



Production, Manufacturing and Logistics

A search method for optimal control of a flow shop system of traditional machines

Omer Selvi, Kagan Gokbayrak*

Department of Industrial Engineering, Bilkent University, Ankara 06800, Turkey

ARTICLE INFO

Article history:

Received 18 July 2008

Accepted 26 December 2009

Available online 11 January 2010

Keywords:

Search method

Controllable service times

Convex programming

Trust-region methods

Flow shop

ABSTRACT

We consider a convex and nondifferentiable optimization problem for deterministic flow shop systems in which the arrival times of the jobs are known and jobs are processed in the order they arrive. The decision variables are the service times that are to be set only once before processing the first job, and cannot be altered between processes. The cost objective is the sum of regular costs on job completion times and service costs inversely proportional to the controllable service times. A finite set of subproblems, which can be solved by trust-region methods, are defined and their solutions are related to the optimal solution of the optimization problem under consideration. Exploiting these relationships, we introduce a two-phase search method which converges in a finite number of iterations. A numerical study is held to demonstrate the solution performance of the search method compared to a subgradient method proposed in earlier work.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

We consider flow shop systems consisting of traditional human operated (non-CNC) machines that are processing identical jobs. During mass production, a company cannot afford human interventions to modify the service times because the setup times for these modifications are idle times for these machines. Moreover, these manual modifications are prone to errors. Therefore, we assume that the service times at these traditional machines are initially controllable, i.e., they are set at the start-up time, and are applied to all jobs processed at these machines.

The cost to be minimized is assumed to consist of service costs on machines, which are dependent on service times, and regular completion time costs for jobs, e.g., inventory holding costs. Motivated by the extended Taylor's tool-wear equation (see in Kalpakjian and Schmid (2006)), we assume that faster services increase wear and tear on the machine tools due to increased temperatures and may raise the need for extra supervision, increasing service costs. The degradation of the product quality due to faster services are also lumped into these service costs. Slower services, on the other hand, build up inventory and postpone the completion times increasing the regular completion time costs. Our objective in this study is to determine the cost minimizing service times.

The scheduling problems of flow shops are known to be NP-hard even for fixed service times (see in Pinedo (2002)). In these problems, the objective is to find the best sequence of jobs

to be processed at machines. Except for two-machine systems with the objective of minimizing makespan, the scheduling literature is limited to heuristics and approximate solution methods. Introduction of controllable service times at machines further complicates the problem. Following the pioneering work in Vickson (1980), controllable service times have received great attention over the last three decades. Nowicki and Zdrzalka (1988) studied a two-machine flow shop system with the objective of minimizing a cost formed of makespan and decreasing linear service costs. Through reducing the knapsack problem to it, the problem was proven to be NP-hard even in the case where the service times are controllable only at the first machine. The heuristic algorithm proposed in this work was later extended to flow shops with more than two machines in Nowicki (1993). In Cheng and Shakhlevich (1999), an algorithm for a similar cost structure was presented for proportionate permutation flow shops where each job is associated with a single service time for all machines. Karabati and Kouvelis (1997) addressed the problem of minimizing a cost formed of decreasing linear service and regular cycle time costs, and introduced an iterative solution procedure where the task of selecting the optimal service times for a given sequence was formulated as a linear programming problem solved by a row generation scheme. Furthermore, a genetic algorithm for large problems was presented whose effectiveness was demonstrated through numerical studies. A survey of results on the controllable service times can be found in Nowicki and Zdrzalka (1990), Hoogeveen (2005) and Shabtay and Steiner (2007).

The studies above assumed the service costs to be decreasing linear functions of service times. This linearity assumption, however, fails to reflect the *law of diminishing marginal returns*:

* Corresponding author. Tel.: +90 312 290 3343.

E-mail addresses: selvi@bilkent.edu.tr (O. Selvi), kgokbayr@bilkent.edu.tr (K. Gokbayrak).

productivity increases at a decreasing rate with the amount of resource employed. Therefore, in this study, we adapt the service cost function $\theta_j(\cdot)$ on machine j defined as

$$\theta_j(s_j) = \frac{\beta_j}{s_j^\alpha}, \quad (1)$$

where β_j is a positive parameter, s_j is the service time at machine j , and α is a positive constant. This cost structure was shown to correspond to many industrial operations in Monma et al. (1990). Such nonlinear and convex service costs were also considered in Gurel and Akturk (2007).

The optimal control literature assumed that jobs are served in a given sequence, and concentrated on determining the optimal control inputs which in turn determine the optimal service times. Pepyne and Cassandras (1998) formulated a nonconvex and nondifferentiable optimal control problem for a single machine system with the objective of completing jobs as fast as possible with the least amount of control effort. The results were extended in Pepyne and Cassandras (2000) for jobs with completion due dates and a cost structure penalizing both earliness and tardiness. In Cassandras et al. (2001), the task of solving these problems was simplified by exploiting structural properties of the optimal sample path. Further exploiting the structural properties of the optimal sample path, “backward in time” and “forward in time” algorithms based on the decomposition of the original nonconvex and nondifferentiable optimization problem into a set of smaller convex optimization problems with linear constraints were presented in Wardi et al. (2001) and Cho et al. (2001), respectively. The “forward in time” algorithm presented in Cho et al. (2001) was then improved in Zhang and Cassandras (2002). Mao et al. (2004) removed the completion time costs and introduced due date constraints. Some optimal solution properties of the resulting problem were identified leading to a highly efficient solution algorithm.

The work on two-machine systems started out with Cassandras et al. (1999), which derived some necessary conditions for optimality and introduced a solution technique using the Bezier approximation method. Extending the work in Mao et al. (2004), Mao and Cassandras (2006) considered a two-machine flow shop system with service costs that are decreasing on service times, and derived some optimality properties that led to an iterative algorithm, which was shown to converge. Gokbayrak and Selvi (2006) studied a two-machine flow shop system with a regular cost on completion times and decreasing costs on service times, and identified some optimal sample path characteristics to simplify the problem. In particular, no waiting was observed between machines on the optimal sample path leading to the transformation of the nonconvex discrete-event optimal control problem into a simple convex programming problem. Gokbayrak and Selvi (2007) extended the no-wait property to multimachine flow shop systems. Using this property, simpler equivalent convex programming formulations were presented and “forward in time” solution algorithms were developed under strict convexity assumptions on service and completion time costs. Gokbayrak and Selvi (2010) generalized the results to multimachine mixed line flow shop systems with Computer Numerical Control (CNC) and traditional machines. The no-wait property was shown to exist for the downstream of the first controllable (CNC) machine of the system. Employing this result, a simplified convex optimization problem along with a “forward in time” decomposition algorithm were introduced enabling for solving large systems in short times and with low memory requirements.

Employing the cost structure in Gokbayrak and Selvi (2007), Gokbayrak and Selvi (2008) and Gokbayrak and Selvi (2009) considered a deterministic flow shop system where the service times at machines are set only once, and cannot be altered between processes.

Gokbayrak and Selvi (2008) derived a set of waiting characteristics in such systems and presented an equivalent simple convex optimization problem employing these characteristics. In order to eliminate the need for convex programming solvers, Gokbayrak and Selvi (2009) derived additional waiting characteristics and introduced a minmax problem, which is almost everywhere differentiable, of a finite set of convex functions along with its subgradient descent solution algorithm. In this study, we propose an alternative solution method for the minmax problem in Gokbayrak and Selvi (2009). The relationships between the minimizers of the convex functions in the minmax problem and the optimal solution are derived. These relationships suggest a two-phase search algorithm that determines the optimal solution in a finite number of iterations. In each iteration a convex optimization problem needs to be solved. For the special case where the service cost structure is as in (1) allowing us to sort the service times of the machines, these convex optimization problems are solved by trust-region methods.

The rest of the paper is organized as follows: In Section 2, we describe the problem and present the minmax formulation given in Gokbayrak and Selvi (2009). In Section 3, we derive the relationships between the optimal solution and the minimizers of the convex functions in the minmax formulation. Consequently, the two-phase search algorithm is presented in this section. Implementation details of this search algorithm are given in Section 4 for the service cost structure in (1). Section 5 demonstrates the solution performance of the proposed methodology by a numerical study. Finally, Section 6 concludes the paper.

2. Problem formulations

Let us consider an M -machine flow shop system with unlimited buffer spaces between machines. A sequence of N identical jobs, denoted by $\{C_i\}_{i=1}^N$, arrive at this system at known times $0 \leq a_1 \leq a_2 \leq \dots \leq a_N$. Machines process one job at a time on a first-come first-served non-preemptive basis, i.e., a job in service can not be interrupted until its service is completed. The service time at each machine j , denoted by s_j , is the same for all jobs and is the j th entry of the service time vector $s = (s_1, \dots, s_M)$.

We consider the discrete-event optimal control problem P , which has the following form:

$$P : \min \left\{ J(s) = \sum_{j=1}^M \theta_j(s_j) + \sum_{i=1}^N \phi_i(x_{i,M}) \right\} \quad (2)$$

subject to

$$x_{i,j} = \max(x_{i,j-1}, x_{i-1,j}) + s_j \quad (3)$$

$$x_{i,0} = a_i, \quad x_{0,j} = -\infty \quad (4)$$

$$s_j \geq 0 \quad (5)$$

for $i = 1, \dots, N$ and $j = 1, \dots, M$, where $x_{i,j}$ denotes the departure time of job C_i from machine j .

In this formulation, θ_j denotes the service cost at machine j and ϕ_i denotes the completion time cost for job C_i . 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 services will decrease the service costs, while increasing the completion times, hence the completion time costs. This trade-off is what makes our problem interesting.

By the nature of the event-driven dynamics given by (3), the problem is inherently nonconvex and nondifferentiable. In the following subsection, the equivalent convex optimization formulation presented in Gokbayrak and Selvi (2009) is revisited.

2.1. Minmax problem

For each job C_i , let us define

$$\sigma_i = \begin{cases} \infty & i = 1, \\ \min_{n=1, \dots, i-1} \left\{ \frac{a_i - a_n}{i - n} \right\} & i > 1, \end{cases} \quad (6)$$

and form the set

$$\Psi = \{0\} \cup \{\sigma_i : i = 1, \dots, N\}.$$

Let us sort and re-index the elements of Ψ with cardinality $\bar{N} + 1$ so that

$$\sigma_{(0)} < \sigma_{(1)} < \sigma_{(2)} < \dots < \sigma_{(\bar{N})},$$

where $\sigma_{(0)}$ and $\sigma_{(\bar{N})}$ are defined to be zero and infinity, respectively. For some distinct jobs C_k and C_l , we may have $\sigma_k = \sigma_l$, so the cardinality of Ψ is at most $N + 1$, i.e., $\bar{N} \leq N$.

Next, we define $r_i(k)$ values as

$$r_i(k) = \begin{cases} \max\{j : \sigma_j \geq \sigma_{(k)}, j \leq i\} & k < \bar{N} \\ 1 & k = \bar{N} \end{cases} \quad (7)$$

for all $i = 1, \dots, N$. Employing these $r_i(k)$ values, we define $y_i^k(s)$ as

$$y_i^k(s) = a_{r_i(k)} + (i - r_i(k))s_{\max} + s_{\text{total}}, \quad (8)$$

where

$$s_{\max} = \max_{j=1, \dots, M} s_j,$$

and

$$s_{\text{total}} = \sum_{j=1}^M s_j.$$

Having defined $y_i^k(s)$, we formulate the equivalent minmax problem of at most N functions:

$$R : \min_{\substack{s_j \geq 0 \\ j=1, \dots, M}} \left\{ J_R(s) = \max_{k=1, \dots, \bar{N}} \{J_k(s)\} \right\}, \quad (9)$$

where $J_k(s)$ functions can be written as

$$J_k(s) = \sum_{j=1}^M \theta_j(s_j) + \sum_{i=1}^N \phi_i(y_i^k(s)). \quad (10)$$

Employing Assumptions 1 and 2, we can show that $\{J_k\}_{k=1}^{\bar{N}}$ and $J_R(s)$ functions are continuous and strictly convex. Borrowed from Gokbayrak and Selvi (2009), the following lemma states that $J_k(s)$ exceeds all other cost functions $\{J_t(s)\}_{t=1}^{\bar{N}}$ when s_{\max} is in the k th interval $[\sigma_{(k-1)}, \sigma_{(k)}]$.

Lemma 1. $J_k(s) \geq J_t(s)$ for all $t \in \{1, \dots, \bar{N}\}$ and for all s satisfying $s_{\max} \in [\sigma_{(k-1)}, \sigma_{(k)}]$.

It follows from (9) and Lemma 1 that $J_R(s) = J_k(s)$ when $s_{\max} \in [\sigma_{(k-1)}, \sigma_{(k)}]$.

Unfortunately, $J_k(s)$ cost functions are nondifferentiable functions: For any service vector $s = (s_1, \dots, s_M)$, the sensitivities of the cost function J_k are given as

$$\frac{\partial J_k}{\partial s_j} = \begin{cases} \theta'_j(s_j) + \sum_{i=1}^N \phi'_i(y_i^k(s)) & s_j < s_{\max} \\ \theta'_j(s_j) + \sum_{i=1}^N [\phi'_i(y_i^k(s))(1 + i - r_i(k))] & s_j > \max_{i \neq j} s_i \end{cases} \quad (11)$$

for $j = 1, \dots, M$. Due to the s_{\max} term in (8), when there are multiple machines with the maximum service time s_{\max} , i.e., when $s_j = \max_{i \neq j} s_i$, nondifferentiability is observed. Consequently, a subgradient algorithm was proposed in Gokbayrak and Selvi (2009).

In this paper, we derive relationships between the minimizers of J_k and J_R functions and propose a search algorithm as an alternative solution method.

3. Two-phase search algorithm

Since $\{J_k\}_{k=1}^{\bar{N}}$ and J_R are strictly convex, they have unique minimizers. Let us denote these minimizers by $\{s^k\}_{k=1}^{\bar{N}}$ and s^* , respectively. In the next theorem, we present some relationships between these minimizers.

Theorem 2. For any $k \in \{1, \dots, \bar{N}\}$, the minimizer of J_k carries the following information about the minimizer of J_R :

- (i) If $s_{\max}^k \in [\sigma_{(k-1)}, \sigma_{(k)}]$, then $s^* = s^k$,
- (ii) if $s_{\max}^k > \sigma_{(k)}$, then $s_{\max}^* \geq \sigma_{(k)}$,
- (iii) if $s_{\max}^k < \sigma_{(k-1)}$, then $s_{\max}^* \leq \sigma_{(k-1)}$.

Proof. (i) If $s_{\max}^k \in [\sigma_{(k-1)}, \sigma_{(k)}]$, then from Lemma 1, $J_k(s^k) = J_R(s^k)$. Since s^k minimizes J_k , we have

$$J_R(s^k) = J_k(s^k) \leq J_k(s) \leq \max_{t=1, \dots, \bar{N}} J_t(s) = J_R(s)$$

for all s , i.e., s^k is the optimal solution of R .

(ii) For a contradiction, let us assume that $s_{\max}^k > \sigma_{(k)}$ and $s_{\max}^* < \sigma_{(k)}$. Hence, we can define a nonempty set I_1 as

$$I_1 = \{i : s_i^k > \sigma_{(k)}\}.$$

Let us define an alternate solution \bar{s} as

$$\bar{s} = s^* + \gamma_1(s^k - s^*),$$

where γ_1 is defined as

$$\gamma_1 = \min_{j \in I_1} \left\{ \frac{\sigma_{(k)} - s_j^*}{s_j^k - s_j^*} \right\}$$

so that $\bar{s}_{\max} = \sigma_{(k)}$. Then, from Lemma 1 and strict convexity of J_k , we have

$$J_k(s^k) < J_k(\bar{s}) = J_R(\bar{s}) < J_k(s^*) \leq \max_{t=1, \dots, \bar{N}} J_t(s^*) = J_R(s^*),$$

which contradicts the optimality of s^* for J_R . Hence, the optimal solution for R satisfies $s_{\max}^* \geq \sigma_{(k)}$.

(iii) For a contradiction, let us assume that $s_{\max}^k < \sigma_{(k-1)}$ and $s_{\max}^* > \sigma_{(k-1)}$. Hence, we can define a nonempty set I_2 as

$$I_2 = \{i : s_i^* > \sigma_{(k-1)}\}.$$

Let us define an alternate solution \hat{s} as

$$\hat{s} = s^k + \gamma_2(s^* - s^k)$$

where γ_2 is defined as

$$\gamma_2 = \min_{j \in I_2} \left\{ \frac{\sigma_{(k-1)} - s_j^k}{s_j^* - s_j^k} \right\}$$

so that $\hat{s}_{\max} = \sigma_{(k-1)}$. Then, from Lemma 1 and strict convexity of J_k , we have

$$J_k(s^k) < J_k(\hat{s}) = J_R(\hat{s}) < J_k(s^*) \leq \max_{t=1, \dots, \bar{N}} J_t(s^*) = J_R(s^*),$$

which contradicts the optimality of s^* for J_R . Hence, the optimal solution for R satisfies $s_{\max}^* \leq \sigma_{(k-1)}$. \square

The direct corollary of Theorem 2 is the following:

Corollary 3. The minimizer of J_k yields the following information:

- (i) If $s_{\max}^k > \sigma_{(k)}$, then $s_{\max}^l \notin [\sigma_{(l-1)}, \sigma_{(l)}]$ for all $l \leq k$,
- (ii) if $s_{\max}^k < \sigma_{(k-1)}$, then $s_{\max}^l \notin [\sigma_{(l-1)}, \sigma_{(l)}]$ for all $l \geq k$.

Proof. (i) If $s_{\max}^l \in [\sigma_{(l-1)}, \sigma_{(l)}]$ for some $l < k$ satisfying $\sigma_{(l)} < \sigma_{(k)}$, then $s^* = s^l$, i.e., $s_{\max}^* \leq \sigma_{(l)} < \sigma_{(k)}$. However, if $s_{\max}^k > \sigma_{(k)}$, then we should have $s_{\max}^* \geq \sigma_{(k)}$, which yields a contradiction. Having $s_{\max}^k \in [\sigma_{(k-1)}, \sigma_{(k)}]$ and $s_{\max}^k > \sigma_{(k)}$ at the same time is also a contradiction.

(ii) If $s_{\max}^l \in [\sigma_{(l-1)}, \sigma_{(l)}]$ for some $l > k$ satisfying $\sigma_{(l-1)} > \sigma_{(k-1)}$, then $s^* = s^l$, i.e., $s_{\max}^* \geq \sigma_{(l-1)} > \sigma_{(k-1)}$. However, if $s_{\max}^k < \sigma_{(k-1)}$, then we should have $s_{\max}^* \leq \sigma_{(k-1)}$, which yields a contradiction. Having $s_{\max}^k \in [\sigma_{(k-1)}, \sigma_{(k)}]$ and $s_{\max}^k < \sigma_{(k-1)}$ at the same time is also a contradiction. \square

Motivated by Theorem 2 and Corollary 3, we develop a search algorithm that operates in two phases: In Phase 1, we search for a J_k whose minimizer satisfies $s_{\max}^k \in [\sigma_{(k-1)}, \sigma_{(k)}]$. Corollary 3 suggests a bisection search for this phase. This phase can yield two different results: If the search is successful to find an $s_{\max}^l \in [\sigma_{(l-1)}, \sigma_{(l)}]$ for some $l = 1, \dots, \bar{N}$, then it will terminate with the optimal solution s^* . If, on the other hand, $s_{\max}^l \notin [\sigma_{(l-1)}, \sigma_{(l)}]$ for all $l = 1, \dots, \bar{N}$, then this phase will yield a $k \in \{1, \dots, \bar{N} - 1\}$ satisfying

$$s_{\max}^k > \sigma_{(k)} > s_{\max}^{k+1}.$$

In this case, from Theorem 2, we conclude that $s_{\max}^* = \sigma_{(k)}$. Since $J_R(s) = J_k(s)$ when $s_{\max} = \sigma_{(k)}$, we proceed to Phase 2, which searches for the solution that minimizes J_k under the constraint that $s_{\max} = \sigma_{(k)}$.

The search algorithm we describe above requires us to determine the minimizers of $\{J_k\}_{k=1}^{\bar{N}}$ with or without the additional $s_{\max} = \sigma_{(k)}$ constraint. Efficient methods are available if we limit the service cost structure to (1).

4. Determining the minimizers of J_k functions

As discussed before, J_k functions are nondifferentiable at points where multiple machines have the maximum service time. If we limit the service cost structure to (1), we can determine the machines that should be assigned the maximum service time beforehand to introduce differentiable cost functions. The next lemma states that there exists an ordering among the optimal service times s_j^k determined by the β_j values of the service cost structure given in (1).

Lemma 4. For any two machines u and v , if $\beta_u \geq \beta_v$ then $s_u^k \geq s_v^k$.

Proof. For a contradiction, let us assume that while $\beta_u \geq \beta_v$, the optimal service times satisfy

$$s_u^k < s_v^k, \tag{12}$$

and define the perturbed service times \bar{s}_j for $j = 1, \dots, M$ as

$$\bar{s}_j = \begin{cases} s_u^k + \Delta & j = u \\ s_v^k - \Delta & j = v \\ s_j^k & \text{otherwise} \end{cases} \tag{13}$$

with $0 < \Delta < \frac{s_u^k - s_v^k}{2}$. Note that, from (12) and (13), we have $s_u^k < \bar{s}_u < \bar{s}_v < s_v^k$, therefore we can write

$$s_{\max}^k \geq \bar{s}_{\max}, \tag{14}$$

and

$$s_{\text{total}}^k = \bar{s}_{\text{total}}. \tag{15}$$

Then, from (8), (14) and (15), we have

$$y_i^k(\bar{s}) \leq y_i^k(s^k) \tag{16}$$

for all $i = 1, \dots, N$.

Moreover, since $\beta_u \geq \beta_v$ and $s_u^k < s_v^k$, the inequality

$$\theta'_u(s_u^k) \leq \theta'_v(s_v^k) \tag{17}$$

is satisfied.

If we denote the cost of the perturbed solution as \bar{J}_k and the cost of the unique minimizer s^k as J_k^* , by Assumptions 1 and 2, and from (12), (13), (16) and (17), we have

$$\begin{aligned} \bar{J}_k - J_k^* &= \theta_u(s_u^k + \Delta) - \theta_u(s_u^k) - \theta_v(s_v^k) + \theta_v(s_v^k - \Delta) \\ &+ \sum_{i=1}^N [\phi_i(y_i^k(\bar{s})) - \phi_i(y_i^k(s^k))] < 0, \end{aligned}$$

which contradicts the optimality assumption and concludes the proof. \square

Employing Lemma 4, we conclude that there exists a $\beta^k \in \{\beta_j\}_{j=1}^M$ threshold value such that if $\beta_j \geq \beta^k$ then $s_j^k = s_{\max}^k$. In the following subsection, we propose a method to determine β^k so that we can form a differentiable cost function and apply calculus of variations techniques to determine s^k .

4.1. Locating minimizers in Phase 1

We define a cost function J_k^β as

$$J_k^\beta(s) = \sum_{j \in I_\beta} \theta_j(s_m) + \sum_{j \notin I_\beta} \theta_j(s_j) + \sum_{i=1}^N \phi_i(y_i^k(s, \beta)), \tag{18}$$

where I_β is the set $\{j : \beta_j \geq \beta\}$ with cardinality K_β , s_m is the service time of the most upstream machine m with $\beta_m = \max_j \beta_j$, and $y_i^k(s, \beta)$ is defined as

$$y_i^k(s, \beta) = a_{r_i(k)} + (K_\beta + i - r_i(k))s_m + \sum_{j \notin I_\beta} s_j. \tag{19}$$

Employing these differentiable cost functions, we define a family of problems Q_k^β as

$$Q_k^\beta : \min_s J_k^\beta(s)$$

subject to

$$\begin{aligned} s_j &= s_m & \text{for } j \in I_\beta \setminus \{m\} \\ s_j &\geq 0 & \text{for } j \in \{1, \dots, M\}. \end{aligned}$$

A specific member of this family, $Q_k^{\beta^k}$ will be of interest to us, as its optimal solution will be s^k .

By the cost structure in (1) and Assumption 2, the optimal solution should be finite and nonzero. Hence, applying calculus of variations techniques to solve $Q_k^{\beta^k}$, we obtain the following set of equations satisfied by the optimal solution s^{β^k} :

$$\begin{aligned} \theta'_j(s_j^{\beta^k}) + \sum_{i=1}^N \phi'_i(y_i^k(s^{\beta^k}, \beta)) &= 0 \quad j \notin I_\beta, \\ \sum_{j \in I_\beta} \theta'_j(s_m^{\beta^k}) + \sum_{i=1}^N [\phi'_i(y_i^k(s^{\beta^k}, \beta))(K_\beta + i - r_i(k))] &= 0 \quad j = m, \\ s_j^{\beta^k} &= s_m^{\beta^k} \quad j \in I_\beta \setminus \{m\}. \end{aligned} \tag{20}$$

For the cost structure in (1), the first equality in (20) suggests that we can pick an arbitrary machine $u \notin I_\beta$ and write

$$s_v^{\beta^k} = c_{u,v} s_u^{\beta^k} \tag{21}$$

for all machines $v \notin I_\beta$ where $c_{u,v} = \left(\frac{\beta_u}{\beta_v}\right)^{1/(\alpha+1)}$. Employing (21) in (20), we need to solve the following two nonlinear equations with two unknowns s_m^β and s_u^β

$$\theta'_u(s_u^\beta) + \sum_{i=1}^N \phi'_i(\bar{y}_i^k(s_m^\beta, s_u^\beta, \beta)) = 0, \tag{22}$$

$$\sum_{j \in I_\beta} \theta'_j(s_m^\beta) + \sum_{i=1}^N [\phi'_i(\bar{y}_i^k(s_m^\beta, s_u^\beta, \beta))(K_\beta + i - r_i(k))] = 0, \tag{23}$$

where $\bar{y}_i^k(s_m^\beta, s_u^\beta, \beta)$ is defined as

$$\bar{y}_i^k(s_m^\beta, s_u^\beta, \beta) = a_{r_i(k)} + (K_\beta + i - r_i(k))s_m^\beta + \sum_{j \notin I_\beta} c_{u,j} s_u^\beta \tag{24}$$

for all $i = 1, \dots, N$.

Note that, independent of M , there are only two unknowns s_m^β and s_u^β in the Eqs. (22) and (23). This system can be solved by well-known solution techniques such as Trust-Region methods (see in Conn et al. (1987)).

Being able to solve for s^β , the β^k value can be determined by a one-directional search, as motivated by the following theorem:

Theorem 5. *If $\beta > \beta^k$, then the optimal solution s^β of Q_k^β satisfies $\max_{j \notin I_\beta} s_j^\beta \geq s_m^\beta$.*

Proof. For a contradiction, assume that $s_j^\beta < s_m^\beta$ is satisfied for all $j \notin I_\beta$ so that $s_j^\beta = s_m^\beta = s_{\max}^\beta$ for all $j \in I_\beta$. If $\beta > \beta^k$ so that $I_\beta \subset I_{\beta^k}$, then $s_j^k = s_m^k = s_{\max}^k$ for all $j \in I_\beta$. Hence, from (10) and (18), we have

$$J_k(s^\beta) = J_k^k(s^\beta), \tag{25}$$

$$J_k(s^k) = J_k^k(s^k). \tag{26}$$

If $\beta > \beta^k$, then there exists a machine u with $\beta > \beta_u \geq \beta^k$, i.e., $u \in I_{\beta^k} \setminus I_\beta$. By the contradiction assumption, we have $s_u^\beta < s_m^\beta = s_{\max}^\beta$ while $s_u^k = s_m^k = s_{\max}^k$, therefore $s^k \neq s^\beta$. Since s^k is the unique minimizer of J_k , we have

$$J_k(s^\beta) > J_k(s^k). \tag{27}$$

It follows from (25)–(27) that

$$J_k^\beta(s^\beta) > J_k^\beta(s^k),$$

which contradicts with the optimality of s^β for J_k^β . Hence the result follows. \square

In our search for the β^k value, we start with $\beta = \beta_m$, and solve for s^β to check the condition in Theorem 5. If the optimal solution s^β satisfies $\max_{j \notin I_\beta} s_j^\beta \geq s_m^\beta$, then we lower the β value to $\max_{j \notin I_\beta} \beta_j$, the largest element of the set $\{\beta_1, \dots, \beta_M\}$ smaller than β , and continue until $\max_{j \notin I_\beta} s_j^\beta < s_m^\beta$ is satisfied. The search algorithm results with the β^k value along with the minimizer s^k of J_k .

As discussed above, in some cases, instead of the optimal solution s^* , the search in Phase 1 may result with the information that $s_{\max}^* = \sigma_{(k)}$ for some k . Next, we present how to obtain the optimal solution s^* employing this information.

4.2. Locating the optimal solution in Phase 2

Since $J_R = J_k$ when $s_{\max} = \sigma_{(k)}$, in Phase 2, we consider a family of problems \hat{Q}_k^β defined as

$$\hat{Q}_k^\beta : \min_s J_k^\beta(s)$$

subject to

$$s_j = \sigma_{(k)} \quad \text{for } j \in I_\beta$$

$$s_j \geq 0 \quad \text{for } j \in \{1, \dots, M\}.$$

A specific member of this family, $\hat{Q}_k^{\beta^*}$ will be of interest to us as its optimal solution will be s^* .

By the cost structure in (1) and Assumption 2, the optimal solution should be finite and nonzero. Hence, applying calculus of variations techniques to solve \hat{Q}_k^β , we obtain the following set of equations satisfied by the optimal solution \hat{s}^β :

$$\theta'_j(\hat{s}_j^\beta) + \sum_{i=1}^N \phi'_i(y_i^k(\hat{s}^\beta, \beta)) = 0 \quad j \notin I_\beta, \tag{28}$$

$$\hat{s}_j = \sigma_{(k)} \quad j \in I_\beta.$$

For the cost structure in (1), the first equality in (28) suggests that we can pick an arbitrary machine $u \notin I_\beta$ and write

$$\hat{s}_v^\beta = c_{u,v} \hat{s}_u^\beta \tag{29}$$

for all machines $v \notin I_\beta$ where $c_{u,v} = \left(\frac{\beta_v}{\beta_u}\right)^{1/(\alpha+1)}$. Employing (29) in (28), we end up with the following nonlinear equation with only one unknown \hat{s}_u^β

$$\theta'_u(\hat{s}_u^\beta) + \sum_{i=1}^N \phi'_i(\bar{y}_i^k(\hat{s}_u^\beta, \beta, \sigma_{(k)})) = 0, \tag{30}$$

where $\bar{y}_i^k(\hat{s}_u^\beta, \beta, \sigma_{(k)})$ is defined as

$$\bar{y}_i^k(\hat{s}_u^\beta, \beta, \sigma_{(k)}) = a_{r_i(k)} + (K_\beta + i - r_i(k))\sigma_{(k)} + \sum_{j \notin I_\beta} c_{u,j} \hat{s}_u^\beta \tag{31}$$

for all $i = 1, \dots, N$.

Note that, independent of M , there is only one unknown in the Eq. (30) that can be solved by Trust-Region methods.

Having the \hat{s}^β solution available, we can determine the relationship between β and β^* , the value corresponding to s^* , by the following theorem whose proof is skipped as it is very similar to the proof of Theorem 5:

Theorem 6. *If $\beta > \beta^*$, then the optimal solution \hat{s}^β of \hat{Q}_k^β satisfies $\max_{j \notin I_\beta} \hat{s}_j^\beta \geq \sigma_{(k)}$.*

The β^* value required to determine the optimal solution s^* is obtained by the same search method as in Phase 1: Employing Theorem 6, we start with $\beta = \beta_m$ and solve \hat{Q}_k^β . If the optimal solution \hat{s}^β satisfies $\max_{j \notin I_\beta} \hat{s}_j^\beta \geq \sigma_{(k)}$, then we lower the β value to $\max_{j \notin I_\beta} \beta_j$, the largest element of the set $\{\beta_1, \dots, \beta_M\}$ smaller than β , and continue until $\max_{j \notin I_\beta} \hat{s}_j^\beta < \sigma_{(k)}$ is satisfied. The search algorithm results with the β^* value along with the optimal solution s^* of J_R .

4.3. Resulting search algorithm

Under the light of the previous discussions, we develop a two-phase search algorithm as depicted in Fig. 1.

In the *Initialization* step, we first determine the most upstream machine m with the largest β value β_m . Then, we determine σ_i values for $i = 1, \dots, N$ by employing (6), form the Ψ set, and determine $\sigma_{(k)}$ values for $k = 0, \dots, \bar{N}$. Finally, in order to search by bisectioning, we set the function index to $k = \lceil (\bar{N} + 1)/2 \rceil$. The variables lb and ub are employed to keep track of the lower and upper bounds of the index search space.

In Phase 1, we search for a cost function J_k whose minimizer satisfies $s_{\max}^k \in [\sigma_{(k-1)}, \sigma_{(k)}]$. In order to obtain the service times minimizing the nondifferentiable cost function J_k , we need to solve a series of differentiable problems $\{Q_k^\beta\}_{\beta=\beta^k}^{\beta=\beta_m}$: Employing Theorem 5, we start with $\beta = \beta_m$ and solve Q_k^β . If the optimal solution s^β satisfies $\max_{j \notin I_\beta} s_j^\beta \geq s_m^\beta$, then we lower the β value to $\max_{j \notin I_\beta} \beta_j$, the largest element of the set $\{\beta_1, \dots, \beta_M\}$ smaller than β , and continue until $\max_{j \notin I_\beta} s_j^\beta < s_m^\beta$ is satisfied. The search algorithm results with the β^k value along with the optimal solution s^k of J_k . If we obtain a solution $s_{\max}^k \in [\sigma_{(k-1)}, \sigma_{(k)}]$ for some $k = 1, \dots, \bar{N}$, by Theorem 2, we conclude that it is the optimal solution of J_R , and stop. Otherwise, we conclude that $s_{\max}^* = \sigma_{(k)}$ for some k and proceed with Phase 2.

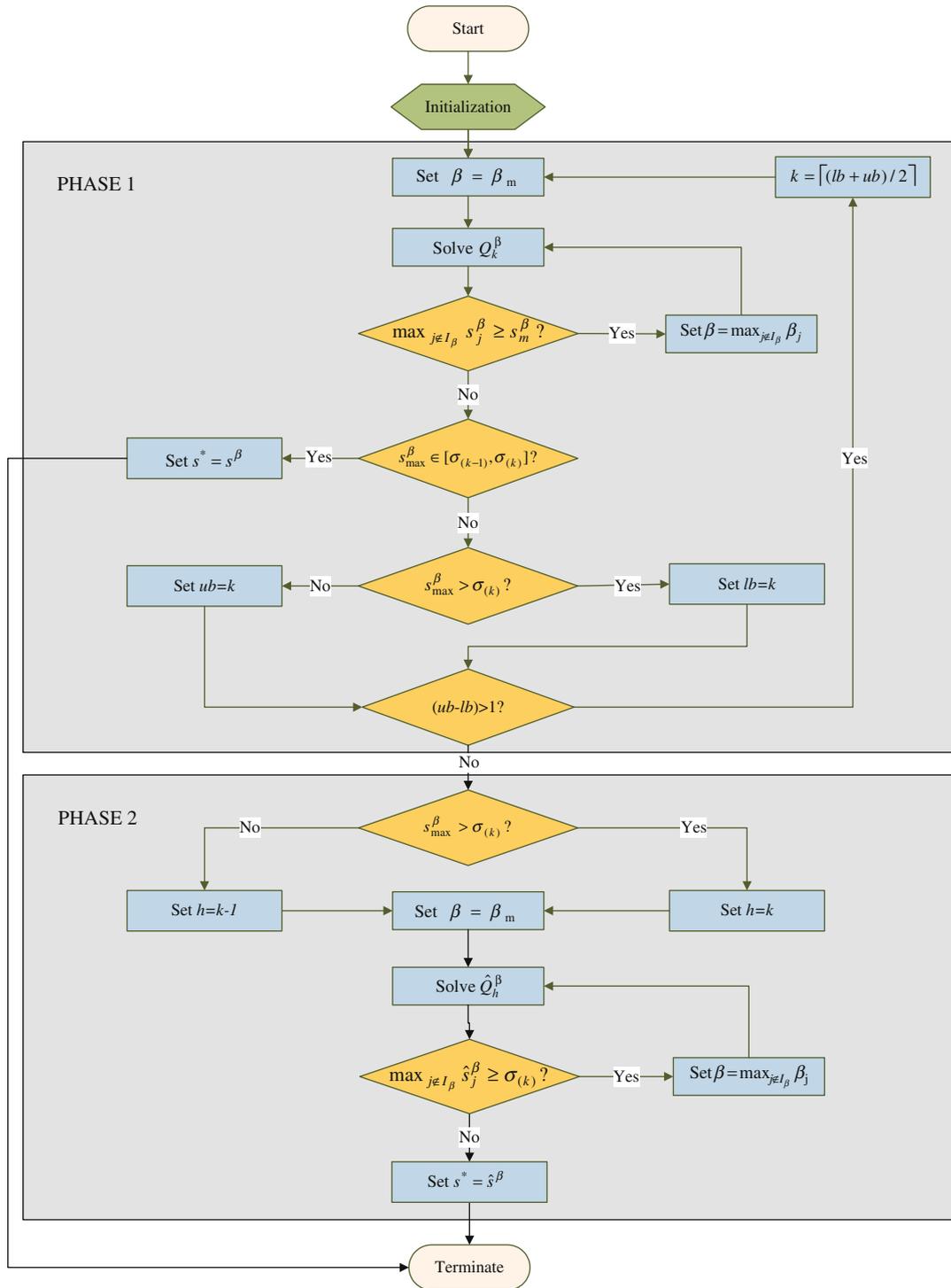


Fig. 1. Flowchart of the two-phase search algorithm.

In Phase 2, we solve a series of differentiable problems $\{Q_k^\beta\}_{\beta=\beta^*}^{\beta=\beta_m}$ to obtain the optimal solution s^* of J_R : Employing Theorem 6, we start with $\beta = \beta_m$ and solve \hat{Q}_k^β . If the optimal solution \hat{s}^β satisfies $\max_{j \in I_\beta} \hat{s}_j^\beta \geq \sigma_{(k)}$, then we lower the β value to $\max_{j \in I_\beta} \beta_j$, the largest element of the set $\{\beta_1, \dots, \beta_M\}$ smaller than β , and continue until $\max_{j \in I_\beta} \hat{s}_j^\beta < \sigma_{(k)}$ is satisfied. The search algorithm results with the β^* value along with the optimal solution s^* of J_R .

In the worst case, the two-phase search algorithm solves $M \lceil \log_2 N \rceil$ of Q_k^β problems involving two nonlinear equations and two unknowns, and M of \hat{Q}_k^β problems involving one nonlinear

equation and one unknown. Hence, it determines the optimal solution in a finite number of steps.

We continue with a numerical study that compares the performances of the two-phase search algorithm and the subgradient descent algorithm in Gokbayrak and Selvi (2009).

5. Numerical example

Let us consider an M -machine flow shop system processing an identical set of N jobs. The service cost $\theta_j(s_j)$ at machine j is given as

Table 1
Average CPU times for subgradient descent and two-phase search algorithms.

N	M = 20		M = 40		M = 60	
	SD	2PS	SD	2PS	SD	2PS
500	0.83	0.32	1.28	0.34	1.70	0.35
1000	1.72	0.33	2.97	0.35	4.22	0.37
1500	3.24	0.35	5.34	0.37	7.71	0.39
2000	6.23	0.37	9.45	0.39	12.60	0.41
2500	9.75	0.39	13.77	0.42	18.53	0.44
3000	13.27	0.41	19.55	0.45	25.83	0.48
4000	20.63	0.50	31.16	0.54	41.30	0.59
000	29.52	0.66	46.75	0.71	63.47	0.77
6000	41.57	0.83	65.82	0.89	89.39	0.96
10,000	106.87	1.72	177.68	1.79	235.99	1.87

$$\theta_j(s_j) = \frac{\beta_j}{s_j} \quad (32)$$

for some constant β_j with α in (1) is set to 1. The completion time cost for job C_i , on the other hand, is given as

$$\phi_i(x_{i,M}) = 10(x_{i,M} - a_i)^2, \quad (33)$$

which satisfies Assumption 2.

In order to compare the solution performances of the search algorithm and the subgradient descent algorithm, we study problems with different M and N settings. The β_j values are randomly selected from the set $\{5i : i = 1, \dots, 20\}$ and the job interarrival times are realized from an exponential distribution with a mean of 2 units.

The problems are solved in Matlab running on a machine with a 2.0GHz Intel Core2Duo T7200 processor and 2GB of RAM. The subgradient descent algorithm (SD) employs the precision measure ϵ with a value of 10^{-5} and the step sizes $\eta_k = \frac{10^{-5}}{k}$. The two-phase search algorithm (2PS) uses *fsolve* function employing a variant of the Powell dogleg method described in Powell (1970) to solve the Q_k^{β} and \hat{Q}_k^{β} problems. Averaged over 10 optimization problems (obtained by varying arrival sequences $\{a_i\}_{i=1}^N$ and the cost parameters $\{\beta_j\}_{j=1}^M$), the computation times for the alternative methodologies for different M and N settings are presented in Table 1.

The two-phase search algorithm improved on the solution times of the subgradient descent algorithm as seen in Table 1. These improvements get drastic as the problem size increases, e.g., for 100 machines and 50,000 jobs, the average computation time over 10 sample problems is 11352.26 seconds for the subgradient descent algorithm, while this value is only 33.06 seconds for the two-phase search algorithm.

6. Conclusion

We considered a service time optimization problem of flow shop systems with fixed service times. As an alternative to the subgradient algorithm in Gokbayrak and Selvi (2009), we proposed a search algorithm that finds the optimal solution in a finite number of iterations. For a specific service cost structure, which allowed us to sort the optimal service times, this search algorithm proved to be extremely efficient. Instead of applying subgradient descent methods on an M -dimensional solution space, we applied trust-region methods to solve at most two nonlinear equations with two

unknowns at each iteration. As a result, it improved the solution times drastically compared to the subgradient descent algorithm proposed in Gokbayrak and Selvi (2009).

References

- Cassandras, C.G., Liu, Q., Pepyne, D.L., Gokbayrak, K., 1999. Optimal control of a two-stage hybrid manufacturing system model. In: Proceedings of 38th IEEE Conference on Decision and Control, pp. 450–455.
- Cassandras, C.G., Pepyne, D.L., Wardi, Y., 2001. Optimal control of a class of hybrid systems. IEEE Transactions on Automatic Control 46 (3), 398–415.
- Cheng, T.C.E., Shakhlevich, N., 1999. Proportionate flow shop with controllable processing times. Journal of Scheduling 2 (6), 253–265.
- Cho, Y.C., Cassandras, C.G., Pepyne, D.L., 2001. Forward decomposition algorithms for optimal control of a class of hybrid systems. International Journal of Robust and Nonlinear Control 11, 497–513.
- Conn, A.R., Gould, N.I.M., Toint, P.L., 1987. Trust-Region Methods. Society for Industrial Mathematics.
- Gokbayrak, K., Selvi, O., 2006. Optimal hybrid control of a two-stage manufacturing system. Proceedings of ACC, 3364–3369.
- Gokbayrak, K., Selvi, O., 2007. Constrained optimal hybrid control of a flow shop system. IEEE Transactions on Automatic Control 52 (12), 2270–2281.
- Gokbayrak, K., Selvi, O., 2008. Optimization of a flow shop system of initially controllable machines. IEEE Transactions on Automatic Control 53 (11), 2665–2668.
- Gokbayrak, K., Selvi, O., 2009. A subgradient descent algorithm for optimization of initially controllable flow shop systems. Discrete Event Dynamic Systems: Theory and Applications 19 (2), 267–282.
- Gokbayrak, K., Selvi, O., 2010. Service time optimization of mixed line flow shop systems. IEEE Transactions on Automatic Control 55 (2).
- Gurel, S., Akturk, M.S., 2007. Considering manufacturing cost and scheduling performance on a CNC turning machine. European Journal of Operational Research 177, 325–343.
- Hoogeveen, H., 2005. Multicriteria scheduling. European Journal of Operational Research 167, 592–623.
- Kalpakjian, S., Schmid, S.R., 2006. Manufacturing Engineering and Technology. Pearson Prentice-Hall.
- Karabati, S., Kouvelis, P., 1997. Flow-line scheduling problem with controllable processing times. IIE Transactions 29 (1), 1–14.
- Mao, J., Cassandras, C.G., 2006. Optimal control of two-stage discrete event systems with real-time constraints. In: Proceedings of Eighth International Workshop on Discrete Event Systems, pp. 125–130.
- Mao, J., Zhao, Q., Cassandras, C.G., 2004. Optimal dynamic voltage scaling in power-limited systems with real-time constraints. In: Proceedings of 43rd IEEE Conference on Decision and Control, pp. 1472–1477.
- Monma, C.L., Schrijver, A., Todd, M.J., Wei, V.K., 1990. Convex resource allocation problems on directed acyclic graphs: Duality, complexity, special cases, and extensions. Mathematics of Operations Research 15 (4), 736–748.
- Nowicki, E., 1993. An approximation algorithm for the m -machine permutation flow shop scheduling problem with controllable processing times. European Journal of Operational Research 70, 342–349.
- Nowicki, E., Zdrzalka, S., 1988. A two-machine flow shop scheduling problem with controllable job processing times. European Journal of Operational Research 34 (2), 208–220.
- Nowicki, E., Zdrzalka, S., 1990. A survey of results for sequencing problems with controllable processing times. Discrete Applied Mathematics 26, 271–287.
- Pepyne, D.L., Cassandras, C.G., 1998. Modeling, analysis, and optimal control of a class of hybrid systems. Discrete Event Dynamic Systems: Theory and Applications 8 (2), 175–201.
- Pepyne, D.L., Cassandras, C.G., 2000. Optimal control of hybrid systems in manufacturing. Proceedings of the IEEE 88 (7), 1108–1123.
- Pinedo, M., 2002. Scheduling: Theory, Algorithms, and Systems. Prentice-Hall.
- Powell, M.J.D., 1970. A fortran subroutine for solving systems of nonlinear algebraic equations. Numerical Methods for Nonlinear Algebraic Equations.
- Shabtay, D., Steiner, G., 2007. A survey of scheduling with controllable processing times. Discrete Applied Mathematics 155, 1643–1666.
- Vickson, R.G., 1980. Choosing the job sequence and processing times to minimize processing plus flow cost on a single machine. Operations Research 28 (5), 1155–1167.
- Wardi, Y., Cassandras, C.G., Pepyne, D.L., 2001. A backward algorithm for computing optimal controls for single-stage hybrid manufacturing systems. International Journal of Production Research 39 (2), 369–393.
- Zhang, P., Cassandras, C.G., 2002. An improved forward algorithm for optimal control of a class of hybrid systems. IEEE Transactions on Automatic Control 47 (10), 1735–1739.