

Constrained Optimal Hybrid Control of a Flow Shop System

Kagan Gokbayrak, *Member, IEEE*, and Omer Selvi

Abstract—We consider an optimal control problem for the hybrid model of a deterministic flow shop system, in which the jobs are processed in the order they arrive at the system. The problem is decomposed into a higher-level discrete-event system control problem of determining the optimal service times, and a set of lower-level classical control problems of determining the optimal control inputs for given service times. We focus on the higher-level problem which is nonconvex and nondifferentiable. The arrival times are known and the decision variables are the service times that are controllable within constraints. We present an equivalent convex optimization problem with linear constraints. Under some cost assumptions, we show that no waiting is observed on the optimal sample path. This property allows us to simplify the convex optimization problem by eliminating variables and constraints. We also prove, under an additional strict convexity assumption, the uniqueness of the optimal solution and propose two algorithms to decompose the simplified convex optimization problem into a set of smaller convex optimization problems. The effects of the simplification and the decomposition on the solution times are shown on an example problem.

Index Terms—Constrained hybrid control, controllable processing times, flow shop, hierarchical decomposition, optimal control.

I. INTRODUCTION

WE consider a deterministic flow shop system consisting of M stages that are processing identical jobs. Associated with each job are physical and temporal states. The physical state of a job i at stage j denoted by $z_{i,j}$ describes a measure of quality of the job such as temperature, chemical composition, bacteria level, etc. At each stage j , the physical state $z_{i,j}$ at time $\tau_{i,j}$ starts out at the initial state ζ_j^0 and a control input $u_{i,j}$ is applied. The physical state is modified to reach the prespecified target state ζ_j^d according to the time-driven dynamics described by the differential equations

$$\dot{z}_{i,j}(t) = f_j(z_{i,j}(t), u_{i,j}(t)) \quad (1)$$

$$z_{i,j}(\tau_{i,j}) = \zeta_j^0, \quad z_{i,j}(\tau_{i,j} + s_{i,j}) = \zeta_j^d \quad (2)$$

over the time period $(\tau_{i,j}, \tau_{i,j} + s_{i,j}]$. If and when the physical state reaches ζ_j^d clearly depends on the control input $u_{i,j}$ and the physical dynamics (1), (2). We assume that control inputs $u_{i,j}$ that can bring the physical state to the target state are available and denote the corresponding service times as $s_{i,j}(u_{i,j})$.

We also assume that the physical states are not altered during the waiting periods; hence, the initial physical states for these identical jobs at each stage are independent of the job index.

A service cost $\psi_j(\cdot)$, given as

$$\psi_j(z_{i,j}(t), u_{i,j}(t), s_{i,j}, \tau_{i,j}) = \int_{\tau_{i,j}}^{\tau_{i,j} + s_{i,j}} F(z_{i,j}(t), u_{i,j}(t)) dt \quad (3)$$

is associated with the physical process applied on job i at stage j . Since the final value ζ_j^d is prespecified, we do not have an additional cost term for the final state. In order to simplify the analysis, we assume time invariance of the system and drop the $\tau_{i,j}$ term from $\psi_j(\cdot)$. Hence, in what follows, the service cost is denoted by $\psi_j(z_{i,j}(t), u_{i,j}(t), s_{i,j})$.

The temporal state $x_{i,j}$, on the other hand, keeps the departure time information of job i from stage j and evolves according to the event-driven dynamics given by the Lindley equation (see [1])

$$x_{i,j} = \max(x_{i,j-1}, x_{i-1,j}) + s_{i,j}(u_{i,j}(t)) \quad (4)$$

$$x_{i,0} = a_i, \quad x_{0,j} = -\infty \quad (5)$$

where a_i is the arrival time of job i at the system, assumed to be prespecified. A completion-time cost $\phi_i(x_{i,M})$, which is assumed to be regular (increasing in $x_{i,M}$), is incurred for each job i departing the system.

In what follows, we consider the minimization of a bicriteria objective function, consisting of service and completion-time costs, subject to physical dynamics (1), (2) and temporal dynamics (4), (5). Our decision variables are the control inputs $u_{i,j}$ that are observed both in physical and temporal dynamics; therefore, the problem considered can be classified as a hybrid control problem. The hierarchical method, proposed in [2], [3], and [4], allows us to decompose the original hybrid control problem into several lower-level continuous-time optimal control problems and a higher-level discrete-event control problem of determining the optimal service times. The lower-level problems, rather simple applications of the classical optimal control theory, define service costs that depend only on the processing times, as well as mappings from the processing times to the optimal control inputs. The higher-level problem with service times $s_{i,j}$ as the decision variables, on the other hand, is challenging, therefore, attracted attention in the scheduling and the hybrid system control contexts.

In the scheduling literature, sequencing problems with controllable processing times have been considered over the

Manuscript received May 15, 2006; revised March 3, 2007. Recommended by Associate Editor J. Hespanha.

The authors are with the Department of Industrial Engineering, Bilkent University, Ankara, 06800 Turkey (e-mail: kgokbayr@bilkent.edu.tr; selvi@bilkent.edu.tr).

Digital Object Identifier 10.1109/TAC.2007.910668

last three decades. Vickson, in [5], considered the single-machine sequencing problem with the objective of minimizing a total schedule cost consisting of service costs, assumed to be decreasing linear functions of processing times, and weighted completion-time costs. A heuristic algorithm was proposed to determine the optimal job sequence and the corresponding processing times, which are chosen from continuous sets. Chen *et al.*, in [6], considered the single-machine sequencing problem with service costs that are decreasing (not necessarily linear) functions of discrete processing times. It was shown that the problems with total schedule costs consisting of service costs and weighted completion-time or earliness-tardiness-penalty costs can be formulated as assignment problems, hence, can be solved in polynomial time. Similar scheduling problems for parallel machines were considered by Alidaee and Ahmadian in [7] and Cheng *et al.* in [8]. Alidaee and Ahmadian assumed decreasing linear service costs and formulated the problems as transportation problems that can be solved by polynomial time algorithms. Cheng *et al.*, on the other hand, considered decreasing convex service costs, and formulated the optimization problems as assignment problems, which can also be solved in polynomial time. A survey of results on the controllable processing times can be found in [9] and [10].

The scheduling problems of permutation flow shops are known to be NP-hard even for fixed processing times (see [11]). Therefore, the scheduling literature for multistage systems is limited to heuristics and approximate solution methods. Nowicki in [12], assumed decreasing linear service costs on each machine and provided an approximation algorithm for minimizing the sum of service and weighted makespan costs. In the flow shop scheduling problems, the task of selecting the optimal processing times for a given sequence is viewed as a subproblem, e.g., by Karabati and Kouvelis in [13]. Karabati and Kouvelis considered a scheduling cost, which is a weighted sum of service and cycle time costs for a minimum product set. In their iterative algorithm, the service cost considered is a decreasing linear function of the processing times, so that a linear programming formulation is obtained and solved by a row generation scheme at each iteration.

The related hybrid system control literature, on the other hand, assumes that jobs are served in a given sequence, and concentrates on determining the optimal control inputs, which in turn determine the optimal processing times. Even though this line of research seems to be solving the subproblem in scheduling, an important difference is that the service costs depend on the time-driven dynamics (1), (2); hence, they cannot be assumed to have an arbitrary function form. Moreover, the optimal control inputs for the physical processes are also determined. Pepyne and Cassandras in [14] formulated an optimal control problem for a single-stage manufacturing process and used calculus of variations techniques to obtain structural properties of the optimal solution. The quadratic objective function was designed to complete jobs as fast as possible with the least amount of control effort. In [15], they extended their results to nonregular completion-time costs penalizing earliness and tardiness with given due dates. The uniqueness of the optimal solution of the generalized problem was shown in

[16]. Exploiting the structural properties of the optimal sample path, efficient algorithms were developed in [17], [18], and [19] for solving the higher-level problem for single-stage systems. In these “backward-in-time” and “forward-in-time” algorithms, the original higher-level problem was decomposed into a set of smaller convex optimization problems with linear constraints. In the case of blocking, an efficient algorithm was presented in [20]. Similar models exist for optimal release time (see [21] and [22]) and lot-sizing (see [23]) problems of manufacturing systems.

The application of the hybrid systems framework to multistage systems was proposed in [24] where approximate solutions for two-stage systems were obtained using the Bezier approximation method to smooth out the max functions in the event-driven dynamics. Earlier work in [25] considered a multistage hybrid system model with constrained service times and presented some optimal sample path characteristics. In [26], we considered two-stage manufacturing systems with no constraints on the service times and identified some new optimal sample path characteristics to simplify the discrete-event control problem. In particular, we showed that no waiting is observed between stages on the optimal sample path. The transformation of the nonsmooth discrete-event optimal control problem into an equivalent convex optimization problem with linear constraints was also presented in [26], which was then simplified utilizing the no-wait property. In this paper, we combine and extend the results from [18], [25], and [26] for a multistage model with constrained service times. We show that the no-wait property extends to these systems for which simplified convex optimization problems are determined. Under an additional strict convexity assumption, these optimization problems are decomposed by “forward-in-time” algorithms into smaller convex optimization problems with linear constraints.

The rest of the paper is organized as follows. In Section II, we set up a constrained optimal hybrid control problem and decompose it into several lower-level continuous-time optimal control problems and a higher-level discrete-event optimal control problem. Since the continuous-time optimal control methods are well established, we focus on the discrete-event optimal control problem, which is nonconvex and nondifferentiable, therefore, challenging. In this section, we also present a convex optimization problem, which has the same solution as the discrete-event optimal control problem. Section III presents some characteristics of the optimal solution and shows that no waiting is observed between stages on the optimal sample path. The convex optimization problem is, then, simplified employing the no-wait property. Also in Section III, we give independent period definitions and present a decoupling property required for the decomposition algorithms. Two forward decomposition algorithms are also presented in this section to decompose the simplified convex problem into smaller convex optimization problems with linear constraints. In order to demonstrate the solution method, Section IV presents an example system, where each job goes through a sequence of linear time invariant processes. Numerical results are also given in this section to show the benefits of the simplification and the decomposition in terms of solution times. Finally, Section V concludes the paper.

II. PROBLEM FORMULATION

Let us consider a sequence of N identical jobs arriving at an M -stage flow shop system at prespecified times $0 \leq a_1 \leq a_2 \leq \dots \leq a_N$. Servers process one job at a time on a first-come first-served nonpreemptive basis. The service times $s_{i,j}$ are constrained to be $s_{i,j} \geq S_j$, i.e., the process at stage j takes at least S_j amount of time possibly due to bounded inputs $|u_{i,j}(t)| \leq U_j$.

We consider the optimal hybrid control problem, denoted by H , which has the following form:

$$H : \min_{\substack{u_{i,j} \\ |u_{i,j}| \leq U_j \\ i=1, \dots, N \\ j=1, \dots, M}} \left\{ \sum_{i=1}^N \sum_{j=1}^M \psi_j(z_{i,j}, u_{i,j}, s_{i,j}) + \sum_{i=1}^N \phi_i(x_{i,M}) \right\} \quad (6)$$

subject to both time-driven dynamics (1), (2) and event-driven dynamics (4), (5).

Let us now consider the process of job i at stage j and minimize its service cost $\psi_j(z_{i,j}, u_{i,j}, s_{i,j})$ for a given $s_{i,j} \geq S_j$ subject to (1) and (2). This is a classical optimal control problem with state specified at fixed terminal time (see [27]). Let the optimal control input be denoted as $u_{i,j}^*(s_{i,j})$ and the optimal service cost be denoted as $\theta_j(s_{i,j})$ where

$$\theta_j(s_{i,j}) = \min_{\substack{u_{i,j} \\ |u_{i,j}| \leq U_j}} \psi_j(z_{i,j}, u_{i,j}, s_{i,j}) \quad (7)$$

subject to (1) and (2). Following the same arguments as in [3], (6) can be reduced to the higher-level discrete-event optimal control problem denoted by P

$$P : \min_{\substack{s_{i,j} \\ s_{i,j} \geq S_j \\ i=1, \dots, N \\ j=1, \dots, M}} \left\{ J = \sum_{i=1}^N \sum_{j=1}^M \theta_j(s_{i,j}) + \sum_{i=1}^N \phi_i(x_{i,M}) \right\} \quad (8)$$

subject to (4) and (5). Note that the solution $\{s_{i,j}^*\}$ to the higher-level problem P will then help determine the optimal control inputs

$$u_{i,j}^*(s_{i,j}^*) = \arg \min_{\substack{u_{i,j} \\ |u_{i,j}| \leq U_j}} \psi_j(z_{i,j}, u_{i,j}, s_{i,j}^*) \quad (9)$$

subject to (1) and (2).

Lower-level solution costs (optimal service costs) θ_j and optimal control inputs $u_{i,j}^*$ can be determined via classical optimal control methods (e.g., see [28]). These methods are well established and are not the focus of this study. We will, instead, dwell on the higher-level problem P . This optimization problem is nonconvex and nondifferentiable over the service times space due to the "max" function in (4).

In this setup, the following assumptions are necessary to make the problem somewhat more tractable while preserving the originality of the problem.

Assumption 1: $\theta_j(\cdot)$, for $j = 1, \dots, M$, is continuously differentiable, monotonically decreasing, and strictly convex.

Assumption 2: $\phi_i(\cdot)$, for $i = 1, \dots, N$, is continuously differentiable, monotonically increasing, and convex.

Note that, for the costs satisfying these assumptions, longer processing times will decrease the service costs, while in-

creasing the departure times; hence, the completion-time costs. This tradeoff is what makes our problem interesting.

Since the service costs $\theta_j(\cdot)$ in (7) are the optimal costs of the lower-level problems, unlike previous work in scheduling and discrete-event system control literature, they are derived costs: We can select the form of the instantaneous cost function $F(z_{i,j}(t), u_{i,j}(t))$ in (3), however, the system dynamics (1), (2) also play a role in determining the service cost $\theta_j(\cdot)$. We assume that a faster service comes at the expense of more resources, therefore, is more expensive. For example, in a turning operation, a faster process will increase the tooling costs and will require extra supervision.

The completion-time costs $\phi_i(\cdot)$, on the other hand, can be viewed as inventory holding costs. Typically, they are linear functions (e.g., see [29]); however, under a continuously compounded interest environment, they can also be strictly convex functions of the system time. Note that a convex cost penalizing tardiness also satisfies Assumption 2. In Section IV, we give an example system with costs that satisfy both assumptions.

As shown in the next section, Assumptions 1 and 2 will suffice for the no-wait property. While proving the uniqueness of the optimal solution, however, we will need strict convexity of the completion-time cost $\phi_i(\cdot)$.

A. Equivalent Convex Optimization Problem

Let us determine an equivalent convex optimization problem for the nonconvex and nondifferentiable higher-level problem P . We obtain it by replacing the constraints $x_{i,j} = \max(x_{i,j-1}, x_{i-1,j}) + s_{i,j}$ in (4) by the constraints

$$\begin{aligned} x_{i,j} &\geq x_{i,j-1} + s_{i,j} \\ x_{i,j} &\geq x_{i-1,j} + s_{i,j} \end{aligned}$$

resulting with the surrogate convex optimization problem \bar{P} defined as

$$\bar{P} : \min_{\substack{s_{i,j} \\ s_{i,j} \geq S_j, x_{i,j} \\ i=1, \dots, N \\ j=1, \dots, M}} \left\{ \bar{J} = \sum_{i=1}^N \sum_{j=1}^M \theta_j(s_{i,j}) + \sum_{i=1}^N \phi_i(x_{i,M}) \right\}$$

subject to

$$\begin{aligned} x_{i,j} &\geq x_{i,j-1} + s_{i,j} \\ x_{i,j} &\geq x_{i-1,j} + s_{i,j} \\ x_{i,0} &= a_i, x_{0,j} = -\infty \end{aligned} \quad (10)$$

for all $i = 1, \dots, N$ and $j = 1, \dots, M$.

Since the feasible set of the surrogate optimization problem \bar{P} contains the feasible set of the higher-level optimization problem P , its optimal cost \bar{J}^* is upper bounded by the optimal cost of P denoted by J^* , i.e., $\bar{J}^* \leq J^*$.

Theorem 1: The optimal solution of \bar{P} satisfies

$$x_{i,j}^* = \max(x_{i,j-1}^*, x_{i-1,j}^*) + s_{i,j}^*$$

for all $i = 1, \dots, N$ and $j = 1, \dots, M$.

Proof: For a contradiction, assume that the optimal solution satisfies

$$\begin{aligned} x_{i,j}^* &> x_{i,j-1}^* + s_{i,j}^* \\ x_{i,j}^* &> x_{i-1,j}^* + s_{i,j}^* \end{aligned}$$

for some i, j and define $\Delta_{i,j} = x_{i,j}^* - \max(x_{i,j-1}^*, x_{i-1,j}^*) - s_{i,j}^* > 0$. If we perturb the optimal solution so that $s_{i,j}^*$ is replaced by $s_{i,j} = s_{i,j}^* + \Delta_{i,j}$, the feasibility will be preserved while the cost variation is given by

$$\Delta \bar{J} = \theta_j(s_{i,j}^* + \Delta_{i,j}) - \theta_j(s_{i,j}^*).$$

By Assumption 1, the process cost θ_j is monotonically decreasing, therefore, $\Delta \bar{J} < 0$ which contradicts the optimality assumption. Hence

$$x_{i,j}^* = \max(x_{i,j-1}^*, x_{i-1,j}^*) + s_{i,j}^*$$

for all $i = 1, \dots, N$ and $j = 1, \dots, M$. ■

Since the optimal solution of \bar{P} is in the feasible set of P , the optimal costs J^* and \bar{J}^* are equal.

III. CHARACTERISTICS OF THE OPTIMAL SOLUTION

Applying calculus of variations techniques (see [28]) on the convex optimization problem \bar{P} , we obtain a set of necessary conditions for optimality. Let us start with introducing the Lagrangian multipliers $\lambda_{i,j}$, $v_{i,j}$, $\mu_{i,j}$ to form the augmented cost

$$\begin{aligned} J_A = & \sum_{i=1}^N \phi_i(x_{i,M}) + \sum_{i=1}^N \sum_{j=1}^M \theta_j(s_{i,j}) \\ & + \sum_{i=1}^N \sum_{j=1}^M \lambda_{i,j} [x_{i,j-1} + s_{i,j} - x_{i,j}] \\ & + \sum_{i=1}^N \sum_{j=1}^M v_{i,j} [x_{i-1,j} + s_{i,j} - x_{i,j}] \\ & + \sum_{i=1}^N \sum_{j=1}^M \mu_{i,j} [S_j - s_{i,j}] \end{aligned}$$

which will be used in the following lemma.

Lemma 1: The optimal solution $\{s_{i,j}^*\}$ must satisfy the following conditions.

1) For all $i = 1, \dots, N$ and $j = 1, \dots, M$

$$x_{i,j}^* = \max(x_{i,j-1}^*, x_{i-1,j}^*) + s_{i,j}^* \quad (11)$$

$$s_{i,j}^* \geq S_j \quad (12)$$

$$\theta_j'(s_{i,j}^*) + \lambda_{i,j}^* + v_{i,j}^* - \mu_{i,j}^* = 0 \quad (13)$$

$$\mu_{i,j}^* \geq 0, \quad \mu_{i,j}^* (S_j - s_{i,j}^*) = 0 \quad (14)$$

$$\lambda_{i,j}^* \geq 0, \quad \lambda_{i,j}^* [x_{i,j-1}^* + s_{i,j}^* - x_{i,j}^*] = 0 \quad (15)$$

$$v_{i,j}^* \geq 0, \quad v_{i,j}^* [x_{i-1,j}^* + s_{i,j}^* - x_{i,j}^*] = 0 \quad (16)$$

$$\lambda_{i,j}^* + v_{i,j}^* > 0. \quad (17)$$

2) For $i = 1, \dots, N-1$ and $j = 1, \dots, M-1$

$$\lambda_{i,j}^* + v_{i,j}^* = \lambda_{i,j+1}^* + v_{i+1,j}^*. \quad (18)$$

3) For $i = 1, \dots, N-1$

$$\lambda_{i,M}^* + v_{i,M}^* = \phi_i'(x_{i,M}^*) + v_{i+1,M}^*. \quad (19)$$

4) For $j = 1, \dots, M-1$

$$\lambda_{N,j}^* + v_{N,j}^* = \lambda_{N,j+1}^*. \quad (20)$$

5)

$$\lambda_{N,M}^* + v_{N,M}^* = \phi_N'(x_{N,M}^*). \quad (21)$$

Proof: By Theorem 1, (11) is satisfied. The optimal solution should also satisfy the service constraints (12). From the augmented cost formulation, we obtain the necessary conditions for optimality

$$\left. \frac{\partial J_A}{\partial s_{i,j}} \right|_{s_{i,j}^*} = \theta_j'(s_{i,j}^*) + \lambda_{i,j}^* + v_{i,j}^* - \mu_{i,j}^* = 0 \quad (22)$$

$$\left. \frac{\partial J_A}{\partial x_{i,j}} \right|_{x_{i,j}^*} = 0 \quad (23)$$

and (14)–(16) for all $i = 1, \dots, N$ and $j = 1, \dots, M$. The condition (23) leads to (18)–(21). Finally, from (14), (22), and since by Assumption 1, $\theta_j'(s_{i,j}^*) < 0$, we have for all $i = 1, \dots, N$ and $j = 1, \dots, M$

$$\lambda_{i,j}^* + v_{i,j}^* = \mu_{i,j}^* - \theta_j'(s_{i,j}^*) > 0. \quad \blacksquare$$

This lemma provides the results needed for the following proofs.

A. Optimality of No-Wait Systems

Let us denote job i by C_i , $i = 1, \dots, N$ and give definitions of blocks and busy periods needed to decompose the sample path at any stage j .

Definition 1: A contiguous set of jobs $\{C_k, \dots, C_n\}$ is said to form a *block* at stage j if:

1) $x_{k-1,j} \leq x_{k,j-1}$ and $x_{n,j} \leq x_{n+1,j-1}$;

2) $x_{i-1,j} > x_{i,j-1}$ for $i = k+1, \dots, n$.

Definition 2: A contiguous set of jobs $\{C_k, \dots, C_n\}$ is said to form a *busy period* at stage j if:

1) $x_{k-1,j} < x_{k,j-1}$ and $x_{n,j} < x_{n+1,j-1}$;

2) $x_{i-1,j} \geq x_{i,j-1}$ for $i = k+1, \dots, n$.

Note that busy periods are formed of single or several blocks.

Before proceeding with the next lemma, we would like to make a note that its conditions are never satisfied on the optimal sample path. However, this lemma is needed for the contradiction argument for proving Theorem 2.

Using Lemma 1, we can show the following monotonicity properties of the optimal service times.

Lemma 2: (Monotonicity properties) If jobs C_{i-1} and C_i are in the same block of the j th stage on the optimal sample path then, for $i = 2, \dots, N-1$ and $j = 2, \dots, M$, the optimal service times satisfy:

i) $s_{i,j}^* \geq s_{i-1,j}^*$;

ii) $s_{i,j-1}^* \geq s_{i+1,j-1}^*$.

Proof: We show i) and ii) by contradiction arguments as follows.

i) Let us assume that jobs C_{i-1} and C_i are in the same block of the j th stage on the optimal sample path and $s_{i-1,j}^* > s_{i,j}^*$. From (12), there are two possible cases.

Case 1) $s_{i-1,j}^* > s_{i,j}^* = S_j$: From (14), $\mu_{i-1,j}^* = 0$ and $\mu_{i,j}^* \geq 0$.

Case 2) $s_{i-1,j}^* > s_{i,j}^* > S_j$: From (14), $\mu_{i-1,j}^* = \mu_{i,j}^* = 0$.

From both cases, for the assumption $s_{i-1,j}^* > s_{i,j}^*$, we get

$$\mu_{i,j}^* - \mu_{i-1,j}^* \geq 0. \quad (24)$$

Since jobs C_{i-1} and C_i are in the same block of the j th stage on the optimal sample path, we have

$$x_{i,j-1}^* < x_{i-1,j}^*. \quad (25)$$

From (11), (15), and (25), we have $\lambda_{i,j}^* = 0$.

For $j = 2, \dots, M-1$, from (18) and (15)

$$\lambda_{i-1,j}^* + v_{i-1,j}^* - v_{i,j}^* = \lambda_{i-1,j+1}^* \geq 0$$

and for $j = M$, from (19)

$$\lambda_{i-1,M}^* + v_{i-1,M}^* - v_{i,M}^* = \phi'_{i-1}(x_{i-1,M}^*) > 0.$$

Hence, for $j = 2, \dots, M$

$$\lambda_{i-1,j}^* + v_{i-1,j}^* - \lambda_{i,j}^* - v_{i,j}^* \geq 0. \quad (26)$$

It follows from (13), (24), and (26) that

$$\begin{aligned} \theta'_j(s_{i,j}^*) - \theta'_j(s_{i-1,j}^*) &= \lambda_{i-1,j}^* + v_{i-1,j}^* - \lambda_{i,j}^* - v_{i,j}^* \\ &\quad + (\mu_{i,j}^* - \mu_{i-1,j}^*) \\ &\geq 0 \end{aligned}$$

for $i = 2, \dots, N-1$ and $j = 2, \dots, M$. By Assumption 1, $\theta'_j(\cdot)$ is monotonically increasing; therefore, $s_{i,j}^* \geq s_{i-1,j}^*$, which contradicts the initial $s_{i-1,j}^* > s_{i,j}^*$ assumption.

ii) Let us assume that jobs C_{i-1} and C_i are in the same block of the j th stage on the optimal sample path and $s_{i,j-1}^* < s_{i+1,j-1}^*$. Following the same argument as above, we can write

$$\mu_{i,j-1}^* - \mu_{i+1,j-1}^* \geq 0. \quad (27)$$

It follows from (13), (15), (18), and (27), and that $\lambda_{i,j}^* = 0$

$$\begin{aligned} \theta'_{j-1}(s_{i,j-1}^*) - \theta'_{j-1}(s_{i+1,j-1}^*) &= \lambda_{i+1,j-1}^* + v_{i+1,j-1}^* - \lambda_{i,j-1}^* - v_{i,j-1}^* \\ &\quad + (\mu_{i,j-1}^* - \mu_{i+1,j-1}^*) \\ &\geq 0 \end{aligned}$$

for $i = 2, \dots, N-1$ and $j = 2, \dots, M$. By Assumption 1, $\theta'_{j-1}(\cdot)$ is monotonically increasing, therefore $s_{i,j-1}^* \geq s_{i+1,j-1}^*$, which contradicts the initial $s_{i,j-1}^* < s_{i+1,j-1}^*$ assumption. ■

The following lemma establishes that, on the optimal sample path, the last job of a busy period does not wait for service in the following stage.

Lemma 3: Consider the job sequence $\{C_k, \dots, C_n\}$ forming a busy period at stage $(j-1)$ where $j \in \{2, \dots, M\}$ on the optimal sample path. Then, the inequality

$$x_{n,j-1}^* \geq x_{n-1,j}^*$$

is satisfied.

Proof: Let us start with the case where $n = N$, i.e., jobs C_k, \dots, C_n are forming the last busy period at stage $(j-1)$ and assume that $x_{N,j-1}^* < x_{N-1,j}^*$. Then, from (11), (15), and (20)

$$\lambda_{N,j-1}^* + v_{N,j-1}^* = \lambda_{N,j}^* = 0$$

which contradicts (17) in Lemma 1. Hence, $x_{N,j-1}^* \geq x_{N-1,j}^*$ for all $j = 2, \dots, M$.

Next, let us consider the case where $n < N$. Since C_n is the last job of the busy period at stage $(j-1)$

$$x_{n,j-1}^* < x_{n+1,j-2}^*. \quad (28)$$

Let us assume that $x_{n,j-1}^* < x_{n-1,j}^*$. From (11), (15), (16), and (18)

$$\lambda_{n,j-1}^* + v_{n,j-1}^* = v_{n+1,j-1}^* + \lambda_{n,j}^* = 0$$

which also contradicts (17) in Lemma 1. Hence, $x_{n,j-1}^* \geq x_{n-1,j}^*$, if C_n ends a busy period at stage $(j-1)$. ■

The next theorem, which shows that it is never optimal to have buffering between stages, is employed to simplify the convex optimization problem \bar{P} .

Theorem 2: (No-wait property) On the optimal sample path, the departure times satisfy

$$x_{i,j-1}^* \geq x_{i-1,j}^*$$

for all $i = 1, \dots, N$ and $j = 2, \dots, M$.

Proof: (By induction) Let us pick an arbitrary stage $j \in \{2, \dots, M\}$

For $i = 1$, we have

$$x_{1,j-1}^* = a_1 + \sum_{l=1}^{j-1} s_{1,l}^* \geq x_{0,j}^* = -\infty.$$

Next, let us assume that, for some $k \in \{2, \dots, N\}$, the inequalities

$$x_{i,j-1}^* \geq x_{i-1,j}^* \quad (29)$$

hold for all $i \in \{1, \dots, k-1\}$. We need to show that $x_{k,j-1}^* \geq x_{k-1,j}^*$ also holds so let us assume for a contradiction that the inequality

$$x_{k,j-1}^* < x_{k-1,j}^* \quad (30)$$

is satisfied.

Let job C_n end the busy period at stage $(j-1)$ in which job C_k resides. Note that if $k = n$, then we already have a contradiction

by Lemma 3, so let us consider the nontrivial case where $k < n$. We will show by induction that the assumptions (29) and (30) lead to

$$x_{t,j-1}^* < x_{t-1,j}^*$$

for all $t = k+1, \dots, n$. The inequality $x_{n,j-1}^* < x_{n-1,j}^*$ will contradict the result from Lemma 3 concluding the proof.

From (30), we can conclude that C_{k-1} and C_k are in the same block of stage j . Then, from Lemma 2, we have

$$s_{k,j}^* \geq s_{k-1,j}^* \quad (31)$$

and

$$s_{k,j-1}^* \geq s_{k+1,j-1}^*. \quad (32)$$

Note that

$$x_{k,j-1}^* = \max(x_{k,j-2}^*, x_{k-1,j-1}^*) + s_{k,j-1}^*$$

and from (29)

$$\begin{aligned} x_{k-1,j}^* &= \max(x_{k-1,j-1}^*, x_{k-2,j}^*) + s_{k-1,j}^* \\ &= x_{k-1,j-1}^* + s_{k-1,j}^* \end{aligned}$$

It follows from (30) that

$$\max(x_{k-1,j-1}^*, x_{k,j-2}^*) + s_{k,j-1}^* < x_{k-1,j-1}^* + s_{k-1,j}^*$$

therefore

$$s_{k-1,j}^* > s_{k,j-1}^*. \quad (33)$$

From (31)–(33), we obtain

$$s_{k,j}^* \geq s_{k-1,j}^* > s_{k,j-1}^* \geq s_{k+1,j-1}^*. \quad (34)$$

Since we are considering the case where $k < n$, jobs C_k and C_{k+1} reside in the same busy period at stage $(j-1)$, therefore

$$x_{k+1,j-1}^* = x_{k,j-1}^* + s_{k+1,j-1}^*.$$

Since jobs C_{k-1} and C_k reside in the same block of stage j

$$x_{k,j}^* = x_{k-1,j}^* + s_{k,j}^*.$$

From (30) and (34), we have

$$x_{k+1,j-1}^* < x_{k,j}^*$$

which concludes the basis part of the induction proof.

Next, let us assume that the inequalities

$$x_{t,j-1}^* < x_{t-1,j}^* \quad (35)$$

hold for all $t = k+1, \dots, r$, where $r < n$. Since jobs C_k, \dots, C_r are in the same block at stage j , it follows from Lemma 2 and (34) that

$$s_{r,j}^* \geq s_{k-1,j}^* > s_{k,j-1}^* \geq s_{r+1,j-1}^*. \quad (36)$$

Since jobs C_r and C_{r+1} are in the same busy period at stage $(j-1)$

$$x_{r+1,j-1}^* = x_{r,j-1}^* + s_{r+1,j-1}^*.$$

Since, from (35), jobs C_{r-1} and C_r are in the same block at stage j

$$x_{r,j}^* = x_{r-1,j}^* + s_{r,j}^*.$$

Hence, from (35) and (36)

$$x_{r+1,j-1}^* < x_{r,j}^*$$

which concludes the inductive step and the induction proof.

Since the assumption (30) leads to

$$x_{n,j-1}^* < x_{n-1,j}^*$$

which contradicts the result from Lemma 3, we can state that, on the optimal sample path, the departure times satisfy

$$x_{i,j-1}^* \geq x_{i-1,j}^*$$

for all $i = 1, \dots, N$ and $j = 2, \dots, M$. ■

Note that in Theorem 2, the result from Lemma 3 is generalized to all jobs.

Applying the no-wait property to the higher-level problem P , we get

$$\hat{P}: \min_{\substack{s_{i,j} \geq s_j \\ i=1, \dots, N \\ j=1, \dots, M}} \left\{ J = \sum_{i=1}^N \sum_{j=1}^M \theta_j(s_{i,j}) + \sum_{i=1}^N \phi_i \left(x_{i,1} + \sum_{j=2}^M s_{i,j} \right) \right\}$$

subject to

$$x_{1,1} = a_1 + s_{1,1} \quad (37)$$

$$x_{i,1} = \max(a_i, x_{i-1,1}) + s_{i,1} \quad (38)$$

$$x_{i,1} \geq x_{i-1,1} + s_{i-1,2} \quad (39)$$

$$x_{i,1} + \sum_{k=2}^{j-1} s_{i,k} \geq x_{i-1,1} + \sum_{k=2}^j s_{i-1,k} \quad (40)$$

for $i = 2, \dots, N$ and $j = 3, \dots, M$.

Let us denote the cost incurred by the jobs $\{C_k, \dots, C_n\}$ as

$$J(k, n) = \sum_{i=k}^n \sum_{j=1}^M \theta_j(s_{i,j}) + \sum_{i=k}^n \phi_i \left(x_{i,1} + \sum_{j=2}^M s_{i,j} \right) \quad (41)$$

and define the convex optimization problem

$$Q(k, n): \min_{\substack{s_{i,j} \geq s_{i-1,j} \\ s_{i,j} \geq s_j \\ i=k, \dots, n \\ j=1, \dots, M}} J(k, n) \quad (42)$$

subject to

$$x_{k,1} = a_k + s_{k,1} \quad (43)$$

$$x_{i,1} \geq a_i + s_{i,1} \quad (44)$$

$$x_{i,1} \geq x_{i-1,1} + s_{i,1} \quad (45)$$

$$x_{i,1} \geq x_{i-1,1} + s_{i-1,2} \quad (46)$$

$$x_{i,1} + \sum_{k=2}^{j-1} s_{i,k} \geq x_{i-1,1} + \sum_{k=2}^j s_{i-1,k} \quad (47)$$

for $i = k+1, \dots, n$ and $j = 3, \dots, M$. Following the same reasoning as in Theorem 1, we can show that the convex optimization problem $Q(1, N)$ yields the optimal solution for \hat{P} . This simplified convex optimization problem $Q(1, N)$ has $(M+1)N$ decision variables, one equality and $(N-1)(M+1)$ inequality constraints (excluding the boundary value constraints). In the next subsection, we give the definition of an independent period structure and present a decoupling property which allows for the decomposition of the simplified problem $Q(1, N)$ into smaller convex optimization problems, one for each independent period.

B. Forward Decomposition Algorithms

The decomposition algorithms that follow will require $Q(k, n)$ to have a unique optimal solution. A sufficient condition for this to be satisfied is given in the next assumption replacing Assumption 2.

Assumption 3: $\phi_i(\cdot)$, for $i = 1, \dots, N$, is continuously differentiable, monotonically increasing, and strictly convex.

Lemma 4: The convex optimization problem $Q(k, n)$ has a unique solution.

Proof: The feasible set defined by the constraints (43)–(47) is convex. Recalling that

$$x_{i,M} = x_{i,1} + \sum_{j=2}^M s_{i,j} \quad (48)$$

a sufficient condition for $Q(k, n)$ to have a unique optimal solution is that the cost

$$J = \sum_{i=k}^n \sum_{j=1}^M \theta_j(s_{i,j}) + \sum_{i=k}^n \phi_i(x_{i,M})$$

is strictly convex.

Let us define two distinct feasible solutions y^1 and y^2 such that

$$y^l = [x_{k,1}^l, \dots, x_{n,1}^l, s_{k,1}^l, \dots, s_{n,M}^l]$$

for $l = 1, 2$. Due to convexity of $\theta_j(\cdot)$ and $\phi_i(\cdot)$ (as in Assumptions 1 and 3), we can write for $0 < \delta < 1$

$$\delta J(y^1) + (1 - \delta)J(y^2) \geq J(\delta y^1 + (1 - \delta)y^2).$$

For strict inequality (and strict convexity), it suffices to show that for some i and j , $s_{i,j}^1 \neq s_{i,j}^2$ or $x_{i,M}^1 \neq x_{i,M}^2$.

Since y^1 and y^2 are distinct, they should differ in at least one component. If $s_{i,j}^1 \neq s_{i,j}^2$ for some i and j , since $\theta_j(\cdot)$ is strictly convex by Assumption 1, strict inequality is obtained. If, on the other hand, $s_{i,j}^1 = s_{i,j}^2$ for all i and j , then for some i , we should have $x_{i,1}^1 \neq x_{i,1}^2$. From (48), it follows that $x_{i,M}^1 \neq x_{i,M}^2$.

Since $\phi_i(\cdot)$ is strictly convex by Assumption 3, strict inequality is obtained. Hence, for distinct feasible solutions y^1 and y^2

$$\delta J(y^1) + (1 - \delta)J(y^2) > J(\delta y^1 + (1 - \delta)y^2)$$

i.e., $J(\cdot)$ is strictly convex and, therefore, $Q(k, n)$ has a unique optimal solution. ■

In what follows, we determine the unique optimal solution for $Q(1, N)$. Independent period structures, defined next, simplify our task.

Definition 3: A contiguous set of jobs $\{C_k, \dots, C_n\}$ is said to form an *independent period* for the system if:

- 1) $x_{k-1,1} < a_k$ and $x_{k-1,j} < a_k + \sum_{l=1}^{j-1} S_l$ for all $j = 2, \dots, M$;
- 2) $x_{n,1} < a_{n+1}$ and $x_{n,j} < a_{n+1} + \sum_{l=1}^{j-1} S_l$ for all $j = 2, \dots, M$;
- 3) for all $i \in \{k, \dots, n-1\}$, $x_{i,1} \geq a_{i+1}$ or $x_{i,j} \geq a_{i+1} + \sum_{l=1}^{j-1} S_l$ for some $j = 2, \dots, M$.

Definition 4: An *independent period structure* for the system is a partition of jobs $\{C_1, \dots, C_N\}$ into independent periods.

The next lemma presents the decoupling property between independent periods.

Lemma 5: Consider a contiguous job sequence $\{C_k, \dots, C_n\}$ forming an independent period on the optimal sample path. The optimal service times for these jobs do not depend on the arrival times $\{a_1, \dots, a_{k-1}, a_{n+1}, \dots, a_N\}$ of the other jobs.

Proof: From (12) and the independent period definition, we have for all $j = 1, \dots, M-1$

$$x_{k-1,j+1}^* < a_k + \sum_{l=1}^j S_l \leq x_{k,j}^* \quad (49)$$

$$x_{k-1,1}^* < a_k \quad (50)$$

for $k > 1$ and

$$x_{n,j+1}^* < a_{n+1} + \sum_{l=1}^j S_l \leq x_{n+1,j}^* \quad (51)$$

$$x_{n,1}^* < a_{n+1} \quad (52)$$

for $n < N$.

From (11), (16), (18), (51), and (52), for $j = 1, \dots, M-1$

$$\lambda_{n,j}^* + v_{n,j}^* = v_{n+1,j}^* + \lambda_{n,j+1}^* = \lambda_{n,j+1}^*$$

and

$$\lambda_{n,M}^* + v_{n,M}^* = \phi'_n(x_{n,M}^*) + v_{n+1,M}^* = \phi'_n(x_{n,M}^*).$$

Hence, there is no dependence of $\lambda_{n,j}^*$ for $j = 1, \dots, M$ to jobs $\{C_{n+1}, \dots, C_N\}$, i.e., the co-state equations do not propagate information in the backward direction between independent periods. Note that if $n = N$ then there is no need for checking the backward information propagation.

Similarly, let us employ inequalities (49) and (50) in (11) to observe that

$$\begin{aligned} x_{k,j}^* &= \max(x_{k,j-1}^*, x_{k-1,j}^*) + s_{k,j}^* \\ &= x_{k,j-1}^* + s_{k,j}^* \end{aligned}$$

Hence, there is no dependence of $x_{k,j}^*$ for $j = 1, \dots, M$ to jobs $\{C_1, \dots, C_{k-1}\}$, i.e., the state equations do not propagate information in the forward direction between independent periods. Note that if $k = 1$ then there is no need for checking the forward information propagation.

Hence, independent periods are decoupled from each other. ■

Let us assume that the optimal independent period structure is given. Let B be the number of independent periods in this structure, and let $C_{k(b)}$ and $C_{n(b)}$ denote the first and the last jobs of the b th independent period where $b = 1, \dots, B$. We can rewrite \hat{P} as

$$\hat{P} : \min_{\substack{s_{i,j} \\ s_{i,j} \geq S_j \\ i=1,\dots,N \\ j=1,\dots,M}} \left\{ J = \sum_{b=1}^B J(k(b), n(b)) \right\}$$

subject to (37)–(40). By the independent period definition, the constraints (38)–(40) for $i = k(b)$ can be reduced to $x_{k(b),1} = a_{k(b)} + s_{k(b),1}$ which is satisfied by the optimal solution. Along with the decoupling property shown in Lemma 5, this allows for the decomposition of \hat{P} into B smaller optimization problems $\hat{P}(k(b), n(b))$ defined as

$$\hat{P}(k(b), n(b)) : \min_{\substack{s_{i,j} \\ s_{i,j} \geq S_j \\ i=k(b),\dots,n(b) \\ j=1,\dots,M}} J(k(b), n(b))$$

subject to

$$\begin{aligned} x_{k(b),1} &= a_{k(b)} + s_{k(b),1} \\ x_{i,1} &= \max(a_i, x_{i-1,1}) + s_{i,1} \\ x_{i,1} &\geq x_{i-1,1} + s_{i-1,2} \\ x_{i,1} + \sum_{k=2}^{j-1} s_{i,k} &\geq x_{i-1,1} + \sum_{k=2}^j s_{i-1,k} \end{aligned}$$

for $i = k(b) + 1, \dots, n(b)$ and $j = 3, \dots, M$. Note that, following similar reasoning as in Theorem 1, $Q(k(b), n(b))$ can be shown to have the same optimal solution as of $\hat{P}(k(b), n(b))$.

We denote the optimal solution of $Q(k, n)$ as $s_{i,j}^*(k, n)$ and $x_{i,1}^*(k, n)$, and the corresponding departure times as $x_{i,j}^*(k, n)$ for $i = k, \dots, n$ and $j = 1, \dots, M$. The following corollary follows from Lemma 5 and relates the optimal solution of $Q(k, n)$ to the optimal solution of the higher-level problem.

Corollary 1: If the job sequence $\{C_k, \dots, C_n\}$ forms an independent period on the optimal sample path, then the optimal solution to $Q(k, n)$ satisfies

$$\begin{aligned} s_{i,j}^*(k, n) &= s_{i,j}^* \\ x_{i,j}^*(k, n) &= x_{i,j}^* \end{aligned}$$

for $i = k, \dots, n$ and $j = 1, \dots, M$.

This corollary forms the basis for our decomposition algorithms.

So far, we have shown that if the optimal independent period structure can be identified, $Q(1, N)$ can be decomposed into a set of smaller problems. The following lemma provides necessary and sufficient conditions for identifying the independent periods on the optimal sample path based on the solution of $Q(k, n)$.

Lemma 6: Let C_k initiate an independent period on the optimal sample path. The job C_n ($k \leq n$) ends the independent period if and only if the following conditions are satisfied:

- 1) $x_{n,1}^*(k, n) < a_{n+1}$ and $x_{n,j}^*(k, n) < a_{n+1} + \sum_{l=1}^{j-1} S_l$ for all $j = 2, \dots, M$;
- 2) for all $i \in \{k, \dots, n-1\}$, $x_{i,1}^*(k, n) \geq a_{i+1}$ or $x_{i,j}^*(k, n) \geq a_{i+1} + \sum_{l=1}^{j-1} S_l$ for some $j = 2, \dots, M$.

Proof: (Necessity) Since jobs $\{C_k, \dots, C_n\}$ form an independent period on the optimal sample path, and since $x_{i,j}^*(k, n) = x_{i,j}^*$ by Corollary 1, the result follows from the independent period definition.

(Sufficiency) Assume that even though both conditions are satisfied, C_n does not end the independent period, i.e., for some $n' > n$, jobs $\{C_k, \dots, C_{n'}\}$ form an independent period on the optimal sample path.

Let us define a solution for $Q(k, n')$ as

$$\bar{s}_{i,j} = \begin{cases} s_{i,j}^*(k, n), & i \in \{k, \dots, n\} \text{ and } j \in \{1, \dots, M\} \\ s_{i,j}^*(k, n') + \alpha, & i = n+1, j = 1 \\ s_{i,j}^*(k, n'), & \text{otherwise} \end{cases} \quad (53)$$

and

$$\bar{x}_{i,1} = \begin{cases} x_{i,1}^*(k, n), & i \in \{k, \dots, n\} \\ x_{i,1}^*(k, n'), & i \in \{n+1, \dots, n'\} \end{cases} \quad (54)$$

where $\alpha \geq 0$ is defined as

$$\alpha = \max(x_{n,1}^*(k, n') - a_{n+1}, 0).$$

By the independent period definition, since C_n is not the last job of the independent period, $x_{n,1}^*(k, n') \geq a_{n+1}$ or $x_{n,j}^*(k, n') \geq a_{n+1} + \sum_{l=1}^{j-1} S_l$ for some $j = 2, \dots, M$. It follows from the first condition that

$$s_{i,j}^*(k, n) \neq s_{i,j}^*(k, n') \quad (55)$$

or

$$x_{i,1}^*(k, n) \neq x_{i,1}^*(k, n') \quad (56)$$

for some $i = k, \dots, n$ and $j = 1, \dots, M$. Therefore, from Lemma 4, the solution given in (53) and (54) is not optimal for $Q(k, n')$.

Let us check the feasibility of this nonoptimal solution: Since $\bar{s}_{i,j}$ and $\bar{x}_{i,1}$ for $i = k, \dots, n$ and $j = 1, \dots, M$ are the solutions for the problem $Q(k, n)$, they satisfy the constraints (43)–(47) for $i = k+1, \dots, n$ and $j = 3, \dots, M$. For $i = n+1$, we have

$$\begin{aligned} x_{n+1,1}^*(k, n') &= \max(x_{n,1}^*(k, n'), a_{n+1}) + s_{n+1,1}^*(k, n') \\ &= a_{n+1} + \alpha + s_{n+1,1}^*(k, n') \\ &= a_{n+1} + \bar{s}_{n+1,1} \\ &= \max(a_{n+1}, \bar{x}_{n,1}) + \bar{s}_{n+1,1} \end{aligned}$$

satisfying the constraints (44) and (45). It follows from the first condition and from $\bar{s}_{n+1,j} \geq S_j$ for all $j = 1, \dots, M$ that the constraints (46) and (47) are also satisfied for $i = n+1$. Finally, since $\bar{s}_{i,j}$ and $\bar{x}_{i,1}$ for $i = n+2, \dots, n'$ and $j = 1, \dots, M$ are the solutions for $Q(k, n')$, the constraints (44)–(47) are satisfied for $i = n+2, \dots, n'$ and $j = 3, \dots, M$. Hence, the nonoptimal solution is feasible for the $Q(k, n')$ problem.

Let us recall from Corollary 1 that

$$\begin{aligned} s_{i,j}^* &= s_{i,j}^*(k, n') \\ x_{i,1}^* &= x_{i,1}^*(k, n') \end{aligned}$$

for $i = k, \dots, n'$ and $j = 1, \dots, M$, and consider the problem $Q(k, n')$. Its optimal cost can be written as

$$J^*(k, n') = J_A^*(k, n') + J_B^*(k, n') \quad (57)$$

where

$$J_A^*(k, n') = \sum_{i=k}^n \sum_{j=1}^M \theta_j (s_{i,j}^*) + \sum_{i=k}^n \phi_i \left(x_{i,1}^* + \sum_{j=2}^M s_{i,j}^* \right)$$

and

$$J_B^*(k, n') = \sum_{i=n+1}^{n'} \sum_{j=1}^M \theta_j (s_{i,j}^*) + \sum_{i=n+1}^{n'} \phi_i \left(x_{i,1}^* + \sum_{j=2}^M s_{i,j}^* \right).$$

The cost \bar{J} due to applying the nonoptimal solution in (53) and (54) can be written as

$$\bar{J}(k, n') = J^*(k, n) + \bar{J}_C \quad (58)$$

where

$$\bar{J}_C = J_B^*(k, n') + \theta_1 (s_{n+1,1}^*(k, n') + \alpha) - \theta_1 (s_{n+1,1}^*(k, n')).$$

Since the optimal solution of $Q(k, n)$ is unique by Lemma 4, from (55) and (56)

$$J_A^*(k, n') > J^*(k, n). \quad (59)$$

Moreover, due to Assumption 1

$$J_B^*(k, n') \geq \bar{J}_C \quad (60)$$

which also accounts for the case that $\alpha = 0$. From (57)–(60)

$$J^*(k, n') > \bar{J}$$

in other words, the cost of the nonoptimal solution is lower than the optimal cost, which is a contradiction. Hence, the result follows. ■

Lemma 6 allows us to identify independent periods on the optimal sample path. If C_k is the first job of an independent period, we can identify this independent period by sequentially solving $Q(k, n)$ and checking for all jobs $\{C_k, \dots, C_n\}$ to see if both conditions in Lemma 6 are satisfied.

The next lemma indicates that checking only for job C_n to see if the first condition in Lemma 6 is satisfied suffices to identify the independent period.

Lemma 7: Let C_k initiate an independent period on the optimal sample path. For all $i \geq k$, if $x_{i,1}^*(k, i) \geq a_{i+1}$ or $x_{i,j}^*(k, i) \geq a_{i+1} + \sum_{l=1}^{j-1} S_l$ for some $j = 2, \dots, M$, then for all $n \geq i$, $x_{i,1}^*(k, n) \geq a_{i+1}$ or $x_{i,j}^*(k, n) \geq a_{i+1} + \sum_{l=1}^{j-1} S_l$ for some $j = 2, \dots, M$.

Proof: (By contradiction) Let us assume that there exists $n > i$ such that $x_{i,1}^*(k, n) < a_{i+1}$ and $x_{i,j}^*(k, n) < a_{i+1} +$

$\sum_{l=1}^{j-1} S_l$ for all $j = 2, \dots, M$. In that case job C_i ends an independent period (not necessarily the one that was started by job C_k). From the decoupling property in Lemma 5 and Corollary 1, for all stages j , we should have

$$x_{i,j}^*(k, i) = x_{i,j}^*(k, n).$$

However, since $x_{i,1}^*(k, i) \geq a_{i+1}$ or $x_{i,j}^*(k, i) \geq a_{i+1} + \sum_{l=1}^{j-1} S_l$ for some $j = 2, \dots, M$, a contradiction is observed. Hence, the result follows. ■

Lemma 7 asserts that an independent period formed by jobs $\{C_k, \dots, C_n\}$ is identified as soon as the problem $Q(k, n)$ is solved. This result is formalized in the following theorem.

Theorem 3: Jobs $\{C_k, \dots, C_n\}$ form an independent period on the optimal sample path if and only if the following conditions are satisfied:

- 1) $x_{k-1,1}^* < a_k$ and $x_{k-1,j}^* < a_k + \sum_{l=1}^{j-1} S_l$ for all $j = 2, \dots, M$;
- 2) for all $i \in \{k, \dots, n-1\}$, $x_{i,1}^*(k, i) \geq a_{i+1}$ or $x_{i,j}^*(k, i) \geq a_{i+1} + \sum_{l=1}^{j-1} S_l$ for some $j = 2, \dots, M$;
- 3) $x_{n,1}^*(k, n) < a_{n+1}$ and $x_{n,j}^*(k, n) < a_{n+1} + \sum_{l=1}^{j-1} S_l$ for all $j = 2, \dots, M$.

Proof: It follows from Lemma 6, Lemma 7, and the independent period definition. ■

Our first forward decomposition algorithm is based on Theorem 3. While identifying the optimal independent period structure, this algorithm also determines the optimal solution.

Algorithm 1:

Step 1: (initialization) $k = 1, n = 1, a_{N+1} = \infty$

while $n \leq N$ do

Step 2: solve subproblem $Q(k, n)$

Step 3: (identify independent periods)

if $x_{n,1}^*(k, n) < a_{n+1}$ and $x_{n,j}^*(k, n) < a_{n+1} + \sum_{l=1}^{j-1} S_l$ for all $j = 2$ to M then

$$s_{i,j}^* = s_{i,j}^*(k, n) \text{ for } i = k, \dots, n \text{ and } j = 1, \dots, M$$

$$k = n + 1$$

end if

Step 4: (increment index n)

$$n = n + 1$$

A slightly modified version of this algorithm can be shown to exist. The following theorem forms the basis for this modification.

Theorem 4: Let jobs $\{C_k, \dots, C_n\}$ form an independent period on the optimal sample path. If for some $r \in \{k, \dots, n-1\}$, $x_{r,1}^*(k, r) \leq a_{r+1}$ and $x_{r,j}^*(k, r) \leq a_{r+1} + \sum_{l=1}^{j-1} S_l$ for all $j = 2, \dots, M$, then the optimal solution for $Q(k, n)$ is given by

$$s_{i,j}^* = \begin{cases} s_{i,j}^*(k, r) & i \in \{k, \dots, r\} \\ s_{i,j}^*(r+1, n) & i \in \{r+1, \dots, n\} \end{cases}$$

and

$$x_{i,1}^* = \begin{cases} x_{i,1}^*(k, r) & i \in \{k, \dots, r\} \\ x_{i,1}^*(r+1, n) & i \in \{r+1, \dots, n\} \end{cases}$$

for $j = 1, \dots, M$.

Proof: The given solution is formed by the optimal solutions of $Q(k, r)$ and $Q(r+1, n)$. We need to show its feasibility for the problem $Q(k, n)$.

The optimal solutions for $Q(k, r)$, $s_{i,j}^*(k, r)$ and $x_{i,1}^*(k, r)$, satisfy (43)–(47) for $i = k+1, \dots, r$ and $j = 3, \dots, M$. Since $x_{r,1}^*(k, r) \leq a_{r+1}$ and the solution of $Q(r+1, n)$ satisfies $x_{r+1,1}^*(r+1, n) = a_{r+1} + s_{r+1,1}^*(r+1, n)$, the constraints (44) and (45) are satisfied for $i = r+1$. Since $x_{r,j}^*(k, r) \leq a_{r+1} + \sum_{l=1}^{j-1} S_l$ and $s_{i,j}^*(r+1, n) \geq S_j$ for all $j = 2, \dots, M$, the constraints (46) and (47) are also satisfied for $i = r+1$. The optimal solutions for $Q(r+1, n)$, $s_{i,j}^*(r+1, n)$ and $x_{i,1}^*(r+1, n)$, satisfy (43)–(47) for $i = r+2, \dots, n$ and $j = 3, \dots, M$. Hence, the solution in the theorem statement is feasible and optimal. ■

Our second forward decomposition algorithm is based on Theorem 4.

Algorithm 2:

Step 1: (initialization) $k = 1, n = 1, a_{N+1} = \infty$

while $n \leq N$ do

Step 2: solve subproblem $Q(k, n)$

Step 3:

if $x_{n,1}^*(k, n) \leq a_{n+1}$ and $x_{n,j}^*(k, n) \leq a_{n+1} + \sum_{l=1}^{j-1} S_l$ for all $j = 2$ to M then

$$s_{i,j}^* = s_{i,j}^*(k, n) \text{ for } i = k, \dots, n \text{ and } j = 1, \dots, M$$

$$k = n + 1$$

end if

Step 4: (increment index n)

$$n = n + 1$$

Note that these decomposition algorithms require only N iterations. However, these iterations are not identical in complexity and depend on the arrival sequence along with the cost parameters. The best case for these algorithms would be an optimal sample path where each job forms an independent period of its own. In this case, $Q(i, i)$ for all $i = 1, \dots, N$ are solved. The worst case for these algorithms, on the other hand, would be an optimal sample path where all jobs reside in the same independent period and no decomposition is observed. In the worst case, we solve $Q(1, i)$ for all $i = 1, \dots, N$. If the number of independent periods expected are low, e.g., for the bulk arrivals case we have only one independent period, we may choose to solve $Q(1, N)$ directly.

We would like to point out that Algorithms 1 and 2 are pretty much the same algorithm, except for the inequality in step 3 which is strict in Algorithm 1 and not strict in Algorithm 2. Due to numerical errors in $Q(k, n)$ solutions, we expect both algorithms to work with the same performance.

IV. NUMERICAL EXAMPLE

Let us consider an M -stage serial manufacturing line processing an identical set of N jobs. Let job C_i at stage j have the time-driven dynamics described by

$$\dot{z}_{i,j}(t) = f_j(z_{i,j}(t), u_{i,j}(t)) = b_j u_{i,j}(t)$$

with given initial ζ_j^0 and final (desired) ζ_j^d states. In order to have time-invariant processes at each stage, b_j values are assumed to be constants for $j = 1, \dots, M$. Each stage j is assumed to have a bounded control input, i.e., $|u_{i,j}| \leq U_j$, and to process one job at a time on a first-come first-served nonpreemptive basis. Let ψ_j , the service cost at stage j , be given by the quadratic cost functional defined as

$$\psi_j(z_{i,j}(t), u_{i,j}(t), s_{i,j}) = \int_0^{s_{i,j}} \frac{1}{2} r_j u_{i,j}^2(t) dt$$

for given r_j . Applying classical optimal control methods, the optimal cost of the lower-level problem for a given service time $s_{i,j}$ can be found as

$$\theta_j(s_{i,j}) = \frac{\beta_j}{s_{i,j}}$$

where

$$\beta_j = \left(\frac{1}{2} r_j \frac{(\zeta_j^d - \zeta_j^0)^2}{b_j^2} \right). \quad (61)$$

The optimal control input turns out to be constant

$$u_{i,j}^*(s_{i,j}) = \left(\frac{\zeta_j^d - \zeta_j^0}{b_j} \right) \frac{1}{s_{i,j}} \quad (62)$$

for service times $s_{i,j} \geq S_j$ where

$$S_j = \frac{|\zeta_j^d - \zeta_j^0|}{b_j U_j}. \quad (63)$$

The departure time cost for job C_i , on the other hand, is given by a cost on its system time defined as

$$\phi_i(x_{i,M}) = \alpha(x_{i,M} - a_i)^2 \quad (64)$$

for some constant α .

Note that the lower-level optimal cost $\theta_j(s_{i,j}) = \beta_j/s_{i,j}$ is continuously differentiable, monotonically decreasing, and strictly convex satisfying Assumption 1. Similarly the departure time cost $\phi_i(x_{i,M}) = \alpha(x_{i,M} - a_i)^2$ is continuously differentiable, monotonically increasing and strictly convex for $x_{i,M} \geq a_i + \sum_{j=1}^M S_j$, hence, satisfies Assumption 3. Therefore, we expect to see a unique solution as shown in Lemma 4.

TABLE I
OPTIMAL SERVICE TIMES

Service Times	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8	Job 9	Job 10
Stage 1	0.495	0.348	0.555	0.288	0.419	0.590	0.459	0.546	0.495	0.495
Stage 2	0.350	0.316	0.392	0.280	0.421	0.417	0.325	0.386	0.350	0.350
Stage 3	0.700	0.631	0.657	0.560	0.586	0.666	0.649	0.689	0.700	0.700
Stage 4	0.495	0.524	0.465	0.546	0.488	0.471	0.508	0.487	0.495	0.495

TABLE II
OPTIMAL DEPARTURE TIMES

Departure Times	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8	Job 9	Job 10
Arrivals	0.000	2.300	2.400	4.900	5.000	5.500	9.000	9.500	11.000	13.000
Stage 1	0.495	2.648	3.202	5.188	5.607	6.197	9.459	10.046	11.495	13.495
Stage 2	0.845	2.963	3.595	5.468	6.029	6.614	9.784	10.433	11.845	13.845
Stage 3	1.545	3.595	4.252	6.029	6.614	7.281	10.433	11.121	12.545	14.545
Stage 4	2.040	4.119	4.716	6.575	7.102	7.752	10.940	11.608	13.040	15.040

The convex optimization problem we had before the results of Section III can be written as

$$\bar{P}: \min_{\substack{s_{i,j} \geq S_j, x_{i,j} \\ i=1,\dots,N \\ j=1,\dots,M}} \left\{ \bar{J} = \sum_{i=1}^N \sum_{j=1}^M \theta_j(s_{i,j}) + \sum_{i=1}^N \phi_i(x_{i,M}) \right\}$$

subject to

$$\begin{aligned} x_{1,1} &= a_1 + s_{1,1} \\ x_{1,j} &= a_1 + \sum_{k=1}^j s_{1,k} \\ x_{i,1} &\geq a_i + s_{i,1} \\ x_{i,1} &\geq x_{i-1,1} + s_{i,1} \\ x_{i,j} &\geq x_{i,j-1} + s_{i,j} \\ x_{i,j} &\geq x_{i-1,j} + s_{i,j} \end{aligned}$$

for $i = 2, \dots, N$ and $j = 2, \dots, M$. This problem has $2MN$ decision variables, M equality constraints, and $2M(N-1)$ inequality constraints (excluding the boundary value constraints on service times). Its solution time will be compared against the solution times of $Q(1, N)$ with and without decomposition. The Matlab function *fmincon*, provided in the Optimization Toolbox, solves these convex optimization problems to optimality starting with the same initial solutions. This function implements a sequential quadratic programming (SQP) algorithm (see [30] and [31]). The computing environment has an Intel Pentium M 745 processor [1.80 GHz, 2MB L2 cache, 400 MHz FSB] with 512 MB of RAM.

Example 1: For the system described above, let the number of stages be $M = 4$ and the number of identical jobs be $N = 10$. These jobs arrive at prespecified arrival times $a = [0.0, 2.3, 2.4, 4.9, 5.0, 5.5, 9.0, 9.5, 11.0, 13.0]$. The α parameter in (64) is given to be 10, and the parameters r_j , b_j , ζ_j^d , ζ_j^0 , and U_j for $j = 1, 2, 3, 4$ are given such that from (61) and (63) we get:

- $\beta = [10, 5, 20, 10]$;

- $S = [0.20, 0.20, 0.30, 0.35]$.

All service times are initially set to be $s_{i,j} = S_j$ for all $i = 1, \dots, 10$ and $j = 1, \dots, 4$.

The convex optimization problem \bar{P} is solved in 38.67 s to yield the optimal service times and the optimal departure times, given in Tables I and II, respectively, resulting with an optimal cost of 1290.15. Once the optimal service times are determined, from (62), we can find the optimal control inputs $u_{i,j}^*(s_{i,j}^*)$ for all jobs i and stages j .

In \bar{P} , we have $2MN = 80$ decision variables, $M = 4$ equality constraints, and $2M(N-1) = 72$ inequality constraints (excluding the boundary value constraints). Note that one may observe the no-wait property, $x_{i,j}^* \geq x_{i-1,j+1}^*$ in the optimal solution.

The convex optimization problem $Q(1, 10)$, implementing the simplification due to the no-wait property, can be solved in 18.03 s. It has fewer decision variables and constraints: only $(M+1)N = 50$ decision variables, one equality and $(N-1)(M+1) = 45$ inequality constraints (excluding the boundary value constraints); hence, such an improvement in speed is observed.

In order to illustrate the speed improvement due to decomposition, we solve the same problem using Algorithm 2. (Algorithm 1 has the same performance for this problem). It decomposes the problem into a set of subproblems for jobs $\{C_1\}$, $\{C_2, C_3\}$, $\{C_4, C_5, C_6\}$, $\{C_7, C_8\}$, $\{C_9\}$, and $\{C_{10}\}$, and yields the optimal solution in only 1.84 s.

Note that the additional improvement observed in the example due to the decomposition algorithm is not typical. For congested systems, the decomposition algorithms may take longer to yield the optimal solution. For uncongested systems where the number of jobs and stages are large, however, we expect orders of magnitude improvements due to decomposition. The speed improvement due to employing the no-wait property, on the other hand, is always observed regardless of the system load as it decreases the number of variables and constraints considerably.

V. CONCLUSION

This paper considered a deterministic flow shop system with some physical dynamics at each stage. For prespecified arrival times, the hybrid control problem was reduced to a discrete-event control problem where the control variables are the deterministic service times that are lower-bounded. The original nonsmooth optimization problem was transformed into a convex optimization problem over a larger set with linear constraints. We derived some characteristics of the optimal solution and showed that no waiting between stages is observed on the optimal sample path. The no-wait property eliminates N variables and N constraints from the convex optimization problem at each stage it is observed. Two “forward-in-time” decomposition algorithms were also developed to decompose the simplified convex optimization problem into smaller convex optimization problems. As shown by the numerical example, the simplification on the convex optimization problem and the decomposition improved the solution times considerably.

REFERENCES

- [1] C. G. Cassandras and S. LaFortune, *Introduction to Discrete Event Systems*. Norwell, MA: Kluwer, 1999.
- [2] K. Gokbayrak and C. G. Cassandras, “Hybrid controllers for hierarchically decomposed systems,” in *Proc. Hybrid System Control Conf.*, 2000, pp. 117–129.
- [3] K. Gokbayrak and C. G. Cassandras, “A hierarchical decomposition method for optimal control of hybrid systems,” in *Proc. 39th IEEE Conf. Decision and Control*, 2000, pp. 1816–1821.
- [4] C. G. Cassandras and K. Gokbayrak, “Optimal control for discrete event and hybrid systems,” *Model., Control, Optim. Complex Syst.*, pp. 285–304, 2002.
- [5] R. G. Vickson, “Choosing the job sequence and processing times to minimize processing plus flow cost on a single machine,” *Oper. Res.*, vol. 28, no. 5, pp. 1155–1167, 1980.
- [6] Z. L. Chen, Q. Lu, and G. Tang, “Single machine scheduling with discretely controllable processing times,” *Oper. Res. Lett.*, vol. 21, pp. 69–76, 1997.
- [7] B. Alidaee and A. Ahmadian, “Two parallel machine sequencing problems involving controllable job processing times,” *Eur. J. Oper. Res.*, vol. 70, pp. 335–341, 1993.
- [8] T. C. E. Cheng, Z. L. Chen, and L. Chung-Lun, “Parallel-machine scheduling with controllable processing times,” *IIE Trans.*, vol. 28, no. 2, pp. 177–180, 1996.
- [9] E. Nowicki and S. Zdrzalka, “A survey of results for sequencing problems with controllable processing times,” *Discrete Appl. Math.*, vol. 26, pp. 271–287, 1990.
- [10] H. Hoogeveen, “Multicriteria scheduling,” *Eur. J. Oper. Res.*, vol. 167, pp. 592–623, 2005.
- [11] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
- [12] E. Nowicki, “An approximation algorithm for the m -machine permutation flow shop scheduling problem with controllable processing times,” *Eur. J. Oper. Res.*, vol. 70, pp. 342–349, 1993.
- [13] S. Karabati and P. Kouvelis, “Flow-line scheduling problem with controllable processing times,” *IIE Trans.*, vol. 29, no. 1, pp. 1–14, 1997.
- [14] D. L. Pepyne and C. G. Cassandras, “Modeling, analysis, and optimal control of a class of hybrid systems,” *J. Discrete Event Dyn. Syst.: Theory Appl.*, vol. 8, no. 2, pp. 175–201, 1998.
- [15] D. L. Pepyne and C. G. Cassandras, “Optimal control of hybrid systems in manufacturing,” *Proc. IEEE*, vol. 88, no. 7, pp. 1108–1123, Jul. 2000.
- [16] C. G. Cassandras, D. L. Pepyne, and Y. Wardi, “Optimal control of a class of hybrid systems,” *IEEE Trans. Autom. Control*, vol. 46, no. 3, pp. 398–415, Mar. 2001.
- [17] Y. Wardi, C. G. Cassandras, and D. L. Pepyne, “A backward algorithm for computing optimal controls for single-stage hybrid manufacturing systems,” *Int. J. Prod. Res.*, vol. 39-2, pp. 369–393, 2001.
- [18] Y. C. Cho, C. G. Cassandras, and D. L. Pepyne, “Forward decomposition algorithms for optimal control of a class of hybrid systems,” *Int. J. Robust Nonlin. Control*, vol. 11, pp. 497–513, 2001.
- [19] P. Zhang and C. G. Cassandras, “An improved forward algorithm for optimal control of a class of hybrid systems,” *IEEE Trans. Autom. Control*, vol. 47, no. 10, pp. 1735–1739, Oct. 2002.
- [20] J. Moon and Y. Wardi, “Optimal control of processing times in single-stage discrete event dynamic systems with blocking,” *IEEE Trans. Autom. Control*, vol. 50, no. 6, pp. 880–884, Jun. 2005.
- [21] M. Gazarik and Y. Wardi, “Optimal release times in a single server: An optimal control perspective,” *IEEE Trans. Autom. Control*, vol. 43, no. 7, pp. 998–1002, Jul. 1998.
- [22] J. Moon and Y. Wardi, “Optimal release times in a single-stage manufacturing system with blocking: Optimal control perspective,” *J. Optim. Theory Appl.*, vol. 125, no. 3, pp. 653–672, 2005.
- [23] A. D. Febraro, R. Minciardi, and S. Saccone, “Optimal control laws for lot-sizing and timing of jobs on a single production facility,” *IEEE Trans. Autom. Control*, vol. 47, no. 10, pp. 1613–1623, Oct. 2002.
- [24] C. G. Cassandras, Q. Liu, D. L. Pepyne, and K. Gokbayrak, “Optimal control of a two-stage hybrid manufacturing system model,” in *Proc. 38th IEEE Conf. Decision and Control*, 1999, pp. 450–455.
- [25] K. Gokbayrak and C. G. Cassandras, “Constrained optimal control for multistage hybrid manufacturing system models,” presented at the 8th IEEE Mediterranean Conf. New Directions in Control and Automation, 2000.
- [26] K. Gokbayrak and O. Selvi, “Optimal hybrid control of a two-stage manufacturing system,” presented at the ACC, 2006.
- [27] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*. Bristol, PA: Hemisphere, 1975.
- [28] D. E. Kirk, *Optimal Control Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1970.
- [29] E. A. Silver, D. F. Pyke, and R. Peterson, *Inventory Management and Production Planning and Scheduling*. New York: Wiley, 1998.
- [30] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. New York: Academic, 1981.
- [31] R. Fletcher, *Practical Methods of Optimization*. New York: Wiley, 1987.



Kagan Gokbayrak (M'06) was born in Istanbul, Turkey, in 1972. He received the B.S. degrees in mathematics and in electrical engineering from Bogazici University, Istanbul, the M.S. degree in electrical and computer engineering from the University of Massachusetts, Amherst, and the Ph.D. degree in manufacturing engineering from Boston University, Boston, MA, in 1995, 1995, 1997, and 2001, respectively.

From 2001 to 2003, he was a Network Planning Engineer at Genuity, Inc., Burlington, MA. Since 2003, he has been a faculty member in the Industrial Engineering Department of Bilkent University, Ankara, Turkey. He specializes in the areas of discrete-event and hybrid systems, stochastic optimization, and computer simulation, with applications to inventory and manufacturing systems.



Omer Selvi was born in Nigde, Turkey, in 1976. He received the B.S. and M.S. degrees in industrial engineering from Bilkent University, Ankara, Turkey, in 1999 and 2002, where he is currently pursuing the Ph.D. degree.

His research interests are in the fields of discrete-event systems and stochastic optimization.