

Service Time Optimization of Mixed-Line Flow Shop Systems

Kagan Gokbayrak and Omer Selvi

Abstract—We consider deterministic mixed-line flow shop systems that are composed of controllable and uncontrollable machines. Arrival times and completion deadlines of jobs are assumed to be known, and they are processed in the order they arrive at the machines. We model these flow shops as serial networks of queues operating under a non-preemptive first-come-first-served policy, and employ max-plus algebra to characterize the system dynamics. Defining completion-time costs for jobs and service costs at controllable machines, a non-convex optimization problem is formulated where the control variables are the constrained service times at the controllable machines. In order to simplify this optimization problem, under some cost assumptions, we show that no waiting is observed on the optimal sample path at the downstream of the first controllable machine. We also present a method to decompose the optimization problem into convex subproblems. A solution algorithm utilizing these findings is proposed, and a numerical study is presented to evaluate the performance improvement due to this algorithm.

Index Terms—Controllable service times, manufacturing, optimal control, queueing systems.

I. INTRODUCTION

WE consider a flow shop system consisting of M machines that are processing N identical jobs. The system consists of both *controllable* machines where the service times are adjustable for each process and *uncontrollable* machines with fixed service times. Based on completion-time costs for jobs and service costs at the controllable machines, an optimization problem is formulated where the control variables are the service times at the controllable machines. Since faster services increase wear, tear, and the energy consumption at the machines, and may raise the need for extra supervision, we assume that service costs are decreasing in service times. Slower services, on the other hand, not only build up inventory increasing inventory costs (a form of completion-time cost) but also may delay completion times resulting with missed deadlines. This trade-off is what makes the problem challenging, and our objective in this study is to determine the cost minimizing service times.

Scheduling problems of flow shops with controllable service times consider the job sequencing at each machine along with the service time optimization. The job sequencing problems of

flow shops are known to be NP-hard even for the case of fixed service times (see in [1]). Therefore, the literature is limited to heuristics and approximate solution methods: Nowicki and Zdrzalka, in [2], were the first to analyze flow shop systems with controllable service times. They studied the problem of minimizing the maximum completion-time cost plus the total service cost in a two machine flow shop system. Assuming that the service cost on each machine was a decreasing linear function of the service times, an approximation algorithm was proposed. In [3], Nowicki considered permutation flow shops in which the job sequences were restricted to be identical on each machine, and extended the approximation algorithm to apply at flow shops of more than two machines. For further references, a literature survey on scheduling with controllable service times can be found in [4]. In this paper, we do not consider the job sequencing problem. Instead, we assume that jobs are served in the order they arrive at machines, i.e., the machines operate on a non-preemptive first-come-first-served policy.

The idea of modeling production systems via max-plus algebra and applying control theory for optimization first appeared in [5] where job release times to a single machine system were controlled to minimize the discrepancy between job completion times and desired due dates. Following this work, service time control problems for CNC (Computer Numerical Control) machines, where the service times could be adjusted between processes, were considered. Pepyne and Cassandras, in [6], formulated an 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. In [7], Pepyne and Cassandras extended their results to jobs with completion deadlines penalizing both earliness and tardiness. In [8], the task of solving these problems was simplified by exploiting structural properties of the optimal sample path, and it was shown that, despite the fact that the objective function was non-convex and non-differentiable, the optimal sample path was unique. 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 non-convex and non-differentiable optimization problem into sets of smaller convex optimization problems with linear constraints were presented in [9] and [10], respectively. The “forward-in-time” algorithm presented in [10] was then improved by Zhang and Cassandras in [11].

Gokbayrak and Selvi, in [12], studied a two-machine flow shop system with regular costs 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

Manuscript received October 08, 2008; revised March 15, 2009. First published January 12, 2010; current version published February 10, 2010. Recommended by Associate Editor I. Paschalidis.

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.2009.2037273

sample path that enabled the transformation of the non-smooth discrete-event optimal control problem into a simple convex programming problem. In [13], Gokbayrak and Selvi extended the no-waiting property to multimachine flow shop systems. Employing this property, simpler equivalent convex programming formulations were presented and a forward-in-time solution algorithm was developed under strict convexity assumptions on service and completion-time costs. In [14] and [15], Gokbayrak and Selvi considered the problem in [13] with the additional constraint that the service times at machines were set initially, and could not be altered between processes. For the resulting service time optimization problem of flow shops of traditional (non-CNC) machines, alternative solution methods based on convex programming and subgradient descent methods were presented.

Parallel to the work by Gokbayrak and Selvi, Mao *et al.*, in [16], considered an optimization problem for a single machine system based only on service costs. Instead of defining a completion-time cost as in [13], they introduced completion deadline constraints. For decreasing convex service costs, it was shown that the optimal solution characteristics led to the highly efficient Critical Task Decomposition Algorithm (CTDA). Employing CTDA, they extended their work to multimachine systems in [17] and [18] to obtain an iterative Virtual Deadline Algorithm (VDA). The main idea of this algorithm was to introduce *virtual* deadlines at each machine except the last one so that the flow shop could be decomposed into single machine systems where CTDA could be applied. Determination of these deadlines was performed iteratively and the convergence of VDA was shown.

In this paper, we extend our work in [13] by introducing uncontrollable machines in the flow shop system and completion deadline constraints in the optimization problem. Following the same line of thought in [13], we first formulate a non-convex and non-differentiable optimization problem with max-plus algebra. Employing the standard method of linearization, an equivalent convex optimization problem formulation is also presented. Utilizing both formulations, we generalize the no-waiting property to mixed-line flow shop systems. This property enables the simplification of the non-convex and non-differentiable problem. Then, we introduce a partitioning for the set of jobs and show that the optimization problem can be solved by decomposing it into convex subproblems, one for each part of the partition. An algorithm that forms the partition and obtains the optimal solution is presented.

The rest of the paper is organized as follows: In Section II, we formulate a non-convex and non-differentiable optimization problem and obtain a convex programming formulation by the standard method of linearization. In Section III, we derive a set of waiting characteristics of such systems and show that, on the optimal sample path, no waiting is observed at the downstream of the first controllable machine. The simplified version of the non-convex problem is also presented in this section. In Section IV, we introduce a partitioning for the set of jobs, and show that the optimal solution can be obtained by solving convex subproblems for each part in this partition. A forward decomposition algorithm is also presented in this section that forms the required partition and obtains the optimal solution. In Section V, a

numerical study is presented to demonstrate the benefits gained through a set of example systems. A performance comparison with VDA is also presented in this section. Finally, Section VI concludes the paper.

II. PROBLEM FORMULATION

Let us consider an M -machine flow shop system. The system consists of both *controllable* machines where the service times can be adjusted before each process and *uncontrollable* machines where the service times are fixed, hence it is called as a *mixed-line* system. We define the sets I_C and I_U , disjoint subsets of the set $I = \{1, \dots, M\}$, as the index sets of the controllable and uncontrollable machines, respectively.

A sequence of N identical jobs arrive at the system at known times $0 \leq a_1 \leq a_2 \leq \dots \leq a_N$ and are processed at all machines sequentially. We denote these jobs and their completion deadlines by $\{C_i\}_{i=1}^N$ and $\{d_i\}_{i=1}^N$, respectively. Machines process these jobs one at a time on a first-come-first-served non-preemptive basis. The durations of these processes at each machine j are denoted by the service times $\{s_{i,j}\}_{i=1}^N$. Due to physical limitations of the machines, we assume that each job at any controllable machine $j \in I_C$ needs at least a service of S_j duration. There is no upper bound on the service times. The service times at the uncontrollable machines $j \in I_U$ are fixed to values s_j .

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

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

subject to

$$x_{i,j} = \max(x_{i,j-1}, x_{i-1,j}) + s_{i,j} \quad \forall j \in I \quad (2)$$

$$x_{i,M} \leq d_i \quad (3)$$

$$s_{i,j} \geq S_j \quad \forall j \in I_C \quad (4)$$

$$s_{i,j} = s_j \quad \forall j \in I_U \quad (5)$$

$$x_{i,0} = a_i, \quad x_{0,j} = -\infty \quad \forall j \in I \quad (6)$$

for all $i = 1, \dots, N$. In this formulation, $x_{i,j}$ denotes the departure time of job C_i from machine j , θ_j denotes the service cost for some job processed at machine j , and ϕ_i denotes the completion-time cost for job C_i .

We assume that a feasible solution exists for P . If not, a binary integer programming problem can be formulated as in [19] to reject some of the jobs for feasibility. The job admission problem is a subject of ongoing research, and it is not considered here.

The following standing assumptions are necessary to make the problem somewhat more tractable while preserving the originality of the problem.

Assumption 1: $\theta_j(\cdot)$, for all $j \in I_C$, is continuously differentiable, monotonically decreasing, and strictly convex.

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

Note that for the costs satisfying these assumptions, longer services will decrease the service costs, while possibly increasing the completion times, hence the completion-time costs.

Due to the *max* functions in the constraints, P is non-convex and non-differentiable. Linearizing these constraints, we can formulate the following convex optimization problem:

$$\bar{P} : \min \left\{ \bar{J} = \sum_{i=1}^N \sum_{j \in I_C} \theta_j(s_{i,j}) + \sum_{i=1}^N \phi_i(x_{i,M}) \right\}$$

subject to

$$x_{i,j} \geq x_{i,j-1} + s_{i,j} \quad \forall j \in I \quad (7)$$

$$x_{i,j} \geq x_{i-1,j} + s_{i,j} \quad \forall j \in I \quad (8)$$

$$x_{i,M} \leq d_i \quad (9)$$

$$s_{i,j} \geq S_j \quad \forall j \in I_C \quad (11)$$

$$s_{i,j} = s_j \quad \forall j \in I_U \quad (11)$$

$$x_{i,0} = a_i, \quad x_{0,j} = -\infty \quad \forall j \in I \quad (12)$$

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

As in [13], due to the standing assumptions, the optimal solution to the convex problem \bar{P} includes the optimal service times for P . Hence, solving \bar{P} suffices to determine the optimal service times. For large M and N values, however, convex problem solvers have high requirements on the computing resources to solve \bar{P} , a fact that motivated the work in this paper.

In the next section, we show that no waiting is observed after the first controllable machine, a property that enables the simplification of the optimization problem P . Linearization will be applied to the simplified version to obtain a convex problem formulation with fewer constraints.

III. WAITING CHARACTERISTICS OF THE OPTIMAL SOLUTION

The flow shop can be decomposed into controllable machines and uncontrollable portions formed of sequentially located uncontrollable machines defined as follows:

Definition 1: Machines $\{u, \dots, v\}$ form an uncontrollable portion if

- 1) Machine $(u - 1)$, if exists, is a controllable machine.
- 2) Machines $\{u, \dots, v\}$ are uncontrollable.
- 3) Machine $(v + 1)$, if exists, is a controllable machine.

The jobs, on the other hand, can be decomposed into blocks according to their waiting characteristics at some machine j .

Definition 2: For a given solution $\{s, x\}$, a contiguous set of jobs $\{C_i\}_{i=k}^n$ is said to form a *block* at machine 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 $i = k + 1, \dots, n$.

Note that each job that does not wait at machine j starts a new block.

We state some previously established optimal solution characteristics for uncontrollable portions and controllable machines in the following subsections, and employ them to present the no-waiting property for the mixed-line flow shops.

A. Uncontrollable Portions

Uncontrollable portions can be treated as fixed-service-time flow shop systems previously studied in [14] and [15]; hence, the results therein are applicable. An important result that we

borrow is that waiting can only be observed at local bottleneck machines defined as follows:

Definition 3: Let machines $\{u, \dots, v\}$ form an uncontrollable portion. A machine $t > u$ in this uncontrollable portion is a *local bottleneck* if its service time exceeds the service times of all upstream machines in the uncontrollable portion, i.e., $s_t > \max_{j=u, \dots, t-1} s_j$. Machine u is also defined to be a local bottleneck.

The arrival time of job C_i at the uncontrollable portion formed of machines $\{u, \dots, v\}$ is given as $x_{i,u-1}$. We borrow the following results that employ the interarrival times to determine the block structure at a local bottleneck machine:

It follows from Lemma 5 in [15] that if job C_{k+1} resides in a block started by job C_k at a local bottleneck machine t , then

$$x_{k+1,u-1} - x_{k,u-1} < s_t. \quad (13)$$

Another result that we borrow from [15] is that a necessary condition for C_{n+1} not to wait at a local bottleneck machine t is

$$x_{n+1,u-1} - x_{n,u-1} \geq s_t \quad (14)$$

which follows from Lemma 3 in [15].

We also borrow the following lemma on the departure times of jobs from uncontrollable machines:

Lemma 1: (Lemma 3 in [14]) The departure time of job C_i from machine v within the uncontrollable portion started by the machine u is given by

$$x_{i,v} = \max \left(x_{i,u-1} + \sum_{j=u}^v s_j, x_{i-1,v} + \max_{j=u, \dots, v} s_j \right) \quad (15)$$

where $x_{i,0} = x_{0,j} = -\infty$ for all i and j .

If a job does not wait at machines $\{u, \dots, v\}$, then the first term in (15) dominates. All jobs that experience waiting at one of these machines definitely wait at the machine with the maximum service time $\max_{j=u, \dots, v} s_j$. This machine prevents waiting in its downstream up to machine v , causing the second term in (15) to dominate. (The details can be found in [14].)

Note that the results that we borrow from [14] and [15] present characteristics of fixed-service-time systems that hold for any cost structure; therefore, they are also applicable to our optimization problem where uncontrollable machine service costs are not considered.

We end this subsection by showing that no waiting is observed at uncontrollable portions preceded by controllable machines.

Lemma 2: Let machines $\{u, \dots, v\}$ form an uncontrollable portion and machine $(u - 1)$ be controllable. On the optimal sample path, jobs do not wait at these uncontrollable machines.

Proof: (By Induction) Since the first job does not wait for service at any machine, we have the basis for induction. For a contradiction in the inductive step, let us assume that, on the optimal sample path, jobs $\{C_i\}_{i=1}^k$ do not wait while job C_{k+1} waits for service within the uncontrollable portion. Let m be the most upstream machine that C_{k+1} waits at in the uncontrollable portion. Let jobs $\{C_i\}_{i=k}^n$ form the block at machine m in which

job C_{k+1} resides. Since machine m has to be a local bottleneck, by Lemma 1, we have, for $i = k + 1, \dots, n$

$$x_{i,u-1}^* + \sum_{j=u}^m s_j < x_{i-1,m}^* + s_m. \quad (16)$$

For these jobs $\{C_i\}_{k+1}^n$, let us define the positive differences $\Delta_i^{u,m}$ as

$$\Delta_i^{u,m} = x_{i-1,m}^* - x_{i,u-1}^* - \sum_{j=u}^{m-1} s_j > 0. \quad (17)$$

Since job C_{k+1} resides in the block started by C_k at machine m , from (2) and (13), we have

$$s_{k+1,u-1}^* \leq x_{k+1,u-1}^* - x_{k,u-1}^* < s_m. \quad (18)$$

We analyze two cases:

Case 1: Job C_{n+1} finds machine $(u-1)$ busy, i.e., $x_{n+1,u-2}^* \leq x_{n,u-1}^*$

Since job C_{n+1} starts a new block at machine m , from the case statement, (2), and (14), we have

$$x_{n+1,u-1}^* - x_{n,u-1}^* = s_{n+1,u-1}^* \geq s_m. \quad (19)$$

From (18) and (19)

$$s_{n+1,u-1}^* > s_{k+1,u-1}^*. \quad (20)$$

Let us define the perturbed service times as

$$\hat{s}_{i,j} = \begin{cases} s_{k+1,u-1}^* + \Delta & i = k+1, j = u-1 \\ s_{n+1,u-1}^* - \Delta & i = n+1, j = u-1 \\ s_{i,j}^* & \text{otherwise} \end{cases} \quad (21)$$

where

$$\Delta = \frac{1}{2} \min \left\{ s_{n+1,u-1}^* - s_{k+1,u-1}^*, \min_{i=k+1, \dots, n} \Delta_i^{u,m} \right\}.$$

Note that Δ is positive due to (17) and (20).

We can simply state that the departure times $\hat{x}_{i,j}$ resulting from the application of the perturbed service times $\hat{s}_{i,j}$ satisfy $\hat{x}_{i,m} = x_{i,m}^*$ for all $i = 1, \dots, k$. Similarly, the service time perturbation at machine $(u-1)$ does not affect the departure times from upstream machines, hence, for example, we have $\hat{x}_{i,u-2} = x_{i,u-2}^*$ for all $i = 1, \dots, N$.

The service time perturbation for C_{k+1} causes

$$\hat{x}_{i,u-1} \leq x_{i,u-1}^* + \Delta \quad (22)$$

for $i = k+1, \dots, n$. Consequently, by Lemma 1 and the definition of Δ , we have

$$\begin{aligned} \hat{x}_{k+1,m} &= \max \left(\hat{x}_{k+1,u-1} + \sum_{j=u}^m s_j, x_{k,m}^* + s_m \right) \\ &= x_{k+1,m}^*. \end{aligned}$$

Applying the same argument recursively, we obtain $\hat{x}_{i,m} = x_{i,m}^*$ for all $i = k+1, \dots, n$.

From (21) and (22), the perturbed departure time of job C_{n+1} from machine $(u-1)$ is written as

$$\begin{aligned} \hat{x}_{n+1,u-1} &\leq \max(x_{n+1,u-2}^*, x_{n,u-1}^* + \Delta) + (s_{n+1,u-1}^* - \Delta) \\ &\leq x_{n+1,u-1}^*. \end{aligned}$$

Therefore, we can write

$$\begin{aligned} \hat{x}_{n+1,m} &\leq \max \left(x_{n+1,u-1}^* + \sum_{j=u}^m s_j, x_{n,m}^* + s_m \right) \\ &\leq x_{n+1,m}^* \end{aligned}$$

resulting with a possible decrease in the completion times for jobs $\{C_i\}_{i=n+1}^N$ due to the perturbation in (21).

As a result, we can state that $\hat{x}_{i,M} \leq x_{i,M}^*$ for all jobs, possibly lowering the completion-time costs due to Assumption 2. Similarly, from (20), (21), and by Assumption 1, the perturbed solution has a lower service cost. Hence, the perturbed solution yields a cost lower than the optimal cost resulting with a contradiction.

Case 2: Job C_{n+1} finds machine $(u-1)$ idle, i.e., $x_{n+1,u-2}^* > x_{n,u-1}^*$

Let us define the perturbed service times as

$$\hat{s}_{i,j} = \begin{cases} s_{k+1,u-1}^* + \Delta & i = k+1, j = u-1 \\ s_{i,j}^* & \text{otherwise} \end{cases} \quad (23)$$

where

$$\Delta = \frac{1}{2} \min \left\{ x_{n+1,u-2}^* - x_{n,u-1}^*, \min_{i=k+1, \dots, n} \Delta_i^{u,m} \right\}.$$

Note that Δ is positive due to (17) and this case's statement.

Following the same steps in Case 1, we can obtain $\hat{x}_{i,M} = x_{i,M}^*$ for all jobs. From Assumption 1 and (23), the perturbed solution has a lower service cost. Hence, the perturbed solution yields a cost lower than the optimal cost resulting with a contradiction. ■

B. Controllable Machines

Since the optimal solution of the convex problem \bar{P} includes the optimal service times for P , we apply calculus of variations techniques on \bar{P} to determine optimal service time characteristics for P .

Let us start with introducing Lagrangian multipliers $\lambda_{i,j}$, $v_{i,j}$, $\mu_{i,j}$, and γ_i for all i and j to form the augmented cost

$$\begin{aligned} \bar{J}_A = & \sum_{i=1}^N \sum_{j \in I_C} \theta_j(s_{i,j}) + \sum_{i=1}^N \phi_i(x_{i,M}) \\ & + \sum_{i=1}^N \sum_{j \in I} \lambda_{i,j}(x_{i,j-1} + s_{i,j} - x_{i,j}) \\ & + \sum_{i=1}^N \sum_{j \in I} v_{i,j}(x_{i-1,j} + s_{i,j} - x_{i,j}) \\ & + \sum_{i=1}^N \sum_{j \in I_C} \mu_{i,j}(S_j - s_{i,j}) + \sum_{i=1}^N \gamma_i(x_{i,M} - d_i). \end{aligned}$$

The co-state equations can be stated as

$$\mu_{i,j}^* - \lambda_{i,j}^* - v_{i,j}^* = \theta'_j(s_{i,j}^*) \quad \forall i, \forall j \in I_C \quad (24)$$

$$\lambda_{i,j}^* + v_{i,j}^* - v_{i+1,j}^* = \lambda_{i,j+1}^* \quad \forall i < N, \forall j < M \quad (25)$$

$$\lambda_{i,M}^* + v_{i,M}^* - v_{i+1,M}^* = \phi'_i(x_{i,M}^*) + \gamma_i^* \quad \forall i < N \quad (26)$$

$$\lambda_{N,j}^* + v_{N,j}^* = \lambda_{N,j+1}^* \quad \forall j < M \quad (27)$$

$$\lambda_{N,M}^* + v_{N,M}^* - \gamma_N^* = \phi'_N(x_{N,M}^*) \quad (28)$$

and

$$\lambda_{i,j}^* (x_{i,j-1}^* + s_{i,j}^* - x_{i,j}^*) = 0 \quad \forall i, j \quad (29)$$

$$v_{i,j}^* (x_{i-1,j}^* + s_{i,j}^* - x_{i,j}^*) = 0 \quad \forall i, j \quad (30)$$

$$\mu_{i,j}^* (S_j - s_{i,j}^*) = 0 \quad \forall i, \forall j \in I_C \quad (31)$$

$$\gamma_i^* (x_{i,M}^* - d_i) = 0 \quad \forall i \quad (32)$$

$$\lambda_{i,j}^* \geq 0 \quad \forall i, j \quad (33)$$

$$v_{i,j}^* \geq 0 \quad \forall i, j \quad (34)$$

$$\mu_{i,j}^* \geq 0 \quad \forall i, \forall j \in I_C \quad (35)$$

$$\gamma_i^* \geq 0 \quad \forall i. \quad (36)$$

Employing these optimality conditions, we can prove the following lemma that establishes the monotonicity property of the optimal service times at controllable machines.

Lemma 3: (Monotonicity Property) Let machine j be a controllable machine, i.e., $j \in I_C$. Then, for some $i \in \{2, \dots, N\}$, if jobs C_{i-1} and C_i are in the same block of the j th machine on the optimal sample path, then the optimal service times satisfy $s_{i,j}^* \geq s_{i-1,j}^*$.

Proof: (By contradiction) Let us assume that jobs C_{i-1} and C_i are in the same block of the j th machine on the optimal sample path, and $s_{i,j}^* < s_{i-1,j}^*$. From (10), there are two possible cases:

Case 1: $s_{i-1,j}^* > s_{i,j}^* = S_j$: From (31), $\mu_{i-1,j}^* = 0$ and from (35), $\mu_{i,j}^* \geq 0$.

Case 2: $s_{i-1,j}^* > s_{i,j}^* > S_j$: From (31), $\mu_{i-1,j}^* = \mu_{i,j}^* = 0$.

From both cases, we get

$$\mu_{i,j}^* - \mu_{i-1,j}^* \geq 0. \quad (37)$$

Since jobs C_{i-1} and C_i are in the same block of the j th machine on the optimal sample path, from (7) and (8), we have

$$x_{i,j}^* > x_{i,j-1}^* + s_{i,j}^*$$

so, from (29)

$$\lambda_{i,j}^* = 0. \quad (38)$$

From (25) and (33), we have

$$\lambda_{i-1,j}^* + v_{i-1,j}^* - v_{i,j}^* = \lambda_{i-1,j+1}^* \geq 0 \quad (39)$$

for $j < M$. Similarly, from (26), (36), and by Assumption 2, we have

$$\lambda_{i-1,M}^* + v_{i-1,M}^* - v_{i,M}^* = \phi'_{i-1}(x_{i-1,M}^*) + \gamma_{i-1}^* > 0. \quad (40)$$

It follows from (24), (37), (38), (39), and (40) that:

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

for all $j \in I_C$, which contradicts Assumption 1. Hence, within a block, the optimal service times are non-decreasing in the job index. ■

Lemmas 1–3 will be employed while proving the next theorem, which shows that a controllable machine prevents buffering in the closest downstream controllable machine.

Theorem 1: Let machines h and t , where $h < t$, be two consecutive controllable machines, possibly separated by the uncontrollable portion $\{h+1, \dots, t-1\}$. On the optimal sample path, no waiting is observed at machine t .

Proof: We prove by induction on jobs:

Basis Step: The first job does not wait at any machine.

Inductive Step: Let us assume, for a contradiction, that jobs $\{C_i\}_{i=1}^k$ do not wait at machine t and that jobs $\{C_i\}_{i=k}^n$ form a block at machine t on the optimal sample path so that

$$\Delta_b = \min_{i=k+1, \dots, n} \{x_{i-1,t}^* - x_{i,t-1}^*\} > 0 \quad (41)$$

is satisfied by the block definition.

We denote the total service time for the uncontrollable machines in between h and t by s_{unc} defined as

$$s_{unc} = \begin{cases} \sum_{j=h+1}^{t-1} s_j & t > h+1 \\ 0 & t = h+1 \end{cases}$$

and the maximum service time for this uncontrollable portion is given by

$$s_m = \begin{cases} \max_{j=h+1, \dots, t-1} s_j & t > h+1 \\ 0 & t = h+1. \end{cases}$$

Since jobs C_k and C_n reside in the same block at the controllable machine t , according to Lemma 3

$$s_{n,t}^* \geq s_{k,t}^*. \quad (42)$$

From Lemma 2 and the block definition, we have

$$\begin{aligned} x_{k+1,t-1}^* &= x_{k+1,h}^* + s_{unc} \\ &< x_{k,t}^* = x_{k,h}^* + s_{unc} + s_{k,t}^*. \end{aligned} \quad (43)$$

Since job C_{k+1} does not wait at machine m , from (14), (42), and (43), we obtain

$$s_{n,t}^* \geq s_{k,t}^* > x_{k+1,h}^* - x_{k,h}^* \geq s_m \quad (44)$$

resulting with, from the block definition, (2), and (44)

$$x_{n+1,t-1}^* \geq x_{n,t}^* \geq x_{n,t-1}^* + s_{n,t}^* > x_{n,t-1}^* + s_m. \quad (45)$$

There are two cases to consider:

Case 1: Idleness is observed at machine h after job C_n departs, i.e., $x_{n+1,h-1}^* > x_{n,h}^*$.

We consider the non-optimal solution s defined as

$$s_{i,j} = \begin{cases} s_{k+1,h}^* + \Delta & i = k+1, j = h \\ s_{i,j}^* & \text{otherwise} \end{cases} \quad (46)$$

for a very small Δ evaluated as

$$\Delta = \frac{1}{2} \min \{ x_{n+1,h-1}^* - x_{n,h}^*, \Delta_b, x_{n+1,t-1}^* - x_{n,t-1}^* - s_m \}$$

and show that the departure times resulting from the application of the non-optimal service times $s_{i,j}$ satisfy $x_{i,t} = x_{i,t}^*$ for $i = 1, \dots, N$. Note that Δ is positive by (41), (45), and the statement of the case.

The perturbation in the service time does not affect jobs $\{C_i\}_{i=1}^k$; therefore, we already have $x_{i,t} = x_{i,t}^*$ for all $i = 1, \dots, k$. Similarly, this perturbation does not affect the departure times from the machines upstream to machine h .

Job C_{k+1} does not wait at the uncontrollable portion on the optimal sample path. Increasing its service time at machine h does not change this fact; hence, we have

$$\begin{aligned} x_{k+1,t-1} &= x_{k+1,h-1}^* + s_{k+1,h}^* + \Delta + s_{unc} \\ &= x_{k+1,t-1}^* + \Delta. \end{aligned}$$

Since $\Delta < x_{k,t}^* - x_{k+1,t-1}^*$, job C_{k+1} waits for service at machine t also with this non-optimal solution. Hence

$$\begin{aligned} x_{k+1,t} &= \max(x_{k+1,t-1}^* + \Delta, x_{k,t}^*) + s_{k+1,t}^* \\ &= x_{k,t}^* + s_{k+1,t}^* = x_{k+1,t}^*. \end{aligned} \quad (47)$$

Similarly, jobs $\{C_i\}_{i=k+2}^n$ will be delayed at most by Δ , resulting with

$$x_{i,t-1} \leq x_{i,t-1}^* + \Delta \quad (48)$$

for all $i = k+2, \dots, n$. Following the same argument above, $x_{i,t} = x_{i,t}^*$ for all $i = k+1, \dots, n$.

Δ value is selected small enough not to alter the service starting and departure times of C_{n+1} at machine h ; hence, $x_{n+1,h} = x_{n+1,h}^*$. Since the departure time of job C_n from machine h may have a delay of at most Δ , i.e., $x_{n,h} \leq x_{n,h}^* + \Delta$,

we can not claim that job C_{n+1} does not wait at the uncontrollable portion. Instead, employing Lemma 1, Δ definition, and (48), and observing that $x_{n+1,t-1}^* = x_{n+1,h}^* + s_{unc}$ from Lemma 2, we obtain

$$\begin{aligned} x_{n+1,t-1} &= \max(x_{n+1,h}^* + s_{unc}, x_{n,t-1} + s_m) \\ &= \max(x_{n+1,t-1}^*, x_{n,t-1} + s_m) \\ &= x_{n+1,t-1}^*. \end{aligned}$$

Since $x_{n,t} = x_{n,t}^*$ and $x_{n+1,t-1} = x_{n+1,t-1}^*$, from (2) and (46) we get $x_{n+1,t} = x_{n+1,t}^*$. Following a similar argument recursively, we get $x_{i,t} = x_{i,t}^*$ for all $i = n+1, \dots, N$.

Since $x_{i,t} = x_{i,t}^*$ for all $i = 1, \dots, N$ and since the optimal service times are applied to all jobs at machines downstream to machine t , we have $x_{i,M} = x_{i,M}^*$ for all $i = 1, \dots, N$. Hence, from Assumption 1 and (46)

$$J(s) - J(s^*) = \theta_h (s_{k+1,h}^* + \Delta) - \theta_h (s_{k+1,h}^*) < 0$$

contradicting the optimality of s^* .

Case 2: No idleness is observed at machine h after job C_n departs, i.e., $x_{n+1,h-1}^* \leq x_{n,h}^*$.

It follows from (44) and $x_{k+1,h}^* \geq x_{k,h}^* + s_{k+1,h}^*$ that

$$s_{k,t}^* > s_{k+1,h}^*. \quad (49)$$

By this case's statement, we have

$$x_{n+1,h}^* = x_{n,h}^* + s_{n+1,h}^*. \quad (50)$$

Since job C_n is the last job in a block and C_{n+1} is the first job in the next block at machine t , from (50) and by Lemma 2, we have

$$x_{n+1,t-1}^* = x_{n,h}^* + s_{n+1,h}^* + s_{unc} \geq x_{n,t}^* > x_{n,h}^* + s_{unc} + s_{n,t}^*$$

resulting with

$$s_{n+1,h}^* > s_{n,t}^*. \quad (51)$$

From (42), (49), and (51), we conclude that

$$s_{n+1,h}^* > s_{k+1,h}^*. \quad (52)$$

Now, we consider the non-optimal solution s defined as

$$s_{i,j} = \begin{cases} s_{k+1,h}^* + \Delta & i = k+1, j = h \\ s_{n+1,h}^* - \Delta & i = n+1, j = h \\ s_{i,j}^* & \text{otherwise} \end{cases} \quad (53)$$

for a very small Δ defined as

$$\Delta = \frac{1}{2} \min \{ s_{n+1,h}^* - s_{k+1,h}^*, \Delta_b, x_{n+1,t-1}^* - x_{n,t-1}^* - s_m \}.$$

Note that Δ is positive from (41), (45), and (52).

Following the same reasoning in Case 1, we can claim that $x_{i,M} = x_{i,M}^*$ for all $i = 1, \dots, n$. Moreover, since the service time perturbation occurs at machine h , the upstream departure times are not affected. Hence, for example, $x_{i,h-1} = x_{i,h-1}^*$ for all $i = 1, \dots, N$.

Since $x_{n,h}^* \leq x_{n,h} \leq x_{n,h}^* + \Delta$, from the statement of the case and (53), we get

$$\begin{aligned} x_{n+1,h} &= \max(x_{n+1,h-1}^*, x_{n,h}) + s_{n+1,h} - \Delta \\ &\leq x_{n,h}^* + s_{n+1,h}^* \leq x_{n+1,h}^*. \end{aligned}$$

The optimal service times are applied to job C_{n+1} at machines downstream to machine h and to jobs $\{C_i\}_{i=n+2}^N$ at all machines, so we can conclude that $x_{i,M} \leq x_{i,M}^*$ for all $i = n+1, \dots, N$. Hence, from (52), (53), and by Assumptions 1 and 2

$$\begin{aligned} J(s) - J(s^*) &= (\theta_h (s_{k+1,h}^* + \Delta) - \theta_h (s_{k+1,h}^*)) \\ &\quad + (\theta_h (s_{n+1,h}^* - \Delta) - \theta_h (s_{n+1,h}^*)) \\ &\quad + \sum_{i=n+1}^N (\phi_i(x_{i,M}) - \phi_i(x_{i,M}^*)) \\ &< 0 \end{aligned}$$

contradicting the optimality of s^* .

From these two cases, we conclude that no waiting is observed at the downstream controllable machine t . ■

By Lemma 2 and Theorem 1, we obtain the following result:

Corollary 1: On the optimal sample path, no waiting is observed after the first controllable machine.

This corollary extends the no-waiting property in [13] to mixed-line flow shop systems. Next, we employ this result to simplify the P formulation.

C. Simplified Problem

In the rest of the paper, we assume, for convenience, that the most upstream machine is a controllable machine, i.e., $\{1\} \subset I_C$. The flow shop systems that start out with a sequence of uncontrollable machines can easily be reduced to our setting by calculating the arrival times at the first controllable machine via (2) and (6), and removing the uncontrollable machines upstream to the first controllable machine from the optimal control problem formulation P .

Employing Corollary 1 in the P formulation, we obtain

$$\hat{P} : \min \left\{ J = \sum_{i=1}^N \sum_{j \in I_C} \theta_j(s_{i,j}) + \sum_{i=1}^N \phi_i \left(x_{i,1} + \sum_{k=2}^M s_{i,k} \right) \right\}$$

subject to

$$x_{1,1} = a_1 + s_{1,1} \quad (54)$$

$$x_{i,1} = \max(a_i, x_{i-1,1}) + s_{i,1} \quad (55)$$

$$x_{i,1} \geq x_{i-1,1} + s_{i-1,2} \quad (56)$$

$$x_{i,1} \geq x_{i-1,1} + s_{i-1,j} + \sum_{k=2}^{j-1} (s_{i-1,k} - s_{i,k}) \quad (57)$$

for all $i > 1$ and for all $j > 2$, and

$$x_{i,1} \leq d_i - \sum_{k=2}^M s_{i,k} \quad (58)$$

$$s_{i,j} \geq S_j \quad \forall j \in I_C \quad (59)$$

$$s_{i,j} = s_j \quad \forall j \in I_U \quad (60)$$

for all i .

Analyzing the constraints of \hat{P} , we determine that the coupling between consecutive jobs are through (55)–(57). If we have

$$a_i \geq x_{i-1,1}$$

$$a_i \geq x_{i-1,1} + s_{i-1,2} - s_{i,1}$$

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

inequalities satisfied for all $j > 2$, then coupling is removed between jobs C_i and C_{i-1} . This observation motivates the decomposition method presented next.

IV. PROBLEM DECOMPOSITION

Let us consider an array of service times $s_i = [s_{i,1}, \dots, s_{i,M}]$ for job C_i that is feasible for \hat{P} to define

$$\delta_j(s_i) = \begin{cases} 0 & j = 1 \\ s_{i,2} - S_1 & j = 2 \\ s_{i,j} - S_1 + \sum_{l=2}^{j-1} (s_{i,l} - S_l) & j > 2 \end{cases} \quad (61)$$

and

$$\delta(s_i) = \max_{j=1, \dots, M} \delta_j(s_i) \quad (62)$$

which leads to a partition of the jobs as follows:

Definition 4: A contiguous set of jobs $\{C_i\}_{i=k}^n$ is said to form an *independent period* for the system if

- 1) $x_{k-1,1}^* + \delta(s_{k-1}^*) < a_k$ (for $k > 1$);
- 2) $x_{n,1}^* + \delta(s_n^*) < a_{n+1}$;
- 3) For all $i \in \{k, \dots, n-1\}$, $x_{i,1}^* + \delta(s_i^*) \geq a_{i+1}$.

Definition 5: An *independent period structure* for the system is a partition of jobs $\{C_i\}_{i=1}^N$ into independent periods.

Note that since $\delta_1(s_i^*) = 0$, $\delta(s_i^*)$ is always nonnegative.

The following lemma states that the optimal service time decision for a job depends only on the arrival times and completion deadlines of the jobs residing in the same independent period. The proof is omitted as it is similar to the proof of Lemma 5 in [13].

Lemma 4: Consider a contiguous job sequence $\{C_i\}_{i=k}^n$ forming an independent period on the optimal sample path. The optimal service times for these jobs do not depend on the arrival times $\{a_1, \dots, a_{k-1}, a_{n+1}, \dots, a_N\}$ and the completion deadlines $\{d_1, \dots, d_{k-1}, d_{n+1}, \dots, d_N\}$.

Let us assume that the independent period structure of the optimal solution is known. We can employ Lemma 4 to decompose problem \hat{P} into subproblems one for each independent period $\{C_i\}_{i=k}^n$ obtained simply by substituting C_k for C_1 and C_n for C_N .

From the definition of independent periods, $x_{i,1}^* + \delta(s_i^*) \geq a_{i+1}$ is satisfied for all jobs C_i where $i \in \{k, \dots, n-1\}$. Since $\delta(s_i^*)$ can be positive, it is possible to have $x_{i,1}^* < a_{i+1}$ for some of these jobs. Therefore, the *max* constraint in (55) remains in

the formulation and needs to be linearized for convex problem formulations.

Defining the cost for the independent period formed of jobs $\{C_i\}_k^n$ as

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

the resulting subproblem for the independent period can be formulated as

$$Q(k, n) : \min \{J(k, n)\}$$

subject to

$$x_{k,1} = a_k + s_{k,1} \quad (63)$$

$$x_{i,1} \geq a_i + s_{i,1} \quad (64)$$

$$x_{i,1} \geq x_{i-1,1} + s_{i,1} \quad (65)$$

$$x_{i,1} \geq x_{i-1,1} + s_{i-1,2} \quad (66)$$

$$x_{i,1} \geq x_{i-1,1} + s_{i-1,j} + \sum_{k=2}^{j-1} (s_{i-1,k} - s_{i,k}) \quad (67)$$

for all $i \in \{k+1, \dots, n\}$ and for all $j > 2$, and

$$x_{i,1} \leq d_i - \sum_{k=2}^M s_{i,k} \quad (68)$$

$$s_{i,j} \geq S_j \quad \forall j \in I_C \quad (69)$$

$$s_{i,j} = s_j \quad \forall j \in I_U \quad (70)$$

for all $i \in \{k, \dots, n\}$.

Note that the convex problem $Q(1, N)$, which has $N(M-1)$ fewer variables and constraints compared to \bar{P} , yields the optimal solution for \bar{P} .

So far, we have shown how to obtain the optimal solution when its independent period structure is given. In the next subsection, we develop an algorithm that obtains the optimal solution as it determines independent period structure for the optimal solution.

A. Forward Decomposition Algorithm

We start with replacing Assumption 2 by the following assumption, so that each $Q(k, n)$ problem has a unique optimal solution:

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

We denote the optimal service times for $Q(k, n)$ as $s_{i,j}^*(k, n)$ for $i = k, \dots, n$ and $j = 1, \dots, M$ and the corresponding departure times from the first stage as $x_{i,1}^*(k, n)$. From Lemma 4, if the job sequence $\{C_i\}_{i=k}^n$ forms an independent period on the optimal sample path, then the optimal solution to $Q(k, n)$ satisfies

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

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

Following the same steps in [13], a procedure for identifying the independent period structure of the optimal sample path is formalized in the following theorem.

Theorem 2: Let job C_k initiate an independent period on the optimal sample path. Then, job C_n ends this independent period if and only if the following conditions are satisfied:

- 1) For all $i \in \{k, \dots, n-1\}$, $x_{i,1}^*(k, i) + \delta(s_i^*(k, i)) \geq a_{i+1}$;
- 2) $x_{n,1}^*(k, n) + \delta(s_n^*(k, n)) < a_{n+1}$.

This theorem suggests a forward decomposition algorithm: We assume that all the independent periods before job C_k are identified, hence the optimal service times $s_{i,j}^*$ for all $i = 1, \dots, k-1$ and $j = 1, \dots, M$ are known. Starting with $i = k$ and incrementing the job index i at each iteration, subproblems $Q(k, i)$ are solved until the second condition is satisfied. Once the second condition is satisfied, we not only obtain an independent period $\{C_i\}_{i=k}^n$ on the optimal sample path, but also obtain the optimal service times for these jobs from the solution of $Q(k, n)$.

This forward decomposition algorithm can be given as

Algorithm 1

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

while $n \leq N$ do

Step 2: solve subproblem $Q(k, n)$ and determine $\delta(s_n^*(k, n))$

Step 3:

if $x_{n,1}^*(k, n) + \delta(s_n^*(k, n)) < a_{n+1}$, then

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

$$k = n + 1$$

endif

Step 4: (increment index n)

$$n = n + 1$$

Note that this decomposition algorithm requires only N iterations. However, these iterations are not identical in complexity and depend on the arrival and deadline sequences along with the cost parameters. The best case for this algorithm 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 this algorithm, 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 this case, we solve $Q(1, i)$ for all $i \in \{1, \dots, N\}$. If the expected number of independent periods is small, e.g., for the bulk arrivals case where we have only one independent period, we may choose to solve $Q(1, N)$ directly.

V. NUMERICAL EXAMPLES

We present two numerical examples in this section. In the first example, we demonstrate the benefit of simplifications due to the no-waiting property by analyzing solution times of different

TABLE I
COMPUTATION TIMES FOR \bar{P} , $Q(1, N)$ FORMULATIONS AND FORWARD DECOMPOSITION ALGORITHM (IN SECONDS)

N	M = 20			M = 30			M = 40		
	\bar{P}	Q(1, N)	FDA	\bar{P}	Q(1, N)	FDA	\bar{P}	Q(1, N)	FDA
50	26.8	25.2	49.0	51.3	45.3	68.1	83.7	72.8	88.4
100	82.8	70.8	104.3	175.1	151.0	145.1	-	257.6	188.3
150	173.3	144.9	160.7	-	328.9	224.1	-	582.0	289.2
200	-	246.0	214.4	-	540.1	296.4	-	-	381.5
250	-	376.5	267.7	-	-	370.3	-	-	478.2
300	-	536.6	319.3	-	-	438.9	-	-	570.2
350	-	-	370.3	-	-	506.7	-	-	658.3

problem sizes for \bar{P} and $Q(1, N)$ convex formulations. The improvement due to the Forward Decomposition Algorithm (FDA) is also illustrated in this example. The second example compares the solution performances of our FDA and the Virtual Deadline Algorithm (VDA) in [18] by Mao and Cassandras.

The computing environment for these examples is Matlab (by The Mathworks) running on a computer with 2.0 GHz Intel Core2Duo T7200 processor and 2 GB of RAM. The convex problems are solved using *cvx* (see [20]), a modeling system for convex programming developed at Stanford University.

Example 1: Let us consider the optimization problem P for an M -machine flow shop system processing a set of N jobs. The service cost $\theta_j(s_{i,j})$ for job C_i at the controllable machine j is given as

$$\theta_j(s_{i,j}) = \frac{\beta_j}{s_{i,j}} \quad (71)$$

for some β_j . The completion-time cost for job C_i , on the other hand, is given by a cost defined as

$$\phi_i(x_{i,M}) = 20(x_{i,M} - a_i)^2. \quad (72)$$

Note that the service cost given in (71) is continuously differentiable, monotonically decreasing, and strictly convex satisfying Assumption 1. Similarly, the completion-time cost given in (72) is continuously differentiable, monotonically increasing, and strictly convex for feasible completion times $x_{i,M} \geq a_i$, hence satisfies Assumption 3. Therefore, we expect to see a unique optimal solution.

We study problems with different M and N values. For each M and N setting, we randomly generate ten optimization problems: We randomly create flow shop systems of $M/2$ controllable and $M/2$ uncontrollable machines. The interarrival times for N jobs are realized from an exponential distribution with a mean of 2 units. The lower bounds on the controllable service times S_j and the β_j cost parameters for all $j \in I_C$, the service times s_j for all $j \in I_U$, and the deadlines d_i for all jobs are all randomly assigned.

The average solution times (over ten optimization problems) of the alternative methodologies for different M and N settings are presented in Table I, where a dash sign denotes a crash due to running out of memory. Due to space limitations, the resulting optimal service and departure times are not reported here. However, as expected, no-waiting is observed after the first controllable machine of the system.

Having a smaller number of variables and constraints, $Q(1, N)$ formulation outperforms \bar{P} formulation; the former is not only faster to solve but also enables the solution of larger

problems. Forward Decomposition Algorithm (FDA), on the other hand, outperforms the $Q(1, N)$ solution methodology in terms of solution times for large and uncongested systems where several independent periods are observed. Moreover, due to solving several smaller problems, memory may no longer be an active constraint. For congested or smaller systems, though, solving $Q(1, N)$ should be preferred. For the bulk arrival case, for example, instead of solving just the $Q(1, N)$ problem, FDA solves $Q(1, i)$ for all $i \in \{1, \dots, N\}$, hence, not only it takes longer to obtain the result, but also no memory benefit is observed.

In the next example, we compare the solution performances of our Forward Decomposition Algorithm (FDA) and Virtual Deadline Algorithm (VDA) in [18] by Mao and Cassandras. In order to have VDA applicable, in this example, we study flow shop systems consisting only of controllable machines and with no completion-time costs.

Example 2: We consider flow shop systems, where all machines are controllable, with service costs given in (71) and no completion-time costs. For each M and N combination, ten optimization problems are run with randomly assigned arrival $\{a_i\}_{i=1}^N$ and deadline $\{d_i\}_{i=1}^N$ sequences, $\{\beta_j\}_{j=1}^M$ cost parameters, and $\{S_j\}_{j=1}^M$ lower bounds on service times. The average solution times over ten problems for each setting are presented in Table II.

As seen in Table II, for small values of M , VDA is much faster than FDA. However, as M increases, VDA takes a lot longer to converge compared to FDA, limiting its usage to small flow shops.

VI. CONCLUSION

This paper studied the service time optimization of flow shop systems consisting of both controllable machines, where the service times are bounded below, and uncontrollable machines with fixed service times. The optimization problem revealed a trade-off between selecting faster services to lower completion-time costs (and to meet deadlines) and selecting slower services to lower service costs.

Linearizing the max constraints due to max-plus queueing dynamics, a convex optimization problem was formulated. A set of waiting characteristics of the system was derived and it was shown that no waiting is observed on the optimal sample path after the first controllable machine. Employing this result, a simplified convex optimization formulation was introduced through eliminating N variables and N constraints from the original convex optimization problem at each machine where no waiting is observed. A ‘‘forward-in-time’’ decomposition

TABLE II
COMPUTATION TIMES FOR VIRTUAL DEADLINE ALGORITHM AND FORWARD DECOMPOSITION ALGORITHM (IN SECONDS)

N	M = 10		M = 20		M = 30		M = 40	
	VDA	FDA	VDA	FDA	VDA	FDA	VDA	FDA
50	6.31	27.15	93.71	41.73	440.54	57.54	1363.47	74.53
100	12.32	54.07	189.55	83.39	862.48	115.31	2774.17	148.02
150	18.68	82.79	287.98	127.91	1356.34	177.40	4134.87	228.60
200	24.85	109.44	379.03	168.72	1806.78	238.15	5528.90	301.27
250	31.14	136.14	476.16	209.03	2279.76	294.52	6914.59	375.09
300	37.31	163.43	581.60	249.37	2866.37	352.80	8273.62	451.50
350	43.56	191.93	677.04	291.58	3285.19	412.36	9665.99	529.29
400	49.75	218.73	769.48	334.44	3766.09	474.47	11038.39	607.33

algorithm was also developed to decompose the simplified convex optimization problem into smaller convex optimization problems. As shown by a numerical example, the simplification due to no-waiting property and the decomposition not only improved the solution times considerably but also allowed us to solve larger problems by alleviating computing hardware constraints.

Another numerical example compared the forward decomposition algorithm against a competing virtual deadline algorithm for controllable flow shop systems with no completion-time costs. The decomposition algorithm turned out to be superior for flow shop systems with large number of machines, because the convergence speed of the virtual deadline algorithm decreased considerably as the number of machines increased.

Assuming that all arrival times and completion deadlines are initially available can appear to be a drawback of this study preventing on-line applications. When such job information is only partially available, receding horizon controllers for flow shops, which is a topic of ongoing research, employ solution methods developed in this study. Hence, this work will form the foundation for online optimization methods for the case of random arrival times and completion deadlines.

REFERENCES

- [1] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*. Englewood Cliffs, NJ: Prentice Hall, 2002.
- [2] E. Nowicki and S. Zdrzalka, "A two-machine flow shop scheduling problem with controllable job processing times," *Eur. J. Oper. Res.*, vol. 34, no. 2, pp. 208–220, 1988.
- [3] 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.
- [4] D. Shabtay and G. Steiner, "A survey of scheduling with controllable processing times," *Discrete Appl. Math.*, vol. 155, pp. 1643–1666, 2007.
- [5] 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.
- [6] D. L. Pepyne and C. G. Cassandras, "Modeling, analysis, and optimal control of a class of hybrid systems," *Discrete Event Dynam. Syst.: Theory Appl.*, vol. 8, no. 2, pp. 175–201, 1998.
- [7] 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.
- [8] 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.
- [9] 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, no. 2, pp. 369–393, 2001.
- [10] 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.
- [11] 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.
- [12] K. Gokbayrak and O. Selvi, "Optimal hybrid control of a two-stage manufacturing system," in *Proc. ACC*, 2006, pp. 3364–3369.
- [13] K. Gokbayrak and O. Selvi, "Constrained optimal hybrid control of a flow shop system," *IEEE Trans. Autom. Control*, vol. 52, no. 12, pp. 2270–2281, Dec. 2007.
- [14] K. Gokbayrak and O. Selvi, "Optimization of a flow shop system of initially controllable machines," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2665–2668, Dec. 2008.
- [15] K. Gokbayrak and O. Selvi, "A subgradient descent algorithm for optimization of initially controllable flow shop systems," *Discrete Event Dynam. Syst.: Theory Appl.*, vol. 19, no. 2, pp. 267–282, 2009.
- [16] J. Mao, C. G. Cassandras, and Q. Zhao, "Optimal dynamic voltage scaling in energy-limited nonpreemptive systems with real-time constraints," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 678–688, Jun. 2007.
- [17] J. Mao and C. G. Cassandras, "Optimal control of two-stage discrete event systems with real-time constraints," *Discrete Event Dynam. Syst.: Theory Appl.*, vol. 17, pp. 505–529, 2007.
- [18] J. Mao and C. G. Cassandras, "Optimal control of multi-stage discrete event systems with real-time constraints," in *Proc. 45th IEEE Conf. Decision Control*, 2006, pp. 1057–1062.
- [19] J. Mao and C. G. Cassandras, "Optimal admission control of discrete event systems with real-time constraints," in *Proc. 46th IEEE Conf. Decision Control*, 2007, pp. 3005–3010.
- [20] M. Grant and S. Boyd, CVX: Matlab Software for Disciplined Convex Programming [Online]. Available: <http://stanford.edu/~boyd/cvx> 2007



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

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, 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., M.S., and Ph.D. degrees in industrial engineering from Bilkent University, Ankara, in 1999, 2002, and 2008, respectively.

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