Proceedings of the 45th IEEE Conference on Decision & Control
Manchester Grand Hyatt Hotel
San Diego, CA, USA, December 13-15, 2006

Optimal Control of Multi-Stage Discrete Event Systems with Real-Time Constraints

Jianfeng Mao and **Christos G. Cassandras**
Dept. of Manufacturing Engineering
and Center for Information and Systems Engineering
Boston University
Brookline, MA 02446
jfmao@bu.edu, cgc@bu.edu

***Abstract*— We consider Discrete Event Systems (DES) involving tasks with real-time constraints and seek to control processing times so as to minimize a cost function subject to each task meeting its own constraint. When tasks are processed over a single stage, it has been shown that there are structural properties of the optimal sample path that lead to very efficient solutions of such problems. When tasks are processed over multiple stages and are subject to end-to-end real-time constraints, these properties no longer hold and no obvious extensions are known. We consider a multi-stage problem with not only stage-dependent but also task-dependent cost functions over all tasks at each stage and derive several new optimality properties. These properties lead to the idea of introducing "virtual" deadlines at each stage except the last one, thus partially decoupling the stages so that the known efficient solutions for single-stage problems can be used. We prove that the solution obtained by an iterative Virtual Deadline Algorithm (VDA) converges to the global optimal solution of the multi-stage problem and illustrate the efficiency of the VDA through numerical examples.**
***Keywords:* discrete event system, multi-stage system, optimal control, real-time constraints**

I. INTRODUCTION

A large class of Discrete Event Systems (DES) involves the control of resources allocated to tasks according to certain operating specifications (e.g., tasks may have real-time constraints associated with them). The basic modeling block for such DES is a single-server queueing system operating on a first-come-first-served basis, whose dynamics are given by the well-known max-plus equation

$$x_i = \max(x_{i-1}, a_i) + s_i(u_i) \quad (1)$$

where a_i is the arrival time of task $i = 1, 2, \ldots$, x_i is the time when task i completes service, and $s_i(u_i)$ is its service time which may be controllable through u_i. Examples arise in manufacturing systems, where the operating speed of a machine can be controlled to trade off between energy costs and requirements on timely job completion [11]; in computer systems, where the CPU speed can be controlled to ensure that certain tasks meet specified execution deadlines [6]; and in wireless networks where severe battery limitations call for new techniques aimed at maximizing the lifetime of such a network [10]. A particularly interesting class of problems arises when such systems are subject to *real-time constraints*, i.e., $x_i \le d_i$ for each task i with a given "deadline" d_i. In order to meet such constraints, one typically has to incur a higher cost associated with control u_i. Thus, in a broader context, we are interested in studying optimization problems of the form

$$\min_{u_1,\ldots,u_N} \left\{ \sum_{i=1}^{N} \theta_i(u_i) \right\} \quad (2)$$
$$\text{s.t.} \quad x_i = \max(x_{i-1}, a_i) + s_i(u_i) \le d_i, \quad i = 1, ..., N$$

where $\theta_i(u_i)$ is a given cost function assumed to be monotonically increasing in u_i, $s_i(u_i) \ge 0$ is assumed to be monotonically decreasing in u_i, and all a_i, d_i are known. In general, this is a hard nonlinear optimization problem, complicated by the inequality constraints $x_i \le d_i$ and the nondifferentiable max operator involved. Nonetheless, it was shown in [8] that when $\theta_i(u_i)$ is convex and differentiable the solution to such problems is characterized by attractive structural properties leading to a highly efficient algorithm termed *Critical task Decomposition Algorithm* (CTDA). The CTDA does not require any numerical optimization problem solver, but only needs to identify a set of "critical" tasks in $\{1, \ldots, N\}$. The efficiency of the CTDA is crucial for applications where small, inexpensive devices are required to perform on-line computations with minimal on-board resources.

Extending the problem in (2) to a network environment, where each node in the network is characterized by dynamics of the max-plus form (1) coupled to those of other nodes, presents many challenges. We consider in this paper a serial multi-stage DES where tasks at the first stage satisfy

$$x_{i,1} = \max(x_{i-1,1}, a_i) + s_{i,1}(u_{i,1}) \quad (3)$$

and at the following stages $j = 2, ..., M$:

$$x_{i,j} = \max(x_{i-1,j}, x_{i,j-1}) + s_{i,j}(u_{i,j}),\ j = 2, ..., M \quad (4)$$

In addition, tasks at the last stage satisfy the constraints $x_{i,M} \le d_i$. In other words, tasks are processed in series at the M stages (with departures from stage $j-1$ becoming arrivals at stage j) and the real-time constraint is imposed at the end

The authors' work is supported in part by the National Science Foundation under Grant DMI-0330171, by AFOSR under grants FA9550-04-1-0133 and FA9550-04-1-0208, and by ARO under grant DAAD19-01-0610.

of this M-step process. This turns out to be *not* a simple extension of the single-stage problem (2). The decomposition properties characterizing an optimal sample path of (2) no longer hold and the coupling in (4) significantly complicates any solution methodology. The same is true for single-stage problems with no real-time constraints considered in [3], which can also be efficiently solved as shown in [4]: extending such problems to two or more stages becomes significantly more difficult [5],[2].

In [7] we considered a two-stage system with homogeneous cost functions (i.e., different for each stage but not for each task), in which we identified three structural properties leading to an iterative *Virtual Deadline Algorithm* (VDA) through which we can efficiently obtain a globally optimal solution to the problem described above. In this paper, we consider a multi-stage system with $M \geq 2$ and with nonhomogeneous (i.e., different both for each stage and each task) cost functions. We find that only two of the three structural properties in [7] still hold in this case due to the nonhomogeneous cost functions allowed. Nonetheless, the main idea of introducing a "virtual" deadline at each stage $1, ..., M-1$ still applies, so that the M-stage problem is replaced by M single-stage problems of the form (2), which we know can be very efficiently solved through the CTDA in [8]. The key issue then is determining the appropriate virtual deadline for each stage, which we will show requires the solution of an additional, though simple, M-dimensional convex optimization problem that exploits what we will refer to as the "Q-chain structure" of the system. Finally, we show that the iterative VDA converges to the global optimum of the M-stage problem.

The paper is organized as follows. In Section II, we formulate the M-stage problem with strict end-to-end real-time constraints. In Section III, we establish two structural properties of the optimal solution, leading to the proposed VDA in Section IV and the proof of convergence to a global optimum. We provide numerical examples in Section V and conclude with Section VI.

II. M-STAGE PROBLEM FORMULATION

We consider an M-stage DES, as illustrated in Fig 1, where a sequence of N tasks arrive at known times $0 \leq a_1 \leq \cdots \leq a_N$ at stage 1 and have known hard end-to-end deadlines $d_1, ..., d_N$. The tasks are processed on a first-come-first-served basis by M serial non-preemptive servers. Once a task is finished at stage $j-1$, it immediately enters the queue of stage j for $j = 2, ..., M$. The dynamics describing the process at stages $1, \ldots, M$ are given by (3) and (4), where, by convention, $x_{0,1} = \ldots = x_{0,N} = -\infty$. The deadlines $d_1, ..., d_N$ are imposed so that $x_{i,M} \leq d_i$ for all $i = 1, \ldots, N$.

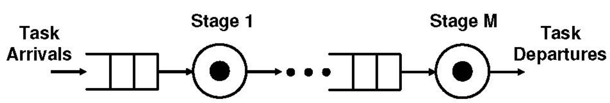

Fig. 1. A multi-stage system

Assuming $s_{i,j}(u_{i,j})$ for all i, j are known monotonically decreasing functions of $u_{i,j}$, we will concentrate on controlling directly the service times $s_{i,j}$ ($u_{i,j}$ can then be recovered) for all i, j. We define vectors $S_j = [s_{1,j}, \cdots, s_{N,j}]^T$ for $j = 1, ..., M$ and formulate the M-stage problem:

$$\begin{aligned} \min_{S_1,...,S_M} \ & \left\{ J(S_1, ..., S_M) \equiv \sum_{j=1}^{M} \sum_{i=1}^{N} \theta_{i,j}(s_{i,j}) \right\} \\ \text{s.t.} \ & x_{i,1} = \max(x_{i-1,1}, a_i) + s_{i,1}, \\ & x_{i,j} = \max(x_{i-1,j}, x_{i,j-1}) + s_{i,j}, \\ & x_{i,M} \leq d_i, \quad i = 1, ..., N, \ j = 2, ..., M; \\ & s_{i,j} \geq 0, \quad i = 1, ..., N, \ j = 1, ..., M; \\ & x_{0,1} = ... = x_{0,M} = -\infty. \end{aligned} \tag{5}$$

We consider the cost functions $\theta_{i,j}(s_{i,j}) = \mu_{i,j}\theta_j\left(\frac{s_{i,j}}{\mu_{i,j}}\right)$ where $\mu_{i,j}$ is the number of operations for task i at stage j used to differentiate tasks and $\theta_j\left(\frac{s_{i,j}}{\mu_{i,j}}\right)$ is the cost for each operation of task i at stage j used to differentiate stages (equivalently, we may think of controlling a processing rate $\rho_{ij} = \mu_{i,j}/s_{i,j}$ for a task whose requirement is expressed as $\mu_{i,j}$). The functions $\theta_j(\cdot)$ are assumed to be continuously differentiable, strictly convex and monotonically decreasing, which is consistent with most applications of interest. For instance, in manufacturing systems the cost of operating a machine is monotonically decreasing and convex in the processing time of a part [11]; in wireless devices, the processing time of a task is a convex monotonically decreasing function of the voltage applied to its CPU and the energy expended is monotonically decreasing and convex in the processing time of a task [8].

As already pointed out, the M-stage problem above is not a simple extension of the single-stage problem studied in [8]. It is much more difficult to solve for three main reasons: (i) it inherits the difficulties of the single-stage problem (described in [8]), (ii) there is an M-fold increase in the dimensionality of the control variables, and (iii) the coupling among the M stage dynamics causes the failure of the structural properties exploited in single-stage problems. In order to overcome these three difficulties and obtain efficient solutions to problem (5), we explore two structural properties of such M-stage systems in the next section.

III. OPTIMALITY PROPERTIES

A. Virtual Deadline Property

The first structural property we identify is one leading to a partial decoupling of the M stages by introducing a "virtual" deadline for tasks at stages $j < M$ and showing that we can replace problem (5) by a set of much simpler problems with a weaker form of coupling between stages.

We begin by defining vectors $X_j = [x_{1,j}, ..., x_{N,j}]^T$ for $j = 1, ..., M$, $A = [a_1, ..., a_N]^T$ and $D = [d_1, ..., d_N]^T$. In what follows, inequalities involving vectors should be understood to apply componentwise. Next, we transform problem (5) into an equivalent problem below by replacing the control variables $S_1, ..., S_M$ by $X_1, ..., X_M$ and incorporating the dynamics into the objective function, where $X_0 = A$:

$$\min_{X_1,\dots,X_M} \Big\{ J(X_1, ..., X_M) = \sum_{j=1}^{M} \sum_{i=1}^{N} \theta_{i,j}\big(x_{i,j} - \max(x_{i,j-1}, x_{i-1,j})\big) \Big\} \tag{6}$$
$$\text{s.t.} \quad X_M \le D,\ x_{i,j} - \max(x_{i,j-1}, x_{i-1,j}) \ge 0, \forall i, j$$

The optimal solution of this problem will henceforth be denoted by $(X_1^*, ..., X_M^*)$.

We can see that the stages in the problem above are strongly coupled because of the end-to-end real time constraints. Now imagine that there exist *virtual deadlines* for all tasks at stage $j = 1, ..., M-1$ and that every stage can independently optimize its control to meet these virtual deadlines. Then, the multi-stage problem (6) would be reduced to M single-stage problems studied in [8]. Let the arrival time vector at stage $j = 1, ..., M$ be $\Gamma_j = [\gamma_{1,j}, ..., \gamma_{N,j}]^T$ and the deadline vector be $\Delta_j = [\delta_{1,j}, ..., \delta_{N,j}]^T$. We then define for each j the single-stage problem:

$$\min_{X_j \in \mathbf{\Phi}(\Gamma_j, \Delta_j)} \Big\{ J_j(X_j|\Gamma_j) \equiv \sum_{i=1}^{N} \theta_{i,j}\big(x_{i,j} - \max(\gamma_{i,j}, x_{i-1,j})\big) \Big\} \tag{7}$$

where $\mathbf{\Phi}(\Gamma_j, \Delta_j)$ is the feasible space of X_j defined as:

$$\mathbf{\Phi}(\Gamma_j, \Delta_j) = \{ X_j : X_j \le \Delta_j;\ x_{i,j} - \max(\gamma_{i,j}, x_{i-1,j}) \ge 0, \forall i \}$$

Let

$$W_j(\Gamma_j, \Delta_j) = \min_{X_j \in \mathbf{\Phi}(\Gamma_j, \Delta_j)} J_j(X_j|\Gamma_j)$$

Since these single-stage problems can be efficiently solved by the CTDA developed in [8], solving M separate single-stage problems is much easier than solving the multi-stage problem (6). To establish the connection between the M single-stage problems and the multi-stage problem, we define the *virtual deadline problem* combining these M problems, with $\Delta_M = D$:

$$\min_{\Delta_1,\dots\Delta_{M-1}} \Big\{ L(\Delta_1, \dots, \Delta_{M-1}) \equiv \sum_{j=1}^{M} W_j(\Gamma_j, \Delta_j) \Big\}$$
$$\text{s.t.} \quad \Gamma_1 = A;\ \Gamma_{j+1} = \arg \min_{X_j \in \mathbf{\Phi}(\Gamma_j, \Delta_j)} J_j(X_j|\Gamma_j), j \ge 1.$$

The following lemma derives a property of the single stage problems (7). This will help in proving Theorem 1, where a connection between the original multi-stage problem (6) and the virtual deadline problem is established.

Lemma 1: Let $X_0^* = A$. The optimal solution of the problem(6), $(X_1^*, ..., X_M^*)$, satisfies for $j = 1, ..., M-1$,

$$X_j^* = \arg \min_{X_j \in \mathbf{\Phi}(X_{j-1}^*, X_j^*)} \Big\{ J_j(X_j|X_{j-1}^*) \Big\}$$

(The proofs in this paper are omitted or just sketched due to space limitations; the full proofs can be found in [9].)

Theorem 1: Let $\Delta_M^* = D$, $\hat{X}_0 = A$ and $\hat{X}_j = \arg\min_{X_j \in \mathbf{\Phi}(\hat{X}_{j-1}, \Delta_j^*)} \{ J_j(X_j|\hat{X}_{j-1}) \}$ for $j = 1, ..., M$. Then, $(\hat{X}_1, ..., \hat{X}_M)$ is the optimal solution of problem (6) if and only if $(\Delta_1^*, ..., \Delta_{M-1}^*)$ is the optimal solution of the virtual deadline problem.

Proof: "$\Longrightarrow$": Assume on the contrary that $\hat{X}_j \ne X_j^*$ for some j. Then,

$$\min_{\Delta_1,\dots,\Delta_{M-1}} \Big\{ L(\Delta_1, ..., \Delta_{M-1}) \Big\} = L(\Delta_1^*, ..., \Delta_{M-1}^*) = \sum_{i=1}^{M} J_j(\hat{X}_j|\hat{X}_{j-1}) = J(\hat{X}_1, ..., \hat{X}_M)$$

Since $\hat{X}_j \ne X_j^*$ for some j, we have

$$\min_{\Delta_1,\dots,\Delta_{M-1}} \Big\{ L(\Delta_1, ..., \Delta_{M-1}) \Big\} = J(\hat{X}_1, ..., \hat{X}_M) > J(X_1^*, ..., X_M^*) \tag{8}$$

We consider $\Delta_j = X_j^*$ for $j = 1, ..., M-1$. By Lemma 1, we have for the virtual deadline problem

$$L(X_1^*, ..., X_{M-1}^*) = \sum_{i=1}^{M} J_j(X_j^*|X_{j-1}^*) = J(X_1^*, ..., X_M^*)$$

Using the equality above, we have

$$\min_{\Delta_1,\dots,\Delta_{M-1}} \Big\{ L(\Delta_1, ..., \Delta_{M-1}) \Big\} \le L(X_1^*, ..., X_{M-1}^*) = J(X_1^*, ..., X_M^*)$$

which contradicts (8).

"$\Longleftarrow$": See [9]. ∎

Based on these results, the optimization of the multi-stage problem (6) is equivalent to first finding the optimal virtual deadlines $(\Delta_1^*, ..., \Delta_{M-1}^*)$ and then solving for M single-stage problems. Since the latter can be efficiently solved by the CTDA, finding $\Delta_1^*, ..., \Delta_{M-1}^*$ in the virtual deadline problem becomes the key to solving the multi-stage problem. Obtaining $\Delta_1^*, ..., \Delta_{M-1}^*$ is facilitated by an additional property discussed next, in which we establish a necessary and sufficient condition for optimality in this problem.

B. Q-Chain Property

The main result of this section is Theorem 2, where we establish a necessary and sufficient condition for optimality in problem (6) that involves a sequence of partially coupled problems defined below. We will refer to these as "Q problems" and collectively as the "Q-chain". The control vector for each Q problem is

$$Z_i = [x_{i,1}, x_{i-1,2}, ..., x_{i-M+1,M}]^T$$

defined for $i = 0, ..., N+M$:

$$\min_{Z_i} \Big\{ Q(Z_i|Z_{i-1}, Z_{i+1}) \equiv \sum_{j=1}^{M} \theta_{i+1-j,j}\big(z_{i,j} - \max(z_{i-1,j-1}, z_{i-1,j})\big) + \sum_{j=1}^{M} \theta_{i+2-j,j}\big(z_{i+1,j} - \max(z_{i,j-1}, z_{i,j})\big) \Big\}$$
$$\text{s.t.} \quad z_{i,j} - \max(z_{i-1,j-1}, z_{i-1,j}) \ge 0,\ \forall j$$
$$z_{i+1,j} - \max(z_{i,j-1}, z_{i,j}) \ge 0,\ \forall j$$
$$z_{i,M} \le d_{i-M+1}.$$

where $z_{i,0} = a_{i+1}$ for $i = 0, ..., N + M - 1$. From the definition of Z_i, we can see that X_j can be recovered by extracting the jth entry of each vector $Z_j, ..., Z_{j+N-1}$ as shown in Fig. 2. Note that the Z_i vectors introduce two sets of dummy variables corresponding to the lower triangular matrix and the upper triangular matrix shown in Fig. 2. These two sets represent the tasks arriving before task 1 and after task N respectively, which are not included in the original problem. In order to eliminate the influence of these dummy variables on our problem, we set, for the lower triangular matrix elements, $d_i = a_1$ and let $x_{i,j}$ be arbitrary constants smaller than a_1 for $i < 1$; that is, we force all tasks before task 1 to leave before a_1 so as to decouple them from tasks $1, ..., N$. Similarly, for the upper triangular matrix elements, we set $a_i = d_N$ and $x_{i,j}$ are arbitrary constants larger than d_N for $i > N$; that is, we force all tasks after task N to arrive after d_N so as to decouple them from tasks $1, ..., N$.

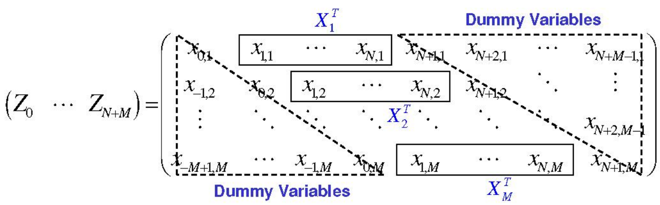

Fig. 2. Relationship between $(Z_0, ..., Z_{N+M})$ and $(X_1, ..., X_M)$

The significance of the ith Q problem can be explained as follows. For a single-stage system, we fix the departure times of the *previous* and *next* task relative to task i and then solve a scalar problem to determine $x_{i,1} \in [\max(x_{i-1,1}, a_i), \min(x_{i+1,1}, d_i)]$ minimizing the total cost incurred by tasks i and $i + 1$ alone. When there are M stages (see Fig 3), we consider the departure times of M consecutive tasks $i, \ldots, i - M + 1$ at stages $1, \ldots, M$ respectively, that is, use $Z_i = [x_{i,1}, x_{i-1,2}, ..., x_{i-M+1,M}]^T$ as the control vector, and fix the departure times of the *previous* and *next* task at each stage, that is, treat $Z_{i-1} = [x_{i-1,1}, x_{i-2,2}, ..., x_{i-M,M}]^T$ and $Z_{i+1} = [x_{i+1,1}, x_{i,2}, ..., x_{i-M+2,M}]^T$ as parameters. We then solve an M-dimensional problem to determine $x_{i,1} \in [\max(x_{i-1,1}, a_i), \min(x_{i+1,1}, x_{i,2})]$, $x_{i-1,2} \in [\max(x_{i-2,2}, x_{i-1,1}), \min(x_{i,2}, x_{i-1,3})], \ldots, x_{i-M+1,M} \in [\max(x_{i-M,M}, x_{i-M+1,M-1}), \min(x_{i-M+2,1}, d_{i-M+1})]$ minimizing the total cost incurred by tasks i and $i+1$ at stage 1, tasks $i - 1$ and i at stage 2, $\ldots$, tasks $i - M + 1$ and $i - M + 2$ at stage M.

In order to obtain our main result, we will need the next two lemmas.

Lemma 2: Problem $Q(Z_i|Z_{i-1}, Z_{i+1})$ is strictly convex in Z_i and problem $J(X_1, ..., X_M)$ is strictly convex in $X_1, ..., X_M$.

Lemma 3: Let $Y_j = [y_{1,j}, ..., y_{N,j}]^T$, and suppose $[Y_1^T, ..., Y_M^T]^T$ is a feasible direction for problem $J(X_1, ..., X_M)$. Consider the directional derivative along $[Y_1^T, ..., Y_M^T]^T$, $J'(X_1, ..., X_M; Y_1, ..., Y_M)$. Similarly, let $V_i = [y_{i,1}, ..., y_{i-M+1,M}]^T$, where $y_{i,j} = 0$ for $i < 1$ or $i > N$, and consider the directional derivative along V_i, $Q'(Z_i; V_i|Z_{i-1}, Z_{i+1})$. Then

$$J'(X_1, ..., X_M; Y_1, ..., Y_M) = \sum_{i=1}^{N+M-1} Q'(Z_i; V_i|Z_{i-1}, Z_{i+1}).$$

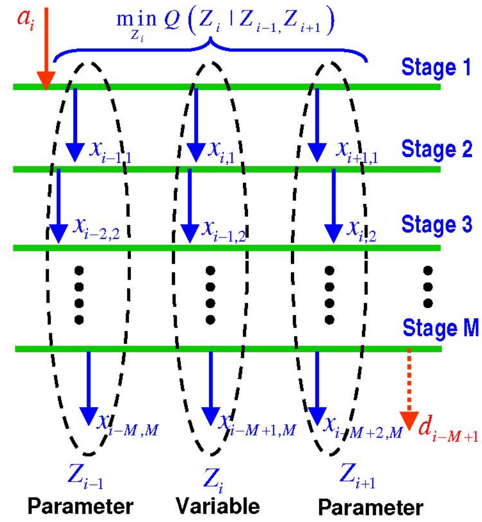

Fig. 3. Illustration of Q Problem

Let $Z_i^* = [x_{i,1}^*, ..., x_{i-M+1,M}^*]^T$ for $i = 0, ..., N + M$ be the solutions of the Q problems we have defined and recall that $X_1^*, ..., X_M^*$ can be recovered from $Z_0^*, ..., Z_{N+M}^*$ as shown in Fig. 2. We then establish the following result.

Theorem 2: $X_1^*, ..., X_M^*$ is the unique global optimum of $J(X_1, ..., X_M)$ if and only if

$$Z_i^* = \arg\min_{Z_i} \left\{ Q(Z_i|Z_{i-1}^*, Z_{i+1}^*) \right\}$$

for $i = 1, ..., N + M - 1$.

Proof: "$\Longrightarrow$": *[Sketch only]* First, from Proposition B.24(f) in [1] and $Z_i^* = \arg\min_{Z_i} \left\{ Q(Z_i|Z_{i-1}^*, Z_{i+1}^*) \right\}$, we can prove that there exists $G_j = [g_{1,j}, ..., g_{N,j}]^T$ such that

$$\sum_{j=1}^{M} G_j^T (X_j - X_j^*) \geq 0 \tag{9}$$

Second, from Proposition B.24(a) in [1], we can prove that

$$[G_1^T, ..., G_M^T]^T \in \partial J(X_1^*, ..., X_M^*) \tag{10}$$

where $\partial J(X_1^*, ..., X_M^*)$ is the subdifferential of $J(X_1, ..., X_M)$ at $X_1^*, ..., X_M^*$.

Third, from (9), (10) and Proposition B.24(f) in [1], it follows that $X_1^*, ..., X_M^*$ is the global optimum of $J(X_1, ..., X_M)$.

Finally, since $J(X_1, ..., X_M)$ is strictly convex by Lemma 2, $X_1^*, ..., X_M^*$ is the unique global optimum of $J(X_1, ..., X_M)$.

"$\Longleftarrow$": See [9]. ■

Theorem 2 provides a way to determine the optimality of problem $J(X_1, ..., X_M)$ by solving a set of M-dimensional convex optimization problems. Then, from Theorem 1, if we can find some $\Delta_1, ..., \Delta_{M-1}$ which result in the optimal departure times $X_1^*, ..., X_M^*$ that satisfy the property in Theorem 2, then these $\Delta_1, ..., \Delta_{M-1}$ must be the optimal solution of the virtual deadline problem. Thus, combining the two theorems, we can determine the optimality of the virtual deadline problem. The final remaining question, which is addressed in the next section, is how to efficiently find the optimal $\Delta_1, ..., \Delta_{M-1}$.

IV. Virtual Deadline Algorithm

In this section, we develop the Virtual Deadline Algorithm (VDA) to derive $\Delta_1^*, ..., \Delta_{M-1}^*$ efficiently. The VDA is outlined in Table I, where k is the index of the current iteration. In every iteration, the single-stage problem in step 2 is very efficiently solved by the CTDA [8] and $Q(Z_i|Z_{i-1}^k, Z_{i+1}^k)$ in step 3 is an M-dimensional convex optimization problem which can be efficiently solved. Note that the vector Δ_j^{k+1} in step 4 is extracted from the solution Λ_j^{k+1} of the jth Q problem obtained in step 3 as shown in Fig. 2. Finally, $(X_1^*, ..., X_M^*)$ obtained in step 5 is guaranteed to be the global optimum in our original problem as shown in Theorem 3 below.

TABLE I

Virtual Deadline Algorithm

Step 1:	$k = 0$, $\Delta_j^k = D$, for $j = 1, ..., M$;
Step 2:	$X_j^k = \arg\min_{X_j \in \Phi(X_{j-1}^k, \Delta_j^k)} \left\{ J_j(X_j \mid X_{j-1}^k) \right\}$, for $j = 1, ..., M$, where $X_0^k = A$;
Step 3:	$\Lambda_i^{k+1} = \arg\min_{Z_i} \left\{ Q(Z_i \mid Z_{i-1}^k, Z_{i+1}^k) \right\}$, for $i = 1, ..., N+M-1$, where $\Lambda_i^{k+1} = [\delta_{i,1}^{k+1}, ..., \delta_{i-M+1,M}^{k+1}]^T$;
Step 4:	if $\sum_{j=1}^M \left\| \Delta_j^{k+1} - \Delta_j^k \right\| / M < \epsilon$, then $k = k+1$ and Goto Step 2;
Step 5:	$\Delta_j^* = \Delta_j^{k+1}$, $X_j^* = \Delta_j^*$ for $j = 1, ..., M-1$ and $X_M^* = \arg\min_{X_M \in \Phi(X_{M-1}^*, D)} \left\{ J_j(X_M \mid X_{M-1}^*) \right\}$.

To establish Theorem 3, we first need two lemmas:

Lemma 4: If $\Delta_j^k \geq \Delta_j^{k+1}$ for $j = 1, ..., M-1$, then $X_j^k \geq X_j^{k+1}$ for $j = 1, ..., M$, $k = 1, 2, \ldots$

Lemma 5: If $Z_{i-1}^k \geq Z_{i-1}^{k+1}$ and $Z_{i+1}^k \geq Z_{i+1}^{k+1}$, then $\Lambda_i^{k+1} \geq \Lambda_i^{k+2}$, $k = 1, 2, \ldots$

Theorem 3: X_j^k and Δ_j^k monotonically converge to X_j^* for $j = 1, ..., M-1$.

Proof: *[Sketch only]* First, we prove the inequality below for $k \geq 0$ by induction using Lemma 4,

$$\Delta_j^k \geq \Delta_j^{k+1}, \; j = 1, ..., M-1 \tag{11}$$

Second, we prove the inequality below for $k \geq 0$ by using (11),

$$\Delta_j^k \geq X_j^k \geq \Delta_j^{k+1}, \; j = 1, ..., M-1 \tag{12}$$

Third, we prove the inequality below for $k \geq 0$ by induction using Lemmas 1, 4, 5 and Theorem 2,

$$\Delta_j^k \geq X_j^* \quad \text{and} \quad X_j^k \geq X_j^*, \; j = 1, ..., M-1 \tag{13}$$

Finally, from (12) and (13), both $(\Delta_1^k, ..., \Delta_{M-1}^k)$ and $(X_1^k, ..., X_{M-1}^k)$ are guaranteed to converge to the same vector $(\hat{X}_1, ..., \hat{X}_{M-1})$, which is bounded from below by $(X_1^*, ..., X_{M-1}^*)$. X_M^k must converge to $\hat{X}_M = \arg\min_{X_M \leq D} \left\{ J_M(X_M \mid \hat{X}_{M-1}) \right\}$. So, we have for all $i = 1, ..., N+M-1$

$$\hat{Z}_i = \arg\min_{Z_i} \left\{ Q(Z_i \mid \hat{Z}_{i-1}, \hat{Z}_{i+1}) \right\} \tag{14}$$

Using (14) and Theorem 2, it follows that $\hat{X}_j = X_j^*$ for $j = 1, ..., M$. ■

V. Numerical results

A. Example

We have applied the VDA to a 3-stage system with $N = 8$, arrival time vector $A = [1\ 2\ 4\ 6\ 7\ 8\ 9\ 12]^T$, deadline vector $D = [20\ 22\ 24\ 26\ 29\ 31\ 33\ 38]^T$, number of operations vectors $\mu_1 = [1\ 3\ 1\ 4\ 3\ 3\ 1\ 3]^T$, $\mu_2 = [2\ 3\ 1\ 1\ 1\ 3\ 2\ 3]^T$, $\mu_3 = [2\ 1\ 3\ 2\ 4\ 2\ 5\ 2]^T$, and cost functions $\theta_{i,j}(s_{i,j}) = k_j \cdot \mu_{i,j}/(s_{i,j}/\mu_{i,j} + 0.001)$ where $K = [k_1\ k_2\ k_3] = [3\ 2\ 4]$. The termination condition in step 4 of the VDA is set so that $\sum_{j=1}^M \left\| \Delta_j^{k+1} - \Delta_j^k \right\| / M = \sum_{j=1}^M \sum_{i=1}^N (\delta_{i,j}^k - \delta_{i,j}^{k+1})/N/M < \epsilon = 0.001$. The optimal

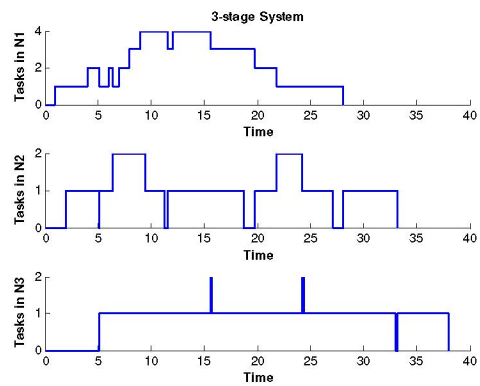

Fig. 4. Optimal sample path obtained by the VDA

sample path obtained by the VDA is shown in Fig. 4, where the y-axis shows the number of tasks in each stage and the x-axis is the time line. Fig. 5 shows the optimal processing

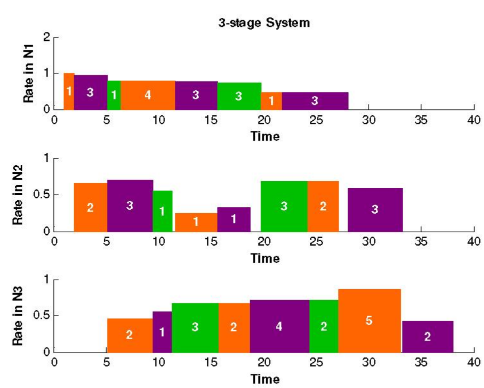

Fig. 5. Optimal processing rate obtained by the VDA

rates, where the length of a block is the service time of the corresponding task, and the number in a block is its number of operations.

B. Complexity

We have tested the complexity of the VDA in terms of CPU time and number of iterations. In these tests, the VDA was programmed using Matlab 7.0 on an Intel Pentium4 3.06GHz, 1.0 GB RAM machine. We tested cases where N varied from 100 to 1000 in increments of 100 and M varied from 2 to 8 in increments of 1. We randomly generated 10 samples for each combination of N and M, in which task interarrival times are exponentially distributed with mean 4, $d_i - a_i$ is uniformly distributed over $[5M, 5M+2]$ and $\mu_{i,j}, k_j$ are selected from $[1,\ 2,\ 3,\ 4]$ with equal probability. For each case, we recorded the elapsed CPU time and required number of iterations, finally averaging them to obtain the corresponding performance.

Fig. 6 and Fig. 7 show the average CPU time (in seconds) as a function of the number of tasks N and the number of stages M. Fig. 8 shows the average number of iterations also as a function of the number of tasks N and the number of stages M. We observe that the VDA complexity scales with N, while the number of iterations is insensitive to N.

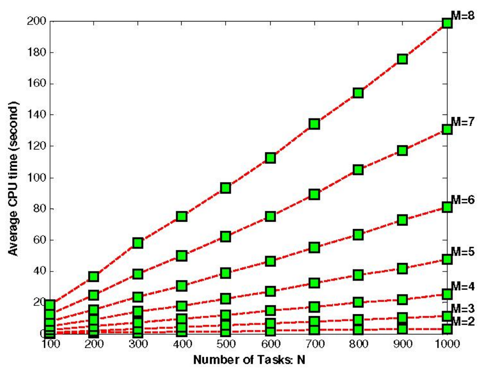

Fig. 6. Average CPU time of VDA as a function of N

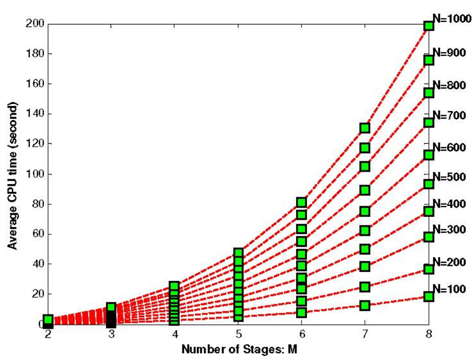

Fig. 7. Average CPU time of VDA as a function of M

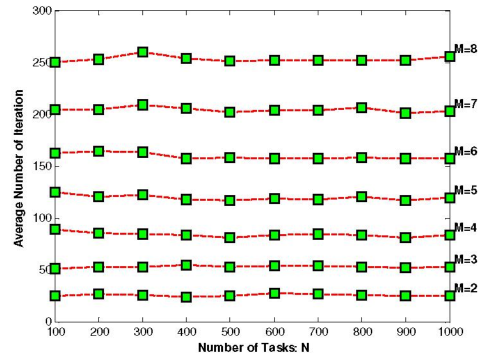

Fig. 8. Average Number of Iterations of VDA

VI. Conclusions and Future Work

As pointed out in the Introduction, it is difficult to extend optimal control problems for DES with real-time constraints from a single stage to $M \geq 2$ stages. We have derived two optimality properties that lead to the idea of introducing "virtual" deadlines at stage $1, ..., M-1$, and then solving partially decoupled single-stage problems whose solutions are known to be efficiently obtained. Based on this idea, we have developed an iterative Virtual Deadline Algorithm (VDA) and shown that it converges to the global optimal solution of the M-stage problem. In practice, task arrival times may not be known at the time problem (5) needs to be solved, in which case one must proceed by repeatedly solving the problem as new arrival information is obtained or by estimating future arrivals or by relying on stochastic optimization techniques making use of distributional information regarding the arrival process. Our ongoing work is focusing on such cases, while also exploring generalizations of the system setting to arbitrary networks rather than the tandem case considered in this paper.

References

[1] D. P. Bertsekas. *Nonlinear programming (Second Edition)*. Athena Scientific, Belmont, Mass., 1999.

[2] C.G. Cassandras, Q. Liu, K. Gokbayrak, and D.L. Pepyne. Optimal control of a two-stage hybrid manufacturing system model. In *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 450–455, Dec. 1999.

[3] C.G. Cassandras, D.L. Pepyne, and Y. Wardi. Optimal control of a class of hybrid system. *IEEE Transactions on Automatic Control*, 46(3):398–415, 2001.

[4] Y.C. Cho, C.G. Cassandras, and D.L. Pepyne. Forward decomposition algorithms for optimal control of a class of hybrid systems. *Intl. J. of Robust and Nonlinear Control*, 11(5):497–513, 2001.

[5] K. Gokbayrak and C.G. Cassandras. Constrained optimal control for multistage hybrid manufacturing system models. In *Proceedings of 8th IEEE Mediterranean Conference on New Directions in Control and Automation*, July 2000.

[6] J.W.S Liu. *Real - Time System*. Prentice Hall Inc., 2000.

[7] J. Mao and C.G. Cassandras. Optimal control of two-stage discrete event systems with real-time constraints. In *8th International Workshop on Discrete Event Systems (WODES'06)*, pages 125–130, July 2006.

[8] J. Mao, Q. Zhao, and C.G. Cassandras. Optimal dynamic voltage scaling in power-limited systems with real-time constraints. In *Proc. of 43rd IEEE Conf. Decision and Control*, pages 1472–1477, Dec. 2004. (Subm. to IEEE Trans. on Mobile Computing).

[9] J.F. Mao and C.G. Cassandras. Optimal control of multi-stage discrete event systems with real-time constraints. *Technical Report, CODES, Boston University*, 2006. See also ftp://dacta.bu.edu:2491/TechReport/2006DVSmHN.pdf.

[10] L. Miao and C. G. Cassandras. Optimal transmission scheduling for energy-efficient wireless networks. To appear in *Proceedings of INFOCOM 2006*.

[11] D.L. Pepyne and C.G. Cassandras. Optimal control of hybrid systems in manufacturing. In *Proceedings of the IEEE*, volume 88, pages 1108–1123, 2000.