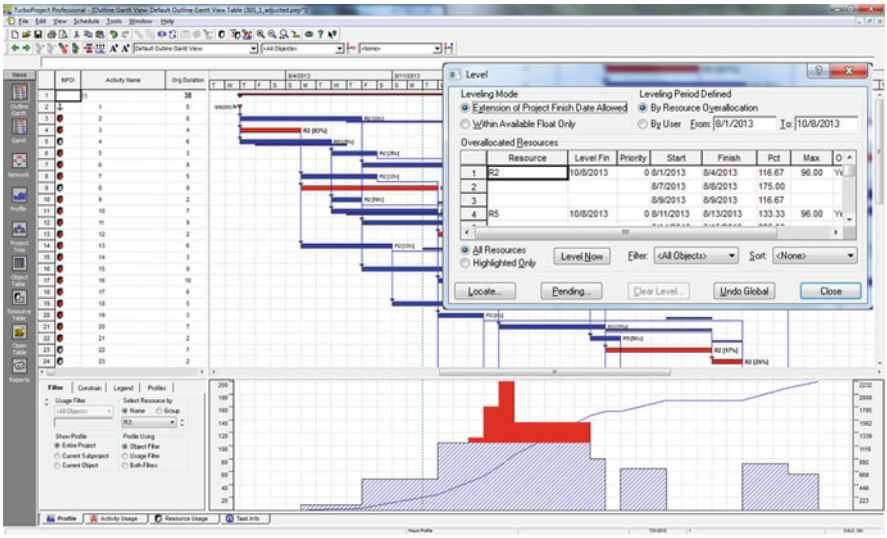


Fig. 62.8 Turbo Project Professional (Instance j301_1)

62.3 Experimental Analysis

For our analysis, we installed the eight software packages on various standard PCs with Windows XP or Windows 8 as operating system; we used Primavera P6 on a Windows 7 system via a remote access. We downloaded the project durations in the optimal (J30 set) and in the best known feasible schedules (J60 and J120 set), respectively, from the Internet page http://129.187.106.231/psplib/ on 2013-08-01. The respective schedules were obtained by the state-of-the-art algorithms from the literature. In our analysis, we used these project durations as reference values.

The CPU time required by the various packages for resource allocation of a single project never exceeded 30 s. With all packages and options, we always obtained a feasible schedule. For none of the 1,560 projects, any software package computed a schedule with a shorter project duration than the reference value.

62.3.1 Results for Default Resource-Allocation Options

First, we analyze the schedules that we obtained for the default resource-allocation options. Tables 62.5 and 62.6, list the mean and the maximum, as well as the variance, respectively, of the relative makespan deviation from the reference values for the three test sets. For all 1,560 instances, Acos Plus.1 (total mean deviation of 7.95 %) computed the best project schedules with regard to the mean deviation, followed by Adept Tracker Professional (8.35 %) and Microsoft Project 2010

Table 62.5 Default options: mean and maximum of relative makespan deviation [%]

	Mean				Maximum			
n	30	60	120	All	30	60	120	All
Acos Plus.1	5.42	5.37	12.03	7.95	35.56	35.63	38.98	38.98
Adept Tracker Professional	5.58	5.76	12.63	8.35	33.90	37.62	36.13	37.62
Microsoft Project 2010	5.03	6.25	15.77	9.54	32.76	38.00	47.40	47.40
CS Project Professional	9.39	10.11	21.05	14.10	51.56	55.17	49.04	55.17
Sciforma 5	9.45	10.47	21.13	14.26	44.44	56.00	71.74	71.74
Primavera P6	9.45	10.54	24.16	15.44	44.44	56.00	50.89	56.00
Turbo Project Professional	8.94	10.22	25.19	15.58	57.14	54.43	72.90	72.90
Microsoft Project 2013	9.60	11.56	25.73	16.40	58.82	75.00	86.05	86.05

Table 62.6 Default options: variance of relative makespan deviation [$\%^2$]

n	30	60	120	All
Acos Plus.1	51.91	67.11	72.05	74.74
Adept Tracker Professional	54.74	73.26	76.91	80.44
Microsoft Project 2010	49.72	81.10	121.04	111.33
Sciforma 5	93.43	119.48	81.40	126.49
Primavera P6	93.43	120.52	99.36	151.70
CS Project Professional	125.38	149.02	130.54	164.96
Turbo Project Professional	123.47	171.43	235.55	239.28
Microsoft Project 2013	144.88	233.80	332.40	299.27

(9.54 %). The worst schedules were obtained by Microsoft Project 2013 (total mean deviation of 16.40 %). Thus, for a user who does not want to decide about any resource-allocation options, one of the packages that offer only few or not any options seems to be a good choice.

62.3.2 Impact of Priority Rules

Next, we analyze how good are the schedules obtained when making use of the various resource-allocation options. If in doing so a software computed several schedules with different objective function values for the same instance, we analyze the schedule with the shortest (best case) and the schedule with the longest (worst case) project duration.

Table 62.7 lists the mean and the maximum of the relative makespan deviation from the reference values for the test sets J30, J60, and J120 and the best-case scenario. In contrast to the situation where the default resource-allocation options were selected, with regard to the mean deviation Primavera P6 (5.69 % in total) and Acos Plus.1 (6.57 %) outperform the other packages. When additionally taking into account the maximum deviation, Acos Plus.1 again seems to be the best choice.

Table 62.7 Priority rules, best case: mean and maximum of relative makespan deviation [%]

	Mean				Maximum			
n	30	60	120	All	30	60	120	All
Primavera P6	2.38	3.75	9.90	5.69	18.64	24.24	48.84	48.84
Acos Plus.1	3.99	4.42	10.34	6.57	27.78	30.10	34.27	34.27
Adept Tracker Professional	5.58	5.76	12.63	8.35	33.90	37.62	36.13	37.62
CS Project Professional	3.32	5.46	14.70	8.35	20.59	23.89	32.89	32.89
Microsoft Project 2010	5.03	6.25	15.77	9.54	32.76	38.00	47.40	47.40
Sciforma 5	9.45	10.47	21.13	14.26	44.44	56.00	71.74	71.74
Turbo Project Professional	8.94	10.22	25.19	15.58	57.14	54.43	72.90	72.90
Microsoft Project 2013	9.60	11.56	25.73	16.40	58.82	75.00	86.05	86.05

Table 62.8 Priority rules, worst case: mean and maximum of relative makespan deviation [%]

	Mean				Maximum			
n	30	60	120	All	30	60	120	All
Adept Tracker Professional	5.58	5.76	12.63	8.35	33.90	37.62	36.13	37.62
Microsoft Project 2010	5.03	6.25	15.77	9.54	32.76	38.00	47.40	47.40
Acos Plus.1	7.55	7.09	15.49	10.46	35.56	37.08	45.16	45.16
Sciforma 5	9.45	10.47	21.13	14.26	44.44	56.00	71.74	71.74
Turbo Project Professional	8.94	10.22	25.19	15.58	57.14	54.43	72.90	72.90
Microsoft Project 2013	9.60	11.56	25.73	16.40	58.82	75.00	86.05	86.05
CS Project Professional	18.02	18.26	32.40	23.63	74.36	60.00	58.00	74.36
Primavera P6	29.02	32.14	56.41	40.51	87.18	113.89	111.23	113.89

Recall that Table 62.7 refers to the case where for each instance and software, a priority rule providing the best schedule was selected. The opposite situation is addressed in Table 62.8; here, we assume that the user always selected the worst priority-rule. Primavera P6 and CS Project Professional deliver rather poor project schedules, whereas Acos Plus.1 again performs rather good. We note that in the worst-case scenario, for a project with 60 activities, Primavera P6 computes a schedule that takes more than 113 % longer than necessary.

62.3.3 Impact of Complexity Scenarios

In this subsection, we evaluate the resource-allocation capabilities for different complexity scenarios, characterized by the mean number of resources used, the scarcity of the resources, and the mean number of precedence relationships. The analysis is based on the best-case scenario discussed in Sect. 62.3.2 and limited to the 600 instances with 120 activities; the results for the smaller instances are similar. Interestingly, all seven software packages behave uniform in this analysis.

Table 62.9 Mean relative makespan deviation in set J120 for various resource factors (best case of all priority rules) [%]

RF	0.25	0.50	0.75	1.00
Acos Plus.1	5.45	12.09	12.41	11.39
Adept Tracker Professional	6.75	15.45	14.93	13.39
CS Project Professional	8.19	16.92	18.03	15.65
Microsoft Project 2010	7.49	19.18	19.48	16.94
Microsoft Project 2013	6.18	38.80	38.71	19.22
Primavera P6	4.70	11.58	12.28	11.03
Sciforma 5	14.90	23.89	24.13	21.59
Turbo Project Professional	16.06	31.91	30.65	22.15

Table 62.10 Mean relative makespan deviation in set J120 for various resource strengths (best case of all priority rules) [%]

RS	0.5	0.4	0.3	0.2	0.1
Acos Plus.1	1.99	5.98	10.19	14.55	18.97
Adept Tracker Professional	3.09	8.06	12.36	17.64	22.00
CS Project Professional	6.15	11.89	15.57	18.63	21.26
Microsoft Project 2010	4.19	10.18	14.89	20.95	28.66
Microsoft Project 2013	12.64	19.33	26.15	31.61	38.91
Primavera P6	1.91	5.94	9.79	13.90	17.94
Sciforma 5	13.32	17.93	22.06	25.42	26.90
Turbo Project Professional	9.39	18.15	24.61	32.58	41.24

Table 62.11 Mean relative makespan deviation in set J120 for various network complexities (best case of all priority rules) [%]

NC	1.5	1.8	2.1
Acos Plus.1	9.61	10.19	11.21
Adept Tracker Professional	11.89	12.33	13.66
CS Project Professional	14.19	14.65	15.25
Microsoft Project 2010	15.55	15.24	16.53
Microsoft Project 2013	25.01	25.65	26.53
Primavera P6	9.23	9.80	10.66
Sciforma 5	20.55	20.73	22.11
Turbo Project Professional	24.55	24.86	26.17

Table 62.9 shows that if all activities use one resource only ($RF = 0.25$) or all resources ($RF = 1$), the mean deviation of makespan is smaller than in the case of two or more resources used.

Table 62.10 lists the mean makespan deviation relative to the resource strength. With increasing resource scarcity (decreasing *RS* value), the deviation increases noticeably.

Table 62.11 indicates that the mean number of precedence relationships *NC* does not affect remarkably the resource-allocation quality of any of the tested packages.

62.4 Conclusions

In this chapter, we have reported on the results of an experimental analysis in which we evaluated the resource-allocation capabilities of eight commercial project management software packages. For the resource- and precedence constrained project scheduling problem RCPSP, it has turned out that when using any of these packages for resource allocation, a project manager must be aware of the risk that the project takes considerably more time than necessary. This gap increases significantly with the number of project activities and the resource scarcity. The results of this study indicate that none of the tested software packages is currently competitive with the best state-of-the-art algorithms from the literature. When working with the default resource-allocation options only, the project durations computed by the software packages Acos Plus.1, Adept Tracker Professional and Microsoft Project 2010 are noticeably shorter than for the other packages. Yet shorter project durations may be obtained by the package Primavera P6; however, this requires in general to investigate a large number of alternative priority rules.

Additional insights could be gained by an analysis of the schedule-generation schemes used in the different software packages. Moreover, the distribution of the makespan deviation for the various priority rules and software packages should be investigated in more detail; this would allow for a more precise comparison of the risk of obtaining an unnecessary large project duration with the different software packages.

References

Farid F, Manoharan S (1996) Comparative analysis of resource-allocation capabilities of project management software packages. Proj Manag J 27:35–44

Herroelen W (2005) Project scheduling–theory and practice. Prod Oper Manag 14:413–432

Johnson R (1992) Resource constrained scheduling capabilities of commercial project management software. Proj Manag J 22:39–43

Kastor A, Sirakoulis K (2009) The effectiveness of resource levelling tools for resource constraint project scheduling problem. Int J Proj Manag 27:493–500

Khattab M, Søyland K (1996) Limited resource-allocation in construction projects. Comput Ind Eng 31:229–232

Kolisch R (1999) Resource allocation capabilities of commercial project management software packages. Interfaces 29:19–31

Lawler E (1976) Combinatorial optimization: networks and Matroids. Holt, Rinehard, and Winston, New York

Liberatore M, Pollack-Johnson B (2003) Factors influencing the usage and selection of project management software. IEEE Trans Eng Manag 50:164–174

Lova A, Tormos P (2001) Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling. Ann Oper Res 102:263–286

Maroto C, Tormos P (1994) Project management: an evaluation of software quality. Int Trans Oper Res 1:209–221

Mellentien C, Trautmann N (2001) Resource allocation with project management software. OR Spectr 23:383–394

Trautmann N, Baumann P (2009) Resource-allocation capabilities of commercial project management software: an experimental analysis. In: Kacem I (ed) Proceedings of the 39th international conference on computers & industrial engineering, Troyes, pp 1155–1160

White D, Fortune J (2002) Current practice in project management–anempirical study. Int J Proj Manag 20:1–11

Index

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 2*, International Handbooks on Information Systems,
DOI 10.1007/978-3-319-05915-0