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

Jianfeng Mao, *Student Member, IEEE*, and Christos G. Cassandras, *Fellow, IEEE*

***Abstract*—We consider Discrete Event Systems 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 state trajectory 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 such 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 a sequence of solutions to single-stage problems with virtual deadlines updated at each step converges to the global optimal solution of the multi-stage problem. This leads to a Virtual Deadline Algorithm (VDA) which is scalable in the number of processed tasks. We illustrate the scalability and efficiency of the VDA through numerical examples.**

***Index Terms*—Discrete event system (DES), 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. The basic modeling block for such DES is a single-server queueing system operating on a first-come-first-served basis, whose event time dynamics are given by the well-known max-plus equation

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

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 energy costs against requirements on timely job completion [23]; in computer systems, where the CPU speed can be controlled to ensure that certain tasks meet specified execution deadlines [17]; and in wireless networks where severe battery limitations call for new techniques aimed at maximizing the lifetime of such a network [10], [21]. 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 general form

$$\begin{aligned} &\min_{u_1,\ldots,u_N} \sum_{i=1}^{N} \theta_i(u_i) \\ \text{s.t. } x_i &= \max(x_{i-1}, a_i) + s_i(u_i) \le d_i, i = 1, \ldots, N; \\ s_i(u_i) &\ge s_{\min,i},\ i = 1, \ldots, N;\ x_0 = -\infty \end{aligned} \tag{2}$$

where $s_{\min,i}$ is the minimal possible service time of task i. In general, the controls $u_1, \ldots, u_N$ may be time-varying. However, as shown in [22], when $\theta_i(\cdot)$ is monotonically increasing and convex and all a_i, d_i are known, then the optimal control of each task is constant during the processing time $s_i(u_i)$. We will consider such cases and also assume that $s_i(u_i) \ge s_{\min,i}$ is monotonically decreasing in u_i for all $i = 1, \ldots, N$. Systems which process tasks with real-time constraints have been extensively studied, mostly in the computer science literature: preemptive tasks are considered, for example, in [1], [24], nonpreemptive periodic tasks in [15], [16], and nonpreemptive aperiodic tasks in [10], [20], [21]. The latter case is of particular interest in wireless communications where nonpreemptive scheduling is necessary to execute aperiodic packet transmission tasks which also happen to be highly energy-intensive.

Even if u_i is constant throughout the service time $s_i(u_i)$, problem (2) is generally a hard nonlinear optimization problem, complicated by the inequality constraints $x_i \le d_i$ and the non-differentiable max operator involved. Although the max operator can be removed by introducing auxiliary variables $w_i, i = 1, \ldots, N$, and adding the constraints $x_i = w_i + u_i, w_i \ge x_{i-1}$, $w_i \ge a_i$, this makes the problem even more inefficient to solve since it doubles its dimensionality and also introduces $2N$ inequality constraints. A max-plus algebra formulation [2], [7], [14] may be utilized to efficiently obtain solutions for some optimal control problems with real-time constraints such as the optimal tracking problem in [6], [8], [9]. However, these problems differ from (2) in two main aspects. First, in these problems release times (arrival times) are controlled and service times are fixed, while in (2) we control service times with arrival times fixed. Second, the problem objectives are different. In optimal tracking the optimal solution is one resulting in the latest possible completion times prior to real-time constraints. In our problem, the objective is to minimize some cost related to

Manuscript received September 28, 2007; revised May 02, 2008. Current version published January 14, 2009. This work was supported in part by the National Science Foundation under Grants DMI-0330171 and EFRI-0735974, by AFOSR under Grants FA9550-07-1-0213 and FA9550-07-1-0361, and by the Department of Energy (DOE) under Grant DE-FG52-06NA27490. Recommended by Associate Editor S. Haar.

The authors are with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446 USA (e-mail: jfmao@bu.edu; cgc@bu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2008.2009572

service times; a feasible solution resulting in the latest completion times before real-time constraints is generally not optimal.

Despite the difficulties above, it was shown in [20] that when $\theta_i(u_i)$ is convex and differentiable the optimal state trajectory in such problems is characterized by attractive structural properties leading to a highly efficient algorithm termed *Critical Task Decomposition Algorithm* (CTDA). The original CTDA assumes that $\theta_i(u_i)$ is of the form $\theta_i(u_i) = \mu_i \theta(u_i/\mu_i)$ and $s_i(u_i) = u_i$, where μ_i is a task-dependent constant (typically representing the "size" of the task). The *Generalized* CTDA (GCTDA) in [21] removes this restriction on $\theta_i(u_i)$. The CTDA completely eliminates the need for a numerical optimization problem solver and reduces the solution of (2) to a simple scalable procedure for identifying a set of "critical" tasks in $\{1,\ldots,N\}$. The efficiency and scalability of the CTDA are crucial for applications where small, inexpensive, often wireless devices are required to perform on-line computations with minimal on-board resources[1].

In this paper, we address the problem of tasks executed in 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. This is a much more challenging problem which cannot be dealt with by merely extending the CTDA. In particular, we consider a serial multi-stage DES where tasks at the first stage satisfy

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

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

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

In addition, tasks at the last stage satisfy the constraints $x_{i,M} \leq d_i$. In other words, tasks are processed in series at the M stages (with departures from stage $j-1$ becoming arrivals at stage j) with an end-to-end real-time constraint imposed at the completion of this M-step process. The decomposition properties characterizing an optimal state trajectory of (2) no longer hold and the coupling in (4) significantly complicates any solution methodology. Incidentally, the same complications arise even in the absence of real-time constraints: extending such single-stage problems solved in [5] even to two stages becomes significantly more difficult [4], [12]. In [19] we considered a two-stage system with homogeneous cost functions (i.e., different for each stage but not for each task) and identified three structural properties 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 one of the key properties when $M = 2$ no longer applies when $M > 2$, so that extending the approach of [19] is infeasible. As in [19], however, the main idea of our approach is to introduce a "virtual" deadline at each stage $1,\ldots,M-1$, 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 [20]. The key issue then is determining the appropriate virtual deadline for each stage, which we will show requires the solution of additional, but simple, M-dimensional convex optimization problems that exploit what we will refer to as the "Q-chain structure" of the system, originally presented in [19]. A complete solution to the multi-stage problem is provided, based on constructing a sequence of solutions to simple single-stage problems as in (2) in which "virtual deadlines" are adjusted at each step. We show that this sequence converges to the global optimum of the original problem and provide an explicit Virtual Deadline Algorithm (VDA) to implement this solution approach.

[1]Numerical examples show [20] that standard nonlinear programming software (e.g., NPSOL 5.0) fails to provide solutions for $N > 400$, while the CTDA gives exact solutions in less than 1 sec for values of N of the order of 10^4. An efficient convex programming solver such as CVX [13] is still (in its default precision) two orders of magnitude slower.

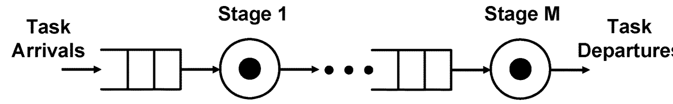


Fig. 1. Multi-stage system.

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, in Section IV, to the construction of a single-stage problem solution sequence and the proof of its convergence to the global optimum of the multi-stage problem. We present the VDA in Section V, followed by numerical examples in Section VI, and conclude with Section VII.

II. Multi-Stage Problem Formulation

We consider a multi-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 $d_1,\ldots,d_N$. The tasks are processed on a first-come-first-served basis by M serial nonpreemptive servers. Changing the processing order at different stages might improve performance, but it adds a layer of complexity that cannot be handled by resource-limited devices. Moreover, in many applications global order preservation is required (e.g., processing data packets through a network requires order preservation at the receiver). Once a task is finished at stage $j-1$, it immediately enters the queue of stage j for $j = 2,\ldots,M$. The dynamics describing the process at stages $1,\ldots,M$ are given by (3) and (4), where, by convention, $x_{0,1} = \cdots = x_{0,M} = -\infty$. The deadlines $d_1,\ldots,d_N$ are imposed so that $x_{i,M} \leq d_i$ for all $i = 1,\ldots,N$.

Let $\mu_{i,j}$ denote the size of task i at stage j and assume the control $u_{i,j}$ is the processing rate of task i at stage j. In the following, we will concentrate on controlling directly the service times $s_{i,j}$ for all i,j because $u_{i,j}$ can then be easily recovered through $u_{i,j} = \mu_{i,j}/s_{i,j}$. Let $\theta_{i,j}(s_{i,j})$ denote the cost function of $s_{i,j}$, which can be expressed as

$$\theta_{i,j}(s_{i,j}) = \mu_{i,j} \cdot \theta_j(u_{i,j}) = \mu_{i,j} \cdot \theta_j\left(\frac{\mu_{i,j}}{s_{i,j}}\right)$$

where $\theta_j(u_{i,j})$ is the cost function of the processing rate $u_{i,j}$ at stage j and is assumed to be continuously differentiable, strictly convex and monotonically increasing with respect to $u_{i,j}$, which is consistent with most applications of interest. For instance, in manufacturing systems the cost of operating a machine is monotonically increasing and convex in the processing rate of a part [23]; in wireless devices, the processing rate of a task is a convex monotonically increasing function of the voltage applied to its CPU and the energy expended is monotonically increasing and convex in the voltage [20]. Based on this fact, $\theta_{i,j}(s_{i,j})$ is continuously differentiable, strictly convex and monotonically decreasing in $s_{i,j}$. Moreover, tasks cannot be processed infinitely fast, that is, $u_{i,j} \leq \rho_{\max,j}$, where $\rho_{\max,j}$ is the maximal processing rate at stage j. This constraint can be equivalently expressed in terms of the service time $s_{i,j}$ as $s_{i,j} \geq s_{\min,i,j} = \mu_{i,j}/\rho_{\max,j}$. Finally, we can formulate the multi-stage problem as follows:

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

Due to the constraints "$s_{i,j} \geq s_{\min,i,j}$", the system cannot guarantee that all tasks meet their associated deadlines, that is, the problem above may be infeasible. In this paper, we will study the feasible case (if that is not the case, then a separate admission control problem has to precede our analysis so as to eliminate certain tasks and lead to a feasible problem, as described for the single-stage case in [18]).

As already pointed out, the M-stage problem above is not a simple extension of the single-stage problem studied in [20] or even the two-stage problem in [19]. It is much more difficult to solve for three main reasons: *(i)* it inherits the difficulties of the single-stage problem (described in [20]), *(ii)* there is an M-fold increase in the dimensionality of the control variables, and *(iii)* the coupling among the $M > 2$ stage dynamics causes the failure of the structural properties exploited in single or two-stage problems. We note that Dynamic Programming (DP) can, in principle, be utilized to reduce the dimensionality of the multi-stage problem (5) by focusing on a portion of the control variables through the introduction of a cost-to-go function. However, the recursion of the cost-to-go function in this problem is too complicated to lead to a closed-form solution. Thus, we have found DP to be quite inefficient in obtaining the optimal solution. In order to overcome these difficulties and obtain efficient solutions to problem (5), we introduce and analyze 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 show 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 N-dimensional vectors $A = [a_1,\ldots,a_N]^T$, $D = [d_1,\ldots,d_N]^T$, and $X_j = [x_{1,j},\ldots,x_{N,j}]^T$ for $j = 1,\ldots,M$, as well as the matrix $X = [X_1,\ldots,X_M]$. In what follows, inequalities involving vectors should be understood to apply componentwise. Next, we transform problem (5) into an equivalent problem below by setting the control variables to be the entries of X and incorporating the dynamics into the objective function. In what follows, we will omit the subscripts i,j from the function $\theta_{i,j}(\cdot)$ only to simplify the unavoidable notational burden necessitated by indexing tasks and nodes. It will be seen that this causes no loss of generality, as all subsequent proofs do not depend on any differences among cost functions associated with tasks or nodes as long as all such functions remain convex and monotonic. The transformed problem (5) becomes

$$\begin{aligned}
&\min_X J(X) = \sum_{j=1}^{M}\sum_{i=1}^{N} \theta\big(x_{i,j} - \max(x_{i,j-1}, x_{i-1,j})\big) \qquad (6)\\
&\text{s.t. } x_{i,j} - \max(x_{i,j-1}, x_{i-1,j}) \geq s_{\min,i,j}, \forall i,j;\ X_M \leq D
\end{aligned}$$

where $X_0 = A$ for notational consistency.

We can see that the M 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 each stage j, denoted by $\Delta_j = [\delta_{1,j},\ldots,\delta_{N,j}]^T$, 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 of the form studied in [20], where the arrival time vector at stage j is just the departure time of stage $j-1$, X_{j-1}. Define

$$L(X_j|X_{j-1}) \equiv \sum_{i=1}^{N} \theta\big(x_{i,j} - \max(x_{i,j-1}, x_{i-1,j})\big) \tag{7}$$

and formulate a single-stage problem for each stage j as

$$\min_{X_j \in \Phi(X_{j-1},\Delta_j)} L(X_j|X_{j-1}) \tag{8}$$

where the feasible space $\Phi(X_{j-1},\Delta_j)$ is defined as

$$\begin{aligned}
&\Phi(X_{j-1},\Delta_j)\\
&= \big\{X_j \leq \Delta_j; x_{i,j} - \max(x_{i,j-1}, x_{i-1,j}) \geq s_{\min,i,j},\ \forall i\big\}.
\end{aligned} \tag{9}$$

Thus, in (8) we fix the vector X_{j-1} with all arrival times at stage j and control the departure times in X_j. Since these single-stage problems can be efficiently solved by the CTDA [20] or its generalized version, the GCTDA, solving M separate single-stage problems is much easier than solving the multi-stage problem (6). If we can obtain the optimal solution of (6) by solving M single-stage problems above, then the complexity of (6) will be greatly reduced. We show through Theorem 1 that this is indeed possible.

Theorem 1: Let $X^* = [X_1^*,\ldots,X_M^*]^T$ denote the optimal solution of Problem (6) and $X_0^* = A$, $\Delta_j = X_j^*$ for $j = 1,\ldots,M$. Then, for $j = 1,\ldots,M$

$$X_j^* = \arg\min_{X_j \in \Phi(X_{j-1}^*,\Delta_j)} L(X_j|X_{j-1}^*).$$

Proof: We only consider the case where $j < M$; the case where $j = M$ can be similarly proved. Proceeding through a contradiction argument, let $\hat{X}_j = [\hat{x}_{1,j}, \ldots, \hat{x}_{N,j}]^T \in \Phi(X^*_{j-1}, \Delta_j)$ and assume that for some $j < M$

$$\arg \min_{X_j \in \Phi(X^*_{j-1}, X^*_j)} L(X_j | X^*_{j-1}) = \hat{X}_j \neq X^*_j.$$

Then

$$L(\hat{X}_j | X^*_{j-1}) < L(X^*_j | X^*_{j-1}). \tag{10}$$

Since $\hat{X}_j$ is the solution of (8), it must satisfy $\hat{X}_j \leq \Delta_j = X^*_j$ and the following inequality holds for all $i = 1, \ldots, N$:

$$x^*_{i,j+1} - \max(\hat{x}_{i,j}, x^*_{i-1,j+1}) \geq x^*_{i,j+1} - \max(x^*_{i,j}, x^*_{i-1,j+1}).$$

Because of the inequality above and the fact that $\theta(\cdot)$ is monotonically decreasing, we have for all $i = 1, \ldots, N$

$$\theta\big(x^*_{i,j+1} - \max(\hat{x}_{i,j}, x^*_{i-1,j+1})\big) \leq \theta\big(x^*_{i,j+1} - \max(x^*_{i,j}, x^*_{i-1,j+1})\big).$$

Using this inequality, it follows from (7) that:

$$L(X^*_{j+1} | \hat{X}_j) \leq L(X^*_{j+1} | X^*_j). \tag{11}$$

Let $\hat{X} = [X^*_1, \ldots, X^*_{j-1}, \hat{X}_j, X^*_{j+1} \ldots, X^*_M]^T$. Note that the only difference between $\hat{X}$ and X^* is the departure time vector at stage j which only affects the cost functions of stages j and $j+1$. Therefore, in evaluating $J(\hat{X}) - J(X^*)$ the only terms remaining are

$$J(\hat{X}) - J(X^*) = L(\hat{X}_j | X^*_{j-1}) + L(X^*_{j+1} | \hat{X}_j) - L(X^*_j | X^*_{j-1}) - L(X^*_{j+1} | X^*_j).$$

Combining this with (10) and (11), we have

$$J(\hat{X}) - J(X^*) < 0$$

which contradicts the optimality of X^* and completes the proof. ■

Based on Theorem 1, it is possible that the whole multi-stage system reaches optimality when each single stage $j = 1, \ldots, M$ reaches its own optimality by setting its virtual deadlines to $\Delta_j = X^*_j$. However, for arbitrary Δ_j, the optimality for each stage does not correspond to the optimality of the whole multi-stage system. Thus, this decomposition applies only at the optimal point X^*, making the theorem of little apparent use. However, we will see that combining this result with an additional property discussed in the next section leads to an efficient iterative process which still reduces the solution of the original problem to solving M partially coupled single-stage problems.

Theorem 1 also provides us an opportunity to simplify the lower bound constraints $s_{i,j} \geq s_{\min,i,j}$ in the multi-stage problem (5). From Theorem 1 we know the optimal solution can be obtained by solving M single-stage problems with the introduction of virtual deadlines. Since the original multi-stage problem is assumed to be feasible, the M corresponding single-stage problems must also be feasible. Then, from Proposition 3 in [20], we know that if a single-stage problem is feasible, then the lower bound constraints can be equivalently replaced by simple nonnegativity constraints. In particular, Proposition 3 in [20] asserts that if (8) is solved without the constraints $s_{i,j} \geq s_{\min,i,j}$ and its solution results in some $s^*_{i,j} < s_{\min,i,j}$, then the single-stage problem is in fact infeasible, hence (5) is infeasible. Therefore, we can replace the lower bound constraint $s_{i,j} \geq s_{\min,i,j}$ in (5) by $s_{i,j} \geq 0$ and we only need to consider these nonnegativity constraints in the rest of the paper.

B. Q-Chain Property

As already mentioned, the *virtual deadline* property can partially decompose the multi-stage problem (6) into M single stage problems (8), but this property alone cannot decouple the whole multi-stage system into M single-stage problems. In this section, we will introduce what we refer to as the *Q-chain property*, which can decompose the multi-stage problem into a sequence of partially coupled problems, referred to as *Q-problems*. This property leads to the main result of this section, Theorem 2, where we show that the solution of the multi-stage problem is equivalent to solving all *Q*-problems.

To fully understand the *Q*-chain property, we begin with the single-stage problem, where $X = X_1$

$$\min_X J(X) = \sum_{i=1}^{N} \theta\big(x_i - \max(a_i, x_{i-1})\big)$$
$$\text{s.t. } x_i - \max(a_i, x_{i-1}) \geq 0, \ \forall i; \quad X \leq D. \tag{12}$$

Define

$$Q(x_i | x_{i-1}, x_{i+1}) \equiv \theta\left(x_i - \max(a_i, x_{i-1})\right) + \theta\left(x_{i+1} - \max(a_{i+1}, x_i)\right)$$

and introduce the *Q*-problem for this simple scalar case for $i = 1, \ldots, N$ as follows:

$$\min_{x_i} Q(x_i | x_{i-1}, x_{i+1})$$
$$\text{s.t. } x_i - \max(a_i, x_{i-1}) \geq 0;$$
$$x_{i+1} - \max(a_{i+1}, x_i) \geq 0; \ \ x_i \leq d_i$$

where x_{N+1} and a_{N+1} are "dummy variables." In order to eliminate the influence of these dummy variables, we set a_{N+1} and x_{N+1} to be arbitrary constants larger than d_N, that is, we force the "dummy" task $N+1$ to arrive after d_N so as to decouple them from tasks $1, \ldots, N$.

Observe that in the *Q*-problem x_i is controllable while x_{i-1}, x_{i+1} are treated as fixed. There are N such *Q*-problems. Each one is a small piece of the single-stage problem (12) and is only coupled to its two neighboring *Q*-problems, $Q(x_{i-1} | x_{i-2}, x_i)$ and $Q(x_{i+1} | x_i, x_{i+2})$. We refer to these *Q*-problems collectively as the "*Q-chain*".

Let $\mathcal{D}_Y J(X)$ denote the directional derivative of $J(X)$ at X along a feasible direction $Y = [y_1, \ldots, y_N]^T$ and let $\mathcal{D}_{y_i} Q(x_i | x_{i-1}, x_{i+1})$ denote the directional derivative of $Q(x_i | x_{i-1}, x_{i+1})$ at x_i along a feasible direction y_i (the definition of the directional derivative can be found in [3] and is also given in the Appendix; its use is motivated by the presence of the nondifferentiable max function in our problem). It can be easily verified that

$$\mathcal{D}_Y J(X) = \sum_{i=1}^{N} \mathcal{D}_{y_i} Q(x_i | x_{i-1}, x_{i+1}). \tag{13}$$

Assume X^* is the optimal solution of the single-stage problem (12). From Theorem 4.3.2 in [11] (quoted in the Appendix), optimality for a convex programming problem such as (12) means that $\mathcal{D}_Y J(X^*) \geq 0$ for any feasible direction Y at X^*. Suppose we can obtain solutions to all *Q*-problems and define a vector $\hat{X} = [\hat{x}_i, \ldots, \hat{x}_N]^T$ such that

$$\mathcal{D}_{y_i} Q(\hat{x}_i | \hat{x}_{i-1}, \hat{x}_{i+1}) \geq 0, \quad \text{for all} \quad i = 1, \ldots, N.$$

Then, combining this condition with (13), we have

$$\mathcal{D}_Y J(\hat{X}) = \sum_{i=1}^{N} \mathcal{D}_{y_i} Q(\hat{x}_i | \hat{x}_{i-1}, \hat{x}_{i+1}) \geq 0$$

which implies that $\hat{X}$ is also the optimal solution of the single-stage problem (12). Each *Q*-problem above is a scalar problem which is much easier to solve than (12). This relationship provides an opportunity to obtain the optimal solution of a large-scale problem by solving a set of much simpler *Q*-problems.

Based on this analysis, we see that the key to establishing the equivalence between the optimality of *Q*-problems and the original problem in the single-stage case is the summation form in (13) satisfied by directional derivatives. If we can similarly express the directional derivative of the multi-stage problem (6) as the summation of the directional derivatives of cost functions for some properly defined *Q*-problems, then we can establish a similar equivalence of the multi-stage problem and a set of simpler *Q*-problems. Towards this goal, we define the natural extension of the single-stage *Q*-problem for the multi-stage system (6) as follows. Introducing the shorthand notation $[x_{i,j}]_{\forall j} \equiv [x_{i,1}, \ldots, x_{i,M}]$, let

$$\begin{aligned} Q\big([x_{i,j}]_{\forall j} \big| [x_{i-1,j}]_{\forall j}, [x_{i+1,j}]_{\forall j}\big) &= \theta\left(x_{i,j} - \max(x_{i,j-1}, x_{i-1,j})\right) \\ &\quad + \theta\left(x_{i+1,j} - \max(x_{i+1,j-1}, x_{i,j})\right) \end{aligned}$$

and define a potential *Q*-problem

$$\begin{aligned} &\min_{[x_{i,j}]_{\forall j}} Q\big([x_{i,j}]_{\forall j} \big| [x_{i-1,j}]_{\forall j}, [x_{i+1,j}]_{\forall j}\big) \\ \text{s.t.} \quad & x_{i,j} - \max(x_{i,j-1}, x_{i-1,j}) \geq 0, \ \forall j; \\ & x_{i+1,j} - \max(x_{i+1,j-1}, x_{i,j}) \geq 0, \ \forall j; \ x_{i,M} \leq d_i. \end{aligned}$$

Unfortunately, we can easily verify that the definition of $Q(\cdot|\cdot,\cdot)$ above cannot satisfy the summation form in (13) with $Y = [Y_1, \ldots, Y_M]^T$ and $Y_j = [y_{1,j}, \ldots, y_{N,j}]^T$ for $j = 1, \ldots, M$, that is

$$\mathcal{D}_Y J(X) \neq \sum_{i=1}^{M} \mathcal{D}_{[y_{i,j}]_{\forall j}} Q\big([x_{i,j}]_{\forall j} \big| [x_{i-1,j}]_{\forall j}, [x_{i+1,j}]_{\forall j}\big). \tag{14}$$

Thus, even if all *Q*-problems above reach optimality simultaneously, the corresponding multi-stage problem (6) may not be optimized. This failure is due to the presence of the function $\max(x_{i-1,j}, x_{i,j-1})$ which introduces a coupling between $x_{i-1,j}$ and $x_{i,j-1}$ for any i, j. To satisfy a summation form as in (13), we have to include both $x_{i-1,j}$ and $x_{i,j-1}$ as controllable variables when defining the *Q*-problem. With this motivation, define

$$\begin{aligned} Q\big([x_{i-j,j}]_{\forall j} \big| [x_{i-j-1,j}]_{\forall j}, [x_{i-j+1,j}]_{\forall j}\big) &\equiv \sum_{j=1}^{M} \theta\big(x_{i-j,j} - \max(x_{i-j,j-1}, x_{i-j-1,j})\big) \\ &\quad + \sum_{j=1}^{M} \theta\big(x_{i-j+1,j} - \max(x_{i-j+1,j-1}, x_{i-j,j})\big). \end{aligned}$$

We then formulate, for $i = 2, \ldots, N+M$, the *Q*-problem

$$\begin{aligned} &\min_{[x_{i-j,j}]_{\forall j}} Q\big([x_{i-j,j}]_{\forall j} \big| [x_{i-j-1,j}]_{\forall j}, [x_{i-j+1,j}]_{\forall j}\big) \\ \text{s.t.} \quad & x_{i-j,j} - \max(x_{i-j,j-1}, x_{i-j-1,j}) \geq 0, \ \forall j; \\ & x_{i-j+1,j} - \max(x_{i-j+1,j-1}, x_{i-j,j}) \geq 0, \ \forall j; \\ & x_{i-M,M} \leq d_{i-M}; \quad x_{i-1,0} = a_{i-1}, \ x_{i,0} = a_i. \end{aligned} \tag{15}$$

Note that $x_{i,j}$, d_i and a_i are only defined for $i = 1, \ldots, N$. Therefore, for $i < 1$ or $i > N$, $x_{i,j}$, d_i and a_i are "dummy variables". In order to eliminate the influence of these dummy variables, we set $d_i = a_1$ for $i < 1$ and let $x_{i,j}$ be arbitrary constants smaller than a_1 for all i such that $i < 1$; that is, we force all "dummy" tasks before task 1 to leave before a_1 so as to decouple them from $1, \ldots, N$. Similarly, we set a_i and $x_{i,j}$ to be arbitrary constants larger than d_N for all $i > N$, that is, we force all tasks after N to arrive after d_N so as to decouple them from tasks $1, \ldots, N$.

In problem (15), the control vector $[x_{i-j,j}]_{\forall j}$ includes all pairs $(x_{i-j,j}, x_{i-j+1,j-1})$ coupled through $\max(\cdot)$, which, as shown next, makes it possible to satisfy a summation form such as (13). To simplify notation, we define the M-dimensional vector for $i = 2, \ldots, N+M$

$$\tilde{X}_i = [\tilde{x}_{i,1}, \ldots, \tilde{x}_{i,M}] = [x_{i-1,1}, \ldots, x_{i-M,M}]^T. \tag{16}$$

Then, the *Q*-problem becomes

$$\min_{\tilde{X}_i \in \Psi(\tilde{X}_{i-1}, \tilde{X}_{i+1})} Q(\tilde{X}_i | \tilde{X}_{i-1}, \tilde{X}_{i+1})$$

where $\Psi(\tilde{X}_{i-1}, \tilde{X}_{i+1})$ is its corresponding feasible space. Fig. 2 provides a visual representation of a *Q*-problem structure in terms of its controllable vector $\tilde{X}_i$ and the fixed vectors $\tilde{X}_{i-1}$ and $\tilde{X}_{i+1}$ that its solution will depend on.

It should be clear that not only can we construct $\tilde{X}_i$ from the matrix $X = [X_1, \ldots, X_M]$, but the converse is also true. In particular, we construct $\tilde{X}_i$ in (16) through

$$\tilde{x}_{i,j} = x_{i-j,j}, \quad i = 1, \ldots, N+M+1, \ j = 1, \ldots, M \tag{17}$$

and, conversely, we obtain the elements of X_j from the matrix $\tilde{X} = [\tilde{X}_2, \ldots, \tilde{X}_{N+M}]$, through

$$x_{i,j} = \tilde{x}_{i+j,j}, \quad i = 1, \ldots, N, \ j = 1, \ldots, M. \tag{18}$$

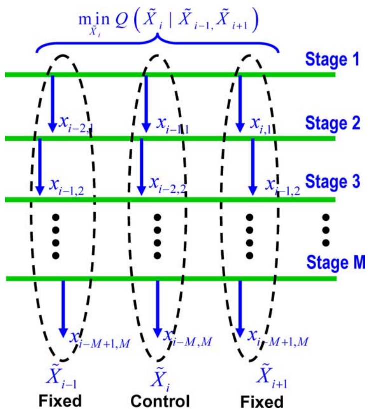


Fig. 2. Illustration of Q-problem.

For convenience, we will make use of an operator $\mathcal{T}$ such that

$$\tilde{X} = \mathcal{T}(X) \quad \text{and} \quad X = \mathcal{T}^{-1}(\tilde{X}) \tag{19}$$

where each element of $\mathcal{T}(X)$ is defined through (17) and each element of $\mathcal{T}^{-1}(\tilde{X})$ is defined through (18).

In what follows, Lemma 1 shows that the definition of the Q-problem in (15) satisfies a desirable summation form similar to (13) relating the directional derivatives of $J(X)$ and of $Q(\tilde{X}_i|\tilde{X}_{i-1},\tilde{X}_{i+1})$. Let $\tilde{Y}_i = [y_{i-1,1},\ldots,y_{i-M,M}]^T$, where $y_{i,j}=0$ for all $i<1$ or $i>N$.

Lemma 1: The directional derivatives of $J(X)$ and of $Q(\tilde{X}_i|\tilde{X}_{i-1},\tilde{X}_{i+1})$ satisfy

$$\mathcal{D}_Y J(X) = \sum_{i=2}^{N+M} \mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}_i|\tilde{X}_{i-1},\tilde{X}_{i+1}).$$

Proof: Applying Lemma 6 in the Appendix to $\theta(x_{i,j} - \max(x_{i,j-1},x_{i-1,j}))$, we have

$$\begin{aligned}&\mathcal{D}_Y \theta\big(x_{i,j} - \max(x_{i,j-1},x_{i-1,j})\big)\\ &= y_{i,j}\theta'(x_{i,j} - \max(x_{i,j-1},x_{i-1,j}))\\ &\quad - \big(\mathcal{D}_{[y_{i,j-1},y_{i-1,j}]} \max(x_{i,j-1},x_{i-1,j})\big)\\ &\quad \cdot \theta'\big(x_{i,j} - \max(x_{i,j-1},x_{i-1,j})\big).\end{aligned}$$

Therefore, from (6), the directional derivative of $J(X)$ along Y is given by

$$\begin{aligned}\mathcal{D}_Y J(X) = \sum_{j=1}^{M}\sum_{i=1}^{N}\Big[& y_{i,j}\theta'\big(x_{i,j}-\max(x_{i,j-1},x_{i-1,j})\big)\\ &-\big(\mathcal{D}_{[y_{i,j-1},y_{i-1,j}]}\max(x_{i,j-1},x_{i-1,j})\big)\\ &\cdot\theta'\big(x_{i,j}-\max(x_{i,j-1},x_{i-1,j})\big)\Big]\end{aligned} \tag{20}$$

and, similarly

$$\begin{aligned}&\mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}_i|\tilde{X}_{i-1},\tilde{X}_{i+1})\\ &= \sum_{j=1}^{M}\Big[y_{i-j,j}\theta'\big(x_{i-j,j}-\max(x_{i-j,j-1},x_{i-j-1,j})\big)\\ &\quad - \big(\mathcal{D}_{[y_{i-j+1,j-1},y_{i-j,j}]}\max(x_{i-j+1,j-1},x_{i-j,j})\big)\\ &\quad \cdot\theta'\big(x_{i-j+1,j}-\max(x_{i-j+1,j-1},x_{i-j,j})\big)\Big].\end{aligned}$$

It follows that:

$$\begin{aligned}&\sum_{i=2}^{N+M}\mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}_i|\tilde{X}_{i-1},\tilde{X}_{i+1})\\ &= \sum_{j=1}^{M}\sum_{i=2}^{N+M}\Big[y_{i-j,j}\theta'\big(x_{i-j,j}-\max(x_{i-j,j-1},x_{i-j-1,j})\big)\\ &\quad - \big(\mathcal{D}_{[y_{i-j+1,j-1},y_{i-j,j}]}\max(x_{i-j+1,j-1},x_{i-j,j})\big)\\ &\quad \cdot\theta'\big(x_{i-j+1,j}-\max(x_{i-j+1,j-1},x_{i-j,j})\big)\Big].\end{aligned}$$

Using Lemma 7 in the Appendix and the facts that $x_{N+1,j-1} > x_{N,M} \geq x_{N,j}$, and $y_{N+1,j-1}=0$, we have

$$\mathcal{D}_{[y_{N+1,j-1},y_{N,j}]}\max(x_{N+1,j-1},x_{N,j}) = 0.$$

Combining this with $y_{i,j}=0$ for all $i<1$ or $i>N$, we get

$$\begin{aligned}&\sum_{i=2}^{N+M}\mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}_i|\tilde{X}_{i-1},\tilde{X}_{i+1})\\ &= \sum_{j=1}^{M}\sum_{i=j+1}^{N+j}\Big[y_{i-j,j}\theta'\Big(x_{i-j,j}-\max(x_{i-j,j-1},x_{i-j-1,j})\Big)\Big]\\ &\quad - \sum_{j=1}^{M}\sum_{i=j}^{N+j-1}\Big[\Big(\mathcal{D}_{[y_{i-j+1,j-1},y_{i-j,j}]}\max(x_{i-j+1,j-1},x_{i-j,j})\Big)\\ &\quad \cdot\theta'\Big(x_{i-j+1,j}-\max(x_{i-j+1,j-1},x_{i-j,j})\Big)\Big].\end{aligned}$$

Letting $k=i-j$ and $l=i-j+1$ in the two inner sums above, we have

$$\begin{aligned}&\sum_{i=2}^{N+M}\mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}_i|\tilde{X}_{i-1},\tilde{X}_{i+1})\\ &= \sum_{j=1}^{M}\sum_{k=1}^{N}\Big[y_{k,j}\theta'\big(x_{k,j}-\max(x_{k,j-1},x_{k-1,j})\big)\Big]\\ &\quad - \sum_{j=1}^{M}\sum_{l=1}^{N}\Big[\big(\mathcal{D}_{[y_{l,j-1},y_{l-1,j}]}\max(x_{l,j-1},x_{l-1,j})\big)\\ &\quad \cdot\theta'\big(x_{l,j}-\max(x_{l,j-1},x_{l-1,j})\big)\Big]\\ &= \mathcal{D}_Y J(X)\end{aligned}$$

where we have used (20). This completes the proof. ∎

Lemma 1 can only guarantee local optimality for the multi-stage problem (6). To establish global optimality, we need to ensure the convexity of problems (6) and (15), as shown in the next lemma.

Lemma 2: The multi-stage problem (6) is strictly convex in X and the Q-problem (15) is strictly convex in $\tilde{X}_i$.

Proof: Starting with the multi-stage problem (6), we first show that the feasible set is convex. Let $X^1=[X_1^1,\ldots,X_M^1]^T$ and $X^2=[X_1^2,\ldots,X_M^2]^T$ be two arbitrary distinct feasible solutions of the multi-stage problem, that is

$$X_M^1 \leq D, \quad X_M^2 \leq D \tag{21}$$

and for $i = 1, \ldots, N,\ j = 1, \ldots, M$

$$\begin{aligned} x_{i,j}^1 - \max(x_{i,j-1}^1, x_{i-1,j}^1) \geq 0 \\ x_{i,j}^2 - \max(x_{i,j-1}^2, x_{i-1,j}^2) \geq 0. \end{aligned} \tag{22}$$

It follows from (21) that $\alpha X_M^1 + \beta X_M^2 \leq D$ for any $\alpha + \beta = 1$ and $\alpha, \beta \geq 0$. Next, we can write

$$\begin{aligned} &\alpha \max(x_{i,j-1}^1, x_{i-1,j}^1) + \beta \max(x_{i,j-1}^2, x_{i-1,j}^2) \\ &\quad = \max\big(\alpha x_{i,j-1}^1 + \beta x_{i,j-1}^2, \alpha x_{i-1,j}^1 + \beta x_{i-1,j}^2 \\ &\quad\quad \alpha x_{i,j-1}^1 + \beta x_{i-1,j}^2, \alpha x_{i-1,j}^1 + \beta x_{i,j-1}^2\big) \\ &\quad \geq \max\big(\alpha x_{i,j-1}^1 + \beta x_{i,j-1}^2, \alpha x_{i-1,j}^1 + \beta x_{i-1,j}^2\big). \end{aligned} \tag{23}$$

Combining (22) and (23)

$$\begin{aligned} &\alpha x_{i,j}^1 + \beta x_{i,j}^2 - \max\big(\alpha x_{i,j-1}^1 + \beta x_{i,j-1}^2, \alpha x_{i-1,j}^1 + \beta x_{i-1,j}^2\big) \\ &\quad \geq \alpha x_{i,j}^1 + \beta x_{i,j}^2 - \alpha \max(x_{i,j-1}^1, x_{i-1,j}^1) \\ &\quad\quad - \beta \max(x_{i,j-1}^2, x_{i-1,j}^2) \geq 0. \end{aligned}$$

Thus, the feasible set is convex.

Second, we prove that the multi-stage problem (6) is strictly convex in X over the feasible set. Since $\theta(\cdot)$ is strictly convex, we have, for any $\alpha + \beta = 1$ and $\alpha, \beta > 0$

$$\begin{aligned} &\alpha \sum_{j=1}^{M} \sum_{i=1}^{N} \theta\big(x_{i,j}^1 - \max(x_{i,j-1}^1, x_{i-1,j}^1)\big) \\ &\quad + \beta \sum_{j=1}^{M} \sum_{i=1}^{N} \theta\big(x_{i,j}^2 - \max(x_{i,j-1}^2, x_{i-1,j}^2)\big) \\ &> \sum_{j=1}^{M} \sum_{i=1}^{N} \theta\big(\alpha x_{i,j}^1 + \beta x_{i,j}^2 \\ &\quad - \alpha \max(x_{i,j-1}^1, x_{i-1,j}^1) - \beta \max(x_{i,j-1}^2, x_{i-1,j}^2)\big). \end{aligned}$$

Since $\theta(\cdot)$ is also monotonically decreasing, combining this fact with (23) and the inequality above, we have

$$\begin{aligned} &\alpha \sum_{j=1}^{M} \sum_{i=1}^{N} \theta\big(x_{i,j}^1 - \max(x_{i,j-1}^1, x_{i-1,j}^1)\big) \\ &\quad + \beta \sum_{j=1}^{M} \sum_{i=1}^{N} \theta\big(x_{i,j}^2 - \max(x_{i,j-1}^2, x_{i-1,j}^2)\big) \\ &> \sum_{j=1}^{M} \sum_{i=1}^{N} \theta\big(\alpha x_{i,j}^1 + \beta x_{i,j}^2 \\ &\quad - \max(\alpha x_{i,j-1}^1 + \beta x_{i,j-1}^2, \alpha x_{i-1,j}^1 + \beta x_{i-1,j}^2)\big) \end{aligned}$$

which implies that the multi-stage problem (6) is strictly convex in X. The Q-problem (15) can be similarly proved to be strictly convex in $\tilde{X}_i$. ■

Using Lemmas 1 and 2, we can finally obtain the following necessary and sufficient condition for global optimality.

Theorem 2: Let $\tilde{X}^* = \mathcal{T}(X^*)$. X^* is the unique global optimum of the multi-stage problem (6) if and only if it holds for $i = 2, \ldots, N+M$ that

$$\tilde{X}_i^* = \arg \min_{\tilde{X}_i \in \Psi(\tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*)} Q(\tilde{X}_i | \tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*).$$

Proof: "$\Longrightarrow$": Using Lemma 1, we have for any feasible direction Y

$$\mathcal{D}_Y J(X^*) = \sum_{i=2}^{N+M} \mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}_i^* | \tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*). \tag{24}$$

Since $\tilde{X}_i^*$ is the optimal solution of the Q-problem (15) with fixed $\tilde{X}_{i-1}^*$ and $\tilde{X}_{i+1}^*$ for $i = 2, \ldots, N+M$, we have

$$\mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}_i^* | \tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*) \geq 0, \quad i = 2, \ldots, N+M.$$

Combining this inequality with (24), we have $\mathcal{D}_Y J(X^*) \geq 0$ for any feasible direction Y. Based on Lemma 2, the multi-stage problem (6) is strictly convex in X. It follows from this fact and $\mathcal{D}_Y J(X^*) \geq 0$ that X^* is the unique global optimum of the multi-stage problem (6) based on Theorem 4.3.2 in [11].

"$\Longleftarrow$": Assume on the contrary that for some i

$$\tilde{X}_i^* \neq \arg \min_{\tilde{X}_i \in \Psi(\tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*)} Q(\tilde{X}_i | \tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*).$$

Then, we can always find some

$$\tilde{X}_i' = [x_{i-1,1}', \ldots, x_{i-M,M}']^T \neq \tilde{X}_i^*$$

such that

$$Q(\tilde{X}_i' | \tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*) < Q(\tilde{X}_i^* | \tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*). \tag{25}$$

Let $X_j' = [x_{1,j}^*, \ldots, x_{i-j-1,j}^*, x_{i-j+1,j}', x_{i-j+2,j}^*, \ldots, x_{N,j}^*]^T$ for $j = 1, \ldots, M$ and $X' = [X_1', \ldots, X_M']$. Then, recalling the definition of $L(X_j | X_{j-1})$ in (7), we have

$$\begin{aligned} &J(X^*) - J(X') \\ &= \sum_{j=1}^{M} \Big[L(X_j^* | X_{j-1}^*) - L(X_j' | X_{j-1}')\Big] \\ &= \sum_{j=1}^{M} \Big[\theta\big(x_{i-j,j}^* - \max(x_{i-j,j-1}^*, x_{i-j-1,j}^*)\big) \\ &\quad + \theta\big(x_{i-j+1,j}^* - \max(x_{i-j+1,j-1}^*, x_{i-j,j}^*)\big) \\ &\quad - \theta\big(x_{i-j,j}' - \max(x_{i-j,j-1}^*, x_{i-j-1,j}^*)\big) \\ &\quad - \theta\big(x_{i-j+1,j}^* - \max(x_{i-j+1,j-1}', x_{i-j,j}')\big)\Big] \\ &= Q(\tilde{X}_i^* | \tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*) - Q(\tilde{X}_i' | \tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*). \end{aligned}$$

By the equation above and (25), it follows that $J(X^*) > J(X')$, which contradicts the optimality of X^*. ■

Theorem 2 provides a way to determine the optimality of the multi-stage problem (6) by solving a set of M-dimensional convex optimization problems. The final remaining question,

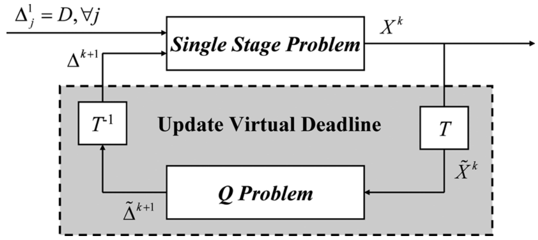


Fig. 3. Generating the sequence $\{X^k\}, k = 1, 2, \ldots$.

which is addressed in the next section, is how to exploit this property in order to efficiently determine X^*.

IV. Convergence Analysis of Single-Stage Solution Sequences

As shown in the previous section, we can partially decompose the original multi-stage problem into M single stage problems or into $N+M-1$ *Q*-problems. Although the optimal solution X^* of the multi-stage problem still cannot be directly obtained by solving these two types of problems, together they can be utilized to solve the multi-stage problem through a sequence of single-stage problem solutions. We will describe next how to construct such a sequence and prove that it monotonically converges to X^*.

Consider a sequence of single-stage problems of the form (8) with solutions defined by

$$X_j^k = \arg \min_{X_j \in \Phi(X_{j-1}^k, \Delta_j^k)} L(X_j | X_{j-1}^k), \quad j = 1, \ldots, M. \tag{26}$$

This gives rise to a sequence $\{X^k\}$, $k = 1, 2, \ldots$ with $X^k = [X_1^k, \ldots, X_M^k]$ and X_j^k dependent on the virtual deadline vector Δ_j^k. Let us initialize these vectors so that $\Delta_j^1 = D$ for all $j = 1, \ldots, M$ and define a sequence $\{\Delta^k\}$, $k = 2, 3, \ldots$ of virtual deadline vectors as follows. Let $\tilde{X}^k = \mathcal{T}(X^k)$ using the definition of $\mathcal{T}$ in (19) and define $\tilde{\Delta}^{k+1}$ to be the solution of the *Q*-problem for $i = 2, \ldots, N+M$

$$\tilde{\Delta}_i^{k+1} = \arg \min_{\tilde{X}_i \in \Psi(\tilde{X}_{i-1}^k, \tilde{X}_{i+1}^k)} Q(\tilde{X}_i | \tilde{X}_{i-1}^k, \tilde{X}_{i+1}^k). \tag{27}$$

Finally, we use the definition of $\mathcal{T}^{-1}$ in (19) to obtain $\Delta^{k+1} = \mathcal{T}^{-1}(\tilde{\Delta}^{k+1})$. This process is illustrated in Fig. 3 where we see that X^k is the input to a collection of *Q*-problems (after being transformed to $\tilde{X}^k$) and Δ^{k+1} consists of the solutions of these problems (after being inverse-transformed from $\tilde{\Delta}^{k+1}$), which are then input to M single-stage problems with virtual deadlines given by Δ^{k+1}.

In the following, we show that $X^k \to X^*$ as $k \to \infty$. We begin with some auxiliary results, i.e., Lemma 3 and Lemma 4, which establish monotonicity properties satisfied by the solutions X^k of the single-stage virtual deadline problems and the solutions $\tilde{X}^k$ of the *Q*-problems, respectively.

Lemma 3: Let

$$X_j^1 = \arg \min_{X_j \in \Phi(X_{j-1}^1, \Delta_j^1)} L(X_j | X_{j-1}^1) \tag{28}$$

$$X_j^2 = \arg \min_{X_j \in \Phi(X_{j-1}^2, \Delta_j^2)} L(X_j | X_{j-1}^2). \tag{29}$$

If, for any $j = 1, \ldots, M$, $X_{j-1}^1 \leq X_{j-1}^2$ and $\Delta_j^1 \leq \Delta_j^2$, then $X_j^1 \leq X_j^2$.

Proof: Assume on the contrary that there exists some i such that $x_{i,j}^1 > x_{i,j}^2$. From Proposition 1 in [20], we have $x_{N,j}^1 = \delta_{N,j}^1$ and $x_{N,j}^2 = \delta_{N,j}^2$ (this is a simple consequence of the monotonicity of $\theta(\cdot)$, implying that any $x_{N,j} < \delta_{N,j}$ would result in a higher cost, hence cannot be optimal). Since $\delta_{N,j}^1 \leq \delta_{N,j}^2$, it follows that $x_{N,j}^1 \leq x_{N,j}^2$. Moreover, $x_{0,j}^1$ and $x_{0,j}^2$ are dummy variables that can be regarded as equal to a common constant. Thus, it holds that $x_{0,j}^1 = x_{0,j}^2$ and $x_{N,j}^1 \leq x_{N,j}^2$. Therefore, to satisfy $x_{i,j}^1 > x_{i,j}^2$ for some i, there must exist some $1 \leq \alpha \leq \beta \leq N$ such that for any j:

$$x_{\alpha-1,j}^1 \leq x_{\alpha-1,j}^2, \quad x_{\beta+1,j}^1 \leq x_{\beta+1,j}^2$$
$$\text{and} \quad x_{i,j}^1 > x_{i,j}^2, \; i = \alpha, \ldots, \beta.$$

Let

$$X_j' = [x_{1,j}^2, \ldots, x_{\alpha-1,j}^2, x_{\alpha,j}^1, \ldots, x_{\beta,j}^1, x_{\beta+1,j}^2, \ldots, x_{N,j}^2].$$

We can easily verify that $X_j' \in \Phi(X_{j-1}^2, \Delta_j^2)$ by establishing the following inequalities:

$$\begin{aligned}
& x_{i,j}^1 \leq \delta_{i,j}^1 \text{ and } \delta_{i,j}^1 \leq \delta_{i,j}^2 \\
& \quad \Longrightarrow x_{i,j}^1 \leq \delta_{i,j}^2, \; \forall\, i = \alpha, \ldots, \beta; \\
& x_{i,j}^1 > x_{i,j}^2 \geq x_{i,j-1}^2 \text{ and } x_{i,j}^1 \geq x_{i-1,j}^1 \\
& \quad \Longrightarrow x_{i,j}^1 - \max(x_{i-1,j}^1, x_{i,j-1}^2) \geq 0, \; \forall\, i = \alpha+1, \ldots, \beta; \\
& x_{\alpha,j}^1 > x_{\alpha,j}^2 \text{ and } x_{\alpha,j}^2 \geq \max(x_{\alpha-1,j}^2, x_{\alpha,j-1}^2) \\
& \quad \Longrightarrow x_{\alpha,j}^1 - \max(x_{\alpha-1,j}^2, x_{\alpha,j-1}^2) \geq 0; \\
& x_{\beta+1,j}^2 \geq x_{\beta+1,j}^1 \geq x_{\beta,j}^1 \text{ and } x_{\beta+1,j}^2 \geq x_{\beta+1,j-1}^2 \\
& \quad \Longrightarrow x_{\beta+1,j}^2 - \max(x_{\beta,j}^1, x_{\beta+1,j-1}^2) \geq 0.
\end{aligned}$$

Next, let $Y = X_j^2 - X_j'$, that is

$$y_i = \begin{cases} x_{i,j}^2 - x_{i,j}^1 < 0 & i = \alpha, \ldots, \beta \\ 0 & \text{otherwise} \end{cases}. \tag{30}$$

Since $X_j', X_j^2 \in \Phi(X_{j-1}^2, \Delta_j^2)$ and $\Phi(X_{j-1}^2, \Delta_j^2)$ is convex, the direction Y at the point X_j' is feasible in $\Phi(X_{j-1}^2, \Delta_j^2)$.

Let

$$X_j'' = [x_{1,j}^1, \ldots, x_{\alpha-1,j}^1, x_{\alpha,j}^2, \ldots, x_{\beta,j}^2, x_{\beta+1,j}^1, \ldots, x_{N,j}^1].$$

We can similarly prove that $X_j'' \in \Phi(X_{j-1}^1, \Delta_j^1)$. Since $Y = X_j'' - X_j^1$ and $X_j'', X_j^1 \in \Phi(X_{j-1}^1, \Delta_j^1)$ and $\Phi(X_{j-1}^1, \Delta_j^1)$ is convex, the direction Y at the point X_j^1 is feasible in $\Phi(X_{j-1}^1, \Delta_j^1)$.

It follows from (29) and $X_j' \in \Phi(X_{j-1}^2, \Delta_j^2)$ that $L(X_j^2 | X_{j-1}^2) < L(X_j' | X_{j-1}^2)$. Then, from the convexity of the single-stage problem (8), Y must be a decreasing direction at X_j', i.e.,

$$\mathcal{D}_Y L\big(X_j' \big| X_{j-1}^2\big) < 0. \tag{31}$$

Using Lemma 6 (see Appendix) and (30), we can compute $\mathcal{D}_Y L\big(X'_j\big|X^2_{j-1}\big)$ as follows:

$$\begin{aligned}&\mathcal{D}_Y L\big(X'_j\big|X^2_{j-1}\big)\\&=y_\alpha\theta'\big(x^1_{\alpha,j}-\max(x^2_{\alpha-1,j},x^2_{\alpha,j-1})\big)\\&\quad+\sum_{i=\alpha+1}^{\beta}y_i\theta'\big(x^1_{i,j}-\max(x^1_{i-1,j},x^2_{i,j-1})\big)\\&\quad-\sum_{i=\alpha}^{\beta-1}\big(\mathcal{D}_{[y_i,0]}\max(x^1_{i,j},x^2_{i+1,j-1})\big)\\&\quad\cdot\theta'\big(x^1_{i+1,j}-\max(x^1_{i,j},x^2_{i+1,j-1})\big)\\&\quad-\big(\mathcal{D}_{[y_\beta,0]}\max(x^1_{\beta,j},x^2_{\beta+1,j-1})\big)\\&\quad\cdot\theta'\big(x^2_{\beta+1,j}-\max(x^1_{\beta,j},x^2_{\beta+1,j-1})\big).\end{aligned}$$

Moreover, from Lemma 7 (see Appendix) and the fact that $y_i<0$ for $i=\alpha,\ldots,\beta$ in (30) we have for $i=\alpha,\ldots,\beta$

$$\begin{aligned}&\mathcal{D}_{[y_i,0]}\max(x^1_{i,j},x^2_{i+1,j-1})\\&\quad=y_i\mathbf{1}_{[x^1_{i,j}>x^2_{i+1,j-1}]}+\max(y_i,0)\mathbf{1}_{[x^1_{i,j}=x^2_{i+1,j-1}]}\\&\quad=y_i\mathbf{1}_{[x^1_{i,j}>x^2_{i+1,j-1}]}.\end{aligned}$$

Combining the last two equations, we have

$$\begin{aligned}&\mathcal{D}_Y L\big(X'_j\big|X^2_{j-1}\big)\\&=y_\alpha\theta'\big(x^1_{\alpha,j}-\max(x^2_{\alpha-1,j},x^2_{\alpha,j-1})\big)\\&\quad+\sum_{i=\alpha+1}^{\beta}y_i\theta'\big(x^1_{i,j}-\max(x^1_{i-1,j},x^2_{i,j-1})\big)\\&\quad-\sum_{i=\alpha}^{\beta-1}y_i\mathbf{1}_{[x^1_{i,j}>x^2_{i+1,j-1}]}\theta'\big(x^1_{i+1,j}-\max(x^1_{i,j},x^2_{i+1,j-1})\big)\\&\quad-y_\beta\mathbf{1}_{[x^1_{\beta,j}>x^2_{\beta+1,j-1}]}\theta'\big(x^2_{\beta+1,j}-\max(x^1_{\beta,j},x^2_{\beta+1,j-1})\big)\\&=y_\alpha\theta'\big(x^1_{\alpha,j}-\max(x^2_{\alpha-1,j},x^2_{\alpha,j-1})\big)\\&\quad+\sum_{i=\alpha+1}^{\beta}y_i\theta'\big(x^1_{i,j}-\max(x^1_{i-1,j},x^2_{i,j-1})\big)\\&\quad-\sum_{i=\alpha}^{\beta-1}y_i\mathbf{1}_{[x^1_{i,j}>x^2_{i+1,j-1}]}\theta'\big(x^1_{i+1,j}-x^1_{i,j}\big)\\&\quad-y_\beta\mathbf{1}_{[x^1_{\beta,j}>x^2_{\beta+1,j-1}]}\theta'\big(x^2_{\beta+1,j}-x^1_{\beta,j}\big).\end{aligned}\tag{32}$$

We can similarly derive

$$\begin{aligned}&\mathcal{D}_Y L\big(X^1_j\big|X^1_{j-1}\big)\\&=y_\alpha\theta'\big(x^1_{\alpha,j}-\max(x^1_{\alpha-1,j},x^1_{\alpha,j-1})\big)\\&\quad+\sum_{i=\alpha+1}^{\beta}y_i\theta'\big(x^1_{i,j}-\max(x^1_{i-1,j},x^1_{i,j-1})\big)\\&\quad-\sum_{i=\alpha}^{\beta-1}y_i\mathbf{1}_{[x^1_{i,j}>x^1_{i+1,j-1}]}\theta'\big(x^1_{i+1,j}-x^1_{i,j}\big)\\&\quad-y_\beta\mathbf{1}_{[x^1_{\beta,j}>x^1_{\beta+1,j-1}]}\theta'\big(x^1_{\beta+1,j}-x^1_{\beta,j}\big).\end{aligned}\tag{33}$$

Let us now compare each term of (33) with the corresponding term in (32). First, since $y_\alpha<0$, $x^1_{\alpha-1,j}\le x^2_{\alpha-1,j}$, $x^1_{\alpha,j-1}\le x^2_{\alpha,j-1}$, $\theta'(\cdot)<0$ (because $\theta(\cdot)$ is monotonically decreasing), and $\theta'(\cdot)$ is monotonically nondecreasing (because $\theta(\cdot)$ is convex), we have

$$\begin{aligned}&y_\alpha\theta'\big(x^1_{\alpha,j}-\max(x^1_{\alpha-1,j},x^1_{\alpha,j-1})\big)\\&\qquad\le y_\alpha\theta'\big(x^1_{\alpha,j}-\max(x^2_{\alpha-1,j},x^2_{\alpha,j-1})\big).\end{aligned}\tag{34}$$

Similarly, since $y_i<0$ for $i=\alpha+1,\ldots,\beta$, $x^1_{i,j-1}\le x^2_{i,j-1}$, $\theta'(\cdot)<0$, and $\theta'(\cdot)$, we have

$$\begin{aligned}&y_i\theta'\big(x^1_{i,j}-\max(x^1_{i-1,j},x^1_{i,j-1})\big)\\&\qquad\le y_i\theta'\big(x^1_{i,j}-\max(x^1_{i-1,j},x^2_{i,j-1})\big).\end{aligned}\tag{35}$$

Third, since $y_i<0$ for $i=\alpha,\ldots,\beta-1$, $x^1_{i+1,j-1}\le x^2_{i+1,j-1}$ and $\theta'(\cdot)<0$, we have

$$\begin{aligned}&-y_i\mathbf{1}_{[x^1_{i,j}>x^1_{i+1,j-1}]}\theta'\big(x^1_{i+1,j}-x^1_{i,j}\big)\\&\qquad\le -y_i\mathbf{1}_{[x^1_{i,j}>x^2_{i+1,j-1}]}\theta'\big(x^1_{i+1,j}-x^1_{i,j}\big).\end{aligned}\tag{36}$$

Finally, since $y_\beta<0$, $x^1_{\beta+1,j-1}\le x^2_{\beta+1,j-1}$, $x^1_{\beta+1,j}\le x^2_{\beta+1,j}$, $\theta'(\cdot)<0$ and $\theta'(\cdot)$ is monotonically nondecreasing, we have

$$\begin{aligned}&-y_\beta\mathbf{1}_{[x^1_{\beta,j}>x^1_{\beta+1,j-1}]}\theta'\big(x^1_{\beta+1,j}-x^1_{\beta,j}\big)\\&\qquad\le -y_\beta\mathbf{1}_{[x^1_{\beta,j}>x^2_{\beta+1,j-1}]}\theta'\big(x^2_{\beta+1,j}-x^1_{\beta,j}\big).\end{aligned}\tag{37}$$

Comparing (32) to (33) in view of (34) through (37), we have

$$\mathcal{D}_Y L\big(X^1_j\big|X^1_{j-1}\big)\le\mathcal{D}_Y L\big(X'_j\big|X^2_{j-1}\big).\tag{38}$$

Recalling (28) and the fact that Y is a feasible direction at X^1_j in $\Phi(X^1_{j-1},\Delta^1_j)$, it follows that:

$$\mathcal{D}_Y L\big(X^1_j\big|X^1_{j-1}\big)\ge 0.$$

Combining this inequality with (38), we have

$$\mathcal{D}_Y L\big(X'_j\big|X^2_{j-1}\big)\ge 0,$$

which contradicts (31) and completes the proof. ■

Lemma 4: Let

$$\tilde{X}^1_i=\arg\min_{\tilde{X}_i\in\Psi(\tilde{X}^1_{i-1},\tilde{X}^1_{i+1})}Q(\tilde{X}_i|\tilde{X}^1_{i-1},\tilde{X}^1_{i+1})\tag{39}$$

$$\tilde{X}^2_i=\arg\min_{\tilde{X}_i\in\Psi(\tilde{X}^2_{i-1},\tilde{X}^2_{i+1})}Q(\tilde{X}_i|\tilde{X}^2_{i-1},\tilde{X}^2_{i+1}).\tag{40}$$

If, for any $i=2,\ldots,N+M$, $\tilde{X}^1_{i-1}\le\tilde{X}^2_{i-1}$ and $\tilde{X}^1_{i+1}\le\tilde{X}^2_{i+1}$, then $\tilde{X}^1_i\le\tilde{X}^2_i$.

Proof: The proof is along the same lines as that of Lemma 3, but we provide it in full for the sake of completeness. Assume on the contrary that there exists $1\le j\le M$ such that $\tilde{x}^1_{j,j}>\tilde{x}^2_{i,j}$, i.e., recalling (17), $x^1_{i-j,j}>x^2_{i-j,j}$. Since $x^1_{i,0}=$

$x_{i,0}^2 = a_i$ and $x_{i-M-1,M+1}^1, x_{i-M-1,M+1}^2$ are dummy variables which can be regarded as the same constant, it naturally holds that $x_{i,0}^1 \le x_{i,0}^2$ and $x_{i-M-1,M+1}^1 \le x_{i-M-1,M+1}^2$. Therefore, there must exist some $1 \le \alpha \le \beta \le M$ such that

$$x_{i-\alpha+1,\alpha-1}^1 \le x_{i-\alpha+1,\alpha-1}^2, \quad x_{i-\beta-1,\beta+1}^1 \le x_{i-\beta-1,\beta+1}^2$$
$$\text{and} \quad x_{i-j,j}^1 > x_{i-j,j}^2,\ j=\alpha,\ldots,\beta.$$

Let

$$\tilde{X}_i' = [x_{i-1,1}^2, \ldots, x_{i-\alpha+1,\alpha-1}^2, x_{i-\alpha,\alpha}^1, \ldots, x_{i-\beta,\beta}^1, x_{i-\beta-1,\beta+1}^2, \ldots, x_{i-M,M}^2].$$

We can easily verify that $\tilde{X}_i' \in \Psi(\tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2)$ by following a procedure similar to the one at the beginning of Lemma 3.

Next, let $Y = \tilde{X}_i^2 - \tilde{X}_i' = [y_1, \ldots, y_M]^T$, that is

$$y_j = \begin{cases} x_{i-j,j}^2 - x_{i-j,j}^1 < 0 & i=\alpha,\ldots,\beta \\ 0 & \text{otherwise} \end{cases}. \tag{41}$$

Since $\tilde{X}_i', \tilde{X}_i^2 \in \Psi(\tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2)$ and $\Psi(\tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2)$ is convex, Y is a feasible direction at $\tilde{X}_i'$ in $\Psi(\tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2)$.

Let

$$\tilde{X}_i'' = [x_{i-1,1}^1, \ldots, x_{i-\alpha+1,\alpha-1}^1, x_{i-\alpha,\alpha}^2, \ldots, x_{i-\beta,\beta}^2, x_{i-\beta-1,\beta+1}^1, \ldots, x_{i-M,M}^1].$$

We can similarly prove that $\tilde{X}_i'' \in \Psi(\tilde{X}_{i-1}^1, \tilde{X}_{i+1}^1)$. Since $Y = \tilde{X}_i'' - \tilde{X}_i^1$ and $\tilde{X}_i'', \tilde{X}_i^1 \in \Psi(\tilde{X}_{i-1}^1, \tilde{X}_{i+1}^1)$ and $\Psi(\tilde{X}_{i-1}^1, \tilde{X}_{i+1}^1)$ is convex, Y is also a feasible direction at $\tilde{X}_i^1$ in $\Psi(\tilde{X}_{i-1}^1, \tilde{X}_{i+1}^1)$.

It follows from (40) that $Q(\tilde{X}_i^2 | \tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2) < Q(\tilde{X}_i' | \tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2)$. From the convexity of the Q-problem, Y is a decreasing direction at $\tilde{X}_i'$, that is

$$\mathcal{D}_Y Q(\tilde{X}_i' | \tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2) < 0. \tag{42}$$

We can compute $\mathcal{D}_Y Q(\tilde{X}_i' | \tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2)$ by using Lemma 6 and Lemma 7 (see Appendix) together with (41) as follows:

$$\begin{aligned}
&\mathcal{D}_Y Q(\tilde{X}_i' | \tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2) \\
&= \sum_{j=\alpha}^{\beta} y_j \theta'\left(x_{i-j,j}^1 - \max(x_{i-j,j-1}^2, x_{i-j-1,j}^2)\right) \\
&\quad - \sum_{j=\alpha+1}^{\beta} \left(\mathcal{D}_{[y_{j-1},y_j]} \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1)\right) \\
&\quad \cdot \theta'\left(x_{i-j+1,j}^2 - \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1)\right) \\
&\quad - \left(\mathcal{D}_{[y_\alpha,0]} \max(x_{i-\alpha,\alpha}^1, x_{i-\alpha+1,\alpha-1}^2)\right) \\
&\quad \cdot \theta'\left(x_{i-\alpha+1,\alpha}^2 - \max(x_{i-\alpha,\alpha}^1, x_{i-\alpha+1,\alpha-1}^2)\right) \\
&\quad - \left(\mathcal{D}_{[y_\beta,0]} \max(x_{i-\beta,\beta}^1, x_{i-\beta-1,\beta+1}^2)\right) \\
&\quad \cdot \theta'\left(x_{i-\beta,\beta+1}^2 - \max(x_{i-\beta,\beta}^1, x_{i-\beta-1,\beta+1}^2)\right) \\
&= \sum_{j=\alpha}^{\beta} y_j \theta'\left(x_{i-j,j}^1 - \max(x_{i-j,j-1}^2, x_{i-j-1,j}^2)\right) \\
&\quad - \sum_{j=\alpha+1}^{\beta} \left(\mathcal{D}_{[y_{j-1},y_j]} \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1)\right) \\
&\quad \cdot \theta'\left(x_{i-j+1,j}^2 - \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1)\right) \\
&\quad - y_\alpha \mathbf{1}_{[x_{i-\alpha,\alpha}^1 > x_{i-\alpha+1,\alpha-1}^2]} \theta'\left(x_{i-\alpha+1,\alpha}^2 - x_{i-\alpha,\alpha}^1\right) \\
&\quad - y_\beta \mathbf{1}_{[x_{i-\beta,\beta}^1 > x_{i-\beta-1,\beta+1}^2]} \theta'\left(x_{i-\beta,\beta+1}^2 - x_{i-\beta,\beta}^1\right).
\end{aligned} \tag{43}$$

Similarly, we can obtain

$$\begin{aligned}
&\mathcal{D}_Y Q(\tilde{X}_i^1 | \tilde{X}_{i-1}^1, \tilde{X}_{i+1}^1) \\
&= \sum_{j=\alpha}^{\beta} y_j \theta'\left(x_{i-j,j}^1 - \max(x_{i-j,j-1}^1, x_{i-j-1,j}^1)\right) \\
&\quad - \sum_{j=\alpha+1}^{\beta} \left(\mathcal{D}_{[y_{j-1},y_j]} \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1)\right) \\
&\quad \cdot \theta'\left(x_{i-j+1,j}^1 - \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1)\right) \\
&\quad - y_\alpha \mathbf{1}_{[x_{i-\alpha,\alpha}^1 > x_{i-\alpha+1,\alpha-1}^1]} \theta'\left(x_{i-\alpha+1,\alpha}^1 - x_{i-\alpha,\alpha}^1\right) \\
&\quad - y_\beta \mathbf{1}_{[x_{i-\beta,\beta}^1 > x_{i-\beta-1,\beta+1}^1]} \theta'\left(x_{i-\beta,\beta+1}^1 - x_{i-\beta,\beta}^1\right).
\end{aligned} \tag{44}$$

Let us now compare each term of (44) with the corresponding term in (43). First, since $y_j < 0$ for $j = \alpha, \ldots, \beta$, $x_{i-j,j-1}^1 \le x_{i-j,j-1}^2$, $x_{i-j-1,j}^1 \le x_{i-j-1,j}^2$, $\theta'(\cdot) < 0$ (because $\theta(\cdot)$ is monotonically decreasing), and $\theta'(\cdot)$ is monotonically nondecreasing (because $\theta(\cdot)$ is convex), we have

$$\begin{aligned}
&y_j \theta'\left(x_{i-j,j}^1 - \max(x_{i-j,j-1}^1, x_{i-j-1,j}^1)\right) \\
&\qquad \le y_j \theta'\left(x_{i-j,j}^1 - \max(x_{i-j,j-1}^2, x_{i-j-1,j}^2)\right).
\end{aligned} \tag{45}$$

Second, since $\mathcal{D}_{[y_{j-1},y_j]} \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1; y_{j-1}, y_j) < 0$ (because $y_{j-1} < 0$ and $y_j < 0$), $x_{i-j+1,j}^1 \le x_{i-j+1,j}^2$, $\theta'(\cdot) < 0$ and $\theta'(\cdot)$ is monotonically nondecreasing, we have

$$\begin{aligned}
&-\left(\mathcal{D}_{[y_{j-1},y_j]} \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1)\right) \\
&\quad \cdot \theta'\left(x_{i-j+1,j}^1 - \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1)\right) \\
&\le -\left(\mathcal{D}_{[y_{j-1},y_j]} \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1)\right) \\
&\quad \cdot \theta'\left(x_{i-j+1,j}^2 - \max(x_{i-j+1,j-1}^1, x_{i-j,j}^1)\right).
\end{aligned} \tag{46}$$

Third, since $x_{i-\alpha+1,\alpha-1}^1 \le x_{i-\alpha+1,\alpha-1}^2$, $y_\alpha < 0$, $x_{i-\alpha+1,\alpha}^1 \le x_{i-\alpha+1,\alpha}^2$, $\theta'(\cdot) < 0$ and $\theta'(\cdot)$ is monotonically nondecreasing, we have

$$\begin{aligned}
&-y_\alpha \mathbf{1}_{[x_{i-\alpha,\alpha}^1 > x_{i-\alpha+1,\alpha-1}^1]} \theta'\left(x_{i-\alpha+1,\alpha}^1 - x_{i-\alpha,\alpha}^1\right) \\
&\qquad \le -y_\alpha \mathbf{1}_{[x_{i-\alpha,\alpha}^1 > x_{i-\alpha+1,\alpha-1}^2]} \theta'\left(x_{i-\alpha+1,\alpha}^2 - x_{i-\alpha,\alpha}^1\right).
\end{aligned} \tag{47}$$

Finally, since $x_{i-\beta-1,\beta+1}^1 \le x_{i-\beta-1,\beta+1}^2$, $y_\beta < 0$, $x_{i-\beta,\beta+1}^1 \le x_{i-\beta,\beta+1}^2$, $\theta'(\cdot) < 0$ and $\theta'(\cdot)$ is monotonically nondecreasing, we have

$$\begin{aligned}
&-y_\beta \mathbf{1}_{[x_{i-\beta,\beta}^1 > x_{i-\beta-1,\beta+1}^1]} \theta'\left(x_{i-\beta,\beta+1}^1 - x_{i-\beta,\beta}^1\right) \\
&\qquad \le -y_\beta \mathbf{1}_{[x_{i-\beta,\beta}^1 > x_{i-\beta-1,\beta+1}^2]} \theta'\left(x_{i-\beta,\beta+1}^2 - x_{i-\beta,\beta}^1\right).
\end{aligned} \tag{48}$$

Comparing (43) to (44) in view of (45) through (48), we get

$$\mathcal{D}_Y Q(\tilde{X}_i^1 | \tilde{X}_{i-1}^1, \tilde{X}_{i+1}^1) \le \mathcal{D}_Y Q(\tilde{X}_i' | \tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2). \tag{49}$$

Recalling (39) and the fact that Y is a feasible solution at $\tilde{X}_i^1$ in $\Psi(\tilde{X}_{i-1}^1, \tilde{X}_{i+1}^1)$, it follows that

$$\mathcal{D}_Y Q(\tilde{X}_i^1 | \tilde{X}_{i-1}^1, \tilde{X}_{i+1}^1) \geq 0.$$

Combining this inequality with (49), we get

$$\mathcal{D}_Y Q(\tilde{X}_i' | \tilde{X}_{i-1}^2, \tilde{X}_{i+1}^2) \geq 0$$

, which contradicts (42) and completes the proof. ■

Before getting to the main convergence result regarding the sequence $\{X^k\}$, we establish one more monotonicity property which applies to the sequence $\{\Delta^k\}$, $k = 1, 2, \ldots$

Lemma 5: The sequence $\{\Delta^k\}$, $k = 1, 2, \ldots$, is monotonically nonincreasing.

Proof: We use an induction argument over $k = 1, 2, \ldots$ For $k = 1$, recall the initial condition $\Delta_j^1 = D$ for $j = 1, \ldots, M$. Thus, Δ_j^2 must be no larger than Δ_j^1 for $j = 1, \ldots, M$, that is, $\Delta^1 \geq \Delta^2$. Assume that $\Delta^n \geq \Delta^{n+1}$. Next, we prove that $\Delta^{n+1} \geq \Delta^{n+2}$.

Since $\Delta^n \geq \Delta^{n+1}$, we have $\Delta_j^n \geq \Delta_j^{n+1}$ for all $j = 1, \ldots, M$. Beginning with stage 1, since $X_0^n = X_0^{n+1} = A$, $\Delta_1^n \geq \Delta_1^{n+1}$ and (26) holds for X_1^n, X_1^{n+1}, we have $X_1^n \geq X_1^{n+1}$ based on Lemma 3. Then, proceeding to stage 2, from $X_1^n \geq X_1^{n+1}$, $\Delta_2^n \geq \Delta_2^{n+1}$ and (26) for X_2^n, X_2^{n+1}, we can similarly apply Lemma 3 to get $X_2^n \geq X_2^{n+1}$. Repeating this process over all stages $j = 1, \ldots, M$, we finally have $X^n \geq X^{n+1}$.

Recalling (19), we have $\tilde{X}^n = \mathcal{T}(X^n)$, $\tilde{X}^{n+1} = \mathcal{T}(X^{n+1})$ and since $X^n \geq X^{n+1}$ it follows that $\tilde{X}_i^n \geq \tilde{X}_i^{n+1}$ for $i = 1, \ldots, N + M + 1$. Combining this with (27) and applying Lemma 4, we get $\tilde{\Delta}_i^{n+1} \geq \tilde{\Delta}_i^{n+2}$ for $i = 2, \ldots, N + M$, that is, $\tilde{\Delta}^{n+1} \geq \tilde{\Delta}^{n+2}$. Finally, since $\Delta^{n+1} = \mathcal{T}^{-1}(\tilde{\Delta}^{n+1})$ and $\Delta^{n+2} = \mathcal{T}^{-1}(\tilde{\Delta}^{n+2})$, we have $\Delta^{n+1} \geq \Delta^{n+2}$ which completes the proof. ■

Theorem 3: The sequence $\{X^k\}$, $k = 1, 2, \ldots$, is monotonically nonincreasing and

$$\lim_{k \to \infty} X^k = X^*.$$

Proof: For convenience, we divide the proof into three parts. First, we prove the inequality $X^k \geq \Delta^k \geq X^{k+1}$ for all $k = 1, 2, \ldots$ Next, we show that $X^k \geq X^*$ for all $k = 1, 2, \ldots$, where X^* is the solution of the original multi-stage problem. Combining these two results, we finally complete the convergence proof.

Part 1: We will prove the inequality below for all $k \geq 1$

$$X^k \geq \Delta^{k+1} \geq X^{k+1}. \tag{50}$$

Since $X_j^{k+1} \in \Phi(X_{j-1}^{k+1}, \Delta_j^{k+1})$ in (9) for $j = 1, \ldots, M$, we immediately get $\Delta^{k+1} \geq X^{k+1}$. Thus, it remains to prove that $X^k \geq \Delta^{k+1}$, or equivalently $\tilde{X}_i^k \geq \tilde{\Delta}_i^{k+1}$ for $i = 2, \ldots, N + M$. We can prove $\tilde{X}_i^k \geq \tilde{\Delta}_i^{k+1}$, that is, recalling (17), $x_{i-j,j}^k \geq \delta_{i-j,j}^{k+1}$ for $j = 1, \ldots, M$, by the following inductive argument over j. Since $x_{i,0}^k = \delta_{i,0}^{k+1} = a_i$, the inequality obviously holds for $j = 0$. If we can prove that, for $j \geq 1$

$$x_{i-j+1,j-1}^k \geq \delta_{i-j+1,j-1}^{k+1} \Longrightarrow x_{i-j,j}^k \geq \delta_{i-j,j}^{k+1} \tag{51}$$

then we can conclude $\tilde{X}_i^k \geq \tilde{\Delta}_i^k$. In what follows, we establish (51). Looking at $x_{i-j,j}^k$, there are two possible cases: *(i)* $\delta_{i-j,j}^k = x_{i-j,j}^k$ and *(ii)* $\delta_{i-j,j}^k > x_{i-j,j}^k$.

For case *(i)*, from Lemma 5, we have $\delta_{i-j,j}^k \geq \delta_{i-j,j}^{k+1}$. Combining this with $\delta_{i-j,j}^k = x_{i-j,j}^k$, we have $x_{i-j,j}^k \geq \delta_{i-j,j}^{k+1}$ as desired.

For case *(ii)*, let $Y = [y_1, \ldots, y_N]^T$ such that

$$y_n = \begin{cases} 1 & n = i - j \\ 0 & \text{otherwise} \end{cases}.$$

We can compute $\mathcal{D}_Y L(X_j^k)$ using Lemma 6 and Lemma 7 (see Appendix) as follows:

$$\begin{aligned}\mathcal{D}_Y L(X_j^k) = &\theta'\left(x_{i-j,j}^k - \max(x_{i-j-1,j}^k, x_{i-j,j-1}^k)\right) \\ &-\mathbf{1}_{[x_{i-j,j}^k \geq x_{i-j+1,j-1}^k]} \theta'\left(x_{i-j+1,j}^k - x_{i-j,j}^k\right).\end{aligned} \tag{52}$$

Since $\delta_{i-j,j}^k > x_{i-j,j}^k$, the direction Y is feasible in $\Phi(X_{j-1}^k, \Delta_j^k)$. Then, from (27), we have

$$\mathcal{D}_Y L(X_j^k) \geq 0. \tag{53}$$

Now let $Y' = [y_1', \ldots, y_M']^T$ and

$$\tilde{\Delta}_i' = [\delta_{i-1,1}^{k+1}, \ldots, \delta_{i-j+1,j-1}^{k+1}, x_{i-j,j}^k, \delta_{i-j-1,j+1}^{k+1}, \ldots, \delta_{i-M,M}^{k+1}]^T$$

such that

$$y_n' = \begin{cases} 1 & n = j \\ 0 & \text{otherwise} \end{cases}. \tag{54}$$

We can similarly obtain $\mathcal{D}_{Y'} Q(\tilde{\Delta}_i' | \tilde{X}_{i-1}^k, \tilde{X}_{i+1}^k)$ using Lemma 6 and Lemma 7 (see Appendix):

$$\begin{aligned}\mathcal{D}_{Y'} &Q(\tilde{\Delta}_i' | \tilde{X}_{i-1}^k, \tilde{X}_{i+1}^k) \\ &= \theta'\left(x_{i-j,j}^k - \max(x_{i-j-1,j}^k, x_{i-j,j-1}^k)\right) \\ &\quad - \mathbf{1}_{[x_{i-j,j}^k \geq \delta_{i-j+1,j-1}^{k+1}]} \theta'\left(x_{i-j+1,j}^k - x_{i-j,j}^k\right) \\ &\quad - \mathbf{1}_{[x_{i-j,j}^k \geq \delta_{i-j-1,j+1}^{k+1}]} \theta'\left(x_{i-j,j+1}^k - x_{i-j,j}^k\right).\end{aligned} \tag{55}$$

Since we have already established the first inequality in (51), we know that $x_{i-j+1,j-1}^k \geq \delta_{i-j+1,j-1}^{k+1}$ which implies that

$$\mathbf{1}_{[x_{i-j,j}^k \geq \delta_{i-j+1,j-1}^{k+1}]} \geq \mathbf{1}_{[x_{i-j,j}^k \geq x_{i-j+1,j-1}^k]}$$

and since $\theta'(\cdot) < 0$ (because $\theta(\cdot)$ is monotonically decreasing), we have

$$\begin{aligned}-\mathbf{1}_{[x_{i-j,j}^k \geq \delta_{i-j+1,j-1}^{k+1}]} &\theta'\left(x_{i-j+1,j}^k - x_{i-j,j}^k\right) \\ &\geq -\mathbf{1}_{[x_{i-j,j}^k \geq x_{i-j+1,j-1}^k]} \theta'\left(x_{i-j+1,j}^k - x_{i-j,j}^k\right).\end{aligned} \tag{56}$$

Combining (52) through (56) and the fact that $\theta'(\cdot) < 0$, we have for $y > 0$:

$$\begin{aligned}\mathcal{D}_{Y'} &Q(\tilde{\Delta}_i' | \tilde{X}_{i-1}^k, \tilde{X}_{i+1}^k) \\ &\geq \mathcal{D}_Y L(X_j^k) - \mathbf{1}_{[x_{i-j,j}^k \geq \delta_{i-j-1,j+1}^{k+1}]} \theta'\left(x_{i-j,j+1}^k - x_{i-j,j}^k\right) \\ &\geq -\mathbf{1}_{[x_{i-j,j}^k \geq \delta_{i-j-1,j+1}^{k+1}]} \theta'\left(x_{i-j,j+1}^k - x_{i-j,j}^k\right) \geq 0.\end{aligned} \tag{57}$$

It follows from (57) that Y' is not a decreasing direction at $\tilde{\Delta}_i'$ when minimizing $Q(\tilde{X}_i | \tilde{X}_{i-1}^k, \tilde{X}_{i+1}^k)$. Recalling (27) and the convexity of the Q-problem, $\tilde{\Delta}_i^{k+1}$ cannot be obtained by searching along the direction Y', that is, $\tilde{\Delta}_i^{k+1} \neq \tilde{\Delta}_i' + \alpha Y'$ for any $\alpha > 0$. By the definition of Y' in (54), this implies that

$$\delta_{i-j,j}^{k+1} \neq x_{i-j,j}^k + \alpha, \quad \forall\, \alpha > 0$$

that is, $x_{i-j,j}^k \geq \delta_{i-j,j}^{k+1}$, which completes the proof of (51) and, hence, the proof of (50).

Part 2: We will now prove the inequalities below for $k \geq 1$

$$\Delta^k \geq X^* \text{ and } \ X^k \geq X^*, \text{ for } \ k = 1, 2, \ldots. \tag{58}$$

We will use an induction argument over $k = 1, 2, \ldots$ We begin with $k = 1$. Since $\Delta_j^1 = D$ and $D \geq X_j^*$ for $j = 1, \ldots, M$, it follows that $\Delta_j^1 \geq X_j^*$ for $j = 1, \ldots, M$, that is, $\Delta^1 \geq X^*$. Let $\Delta = X^*$ and recall Theorem 1, based on which we have

$$X_j^* = \arg \min_{X_j \in \Phi(X_{j-1}^*, \Delta_j)} L(X_j | X_{j-1}^*), \quad \text{for } \ j = 1, \ldots, M. \tag{59}$$

Recall that $X_0^1 = X_0^* = A$. Using $\Delta_1^1 \geq \Delta_1 = X_1^*$, (59) and (26), we can apply Lemma 3 to get $X_1^1 \geq X_1^*$. Then, proceeding to stage 2, from $X_1^1 \geq X_1^*$, $\Delta_2^1 \geq \Delta_2 = X_2^*$, (59) and (26), we can similarly obtain $X_2^1 \geq X_2^*$ from Lemma 3. Repeating the process above over all stages $j = 1, \ldots, M$, we can finally obtain $X_j^1 \geq X_j^*$ for $j = 1, \ldots, M$, that is, $X^1 \geq X^*$.

Next, assume that $\Delta^n \geq X^*$ and $X^n \geq X^*$. We shall prove that $\Delta^{n+1} \geq X^*$ and $X^{n+1} \geq X^*$. From Theorem 2, we know for $i = 2, \ldots, N + M$ that

$$\tilde{X}_i^* = \arg \min_{\tilde{X}_i \in \Psi(\tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*)} Q(\tilde{X}_i | \tilde{X}_{i-1}^*, \tilde{X}_{i+1}^*). \tag{60}$$

Since $X^n \geq X^*$, or equivalently from (19) $\tilde{X}_i^n \geq \tilde{X}_i^*$ for $i = 1, \ldots, N + M + 1$, and since (27) holds, we can apply Lemma 4 to get $\tilde{\Delta}_i^{n+1} \geq \tilde{X}_i^*$ for $i = 2, \ldots, N + M$, that is, $\Delta^{n+1} \geq X^*$. Since $\Delta = X^*$, we have $\Delta^{n+1} \geq \Delta$. Similar to the process above, we start with stage 1 and apply Lemma 3 with $X_0^{n+1} = X_0^* = A$, $\Delta_1^{n+1} \geq \Delta_1$, (59) and (26), to obtain $X_1^{n+1} \geq X_1^*$. Then, repeating the process over all stages $j = 1, \ldots, M$, we finally get $X_j^{n+1} \geq X_j^*$ for all $j = 1, \ldots, M$, that is, $X^{n+1} \geq X^*$.

Part 3: We now finally prove that

$$X^k \to X^* \quad \text{as } \ k \to \infty. \tag{61}$$

Let $W^{2k-1} = \Delta^k$ and $W^{2k} = X^k$. In view of (50), $\{W^k\}$ is a monotonically nonincreasing sequence. Combining it with (58), $\{W^k\}$ must converge to some vector X' such that $X' \geq X^*$ as $k \to \infty$, that is

$$W^k \to X' \quad \text{as } \ k \to \infty. \tag{62}$$

Let $\tilde{W}^{2k+1} = \mathcal{T}(W^{2k+1}) = \mathcal{T}(\Delta^{k+1})$ and $\tilde{W}^{2k} = \mathcal{T}(W^{2k}) = \mathcal{T}(X^k)$. Then, from (27), we have for $i = 2, \ldots, N + M$ that

$$W_i^{2k+1} = \arg \min_{\tilde{X}_i \in \Psi(\tilde{W}_{i-1}^{2k}, \tilde{W}_{i+1}^{2k})} Q(\tilde{X}_i | \tilde{W}_{i-1}^{2k}, \tilde{W}_{i+1}^{2k}).$$

TABLE I
VIRTUAL DEADLINE ALGORITHM

Step 1:	$k = 1$, $\Delta_j^k = D$, for $j = 1, ..., M$;
Step 2:	$X_0^k = A$, $X_j^k = \arg \min_{X_j \in \Phi(X_{j-1}^k, \Delta_j^k)} L(X_j \| X_{j-1}^k)$, for $j = 1, ..., M$; and $\tilde{X}^k = \mathcal{T}(X^k)$;
Step 3:	$\tilde{\Delta}_i^{k+1} = \arg \min_{\tilde{X}_i \in \Psi(\tilde{X}_{i-1}^k, \tilde{X}_{i+1}^k)} Q(\tilde{X}_i \| \tilde{X}_{i-1}^k, \tilde{X}_{i+1}^k)$, for $i = 2, ..., N+M$; and $\Delta^{k+1} = \mathcal{T}^{-1}(\tilde{\Delta}^{k+1})$;
Step 4:	if $\|X^k - X^{k-1}\| > \epsilon$, then $k = k + 1$ and Goto **Step 2**;
Step 5:	$X^* = X^k$.

Combining this with (62), we get for $i = 2, \ldots, N + M$ that

$$\tilde{X}_i' = \arg \min_{\tilde{X}_i \in \Psi(\tilde{X}_{i-1}', \tilde{X}_{i+1}')} Q(\tilde{X}_i | \tilde{X}_{i-1}', \tilde{X}_{i+1}'). \tag{63}$$

Using (63) and Theorem 2, it follows that $\tilde{X}_i' = \tilde{X}_i^*$ for $i = 2, \ldots, N + M$, or equivalently from (19)$X' = X^*$. It then follows from (62) that the sequence $\{W^k\}$ monotonically converges to X^*. Since $\{X^k\}$ is a subsequence of $\{W^k\}$, (61) indeed holds. ■

V. Virtual Deadline Algorithm

Based on the analysis above, the Virtual Deadline Algorithm (VDA) in Table I is a direct implementation of the sequence construction in (26)–(27) also illustrated in Fig. 3. The VDA provides a computationally efficient way to obtain X^* by exploiting the fact that each single-stage problem in Step 2 can be very efficiently solved with the CTDA in [20] (or its generalized version GCTDA [21]), while solving each Q-problem in Step 3, an M-dimensional convex optimization problem, is a relatively simple task. Theorem 3 guarantees that the VDA provides some X^k arbitrarily close to the global optimum of our original problem.

Moreover, the VDA can be further improved. Although each M-dimensional convex optimization problem in Step 3 is quite easy to solve, there are $N + M - 1$ such problems we need to solve in each iteration, a considerable effort, especially when N is very large. Proposition 4 below provides us a way to reduce the number of these problems by utilizing an optimality property of the single-stage problem, which reduces it to a simple procedure of identifying "critical tasks." As in [20] and [21], a "critical task" is defined as some task i such that

$$\theta'(s_i) \neq \theta'(s_{i+1}) \quad \text{or} \quad x_i < a_{i+1}$$

where $s_i = x_i - \max(x_{i-1}, a_i)$ and a_i is the arrival time of task i.

Proposition 4: If task $i - j$ is not a critical task when solving the single-stage problem (8) for all $j = 1, \ldots, M$ in the kth iteration, then $\tilde{\Delta}_i^{k+1} = \tilde{X}_i^k$.

Proof: Since, by assumption, task $i - j$ is not critical when solving the single stage problem (8) for all $j = 1, \ldots, M$ in the kth iteration, then, from the definition of a critical task above, we have for all $j = 1, \ldots, M$ that

$$\theta'(s_{i-j,j}^k) = \theta'(s_{i-j+1,j}^k) \quad \text{and} \quad x_{i-j,j}^k \geq x_{i-j+1,j-1}^k \tag{64}$$

where $s^k_{i-j,j} = x^k_{i-j,j} - \max(x^k_{i-j-1,j}, x^k_{i-j,j-1})$ and $x^k_{i-j,j-1}$ is the departure time of task $i-j$ from stage $j-1$ and regarded as the arrival time of task $i-j$ at stage j.

Proceeding by contradiction, assume that $\tilde{\Delta}^{k+1}_i \neq \tilde{X}^k_i$. Let $\tilde{Y}_i = \tilde{\Delta}^{k+1}_i - \tilde{X}^k_i$. From (27), we have

$$\mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}^k_i | \tilde{X}^k_{i-1}, \tilde{X}^k_{i+1}) < 0. \quad (65)$$

Since $x^k_{i-j,j} \geq x^k_{i-j+1,j-1}, j = 1, \ldots, M$ in (64), we can compute $\mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}^k_i | \tilde{X}^k_{i-1}, \tilde{X}^k_{i+1})$ using Lemmas 6 and 7 in the Appendix as

$$\mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}^k_i | \tilde{X}^k_{i-1}, \tilde{X}^k_{i+1}) = \sum_{j=1}^{M} \left(y_{i-j,j}\theta'(s^k_{i-j,j}) - f(y_{i-j,j})\theta'(s^k_{i-j+1,j}) \right) \quad (66)$$

where

$$f(y_{i-j,j}) = y_{i-j,j}\mathbf{1}_{[x^k_{i-j,j} > x^k_{i-j+1,j-1}]} + \max(y_{i-j,j}, y_{i-j+1,j-1})\mathbf{1}_{[x^k_{i-j,j} = x^k_{i-j+1,j-1}]}.$$

Using Part 1 in Theorem 3, we have $\tilde{\Delta}^{k+1}_i \leq \tilde{X}^k_i$, which implies that $\tilde{Y}_i \leq 0$. Combining this with the definition of $f(y_{i-j,j})$, we have

$$f(y_{i-j,j}) \geq y_{i-j,j}. \quad (67)$$

Since $\theta(\cdot)$ is monotonically decreasing, $\theta'(\cdot) < 0$. Moreover, $\theta'(s^k_{i-j,j}) = \theta'(s^k_{i-j+1,j}), j = 1, \ldots, M$ in (64), which implies that

$$\theta'(s^k_{i-j,j}) = \theta'(s^k_{i-j+1,j}) < 0. \quad (68)$$

Using (67) and (68), we have

$$f(y_{i-j,j})\theta'(s^k_{i-j+1,j}) \leq y_{i-j,j}\theta'(s^k_{i-j,j})$$

that is

$$y_{i-j,j}\theta'(s^k_{i-j,j}) - f(y_{i-j,j})\theta'(s^k_{i-j+1,j}) \geq 0.$$

Combining this with (66), we get

$$\mathcal{D}_{\tilde{Y}_i} Q(\tilde{X}^k_i | \tilde{X}^k_{i-1}, \tilde{X}^k_{i+1}) \geq 0$$

which contradicts (65) and completes the proof. ■

Based on Proposition 4, we only need to solve those Q-problems that involve critical tasks. Qualitatively, looser deadlines cause fewer critical tasks, which results in fewer Q-problems we need to solve. In the extreme case that all tasks share a common arrival time, deadline, and number of operations, we only need to solve $2M - 2$ Q-problems in each iteration, independent of N. Furthermore, even if the condition in Proposition 4 can not be satisfied when solving (27), $\tilde{X}^k_i$ can still be used as a very good initial point because it is feasible and close to the optimal solution $\tilde{\Delta}^{k+1}_i$.

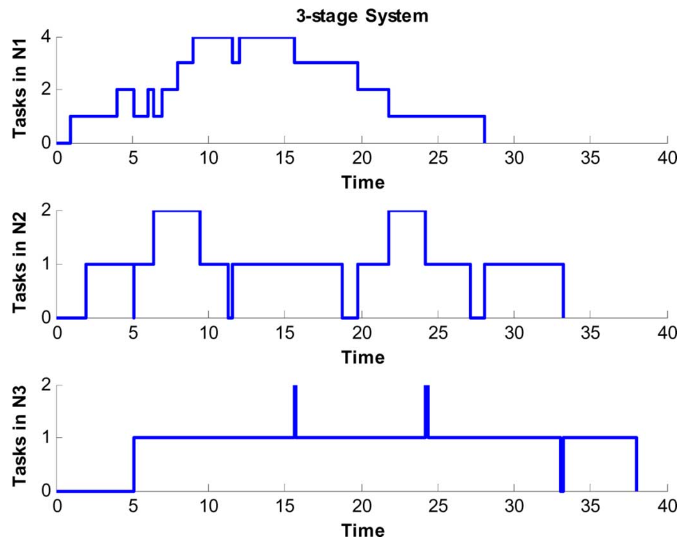


Fig. 4. Optimal sample path obtained by the VDA.

To make this improvement, we need to exploit and store the index of critical tasks when solving M single stage problems in Step 2. Since the CTDA solves the single-stage problem by identifying critical tasks, the only extra effort made in Step 2 is only to store their indices. Then, in Step 3, we only need to introduce a check to determine whether the Q-problem involves critical tasks or not. If so, we solve it by using $\tilde{X}^k_i$ as the initial point; otherwise, we directly set $\tilde{\Delta}^{k+1}_i = \tilde{X}^k_i$. Therefore, the overhead involved is minor and can drastically reduce the complexity of Step 3.

VI. Numerical Results

A. Example

We have applied the VDA to a 3-stage system with $N = 8$, arrival time vector $A = [1\ 2\ 4\ 6\ 7\ 8\ 9\ 12]^T$ and deadline vector $D = [20\ 22\ 24\ 26\ 29\ 31\ 33\ 38]^T$. Each task i is characterized by a number of operations $\mu_{i,j}$ at stage j with corresponding vectors $\mu_1 = [1\ 3\ 1\ 4\ 3\ 3\ 1\ 3]^T$, $\mu_2 = [2\ 3\ 1\ 1\ 1\ 3\ 2\ 3]^T$, $\mu_3 = [2\ 1\ 3\ 2\ 4\ 2\ 5\ 2]^T$. The cost functions are

$$\theta_{i,j}(s_{i,j}) = k_j \cdot \frac{\mu_{i,j}}{\left(\frac{s_{i,j}}{\mu_{i,j}} + 0.001\right)}$$

where $[k_1\ k_2\ k_3] = [3\ 2\ 4]$. The termination condition in step 4 of the VDA is set so that

$$\sum_{j=1}^{M} \frac{\left\| \Delta^{k+1}_j - \Delta^k_j \right\|}{M} = \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{\frac{(\delta^k_{i,j} - \delta^{k+1}_{i,j})}{N}}{M} < \epsilon = 0.001.$$

The optimal sample path obtained by the VDA is shown in Fig. 4, plotting the number of tasks in each stage as a function of time. Fig. 5 shows the optimal processing rates, where the length of a block is the service time of the corresponding task and the block label is its number of operations.

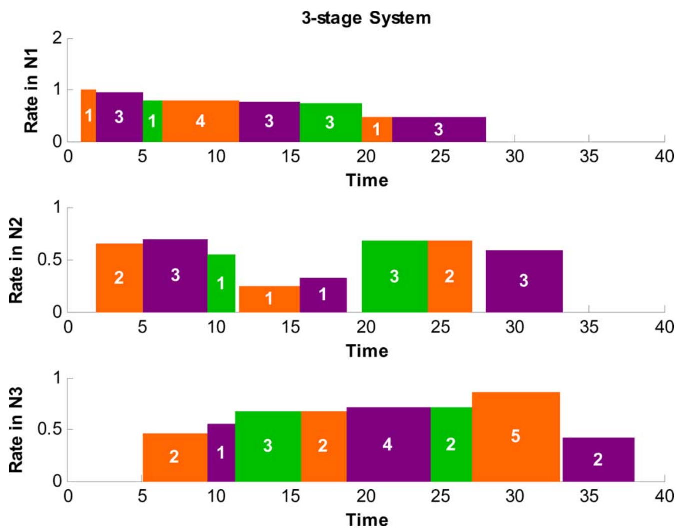


Fig. 5. Optimal processing rate obtained by the VDA.

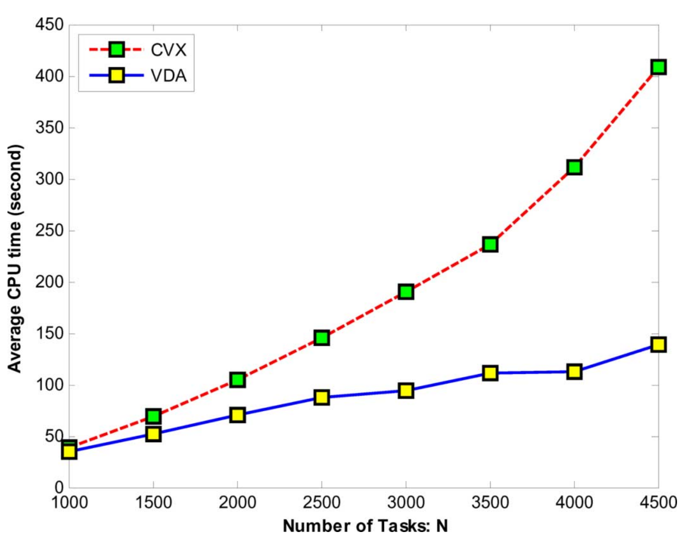


Fig. 6. VDA versus CVX.

B. Complexity

We proceed to a more complicated example with $M = 5$ and test the complexity of the VDA in terms of CPU time compared to CVX, an efficient convex programming solver [13]. In these tests, the VDA was programmed using Matlab 7.0 on an Intel Pentium4 3.06 GHz, 1.0 GB RAM machine. We tested cases where N varied from 1000 to 4500 in increments of 500. We randomly generated 20 samples for each N. For each case, we recorded the elapsed CPU time when both reached the same precision, finally averaging them to obtain the corresponding performance.

Fig. 6 shows the average CPU time (in seconds) as a function of the number of tasks N, where the dash and solid lines represent the results of CVX and VDA respectively. We observe that the VDA's complexity scales with N while CVX does not possess this property. This is because the VDA only needs to solve an M-dimensional ($M = 5$ in this example) convex optimization problem no matter what N is and the CTDA (utilized in the VDA) also scales with N. Furthermore, CVX could not guarantee finding a solution in this example (i.e., its obtained solution may be "NaN" for some sample paths) when N increases to 3000 and beyond, while the VDA can always derive the optimal solution because of the robustness of the CTDA and the M-dimensional convex programming solver.

VII. Conclusion

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 $1, \ldots, M-1$, and then solving partially decoupled single-stage problems whose solutions are known to be efficiently obtained. Based on this idea, we have shown that a sequence of solutions to simpler problems converges to the global optimum of the original problem and have developed an iterative Virtual Deadline Algorithm (VDA) implementing this approach. 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, 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 serial multi-stage case considered in this paper.

Appendix

Theorem 4.3.2 in [11]: Let f be a convex function, S be a convex set and $x^0 \in int(dom(f))$ be a feasible point. Then x^0 is a (global) minimum point of the problem (P_c) if and only if the following condition hold:

$$f'(x^0, y) \geq 0, \quad \forall y \in T(S, x^0).$$

Remark: $dom(f)$ denotes the domain of function f and $int(dom(f))$ is the set of interior points of $dom(f)$. Problem P_c is the convex nonlinear programming problem:

$$\min_{x \in S} f(x)$$

where S is a convex set and $f(x)$ is a convex function with respect to x. $f'(x^0, y)$ is the directional derivative of the function f at x^0 along the direction y, which is equivalent to $\mathcal{D}_y f(x)$ defined in this paper. $T(S, x^0)$ represents a cone defined below:

$$T(S, x^0) = \left\{ y \in \mathbb{R}^n \mid \exists \{t^k\} \subset \mathbb{R}^+, \exists \{x^k\} \subset S \right.$$
$$\left. : \lim_{k \to +\infty} t^k = 0, \lim_{k \to +\infty} \frac{x^k - x^0}{t^k} = y \right\}$$

which can be equivalently regarded as the set of feasible directions at x^0 in the convex optimization problems (6) and (15) because their feasible sets are polyhedral.

Lemma 6: Let $h(x) = f(g(x))$, where $f(\cdot)$ and $g(\cdot)$ are scalar functions, $f(\cdot)$ is differentiable and $g(\cdot)$ is continuous. Then it holds for any feasible direction y that

$$\mathcal{D}_y h(x) = \mathcal{D}_y g(x) f'(x).$$

Proof: By the definition of the directional derivative [3], we have

$$\begin{aligned}\mathcal{D}_y h(x) &= \lim_{\alpha \downarrow 0} \frac{h(x+\alpha y) - h(x)}{\alpha} \\ \mathcal{D}_y g(x) &= \lim_{\alpha \downarrow 0} \frac{g(x+\alpha y) - g(x)}{\alpha}. \end{aligned} \quad (69)$$

Let $H(\alpha) = h(x+\alpha y)$, $G(\alpha) = g(x+\alpha y)$, $H^+(\alpha)$ and let $G^+(\alpha)$ denote the right derivative of $H(\alpha)$ and $G(\alpha)$ respectively. Using (69), we can easily verify that

$$\begin{aligned}H^+(0) &= \lim_{\delta \downarrow 0} \frac{H(\delta) - H(0)}{\delta} = \mathcal{D}_y h(x) \\ G^+(0) &= \lim_{\delta \downarrow 0} \frac{G(\delta) - G(0)}{\delta} = \mathcal{D}_y g(x). \end{aligned} \quad (70)$$

By the definition of $H(\alpha)$ and $G(\alpha)$, we have $H(\alpha) = f(G(\alpha))$. Since $g(x)$ is continuous and y is a feasible direction, the right derivative $G^+(0)$ always exists, which implies that

$$G(\delta) - G(0) = \delta G^+(0) + \epsilon(\delta)\delta, \quad \text{where } \epsilon(\delta) \to 0 \text{ as } \delta \downarrow 0.$$

Since $f(\cdot)$ is differentiable, we have

$$f(G(0)+\beta) - f(G(0)) = \beta f'(G(0)) + \eta(\beta)\beta,$$

where $\eta(\beta) \to 0$ as $\beta \to 0$. Then,

$$\begin{aligned}f(G(\delta)) - f(G(0)) &= f\big(G(0) + \delta G^+(0) + \epsilon(\delta)\delta\big) - f(G(0)) \\ &= \beta_\delta f'(G(0)) + \eta(\beta_\delta)\beta_\delta\end{aligned}$$

where $\beta_\delta = \delta G^+(0) + \epsilon(\delta)\delta$. Observe that as $\delta \downarrow 0$, $\beta_\delta/\delta \to G^+(0)$ and $\beta_\delta \to 0$, thus $\eta(\beta_\delta) \to 0$. Therefore,

$$\begin{aligned}H^+(0) &= \lim_{\delta \downarrow 0} \frac{f(G(\delta)) - f(G(0))}{\delta} \\ &= \lim_{\delta \downarrow 0} \frac{\beta_\delta f'(G(0)) + \eta(\beta_\delta)\beta_\delta}{\delta} = G^+(0) f'(G(0)).\end{aligned}$$

Combining it with (70), we have $\mathcal{D}_y h(x) = \mathcal{D}_y g(x) f'(g(x))$. ■

Lemma 7:

$$\begin{aligned}&\mathcal{D}_{[y_1,y_2]} \max(x_1,x_2) \\ &\quad = y_1 \mathbf{1}_{[x_1>x_2]} + y_2 \mathbf{1}_{[x_1<x_2]} + \max(y_1,y_2)\mathbf{1}_{[x_1=x_2]}\end{aligned}$$

Proof: By the definition of directional derivative, we have

$$\begin{aligned}&\mathcal{D}_{[y_1,y_2]} \max(x_1,x_2) \\ &\quad = \lim_{\alpha \downarrow 0} \frac{\max(x_1+\alpha y_1, x_2+\alpha y_2) - \max(x_1,x_2)}{\alpha}.\end{aligned}$$

There are three possible cases when computing $\mathcal{D}_{[y_1,y_2]} \max(x_1,x_2)$. *Case 1*: $x_1 > x_2$. There always exists some $\epsilon > 0$ such that

$$\max(x_1+\alpha y_1, x_2+\alpha y_2) = x_1 + \alpha y_1, \quad \forall 0 \le \alpha < \epsilon.$$

Therefore

$$\mathcal{D}_{[y_1,y_2]} \max(x_1,x_2) = \lim_{\alpha \downarrow 0} \frac{x_1 + \alpha y_1 - x_1}{\alpha} = y_1$$

Case 2: $x_1 < x_2$. There always exists some $\epsilon > 0$ such that

$$\max(x_1+\alpha y_1, x_2+\alpha y_2) = x_2 + \alpha y_2, \quad \forall 0 \le \alpha < \epsilon.$$

Therefore

$$\mathcal{D}_{[y_1,y_2]} \max(x_1,x_2) = \lim_{\alpha \downarrow 0} \frac{x_2 + \alpha y_2 - x_2}{\alpha} = y_2$$

Case 3: $x_1 = x_2$. It follows that:

$$\max(x_1+\alpha y_1, x_2+\alpha y_2) = x_1 + \alpha \max(y_1,y_2), \quad \forall \alpha \ge 0.$$

Therefore

$$\begin{aligned}\mathcal{D}_{[y_1,y_2]} \max(x_1,x_2) &= \lim_{\alpha \downarrow 0} \frac{x_1 + \alpha \max(y_1,y_2) - x_1}{\alpha} \\ &= \max(y_1,y_2).\end{aligned}$$

Combining these three cases, the lemma immediately follows. ■

REFERENCES

[1] H. Aydin, R. Melhem, D. Mossé, and P. Mejia-Alvarez, "Power-aware scheduling for periodic real-time tasks," *IEEE Trans. Computers*, vol. 53, no. 5, pp. 584–600, May 2004.

[2] F. Baccelli, G. Cohen, G. J. Olsder, and J. P. Quadrat, *Synchronization and linearity: An Algebra for Discrete Event Systems*. New York: Wiley, 1992.

[3] D. P. Bertsekas, *Nonlinear Programming*, Second ed. Belmont, MA: Athena Scientific, 1999.

[4] C. G. Cassandras, Q. Liu, K. Gokbayrak, and D. L. Pepyne, "Optimal control of a two-stage hybrid manufacturing system model," in *Proc. 38th IEEE Conf. Decision Control*, 1999, pp. 450–455.

[5] 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, no. 5, pp. 497–513, 2001.

[6] G. Cohen, S. Gaubert, and J. P. Quadrat, "From first to second-order theory of linear discrete event systems," in *Proc. 1st IFAC World Congress*, Sydney, Australia, Jul. 1993, pp. 18–23.

[7] R. Cunninghame-Green, *Minimax Algebra*. Berlin, Germany: Springer, 1979.

[8] P. Declerck, "Estimation, prediction and control in (max,+) systems," in *Proc. 1st IFAC Symp. Syst. Struct. Control, Workshop Max-Plus Algebras*, 2001, pp. 27–29.

[9] P. Declerck, "Control synthesis using the state equations and the ARMA model in timed event graphs," in *Proc. 5th IEEE Mediterranean Conf. Control Syst.*, Paphos, Cyprus, Jul. 1997, pp. 21–23.

[10] A. E. Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu, and S. Zahedi, "Energy-efficient scheduling of packet transmissions over wireless networks," in *Proc. IEEE INFOCOM*, New York, NY, 2002, vol. 3, 23–27, pp. 1773–1782.

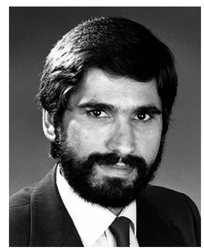
[11] C. Giorgi, A. Guerraggio, and J. Thierfelder, *Mathematics of Optimization: Smooth and Nonsmooth Case*. Murray Hill, NJ: Elsevier, 2004.

[12] K. Gokbayrak and C. G. Cassandras, "Constrained optimal control for multistage hybrid manufacturing system models," in *Proc. 8th IEEE Mediterranean Conf. New Directions Control Autom.*, Jul. 2000, [CD ROM].

[13] M. Grant, S. Boyd, and Y. Ye, "Disciplined convex programming," in *Global Optimization: From Theory to Implementation*, L. Liberti and N. Maculan, Eds. Berlin, Germany: Springer, 2006, pp. 155–210.
[14] B. Heidergott, G. J. Olsder, and J. Woude*, Max-Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and its Applications*. Princeton, NJ: Princeton Univ. Press, 2006.
[15] K. Jeffay, D. F. Stanat, and C. U. Martel, "On non-preemptive scheduling of periodic and sporadic tasks," in *Proc. IEEE Real-Time Systems Symp.*, 1991, pp. 129–139.
[16] J. Jonsson, H. Lonn, and K. G. Shin, "Non-preemptive scheduling of real-time threads on multi-level-context architectures," in *Proc. IEEE Workshop Parallel Distrib. Real-Time Syst.*, 1999, vol. 1586, pp. 363–374.
[17] J. W. S. Liu*, Real – Time Systems*. Englewood Cliffs, MJ: Prentice Hall, 2000.
[18] J. Mao and C. G. Cassandras, "Optimal admission control of discrete event systems with real-time constraints," in *Proc. 46rd IEEE Conf. Decision Control*, Dec. 2007, pp. 3005–3010.
[19] J. Mao and C. G. Cassandras, "Optimal control of two-stage discrete event systems with real-time constraints," *J. Discrete Event Dyn. Syst.*, vol. 17, no. 4, pp. 505–529, 2007.
[20] J. Mao, C. G. Cassandras, and Q. C. Zhao, "Optimal dynamic voltage scaling in power-limited systems with real-time constraints," *IEEE Trans. Mobile Computing*, vol. 6, no. 6, pp. 678–688, Jun. 2007.
[21] L. Miao and C. G. Cassandras, "Optimal transmission scheduling for energy-efficient wireless networks," in *Proc. INFOCOM*, 2006, [CD ROM].
[22] L. Miao and C. G. Cassandras, "Optimality of static control policies in some discrete event systems," *IEEE Trans. Automat. Control*, vol. 50, no. 9, pp. 1427–1431, Sep. 2005.
[23] 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.
[24] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in *Proc. 36th Annu. Symp. Foundations Computer Sci. (FOCS'95)*, 1995, pp. 374–382.

Jianfeng Mao (S'08) received the B.E. degree in automatic control and the M.E. degree in control theory and applications from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and is currently pursuing the Ph.D. degree in systems engineering at Boston University, Boston, MA.

He specializes in the areas of modeling and optimization of complex systems with application to sensor networks, manufacturing systems.



Christos G. Cassandras (F'96) received the B.S. degree from Yale University, New Haven, CT, in 1977, the M.S.E.E. degree from Stanford University, Stanford, CA, in 1978, and the S.M. and Ph.D. degrees from Harvard University, Cambridge, MA, in 1979 and 1982, respectively.

From 1982 to 1984, he was with ITP Boston, Inc. where he worked on the design of automated manufacturing systems. From 1984 to 1996, he was a Faculty Member at the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. Currently, he is Head of the Division of Systems Engineering and Professor of Electrical and Computer Engineering at Boston University, Boston, MA, and a founding member of the Center for Information and Systems Engineering (CISE). He has published over 250 papers in these areas, and four books. He specializes in the areas of discrete event and hybrid systems, stochastic optimization, and computer simulation, with applications to computer networks, sensor networks, manufacturing systems, transportation systems, and command/control systems.

Dr. Cassandras is currently Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and has served on several editorial boards and as Guest Editor for various journals. He has also served on the IEEE Control Systems Society Board of Governors. He received the Distinguished Member Award of the IEEE Control Systems Society (2006), the 1999 Harold Chestnut Prize, and the 1991 Lilly Fellowship. He is a member of Phi Beta Kappa and Tau Beta Pi, and is a Fellow of the IFAC.