A New Model for Supervisory Control of Discrete Event Systems

[a]Qiying Hu and [b]Wuyi Yue

[a]College of International Business and Management
Shanghai University, Shanghai 201800 CHINA, e-mail: huqiying@sina.com
[b]Department of Information Science and Systems Engineering
Konan University, Kobe 658-8501 JAPAN, e-mail: yue@konan-u.ac.jp

(Received October 13, 2005)

Abstract

In this paper, we present a new model for supervisory control of discrete event systems (DESs) with an arbitrary control pattern. Here, a discrete event system is defined as a collection of event sets that depend on strings. When the system generates a string, the next event that may occur should be in the corresponding event set. In the optimal control model, there are rewards for choosing control inputs at strings and the sets of available control inputs depend also on strings. The performance measure is to find a policy under the condition where the discounted total reward among strings from the initial state is maximized. By applying ideas from Markov decision processes, we divide the problem into three sub-cases where the optimal value is respectively finite, positive infinite and negative infinite. For the case with finite optimal values, the optimality equation is shown and further characterized with its solutions. We also characterize the structure of the set of all optimal policies. We present a new supervisory control problem of DESs with the control pattern being dependent of strings. We study the problem in both the event feedback control and the state feedback control by generalizing concepts of invariant and closed languages/predicates. Finally, we apply the above model and results to a job matching problem.

Keywords: Supervisory control, discrete event systems, arbitrary control pattern.

1 Introduction

This paper presents a new model for optimal control of discrete event systems (DESs) and, based on this model, generalizes the control pattern for supervisory control of DESs. The supervisory control of DESs was presented by Ramadge and Wonham [1], [2] and Wonham and Ramadge [3] and has two branches: event feedback control and state feedback control. Since its development approximately twenty years ago, several specific branches of research relating to supervisory control have emerged, such as partially observable information [4]-[6], decentralized supervisory control [7], hierarchical supervisory control [8].

A discrete event system is driven for occurring finite or infinite events. Such a system is often described by strings of occurring events. Every time an event occurs, a new string forms and the next event that can occur is constrained in an event subset. The event set is further divided into two disjoint subsets, called respectively the controllable event set and the uncontrollable event set. A control input is an event subset but includes the uncontrollable event set. We let Γ be the set of all control inputs. In the event feedback control, a control input should be chosen from Γ based on strings. This implies that the occurring of events outside of the control input is prohibited. The system behavior is described by languages, that being sets of strings. With this description, the synthesizing problem (or called the supervisory control problem) for controlling the system is as: whether a given language can be synthesized by a supervisor or not. As shown in [1], a given language can be synthesized if and only if it is controllable and closed. The synthesizing problem together with its result is the essential basis for the supervisory control theory.

Some research relating to optimal control problems of DESs has been presented in the literature. Tsitsiklis [9] presented a dynamic programming model to solve some synthesizing problems in the supervisory control: for given discrete event systems G_1, G_2, G to find a supervisor π that satisfies $L(G_1) \subseteq L(\pi/G) \subseteq L(G_2)$, where $L(\cdot)$ is the language generated in DESs. In regard to the reward for occurring events at states, Passino and Antsaklis [10] studied the optimal control problem of minimizing the total reward among strings from the initial state to some given target state subset and presented a heuristic algorithm to search for the string with the minimal total reward by using a branching-bounding algorithm.

Again, considering the reward for occurring events at states, Brave and Heymann [11] calculated the optimal attraction by minimizing the total reward among all possible strings from an arbitrary state to a given global attraction. They also found conditions for the existence of supervisors achieving optimal attraction and provided efficient algorithms for their synthesis. Kumar and Garg

[12] studied a static designing problem with two reward functions $c(q, \sigma)$ and $p(q)$. They presented algorithms to compute the maximal sub-controllable languages for the supervisory control by using the maximal-flow-minimal-cut theorem. Yamsaki and Ushio [13] proposed a method to construct a supervisor based on reinforcement learning for state feedback control of partially observed DESs. They introduced a probability structure for the system based on some other parameters. Fu, Ray and Lagoa [14] presented a state based method for optimal control of regular languages with the performance measure being a signed real measure of the supervised sublanguage. Ray, Fu and Lagoa [15] generalized the model by considering disabling event cost. The cost in their models is for occurring events at states and the cost for occurring a string is the product of all costs for occurring events in the string.

But all of the above researches either related special optimal control problems with special reward functions to solve special problems in the supervisory control, or, they were concerned with the static control of DESs [12], and were not concerned with general frameworks for optimal control of DESs.

On the other hand, Golaszewski and Ramadge [16] and Li *et al.* [17] studied the supervisory control problems with a general control pattern which is a subset of Γ. This includes the supervisory control problem with forced events (Golaszewski and Ramadge [16]) and control of timed DESs (Brandin and Wonham [18]) as special cases. Golaszewski and Ramadge [16] studied the full observation case while Li *et al.* [17] studied the partial observation case under restricted conditions. Takai [6] also studied the partial observation case but discarded the restricted conditions. All of these papers discussed the event feedback control.

In this paper, we present a new model for optimal control of DESs with an arbitrary control pattern. Here, a discrete event system is defined as a collection of event sets that depend on strings. When the system generates a string, the next event that may occur should be in the corresponding event set. Moveover, the control pattern (which consists of sets of available control inputs) depends on strings. This control pattern is more general than that in papers of Golaszewski and Ramadge [16], Li *et al.* [17] and Takai [6]. In the optimal control model, there are rewards for choosing control inputs at strings. The performance measure is to find a policy under the condition where the discounted total reward among strings from the initial state is maximized.

In this paper, by applying ideas from Markov decision processes (Hu and Xu [19], Hu and Yue [20]), we divide the problem into three sub-cases where the optimal value is respectively finite, positive infinite and negative infinite. For the case with a finite optimal value, the optimality equation is shown and further characterized with its solutions. We also characterize the structure of the set of all optimal policies. Moreover, we show that when the reward function is stationary, the optimality equation and its solution are also stationary. We present a new supervisory control problem of DESs with the control pattern being dependent of strings. We study the problem in both the event feedback control and the state feedback control by generalizing the concepts of invariant and closed languages/predicates. Finally, we apply the above model and its corresponding results to a job-matching problem.

The remainder of the paper is organized as follows. In Section 2, we describe the system model with notations and preliminaries. In Section 3, we discuss the optimality equation with its solutions and optimal policies. In Section 4 we simplify the model and show that better results can be obtained when the discrete event system is described by an automaton. In Section 5, we study the supervisory control problem with a more general control pattern based on the optimal control model. In Section 6, we use the optimal control problem to describe and solve a job-matching problem. Section 7 contains the paper's conclusions.

2 System Model

Let Σ be a finite set of events and Σ^* be the set of all finite length strings formed with elements of Σ, including the null string ϵ. Any subset of Σ^* is called a language on Σ. The discrete event system we considered here is described by

$$G = \{\Sigma(s),\ s \in \Sigma^*\} \tag{1}$$

where $\Sigma(s) \subseteq \Sigma$ is the set of events that can occur after string s. The system evolves as follows. Initially, an event $\sigma \in \Sigma(\epsilon)$ occurs. Inductively, if string $s \in \Sigma^*$ occurs then the next event should be in the event set $\Sigma(s)$. Hence, the language generated by the system, denoted by $L(G)$, is defined recursively as follows: a) $\epsilon \in L(G)$, and b) if $s \in L(G)$ then $s\sigma \in L(G)$ for each $\sigma \in \Sigma(s)$. It is apparent that such a language $L(G)$ is well defined. Moreover, for any string $r \in L(G)$, we define by $L(G, r)$ the language similarly as $L(G)$ where "a) $\epsilon \in L(G)$" is replaced by "a) $r \in L(G)$". Certainly, $L(G, r) = \{s \in \Sigma^* \mid rs \in L(G)\}$.

Remark 1 1) In order to determine the language $L(G)$, it suffices to know $\Sigma(s)$ for $s \in L(G)$. In fact, $\{\Sigma(s),\ s \in L(G)\}$ and $L(G)$ are determined by each other since that if $L(G)$ is given then $\Sigma(s) = \{\sigma \in \Sigma \mid s\sigma \in L(G)\}$ *for* $s \in L(G)$.

2) In literature relating to supervisory control, a system is described by an automaton:

$$G := \{\Sigma,\ Q,\ \delta,\ q_0\}. \tag{2}$$

Here, Q is a countable state space, Σ is a finite event set, $\delta(\sigma, q)$ is a partial state transition function and q_0 is the initial state. Let inductively $\delta(s\sigma, q) = \delta(\sigma, \delta(s, q))$ for $s\sigma \in \Sigma^*$ and write $\delta(s, q)!$ if $\delta(s, q)$ is well defined. It is apparent that the expression given by Eq. (1) for a DES is equivalent to that given by Eq. (2). An automaton is well structured and a DES of Eq. (1) has a universal form. □

For strings s, t, r, if $s = tr$ then we call t a prefix of s and denote it by $t \leq s$. For a language $L \subseteq \Sigma^*$, we define its closure, denoted by $\bar{L}$, to be a set of all prefixes of strings in L. We call L a closed language if $\bar{L} = L$. We will consider infinite strings. Let Σ^ω be the set of all infinite strings on Σ. For strings $s \in \Sigma^\omega$ and $t \in \Sigma^*$, if there is $r \in \Sigma^\omega$ such that $s = tr$, then we call t a prefix of s and denote it also by $t \leq s$. We define the infinite language generated by G as $L^\omega(G) = \{s \in \Sigma^\omega \mid t \in L(G) \text{ for all } t \leq s\}$. Then, $L(G)$ is the closure of $L^\omega(G)$.

As in the supervisory control, the event set Σ is divided into two disjoint subsets: an uncontrollable event set Σ_u and a controllable event set Σ_c. A control input is defined as a subset of Σ satisfying $\Sigma_u \subseteq \gamma \subseteq \Sigma$. Let Γ be the set of all control inputs. The joint operation in Γ is defined as usual: $\gamma_1 \wedge \gamma_2 = \gamma_1 \cap \gamma_2$. There is a control pattern attached with the DES G. Suppose that for $s \in L(G)$ there is $\Gamma(s) \subseteq \Gamma$. $\Gamma(s)$ represents the set of available control inputs at string s. That is, it is required that the control input at string $s \in L(G)$ should be restricted in $\Gamma(s)$. Hence, we define a controlled discrete event system (CDES) by

$$G_c = \{\Sigma(s), \Gamma(s), \quad s \in L(G)\}. \tag{3}$$

A policy for G_c is defined as a map $\pi : L(G) \to \Gamma$ satisfying $\pi(s) \in \Gamma(s)$ for each $s \in L(G)$. The set of policies is denoted by Π. For each policy $\pi \in \Pi$, the system controlled under π, denoted by π/G, is $\pi/G = \{\Sigma(s) \wedge \pi(s),\ s \in \Sigma^*\}$. We denote by $L(\pi/G)$ and $L^\omega(\pi/G)$ respectively the language and the infinite language generated by π/G. Surely, $L(\pi/G) \subseteq L(G)$ and $L^\omega(\pi/G) \subseteq L^\omega(G)$.

As in Golaszewski and Ramadge [16], it is assumed that

$$\Sigma(s) = \bigcup \{\gamma \mid \gamma \in \Gamma(s)\}. \tag{4}$$

Now, we define an optimal control problem for the system G as the following triple,

$$\{\Sigma(s), \Gamma(s), (c(s, \gamma), \gamma \in \Gamma(s)),\ \ s \in L(G)\} \tag{5}$$

where $c(s, \gamma)$ is an extended real-valued reward function $c(s, \gamma) \in [-\infty, +\infty]$ for choosing control input γ at string s for $s \in L(G)$ and $\gamma \in \Gamma(s)$.

We will consider the optimal control problem of Eq. (5) in infinite languages. Hence, we assume that G is alive, i.e., $\Sigma(s)$ is non-empty for each string $s \in L(G)$. This assumption can be relaxed. In fact, we can introduce a fictitious event $\sigma_J \notin \Sigma$ and let $\Sigma_u := \Sigma_u \cup \{\sigma_J\}$, $\Sigma(s) := \{\sigma_J\}$ if $\Sigma(s) = \emptyset$ and let $c(s, \gamma) = c(s, \gamma - \{\sigma_J\})$ for each s and γ.

For a policy $\boldsymbol{\pi} \in \Pi$, we let $c(s, \boldsymbol{\pi}) = c(s, \boldsymbol{\pi}(s))$ be the reward at string s under $\boldsymbol{\pi}$ and let $v_s(\boldsymbol{\pi}, t) = \sum_{k=0}^{\infty} \beta^k c(st_k, \boldsymbol{\pi})$ be the discounted total reward for occurring string $t = \sigma_1\sigma_2 \cdots \in L^{\omega}(G, s)$ under $\boldsymbol{\pi}$ when string s has occurred, where $t_k = \sigma_1\sigma_2 \cdots \sigma_k$ is a prefix of t for $k = 1, 2, \ldots, n$ and $t_0 = \epsilon$. $\beta > 0$ is the discounted factor. We will simply call $v_s(\boldsymbol{\pi}, t)$ *the reward for t at s under $\boldsymbol{\pi}$.*

In general, there are infinite possible strings that may be generated by the system (G or $\boldsymbol{\pi}/G$), but there is only one string that will finally be generated. We cannot know which string will be generated until the end of the system is reached. Thus we consider the maximal discounted total reward for all possible strings that may occur in the system controlled under $\boldsymbol{\pi}$. Formally, we define

$$V(\boldsymbol{\pi}, s) = \sup_{t \in L^{\omega}(\boldsymbol{\pi}/G, s)} v_s(\boldsymbol{\pi}, t), \quad s \in L(G) \tag{6}$$

as the maximal discounted total reward of the system controlled under $\boldsymbol{\pi}$ when $s \in L(G)$ has occurred, where $L^{\omega}(\boldsymbol{\pi}/G, s)$ is the infinite language with the initial string s.

We define the optimal value function by

$$V^*(s) = \sup_{\boldsymbol{\pi} \in \Pi} V(\boldsymbol{\pi}, s), \quad s \in L(G). \tag{7}$$

$V^*(s)$ is the best case we have for the discounted total reward. We call policy $\boldsymbol{\pi}^*$ optimal at string s if $V(\boldsymbol{\pi}^*, s) = V^*(s)$ and call $\boldsymbol{\pi}^*$ optimal if $\boldsymbol{\pi}^*$ is optimal at each $s \in L(G)$.

We introduce the following condition on the reward function.

Condition (GC): The discounted total reward $v_s(\boldsymbol{\pi}, t)$ for string t at s is well defined for each finite string $s \in L(G)$ and each infinite string $t \in \Sigma^{\omega}$ with $st \in L^{\omega}(G)$.

Condition (GC) will be true, for example, when the reward function $c(\cdot, \cdot)$ is non-negative, or is non-positive, or is uniformly bounded and $\beta \in (0, 1)$. Condition (GC) implies that the objective function $V(\boldsymbol{\pi}, s)$ is well defined for each policy $\boldsymbol{\pi}$ and string $s \in L(G)$. So, we say that the optimal control problem is well defined. Surely, condition (GC) is the base for discussing about the optimal control problem and is assumed throughout this paper.

3 Optimality Equation and Optimal Policies

In this section, we will study the optimality equation and optimal policies for the optimal control problem of Eq. (5). First, for $s \in L(G)$, we define

$$\Gamma_1(s) := \{\gamma \in \Gamma(s) \mid c(s, \gamma) > -\infty\} \tag{8}$$

as the set of control inputs at string s with the reward being larger than negative infinity. Let $L^{-\infty}(G) := \{s \in L(G) \mid \Gamma_1(s) = \emptyset\}$ be a sub-language of $L(G)$. $\Gamma_1(s) = \emptyset$ means that $c(s, \gamma) = -\infty$ for all $\gamma \in \Gamma(s)$. Hence, for each $s \in L^{-\infty}(G)$ and $\boldsymbol{\pi} \in \Pi$, $c(s, \boldsymbol{\pi}) = -\infty$ and so $V(\boldsymbol{\pi}, s) = -\infty$. This shows the following lemma.

Lemma 1 *$V^*(s) = -\infty$ for all $s \in L^{-\infty}(G)$, so each policy $\boldsymbol{\pi} \in \Pi$ is optimal at $s \in L^{-\infty}(G)$.* □

From the above lemma, it suffices to discuss the optimality in $L(G) - L^{-\infty}(G)$. Moreover, for each $s \in L(G) - L^{-\infty}(G)$, if $\gamma \in \Gamma(s) - \Gamma_1(s)$ then $c(s, \gamma) = -\infty$. Hence, each policy $\boldsymbol{\pi} \in \Pi$ with $\boldsymbol{\pi}(s) \in \Gamma(s) - \Gamma_1(s)$ must satisfy $V(\boldsymbol{\pi}, s) = -\infty$ and so we would not consider such a policy. Let Π_1 be the set of policies $\boldsymbol{\pi}$ satisfying $\boldsymbol{\pi}(s) \in \Gamma_1(s)$ for all $s \in L(G) - L^{-\infty}(G)$. Surely, $V^*(s) = \sup_{\boldsymbol{\pi} \in \Pi_1} V(\boldsymbol{\pi}, s)$ for $s \in L(G) - L^{-\infty}(G)$. So, we will limit our attention to policies in Π_1.

Hence, we limit our discussion on $L(G) - L^{-\infty}(G)$ with the set of available control inputs at string $s \in L(G) - L^{-\infty}(G)$ being $\Gamma_1(s)$ in the following. For notational simplicity, we let $\Sigma_\pi(s) = \Sigma(s) \cap \boldsymbol{\pi}(s)$ and $\Sigma_\gamma(s) = \Sigma(s) \cap \gamma$ for $\boldsymbol{\pi} \in \Pi$, $\gamma \in \Gamma(s)$ and $s \in L(G)$.

Lemma 2 *For any policy $\boldsymbol{\pi} \in \Pi_1$,*

$$V(\boldsymbol{\pi}, s) = c(s, \boldsymbol{\pi}) + \beta \max_{\sigma \in \Sigma_\pi(s)} V(\boldsymbol{\pi}, s\sigma), \;\; s \in L(G) - L^{-\infty}(G).$$

Proof. If $c(s, \boldsymbol{\pi}) = +\infty$ then both sides of the above equation are infinite and so the above equation holds. Otherwise, $c(s, \boldsymbol{\pi})$ is finite due to $\boldsymbol{\pi} \in \Pi_1$. Then, we have that for $s \in L(G) - L^{-\infty}(G)$,

$$\begin{aligned} V(\boldsymbol{\pi}, s) &= \sup_{t \in L^\omega(\boldsymbol{\pi}/G, s)} \sum_{k=0}^{\infty} \beta^k c(st_k, \boldsymbol{\pi}) \\ &= c(s, \boldsymbol{\pi}) + \beta \max_{\sigma \in \Sigma_\pi(s)} \sup_{t' \in L^\omega(\boldsymbol{\pi}/G, s\sigma)} \sum_{k=0}^{\infty} \beta^k c(s\sigma t'_k, \boldsymbol{\pi}). \end{aligned}$$

This results in the lemma. □

The above lemma separates the discounted total reward for an infinite string into two parts: the reward $c(s, \boldsymbol{\pi})$ for the first period and the maximal discounted total reward for the remaining periods. We divide the language $L(G)$ into three disjoint sub-languages:

$$L(G) = L^+(G) \bigcup L^-(G) \bigcup L^0(G)$$

where $L^+(G) := \{s \in L(G) \mid V^*(s) = +\infty\}$ is the sub-language with the positive infinite optimal value, $L^-(G) := \{s \in L(G) \mid V^*(s) = -\infty\}$ is the sub-language with the negative infinite optimal value and $L^0(G) := \{s \in L(G) \mid V^*(s) \in (-\infty, +\infty)\}$ is the sub-language with finite optimal values. Surely, $L^{-\infty}(G) \subseteq L^-(G)$. Furthermore, we let

$$L^{+\infty}(G) := \{s \in L(G) \mid \text{there is a policy } \boldsymbol{\pi} \in \Pi_1 \text{ such that } V(\pi, s) = +\infty\}$$

be a sub-language of $L^+(G)$. We have the following results.

Theorem 1 *1) Each policy $\boldsymbol{\pi} \in \Pi$ is optimal in $L^-(G)$; there are optimal policies in $L^{+\infty}(G)$; and there is no optimal policy in $L^+(G) - L^{+\infty}(G)$.*

2) V^* satisfies the following optimality equation in the sub-language $L^0(G)$:

$$V(s) = \max_{\gamma \in \Gamma_1(s)} \{c(s, \gamma) + \beta \max_{\sigma \in \Sigma_\gamma(s)} V(s\sigma)\}, \quad s \in L^0(G). \tag{9}$$

Proof. 1) It is easy to see that the result in Lemma 1 is also true for $L^-(G)$ since that when $V^*(s) = -\infty$, $V(\boldsymbol{\pi}, s) = -\infty$ for each policy $\boldsymbol{\pi} \in \Pi$. The remaining results are obvious.

2) For $s \in L^0(G)$, all $V^*(s)$ and $c(s, \gamma)$ for $\gamma \in \Gamma_1(s)$ are finite. Hence, it follows Lemma 2 that

$$\begin{aligned} V^*(s) &= \sup_{\boldsymbol{\pi} \in \Pi_1} V(\boldsymbol{\pi}, s) \\ &= \sup_{\boldsymbol{\pi} \in \Pi_1} \{c(s, \boldsymbol{\pi}) + \beta \max_{\sigma \in \Sigma_{\boldsymbol{\pi}}(s)} V(\boldsymbol{\pi}, s\sigma)\} \\ &= \sup_{\gamma \in \Gamma_1(s), \boldsymbol{\pi} \in \Pi_1} \{c(s, \gamma) + \beta \max_{\sigma \in \Sigma_\gamma(s)} V(\boldsymbol{\pi}, s\sigma)\} \\ &= \max_{\gamma \in \Gamma_1(s)} \{c(s, \gamma) + \beta \sup_{\boldsymbol{\pi} \in \Pi_1} \max_{\sigma \in \Sigma_\gamma(s)} V(\boldsymbol{\pi}, s\sigma)\} \\ &= \max_{\gamma \in \Gamma_1(s)} \{c(s, \gamma) + \beta \max_{\sigma \in \Sigma_\gamma(s)} V^*(s\sigma)\}. \end{aligned}$$

This completes the proof. □

In the following, we characterize solutions of the optimality equation (9). We omit the proof for the lemma.

Lemma 3 *We have the following four statements.*

1) V^* satisfies the following condition:

$$\limsup_{n\to\infty}\beta^n \sup_{t\in L^\omega(\pi/G,s)} V(st_n)\geq 0,\ \ \forall\pi\in\Pi_1, s\in L^0(G)\ with\ V(\pi,s)\neq-\infty. \tag{10}$$

2) *$V\geq V^*$ if V is a solution of the optimality equation (9) and satisfies equation (10).*

3) $V\leq V^*$ if V is a solution of the optimality equation (9) and satisfies

$$\limsup_{n\to\infty}\beta^n \sup_{t\in L^\omega(\pi/G,s)} V(st_n)\leq 0,\ \ \forall\pi\in\Pi_1, s\in L^0(G)\ with\ V(\pi,s)\neq-\infty. \tag{11}$$

4) $V=V^*$ if V is a solution of the optimality equation (9) and satisfies

$$\limsup_{n\to\infty}\beta^n \sup_{t\in L^\omega(\pi/G,s)} V(st_n)= 0,\ \ \forall\pi\in\Pi_1, s\in L^0(G)\ with\ V(\pi,s)\neq-\infty. \tag{12}$$

From the above lemma, especially result 1), condition of Eq. (11) is equivalent to condition of Eq. (12) for $V=V^*$. So, the following theorem is true.

Theorem 2 *1) V^* is the smallest solution of the optimality equation (9) satisfying condition of Eq. (10).*

2) V^* is the unique solution of the optimality equation (9) satisfying condition of Eq. (11) or equivalently condition of Eq. (12) if and only if the optimality equation (9) has a solution satisfying condition of Eq. (11) or equivalent condition of Eq. (12). □

A sufficient condition for Eq. (11) is the following

$$\limsup_{n\to\infty}\beta^n \sup_{t\in L^\omega(\pi/G,s)} V(st_n)\leq 0,\qquad \forall s\in L^0(G).$$

This condition is simpler and may be verified easier than Eq. (11).

The following two theorems relate the optimality of policies to the optimality equation.

Theorem 3 *For each policy π^*, if*

$$\limsup_{n\to\infty}\beta^n \sup_{t\in L^\omega(\pi^*/G,s)} V^*(st_n)= 0,\qquad \forall s\in L^0(G)\ with\ V(\pi^*,s)\neq-\infty, \tag{13}$$

then π^* is optimal if and only if π^* attains the maximum of the optimality equation (9).

Proof. Sufficiency. Similar to the proof of 2) in Lemma 3, we have

$$V^*(s) = c(s,\boldsymbol{\pi}^*) + \beta \max_{\sigma\in\Sigma_{\pi^*}(s)} V^*(s\sigma)$$
$$= V^{n-1}(\boldsymbol{\pi}^*,s) + \beta^n \sup_{t\in L^\omega(\pi^*/G,s)} V^*(st_n), \quad s\in L^0(G),\ n\geq 1. \tag{14}$$

By letting $n\to\infty$ in the above equation, we get $V(\boldsymbol{\pi}^*,\ s) = V^*(s)$ for all $s\in L^0(G)$.

Necessity. If $\boldsymbol{\pi}^*$ is optimal, then $V(\boldsymbol{\pi}^*,\ s) = V^*(s)$ for all $s\in L^0(G)$. With this and Lemma 2 we have

$$\max_{\gamma\in\Gamma_1(s)}\{c(s,\gamma) + \beta \max_{\sigma\in\Sigma_\gamma(s)} V^*(s\sigma)\} = V^*(s) = V(\boldsymbol{\pi}^*,s)$$
$$= c(s,\boldsymbol{\pi}^*) + \beta \max_{\sigma\in\Sigma_{\pi^*}(s)} V^*(s\sigma).$$

This implies that $\boldsymbol{\pi}^*$ attains the maximum of the optimality equation (9). □

The above theorem characterizes the optimal policies with the optimality equation (9), while the following theorem characterizes the structure of the set of optimal policies. We let the set of optimal control inputs at string $s\in L^0(G)$ be

$$\Gamma_1^*(s) = \{\gamma\in\Gamma_1(s)\,|\,\gamma \text{ attains the maximum in Eq. (9)}\}.$$

These sets will play an important role in optimal policies.

Theorem 4 *A policy $\boldsymbol{\pi}^*$ is optimal in $L^0(G)$ if and only if the following two statements are true:*

1) For $s\in L^0(G)$, $\Gamma_1^*(s)$ is non-empty and $\boldsymbol{\pi}^*(s)\in\Gamma_1^*(s)$;

2) $(\boldsymbol{\pi}^*,\ V^*)$ satisfies Eq. (13) or

$$\limsup_{n\to\infty} \beta^n \sup_{t\in L^\omega(\pi^*/G,s)} V^*(st_n) \leq 0, \quad \forall s\in L^0(G) \text{ with } V(\boldsymbol{\pi}^*,s)\neq -\infty. \tag{15}$$

The above theorem 4 can be easily proved from Lemma 2 and Theorem 3.

In this section, we study the optimal control problem by ideas and methods from MDP (Hu and Xu [19]). We divide the language $L(G)$ into three sub-languages: $L^-(G)$, $L^+(G)$ and $L^0(G)$. We also show and characterize the optimality equation in $L^0(G)$. In the next section, we will study some properties of the sub-languages. For this we will use some ideas from language, automaton and the supervisory control of DESs.

4 System Based on Automaton

When the discrete event system is modeled by an automaton, the problem arises whether any better result can be obtained or not.

For the given optimal control problem of Eq. (5), suppose that G can be described by an automaton $G=\{\Sigma, Q, \delta, q_0\}$ and furthermore for $\gamma\in\Gamma(s)$ and $s\in L(G)$,

$$\Sigma(s)=\Sigma(\delta(s, q_0)),\ \Gamma(s)=\Gamma(\delta(s, q_0)),\ c(s, \gamma)=c(\delta(s, q_0), \gamma) \tag{16}$$

where for $q\in Q$, $\Sigma(q)=\{\sigma\in\Sigma \mid \delta(\sigma, q)!\}$, $\Gamma(q)\subseteq\Gamma$ and $c(q, \gamma)$ is an extended real valued reward function defined on $\{(q, \gamma)\mid\gamma\in\Gamma(q),\ q\in Q\}$. When condition of Eq. (16) is true, we say that the controlled discrete event system G_c and the optimal control problem of Eq. (5) are stationary.

We define a state policy f as a map: $Q\to\Gamma$ satisfying $f(q)\in\Gamma(q)$. Obviously, a state policy is also a policy $\pi: L(G)\to\Gamma$ with $\pi(s)=f(\delta(s, q_0))$. Let F be the set of all state policies.

For any given $q\in Q$ and infinite string $t=\sigma_0\sigma_1\cdots$, we define $q_{k+1}=\delta(\sigma_k, q_k)$ for $k\geq 0$ with $q_0=q$, and $v_q(f, t)=\sum_{k=0}^{\infty}\beta^k c(q_k, f)$ for $f\in F$ as the discounted total reward for string t from state q under state policy f. Moreover, we define

$$V(f,q)=\sup_{t\in L^{\omega}(f/G,q)} v_q(f,t)$$

as the maximal discounted total reward from state q under state policy f. The optimal value function within state policies is defined by $V^0(q)=\sup_{f\in F}V(f, q)$ for $q\in Q$.

Similarly, as in Sec. 3, we divide the state set Q into the following three subsets:

$$Q^-=\{q\in Q\mid V^0(q)=-\infty\},$$
$$Q^+=\{q\in Q\mid V^0(q)=+\infty\},$$
$$Q^0=\{q\in Q\mid V^0(q)\in(-\infty, +\infty)\}.$$

Moreover, we define $Q^{+\infty}=\{q\in Q\mid \text{there is a state policy } f \text{ such that } V(f, q)=+\infty\}$. Let $\Gamma_1(q)=\{\gamma\in\Gamma(q)\mid c(q, \gamma)>-\infty\}$ for $q\in Q$. Certainly, $\Gamma_1(s)=\Gamma_1(\delta(s, q))$ and so $\Sigma_1(s)=\Sigma_1(\delta(s, q)):=\bigcup\{\gamma\mid\gamma\in\Gamma_1(\delta(s, q))\}$. Therefore, $G_1=\{\Sigma_1(s),\ s\in\Sigma^*\}$ can be described by an automaton $G_1=\{Q, \Sigma, \delta_1, q_0\}$ where $\delta_1(\sigma, q)=\delta(\sigma, q)$ for $\sigma\in\Sigma_1(q)$ and is otherwise undefined.

If we restrict the problem under the condition of Eq. (16) within the state policy set F and the state set Q, then all the results obtained in Sec. 3 are still true. For example, the following lemma is similar to Theorem 1.

Lemma 4 *1) Each state policy $f \in F$ is optimal in Q^-, there is an optimal state policy in $Q^{+\infty}$ but there is no optimal state policy in $Q^+ - Q^{+\infty}$.*

2) V^0 satisfies the following equation in the subset Q^0:

$$V(q) = \max_{\gamma \in \Gamma_1(q)} \{c(q,\gamma) + \beta \max_{\sigma \in \Sigma_\gamma(q)} V(\delta(\sigma,q))\}, \quad q \in Q^0. \tag{17}$$

This equation is called the stationary optimality equation.

The following theorem discusses relationships between $V^*(s)$ and $V^0(\delta(s, q_0))$ for $s \in L(G)$.

Theorem 5 *$V^*(s) \geq V^0(\delta(s, q_0))$ for all $s \in L(G)$. Moreover, if*

$$\begin{aligned} &\limsup_{n\to\infty} \beta^n \sup_{t \in L^\omega(\pi^*/G,s)} V^0(\delta(st,q_0)) \geq 0, \\ &\forall \pi \in \Pi_1, \forall s \in L(G) \text{ with } \delta(s,q_0) \in Q^0, \end{aligned} \tag{18}$$

then $V^*(s) = V^0(\delta(s, q_0))$ for all $s \in L(G)$ with $\delta(s, q_0) \in Q^0$.

Proof. First, it is apparent that $v_s(f, t) = v_{\delta(s, q_0)}(f, t)$ and $V(f, s) = V(f, \delta(s, q_0))$ for $s \in L(G)$, $st \in L^\omega(G)$, $f \in F$. Then for each $s \in L(G)$,

$$V^*(s) = \sup_\pi V(\boldsymbol{\pi}, s) \geq \sup_f V(f,s) = \sup_f V(f, \delta(s,q_0)) = V^0(\delta(s,q_0)).$$

On the other hand, for each policy $\boldsymbol{\pi}$, we have from Eq. (17) that

$$V^0(\delta(s,q_0)) \geq c(s,\boldsymbol{\pi}) + \beta \max_{\sigma \in \Sigma_\pi(s)} V^0(\delta(s,q_0)), \quad \forall s \in L(G) \text{ with } \delta(s,q_0) \in Q^0.$$

With this, we can prove as in Lemma 3 that for all $s \in L(G)$ with $\delta(s, q_0) \in Q^0$,

$$V^0(\delta(s,q_0)) \geq V^{n-1}(\boldsymbol{\pi}, s) + \beta^n \sup_{t \in L^\omega(\pi/G,s)} V^0(\delta(st,q_0)).$$

By letting $n \to \infty$, and due to Eq. (18), we get that $V^0(\delta(s, q_0)) \geq V(\boldsymbol{\pi}, s)$. Since $\boldsymbol{\pi}$ is arbitrary, $V^0(\delta(s, q_0)) \geq V^*(s)$. Hence, $V^0(\delta(s, q_0)) = V^*(s)$ for all $s \in L(G)$ with $\delta(s, q_0) \in Q^0$. □

5 Supervisory Control Problem

The most basic problem in the supervisory control of DESs is the supervisory control problem. In this section, we will study the problem in the frameworks of both event feedback control and state feedback control.

5.1 Event Feedback Control

For a given language $K \subseteq L(G)$, we consider the problem of whether there is a policy $\boldsymbol{\pi}$ such that $L(\boldsymbol{\pi}/G) = K$. If so, we say that K can be synthesized by $\boldsymbol{\pi}$, or $\boldsymbol{\pi}$ synthesizes K. In the standard model, i.e., $\Gamma(s) = \Gamma$ for all $s \in L(G)$, a necessary and sufficient condition for synthesizing a language K is that K is G-closed and G_u-invariant (Ramadge and Wonham [1]).

In this subsection, we discuss the problem for CDES G_c of Eq. (3). Since the set $\Gamma(s)$ does not equal Γ we need some other conditions for synthesizing a language. We denote by $\boldsymbol{\Gamma} = \{\Gamma(s),\ s \in L(G)\}$ the collection of sets of available control inputs.

Definition 1 *A language K is said to be*

a) $\boldsymbol{\Gamma}$-invariant if for each string $s \in K$ there is $\gamma_s \in \Gamma(s)$ satisfying

$$s\gamma_s \cap L(G) \subseteq K. \tag{19}$$

b) $\boldsymbol{\Gamma}$-closed if for each $s\sigma \in K$ there is $\gamma_s \in \Gamma(s)$ satisfying $\sigma \in \gamma_s$ and Eq. (19).

c) $\boldsymbol{\Gamma}$-controllable if it is $\boldsymbol{\Gamma}$-invariant and $\boldsymbol{\Gamma}$-closed.

Remark 2 When $\Sigma_u \in \Gamma(s)$ for all $s \in K$, Σ_u is the minimal element in all $\Gamma(s)$. In this case, Eq. (19) is equivalent to $s\Sigma_u \cap L(G) \subseteq K$ for all $s \in K$, which is exactly the definition of controllable language for the standard model (Ramadge and Wonham [1]). In fact, a $\boldsymbol{\Gamma}$-invariant language must be controllable but in general the reverse is not true. Hence, the concept of $\boldsymbol{\Gamma}$-invariant languages generalizes that of controllable languages. Similarly, the concept of $\boldsymbol{\Gamma}$-closed languages generalizes that of closed languages: a $\boldsymbol{\Gamma}$-closed language is closed but in general the reverse is not true. □

We have the following result on synthesizing a language.

Theorem 6 *For any given language $K \subseteq L(G)$, there is a policy $\boldsymbol{\pi}_K$ such that $L(\boldsymbol{\pi}_K/G) = K$ if and only if K is $\boldsymbol{\Gamma}$-controllable and the maximal element of $\Gamma_K(s) = \{\gamma \mid s\gamma \cap L(G) \subseteq K,\ \gamma \in \Gamma(s)\}$ exists uniquely for each $s \in K$. Moreover, the policy $\boldsymbol{\pi}_K$ can be taken by*

$$\boldsymbol{\pi}_K(s) = \max \Gamma_K(s) = \max\{\gamma \mid s\gamma \cap L(G) \subseteq K,\ \gamma \in \Gamma(s)\}, \qquad s \in K. \tag{20}$$

Proof. Necessity. For the given language $K \subseteq L(G)$, if there is a policy $\boldsymbol{\pi}_K$ such that $L(\boldsymbol{\pi}_K/G) = K$, then it is easy to see that

$$s\boldsymbol{\pi}_K(s) \cap L(G) = \{s\sigma \mid s\sigma \in K\}, \qquad s \in K. \tag{21}$$

With this and Definition 1 we know that K is $\mathbf{\Gamma}$-controllable and the maximal element of $\Gamma_K(s)$ exists uniquely for each $s\in K$.

Sufficiency. Since K is $\mathbf{\Gamma}$-invariant, the set $\Gamma_K(s)$ is non-empty. Then, $\boldsymbol{\pi}_K(s)=\max\Gamma_K(s)$ satisfies Eq. (19), i.e., $s\ \boldsymbol{\pi}_K(s)\cap L(G)\subseteq K$ for all $s\in K$. On the other hand, for each $s\sigma\in K$, because K is $\mathbf{\Gamma}$-closed and the maximum of $\boldsymbol{\pi}_K(s)$, we have $\sigma\in\boldsymbol{\pi}_K(s)$. Hence, Eq. (21) is true and so $L(\boldsymbol{\pi}_K/G)=K$. □

The above theorem 6 says that the policy $\boldsymbol{\pi}_K$ defined by Eq. (20) synthesizes the given language K.

The above theorem 6 solves the synthesizing problem if the given language K is $\mathbf{\Gamma}$-controllable. Otherwise, we want to know if there is a unique maximal sub-language of K that is $\mathbf{\Gamma}$-controllable. In the following, we assume that $\Gamma(s)$ is closed under union $\vee$.

For any given language K, let K_1 and K_2 be two $\mathbf{\Gamma}$-controllable sub-languages of K. It is easy to see from the definitions that $K_1\cup K_2$ is also a $\mathbf{\Gamma}$-controllable sub-language of K. Hence, the set of $\mathbf{\Gamma}$-controllable sub-languages of K is closed under union and so has the unique maximal element. We denote this maximal element by

$$K^{\uparrow}=\max\{K'\,|\,K'\subseteq K \text{ is } \mathbf{\Gamma}\text{-controllable}\}.$$

This shows the following lemma.

Lemma 5 *For any given language $K\subseteq L(G)$, its maximal $\mathbf{\Gamma}$-controllable sub-language $K^{\uparrow}$ exists uniquely and the policy synthesizing $K^{\uparrow}$ is $\boldsymbol{\pi}_{K^{\uparrow}}$.* □

Similarly, for any given language $K\subseteq L(G)$, its maximal $\mathbf{\Gamma}$-invariant sub-language and maximal $\mathbf{\Gamma}$-closed sub-language exist uniquely.

In the following, we will introduce an optimal control problem to compute $K^{\uparrow}$. In fact, the policy $\boldsymbol{\pi}_{K^{\uparrow}}$ will be constructed. But before doing this, we introduce some concepts.

We define $\gamma_1\le\gamma_2$ by the set inclusion $\gamma_1\subseteq\gamma_2$ for two control inputs γ_1 and γ_2. Then $\le$ is a partial order in Γ and also in $\Gamma(s)$ for each s. Moreover, we define $\boldsymbol{\pi}_1\le\boldsymbol{\pi}_2$ for two policies $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ if $\boldsymbol{\pi}_1(s)\le\boldsymbol{\pi}_2(s)$ for all $s\in L(G)$. For a policy set Π', we call $\boldsymbol{\pi}^*\in\Pi'$ the maximum policy of Π' if $\boldsymbol{\pi}\le\boldsymbol{\pi}^*$ for all $\boldsymbol{\pi}\in\Pi'$. Especially, for an optimal control problem, the maximum policy of the set of all its optimal policies is called the maximum optimal policy, if it exists.

For the given language K, we define an optimal control problem based on CDES G_c with the reward function

$$c(s,\gamma)=\begin{cases}0, & \text{if } s\in K,\ s\gamma\cap L(G)\subseteq K\\ -1, & \text{else.}\end{cases}$$

This reward function is uniformly bounded. We take the discounted factor by any $\beta \in (0, 1)$. Then, $V(\boldsymbol{\pi}, s)$ is well defined and uniformly bounded. Therefore, $L^{+}(G) = L^{-}(G) = \emptyset$, $L^{0}(G) = L(G)$ and $\Gamma_1(s) = \Gamma(s)$. Let

$$K^{*} = \{s \in K \mid V^{*}(s) = 0\}.$$

We have the following result from K^{*}.

Theorem 7 *K^{*} is the maximal $\boldsymbol{\Gamma}$-invariant sub-language of K and $\boldsymbol{\pi}_{K^{*}}$ is the maximum optimal policy of the optimal control problem. Moreover, $K^{\uparrow} = L(\boldsymbol{\pi}_{K^{*}}/G) = \{s \in K^{*} \mid t \in K^{*}, \forall t \leq s\}$ is the maximal $\boldsymbol{\Gamma}$-closed sub-language of K^{*}.*

Proof. 1) For each $s \in K^{*}$, $V^{*}(s) = 0$ and so there is $\gamma \in \Gamma(s)$ such that $c(s, \gamma) = 0$. This results in γ satisfying Eq. (19). Thus, K^{*} is a $\boldsymbol{\Gamma}$-invariant sub-language of K.

Now, if $K' \subseteq K$ is $\boldsymbol{\Gamma}$-invariant, then from the optimality equation (9) and the definition of $\boldsymbol{\Gamma}$-invariant language, we know that $V^{*}(s) = 0$ for $s \in K'$. Thus, $K' \subseteq K^{*}$.

Hence, K^{*} is the maximal $\boldsymbol{\Gamma}$-invariant sub-language of K.

2) It is apparent that $\Gamma_{K^{*}}(s)$ is exactly the set of control inputs that attains the maximum of the optimality equation (9). It is non-empty and closed under union. Hence, $\boldsymbol{\pi}_{K^{*}}(s) = \max \Gamma_{K^{*}}(s)$ is well defined. Then, due to the proof of Theorem 6, $\boldsymbol{\pi}_{K^{*}}$ is the maximal optimal policy of the optimal control problem.

3) We then prove that $L(\boldsymbol{\pi}_{K^{*}}/G) = K^{\uparrow}$. First, it is obvious that $L(\boldsymbol{\pi}_{K^{*}}/G) \subseteq K^{*} \subseteq K$ and $L(\boldsymbol{\pi}_{K^{*}}/G)$ is $\boldsymbol{\Gamma}$-controllable. Second, if $K' \subseteq K$ is $\boldsymbol{\Gamma}$-controllable, then it is obvious that $V(\boldsymbol{\pi}_{K'}, s) = 0$ for $s \in K'$. Thus, $V^{*}(s) = 0$ for $s \in K'$. This shows that $K' \subseteq K^{*}$. Therefore, $\boldsymbol{\pi}_{K'}(s) \leq \boldsymbol{\pi}_{K^{*}}(s)$ for all $s \in K'$. This implies that $K' = L(\boldsymbol{\pi}_{K'}/G) \subseteq L(\boldsymbol{\pi}_{K^{*}}/G)$.

Overall, $K^{\uparrow} = L(\boldsymbol{\pi}_{K^{*}}/G)$. Obviously, $L(\boldsymbol{\pi}_{K^{*}}/G) = \{s \in K^{*} \mid t \in K^{*}, \forall t \leq s\}$ is the maximal $\boldsymbol{\Gamma}$-closed sub-language of K^{*}. □

From the above theorem 7, we know that in order to compute the maximal $\boldsymbol{\Gamma}$-controllable language $K^{\uparrow}$ of K, we can first compute the maximal $\boldsymbol{\Gamma}$-invariant language K^{*} of K and then compute the maximal $\boldsymbol{\Gamma}$-closed language of K^{*} which is exactly $K^{\uparrow}$.

5.2 State Feedback Control

First, for a state policy f, we let $f/G = (Q, \Sigma, \delta_f, q_0)$ be a subsystem of G with $\delta_f(\sigma, q) = \delta(\sigma, q)$ if $\sigma \in f(q)$ and otherwise undefined, and $R(f/G) = \{\delta(s, q_0) \mid s \in L(f/G)\}$ be the reachable state set of the system f/G (Ramadge and Wonham [2]). The supervisory control problem in the state feedback control is that for a given predicate $P \subseteq Q$ whether there is a state policy f such that $R(f/G) = P$. If so, we say that P can be synthesized by f, or f synthesizes P. In the standard model with $\Gamma(q) = \Gamma$ for all $q \in Q$, a necessary and sufficient condition for synthesizing a predicate P is that P is controllable (Li and Wonham [4]).

Similarly, as in the previous subsection, we denote $\mathbf{\Gamma} = \{\Gamma(q),\ q \in Q\}$ as the collection of sets of available control inputs. Moreover, a predicate P is said to be

a) $\mathbf{\Gamma}$-invariant if for each state $q \in P$ there is $\gamma_q \in \Gamma(q)$ such that $\delta(\sigma,\ q) \in P$ for all $\sigma \in \Sigma(q) \wedge \gamma_q$. Let $\Gamma_P(q)$ be the set of all such γ_q.

b) $\mathbf{\Gamma}$-closed if for each state $q \in P$ there are an integer $n \geq 0$, states $q_k \in P$, and control inputs $\gamma_k \in \Gamma_P(q_k)$ for $k = 0,\ 1,\ \ldots,\ n-1$ such that for $k = 0,\ 1,\ \ldots,\ n-1$ there is $\sigma_k \in \gamma_k$ with $q_{k+1} = \delta(\sigma_k,\ q_k)$ with $q_n = p$.

c) $\mathbf{\Gamma}$-controllable if P is $\mathbf{\Gamma}$-invariant and $\mathbf{\Gamma}$-closed.

The above concepts are corresponding to those defined in Definition 1. Also, when $\Gamma(q) = \Gamma$ for all $q \in Q$, $\mathbf{\Gamma}$-controllable predicates are exactly the controllable predicates in the standard model (Ramadge and Wonham [2]).

The following theorem can be similarly proved as in Theorem 3.

Theorem 8 *For any given predicate $P \subseteq Q$, there is a state policy f_P such that $R(f_P/G) = P$ if and only if P is $\mathbf{\Gamma}$-controllable and $\max\Gamma_P(q)$ exists uniquely for each $q \in P$. Moreover, the state policy f_P can be taken by*

$$f_P(q) = \max \Gamma_P(q) = \max\{\gamma \mid \gamma \in \Gamma(q),\ \delta(\sigma,\ q) \in P,\ \forall \sigma \in \Sigma(q) \wedge \gamma\},\quad q \in P. \tag{22}$$

□

When the given predicate P is not $\mathbf{\Gamma}$-controllable, we assume that $\Gamma(q)$ is closed under union $\vee$ for each $q \in P$. Then the maximal $\mathbf{\Gamma}$-controllable sub-predicate of P, denoted by $P^{\uparrow}$, is unique. Also, we introduce a stationary reward function by $c(q,\ \gamma) = 0$ if $q \in P$, $\delta(\sigma,\ q) \in P$, $\forall \sigma \in \Sigma(q) \wedge \gamma$ and $= -1$ else. We still let $V^0(q)$ be the optimal value function and let $P^* = \{q \in P \mid V^0(q) = 0\}$. We have the following theorem about P^* and $P^{\uparrow}$, which can be similarly proved as in Theorem 7.

Theorem 9 *P^* is the maximal $\mathbf{\Gamma}$-invariant sub-predicate of P and f_{P^*} is the maximum optimal policy of the corresponding stationary optimal control problem. Moreover, $P^{\uparrow} = R(f_{P^*}/G)$ is the maximal $\mathbf{\Gamma}$-closed sub-predicate of P^*.* □

6 Job Matching Problem

We consider a shop with two machines, M_1 and M_2, and two job types, J_1 and J_2. A job of J_1 must be processed first in M_1 and then in M_2, while a job of J_2 must be processed first in M_2 and then in M_1. A job is said to be completed if it is completed in both machines. Suppose that completed jobs are output to another system to to be equipped and that the number of jobs of J_1 should equal the number of jobs of J_2 in each output (which is called job matching).

The shop can be modelled by an automaton G as given in Fig. 1.

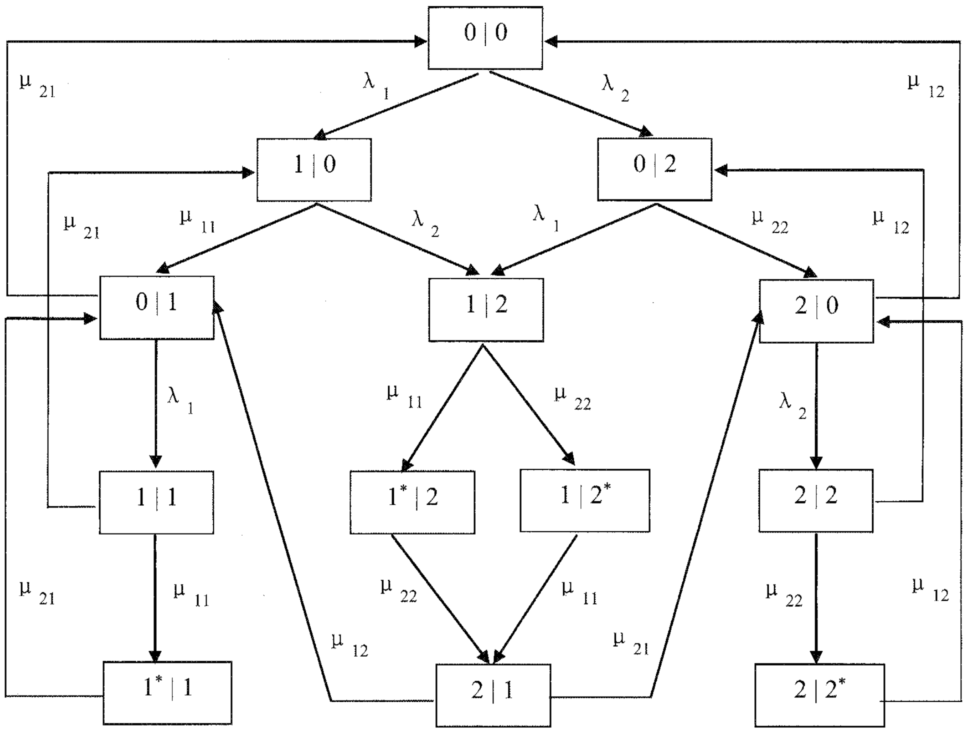

Figure 1. A job matching problem: automaton G.

The state variable is defined as (i, j) with $i \in \{0, 1, 2, 1^*\}$ and $j \in \{0, 1, 2, 2^*\}$. Here, $i=0$ (or $j=0$) represents that Machine 1 (or 2) is empty, $i \in \{1, 2\}$ (or $j \in \{1, 2\}$) represents that a job of J_i (or J_j) is being processed in M_1 (or M_2), while $i=1^*$ (or $j=2^*$) represents that a job of J_1 (or J_2) is completed and waiting in M_1 (or M_2). A job of J_i is waiting in M_i means that there is another job (of J_1 or J_2) being processed or waiting in M_j $(j \neq i)$. Let Q be the set of all states.

The event set is $\Sigma = \{\lambda_i, \mu_{ij} \mid i, j = 1, 2\}$, where event λ_i represents the arrival of a job of J_i, event μ_{ij} represents completion of a job of J_i in machine j. Here it is assumed that only the arrival events can be controlled. Hence, the uncontrollable event set is $\Sigma_u = \{\mu_{ij} \mid i, j = 1, 2\}$, while the controllable event set is $\Sigma_c = \{\lambda_1, \lambda_2\}$.

Suppose that there are N trays in the shop to take jobs. Each tray is common, that is, each tray can take a job either of J_1 or of J_2. When there is an empty tray, then a job can join the shop. The tray will take the job to the two machines and then output the job. It is not allowed that all the N trays take jobs of one type.

For any $s \in L(G)$, if we let $|s|_{12}$ be the difference of the number of event λ_1 in s minus that of event λ_2 in s, then $|s|_{12}$ represents the number of jobs of J_1 that have joined the shop minus that of J_2 that have joined the shop. Due to the requirements of job matching and that there are N trays, $|s|_{12} \in \{-N, -N+1, ..., 0, ..., N-1, N\}$. But if $|s|_{12} = -N$ or N then all the trays are taking jobs of the same type and so the system becomes deadlocked. Hence, we need some control mechanism. Here, we assume that the set of control inputs available at string $s \in L(G)$ is given by

$$\Gamma(s) = \begin{cases} \Gamma^-, & \text{if } |s|_{12} = -N+1 \\ \Gamma^+, & \text{if } |s|_{12} = N-1 \\ \Gamma, & \text{otherwise} \end{cases}$$

where

$$\Gamma^- = \{\gamma \in \Gamma \mid \lambda_2 \notin \gamma\}, \quad \Gamma^+ = \{\gamma \in \Gamma \mid \lambda_1 \notin \gamma\}.$$

Under any control input $\gamma \in \Gamma(s)$ at string s with $||s|_{12}| < N$, the system will never become deadlocked.

Moreover, the cost function is assumed as follows:

$$\begin{aligned} c(s, \gamma) = {} & c_1 \chi(\lambda_1 \notin \gamma, \lambda_1 \in \Sigma(s)) + c_2 \chi(\lambda_2 \notin \gamma, \lambda_2 \in \Sigma(s)) \\ & + \infty \chi(\gamma \wedge \Sigma(s) = \emptyset), \qquad s \in L(G), \gamma \in \Gamma. \end{aligned}$$

Here χ is the indicator function, $c_i \geq 0$ is the cost for prohibiting event λ_i for $i = 1, 2$, and the last term $\chi(\gamma \wedge \Sigma(s) = \emptyset)$ means that the deadlock is not allowed. Furthermore, for string s, if $\lambda_i \notin \Sigma(s)$ then event λ_i could not occur at string s and so there is no cost to prohibit λ_i at s. We take the discounted factor by any $\beta \in (0, 1)$. Since $c(s, \gamma)$ is bounded, $\Gamma_1(s) = \Gamma(s)$ for all string s. It should be noted that if $\Gamma(s) = \Gamma$ then we will have the same result as in the following except that $\Gamma_1(s)$ is just the $\Gamma(s)$ defined previously in the third term in $c(s, \gamma)$.

By noting that the cost function $c(s, \gamma)$ depends on string s only through state $q = \delta(s, q_0)$, we let

$$\begin{aligned} c(q, \gamma) = {} & c_1 \chi(\lambda_1 \notin \gamma, \lambda_1 \in \Sigma(q)) + c_2 \chi(\lambda_2 \notin \gamma, \lambda_2 \in \Sigma(q)) \\ & + \infty \chi(\gamma \wedge \Sigma(q) = \emptyset), \qquad q \in Q, \quad \gamma \in \Gamma. \end{aligned}$$

Then

$$c(s,\gamma)=c(\delta(s,q_0),\gamma),\qquad s\in L(G),\ \gamma\in\Gamma.$$

Let $V^*(s)$ be the minimal discounted total cost when string s has occurred. Since minimizing cost can be mathematically equivalent to maximizing reward, we can use all the previous results except that $\max_\gamma$ should be replaced with $\min_\gamma$ in the optimality equation.

Since $\Gamma(s)$ depends on s only through $|s|_{12}$, it can be proved from Theorems 1 and 2 that $V^*(s)$ depends on s only through $\delta(s,q_0)$ and $|s|_{12}$. It is easy to see that under each policy, $|s|_{12}\in\mathcal{N}:=\{-N+1,\dots,0,\dots,N-1\}$. Then we let the function $V^*(q,n)$, be defined on $Q\times\mathcal{N}$, such that

$$V^*(s)=V^*(\delta(s,q_0),|s|_{12}).$$

We introduce a variable $n=|s|_{12}$, which has the following transition law based on the changes of the system for any $n\in\mathcal{N}$: $\eta(\sigma,n)=n+1$ if $\sigma=\lambda_1$, $=n-1$ if $\sigma=\lambda_2$, and n otherwise.

Since we want to minimize the discounted total cost, "max" in the optimality equation (9) should be replaced by "min". Then, the optimal value function $V^*(i,j;n)$ (here $q=(i,j)$) is the unique bounded solution of the following optimality equation

$$\begin{aligned}&V(i,j;n)=\min_{\gamma\in\Gamma_n}\{c(i,j,\gamma)+\beta\max_{\sigma\in\Sigma_\gamma(i,j)}V(\delta(\sigma,i,j),\eta(\sigma,n))\},\\&(i,j)\in Q;\quad n\in\mathcal{N}\end{aligned}\tag{23}$$

where $\Gamma_{-N+1}=\Gamma^-$, $\Gamma_n=\Gamma$ for $n=-N+2,\dots,N-2$ and $\Gamma_{N-1}=\Gamma^+$.

It suffices to solve the following equations (where $n\in\mathcal{N}$):

$$\begin{aligned}V(i,j;n)&=\min_{\gamma\in\Gamma_n}\{c(i,j,\gamma)+\beta\max_{\sigma\in\Sigma_\gamma(i,j)}V(\delta(\sigma,i,j),\eta(\sigma,n))\},(i,j)\in Q,\\V(1,2;n)&=\beta^2V(2,1;n),\\V(1,1;n)&=\beta\max\{V(1,0;n),\beta V(0,1;n)\},\\V(2,2;n)&=\beta\max\{V(0,2;n),\beta V(2,0;n)\},\\V(2,1;n)&=\beta\max\{V(0,1;n),V(2,0;n)\}.\end{aligned}\tag{24}$$

The above set of equations can be computed by successive approximation. That is, for any given initial value of $V_0(i,j;n)$ (e.g., $V_0(i,j;n)=0$ for all i,j,n), then we iteratively compute $V_{k+1}(i,j)$ for $k=0,1,\dots$ by

$$\begin{aligned}
V_{k+1}(i,j;n) &= \min_{\gamma\in\Gamma_n}\{c(i,j,\gamma)+\beta \max_{\sigma\in\Sigma_\gamma(i,j)} V_k(\delta(\sigma,i,j),\eta(\sigma,n))\},\ \ (i,j)\in Q^*,\\
V_{k+1}(1,2;n) &= \beta^2 V_k(2,1;n),\\
V_{k+1}(1,1;n) &= \beta \max\{V_k(1,0;n), \beta V_k(0,1;n)\},\\
V_{k+1}(2,2;n) &= \beta \max\{V_k(0,2;n), \beta V_k(2,0;n)\},\\
V_{k+1}(2,1;n) &= \beta \max\{V_k(0,1;n), V_k(2,0;n)\}.
\end{aligned} \tag{25}$$

Similarly as in Markov decision processes (Hu and Yue [20]), it can be proven that

$$\lim_{k\to\infty} V_k(i,j;n) = V(i,j;n), \qquad \forall i,j,n.$$

So, for a given small constant $\epsilon > 0$, when $|V_{k+1}(i,j;n) - V_k(i,j;n)| < \epsilon$ for all i, j, n, we stop the above iterative procedure and take $V_{k+1}(i,j;n)$ as an approximating value of $V(i,j;n)$ for i, j, n.

In the following, we solve the optimality equation for $N=2$, i.e., there are only two trays in the shop.

We use the successive approximation to solve the above equations with $c_1=1$, $c_2=5$, $\beta=0.99$ and $\epsilon=0.01$. The successive approximation stops when the iteration step is 517 and the result is given in Table 1. From this, with the optimality equation, we obtain the optimal policy as given in Table 2, where $\Sigma_1=\Sigma-\{\lambda_1\}$ and $\Sigma_2=\Sigma-\{\lambda_2\}$. In fact, the optimal policy is maximal among all available control inputs except at (0, 0; 0) where it takes Σ_2, a real subset of the maximum Σ among all available control inputs.

When the discounted factor β is smaller, the number of steps that are needed will be smaller. For example, with the same parameters as above but with a discount factor $\beta=0.95$, the number of iteration steps is 103 for stopping the successive approximation.

Table 1. Optimal values for $c_1=1$, $c_2=5$ and $\beta=0.99$.

$V(i,j;n)$	(0, 0)	(1, 0)	(0, 2)	(0, 1)	(2, 0)	(1, 2)	(1, 1)	(2, 2)	(2, 1)
$n=0$	163.51	163.00	163.00	161.87	164.65	159.76	161.37	161.37	163.00
$n=-1$	166.37	168.05	168.00	164.71	169.71	164.65	166.36	166.32	168.00
$n=1$	162.37	160.12	160.13	161.75	160.75	156.92	158.52	158.52	160.12

Table 2. Optimal policy for $c_1 = 1$, $c_2 = 5$ and $\beta = 0.99$.

$f^*(i, j; n)$	(0, 0)	(1, 0)	(0, 2)	(0, 1)	(2, 0)	(1, 2)	(1, 1)	(2, 2)	(2, 1)
$n = 0$	Σ_2	Σ	Σ	Σ	Σ	Σ	Σ	Σ	Σ
$n = -1$	Σ_2	Σ_2	Σ	Σ	Σ_2	Σ	Σ	Σ	Σ
$n = 1$	Σ_1	Σ	Σ_1	Σ_1	Σ	Σ	Σ	Σ	Σ

Now we discard the restriction on $\Gamma(s)$ and the finiteness of trays. Suppose that $\Gamma(s) = \Gamma$ for all $s \in L(G)$ and there are infinite trays in the shop. The objective is to control the system such that the number of completed jobs of any type that are waiting for output is, at most, one. That is, the language to be synthesized is $K = \{s \in L(G): ||s|_{12}| \leq 1\}$.

Then from the results in Section 5, it is easy to see that $K^* = K$,

$$\pi_K(s) = \begin{cases} \Sigma - \{\lambda_1\}, & \text{if } |s|_{12} = 1 \\ \Sigma - \{\lambda_2\}, & \text{if } |s|_{12} = -1 \\ \Sigma, & \text{otherwise} \end{cases}$$

and the maximal closed sub-language of K is

$$K^{\uparrow} = L(\pi_K/G) = \{s \in K: ||t|_{12}| \leq 1, \forall t \leq s\}.$$

The optimal policy π_K can be realized by

a) a three states automaton $\mathcal{S} = \{S, \Sigma_c, \eta\}$ with the state space $S = \{-1, 0, 1\}$, the event set $\Sigma_c = \{\lambda_1, \lambda_2\}$ and the state transition function $\eta(\sigma, n)$ which was defined previously, and

b) a map $\phi: S \to \Gamma$ such that $\phi(1) = \Sigma - \{\lambda_1\}$, $\phi(-1) = \Sigma - \{\lambda_2\}$ and $\phi(0) = \Sigma$. This means that the control input is $\phi(n)$ whenever the automaton $\mathcal{S}$ is at state n. The automaton $\mathcal{S}$ is simpler than the original system G described in Fig. 1.

7 Conclusions

In this paper, we presented a new problem for optimal control of discrete event systems with an arbitrary control pattern. Here the discrete event system is defined by a collection of event sets depending on strings, while the control pattern also depends on strings. We solved the problem through the optimality equations by applying ideas from Markov decision processes under necessary conditions. When the reward function is stationary, the optimality equation and the optimal policies are also stationary. Moreover, based on the above model and its results, we studied the supervisory control problem with control pattern depending on strings by generalizing the concepts of invariant and closed languages/predicates. Finally, we used the model for optimal control to solve a job matching-problem.

Further research may include the optimal control and the supervisory control with arbitrary control pattern of DESs with incomplete information (such as partially observable DESs and decentralized supervisory control), etc.

Acknowledgment

The research was supported in part by the National Natural Science Foundation of China (No. 70271021 and No. 69904008) and was supported in part by GRANT-IN-AID FOR SCIENTIFIC RESEARCH (No. 16560350) and by MEXT.ORC (2004-2008), Japan.

References

[1] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event systems," *SIAM J. Control and Optimization*, Vol. 25, pp. 206-230, 1987.

[2] P. J. Ramadge and W. M. Wonham, "Modular feedback logic for discrete event systems," *SIAM J. Control and Optimization*, Vol. 25, pp. 1202-1218, 1987.

[3] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM J. Control and Optimization*, Vol. 25, No. 3, pp. 637-659, 1987.

[4] Y. Li and W. M. Wonham, "Controllability and observability in the state-feedback control of discrete-event systems," *Proc. of IEEE Conf. Decision & Control*, pp. 203-208, 1989.

[5] Y. Lin and W. M. Wonham, "On the observability of discrete-event systems," *Information Sci.*, Vol. 44, pp. 173-198, 1989.

[6] S. Takai, "Supervisory control of partially observed discrete event systems with arbitrary control patterns," *International J. Sys. Sci.*, Vol. 31, pp. 649-656, 2000.

[7] K. Rudie, S. Lafortune and F. Lin, "Minimal communication in a distributed

discrete-event system," *IEEE Trans. on Automatic Control*, Vol. 48, pp. 957-975, 2003.

[8] Z. Chao and Y. Xi, "Necessary conditions for control consistency in hierarchical control of discrete-event systems," *IEEE Trans. on Automatic Control*, Vol. 48, pp. 465-468, 2003.

[9] J. N. Tsitsiklis, "On the control of discrete event dynamic systems," *Math. Control Signals Sys.*, Vol. 2, pp. 95-107, 1989.

[10] M. Passino and P. J. Antsaklis, "On the optimal control of discrete event systems," *Proc. of IEEE Conf. Decision & Control*, pp. 2713-2718, 1989.

[11] Y. Brave and M. Heymann, "On optimal attraction in discrete event systems," *Information Sci.*, Vol. 67, pp. 245-267, 1993.

[12] R. Kumar and V. K. Garg, "Optimal supervisory control of discrete event dynamic systems," *SIAM J. Control and Optimization*, Vol. 33, pp. 419-439, 1995.

[13] T. Yamasaki and T. Ushio, "Supervisory control of partially observed discrete event systems based on reinforcement learning," *Proc. of the 2003 IEEE International Conf. on Sys., Man, and Cybern.*, pp. 2956-2961, 2003.

[14] J. Fu, A. Ray, and C. M. Lagoa, "Unconstrained optimal control of regular languages," *Automatica*, Vol. 40, No. 4, pp. 639-646, 2004.

[15] A. Ray, J. Fu, and C. Lagoa, "Optimal supervisory control of finite state automata," *International J. of Control*, Vol. 77, No. 12, pp. 1083-1100, 2004.

[16] C. H. Golaszewski and P. J. Ramadge, "Control of discrete event processes with forced events," *Proc. of the 26th IEEE Conf. on Decision and Control*, pp. 247-251, 1987.

[17] Y. Li, F. Lin and Z. H. Lin, "A generalized framework for supervisory control of discrete event systems," *International J. of Intelligent Control & Sys.*, Vol. 2, pp. 139-160, 1998.

[18] B. A. Brandin and W. M. Wonham, "Supervisory control of timed discrete event systems," *IEEE Trans. on Automatic Control*, Vol. 39, pp. 329-342, 1994.

[19] Q. Hu and C. Xu, "The finiteness of the reward function and the optimal value function in Markov decision processes," *Math. Methods in Oper. Res.*, Vol. 49, pp. 255-266, 1999.

[20] Q. Hu and W. Yue, "Analysis for some properties of discrete time Markov decision processes," *J. of Optimization*, Vol. 52, pp. 495-505, 2003.

[21] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. of the IEEE*, Vol. 77, pp. 81-98, 1989.