Optimal Control for Resource Allocation in 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 9, 2004)

Abstract

Supervisory control for discrete event systems (DESs) belongs essentially to the logic level for control problems in DESs. Its corresponding control task is hard. In this paper, we study a new optimal control problem in DESs. The performance measure is to maximize the maximal discounted total reward among all possible strings (i.e., paths) of the controlled system. The condition we need for this is only that the performance measure is well defined. We then divide the problem into three sub-cases where the optimal values are respectively finite, positive infinite and negative infinite. We then show the optimality equation in the case with a finite optimal value. Also, we characterize the optimality equation together with its solutions and characterize the structure of the set of all optimal policies. When the reward function is stationary, we show that the optimality equation and its solution are also stationary. All the above results are still true when the performance measure is to maximize the minimal discounted total reward among all possible strings of the controlled system. Finally, we apply these equations and solutions to a resource allocation system. The system may be deadlocked and in order to avoid the deadlock we can either prohibit the occurrence of some events or resolve the deadlock. It is shown that from the view of the maximal discounted total cost, it is better to resolve the deadlock if and only if the cost for resolving the deadlock is less than the threshold value.

Keywords: Optimal control, discrete event systems, resource allocation system.

---

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

1 Introduction

Discrete event systems (DESs) are those systems that are driven by often occurring finite events. Supervisory control of DESs, presented by Ramadge and Wonham [1], [2] and [3], belongs to the logic level of control for DESs [4]. However, there are only a few papers that discuss optimal control problems based on the framework of the supervisory control. The essential task of the supervisory control is to constrain the system's behavior in a given region, where the system's behavior is at most times described by a set of strings of occurring events or, at other times, by a state subset. At the same time, the control task in the supervisory control is categorized as hard, where strings in the given behaviors are allowed and all other strings are strictly prohibited. However, there are many practical problems in attaining optimal control which belong to the performance level (i.e., optimizing some performance measures). And the control tasks in these practical problems are soft where it is better if some behavior occurs.

Considering the reward of occurring events at states, Passino and Antsaklis [5] 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 branching-bounding algorithm.

Tsitsiklis [6] presented a dynamic programming model to solve some synthesizing problems in the supervisory control.

Kumar and Garg [7] studied an optimal static control problem with two reward functions $c(q,\sigma)$ and $p(q)$, but they assumed that these two rewards occurred only once. So the problem is, in fact, a static designing problem. They used the maximal-flow-minimal-cut theorem to solve the problem, based on which they presented algorithms to compute the maximal sub-controllable languages for the supervisory control.

Some studies on optimal control problems have been used to solve stability problem of DESs. Considering the reward for occurring events at states, Brave and Heymann [8] calculated the optimal attraction by minimizing the total reward among all possible strings from an arbitrary state to a given global attraction, found conditions for the existence of supervisors achieving optimal attraction and provided efficient algorithms for their synthesis. Hu and Liu [9] used Markov decision processes to study the static stability problems in DESs.

But all of the above researches either related special optimal control problems with special reward functions to solve some problems in the supervisory control, or, they were concerned with the static control of DESs [7], and were not concerned with general frameworks for optimal control of DESs.

In this paper, we present a new model for optimal control problems in DESs. In the model, there are rewards for occurring events 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. By applying ideas from Markov decision processes [10] and

[11], 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, we show the optimality equation and characterize the optimality equation together 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. All of the above results are still true when the performance measure is to maximize the minimal discounted total reward among all strings of the controlled system. Finally, we apply these equations and solutions to the resource allocation of a system. The system may be deadlocked so we can either prohibit the occurrence of some events in order to avoid a deadlock or we can resolve the deadlock. It is shown that from the view of the maximal discounted total cost, it is better to resolve the deadlock if and only if the cost of resolving the deadlock is less than the threshold value.

The remainder of the paper is organized as follows. In Section 2, we give the system model with notations and preliminaries. In Sections 3, we discuss the optimality equation and its solutions. In Section 4, we use the optimal control analysis to describe and solve the resource allocation problems in a system model. Section 5 contains the paper's conclusions.

2 System Model

For a finite event set Σ, we denote by Σ^* the set of all finite strings on Σ including the empty string ϵ. For strings s,t and r, if $s=tr$ then we call t a prefix of s and denote it by $t\leq s$. We call sets of Σ^* languages. For a language $L\subset\Sigma^*$, we define its closure, denoted by $\bar{L}$, a set of all prefixes of strings in L. We call L closed if $\bar{L}=L$. 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 by $t\leq s$. We call sets of Σ^ω infinite languages.

A discrete event system based on automatons is $G=\{Q,\Sigma,\delta,q_0\}$, where Q is a countable state space, Σ is a finite event set, δ is a partial function from $\Sigma\times Q$ to Q, while $q_0\in Q$ is the initial state. We generalize δ by $\delta(s\sigma,q)=\delta(\sigma,\delta(s,q))$ inductively for $s\sigma\in\Sigma^*$, and when $\delta(s,q)$ is well defined we will denote it by $\delta(s,q)!$. Moreover, we let $\Sigma(q)=\{\sigma\,|\,\delta(\sigma,q)!\}$ be set of events that may occur at state q and $\Sigma(s)=\{\sigma\,|\,\delta(s\sigma,q_0)!\}$ be set of events that may occur after string s. The language generated by G is defined by $L(G)=\{s\in\Sigma^*\,|\,\delta(s,q_0)!\}$ and the infinite language generated by G is $L^\omega(G)=\{s\in\Sigma^\omega\,|\,\delta(t,q_0)!$ for all $t\leq s\}=\{s\in\Sigma^\omega\,|\,t\in L(G)$ for all $t\leq s\}$. It is assumed that G is alive, i.e., $\Sigma(q)$ is non-empty for each state $q\in Q$. This assumption can be relaxed. In fact, if there are some empty $\Sigma(q)$ then we introduce a fictitious event $\sigma_J\notin\Sigma$ and let $\delta(\sigma_J,q)=q$ whenever $\Sigma(q)$ is empty.

The event set Σ is divided into an uncontrollable event set Σ_u and a controllable event set Σ_c. Σ_u and Σ_c are disjoint. A control input is an event subset γ satisfying $\Sigma_u\subset\gamma\subset\Sigma$.

The set of such control inputs is denoted by Γ. A control input γ is chosen at string s (or state q) means that the next event should be in $\Sigma(s) \cap \gamma$ (or $\Sigma(q) \cap \gamma$).

We define a policy as a mapping $\pi : L(G) \to \Gamma$ and a stationary policy as a mapping $f: Q \to \Gamma$. Under the policy $\pi, \pi(s) \in \Gamma$ is chosen whenever the string s occurs, while under the stationary policy f, $f(q) \in \Gamma$ is chosen whenever the state q is visited. A stationary policy f is a special policy π with $\pi(s) = f(\delta(s, q_0))$ for all $s \in L(G)$. We denote the sets of policies and stationary policies by Π and F, respectively. In the supervisory control theory ([1], [2] and [3]), a policy π is called a supervisor, while a stationary policy f is called a state feedback.

We introduce several concepts in DESs (see [1], [2] and [3]). For a policy π, we define the language $L(\pi/G)$ generated by the system supervised under π inductively by: (a) $\epsilon \in L(\pi/G)$ and (b) if $s \in L(\pi/G)$ while $\sigma \in \Sigma(s)$ and $\sigma \in \pi(s)$, then $s\sigma \in L(\pi/G)$. The infinite language generated by the system supervised under π is denoted by $L^{\omega}(\pi/G)$. Let $L^{\omega}(\pi/G, s) = \{t \in \Sigma^{\omega} | s \cdot t \in L^{\omega}(\pi/G)\}$ for $s \in L(G)$. Moreover, for any $f \in F$, we define a system as $f/G := \{Q, \Sigma, \delta_f, q_0\}$, where $\delta_f(\sigma, q) = \delta(\sigma, q)$ is well defined, if and only if $\sigma \in f(q)$ and $\delta(\sigma, q)!$.

For the given discrete event system G, suppose that there is an extended real-valued reward function $c(s, \sigma) \in [-\infty, +\infty]$ for an event σ occurring after string s for $s \in L(G)$ and $\sigma \in \Sigma(s)$. If the fictitious event σ_J has been introduced to ensure that G is alive, then we let $c(s, \sigma_J) = 0$ for string s with that $\Sigma(s)$ is empty.

For any infinite string $t = \sigma_0\sigma_1\sigma_2 \cdots \in \Sigma^{\omega}$, we define its prefix as $t_k = \sigma_0\sigma_1 \cdots \sigma_{k-1}$ for $k \geq 1$ and $t_0 = \epsilon$. Let $\beta > 0$ be a discount factor. For each finite string $s \in L(G)$ and each infinite string $t \in \Sigma^{\omega}$ with $st \in L^{\omega}(G)$, we define by

$$v_s(t) = \sum_{k=0}^{\infty} \beta^k c(s \cdot t_k, \sigma_k)$$

as the discounted total reward for occurring t after s. We will simply call $v_s(t)$ *the reward for t after s*.

In general, there are infinite possible strings that may be generated by the system (G or π/G), but there is only one string that will be generated. We cannot know which string will be generated before the end of the system. Thus we consider respectively the maximal discounted total reward and the minimal discounted total reward of all possible strings that may be generated by the system controlled under π. Formally, we define

$$I(\pi, s) = \sup_{t \in L^{\omega}(\pi/G, s)} v_s(t), \quad s \in L(G), \tag{1}$$

$$J(\pi, s) = \inf_{t \in L^{\omega}(\pi/G, s)} v_s(t), \quad s \in L(G) \tag{2}$$

as respectively the maximal discounted total reward and the minimal discounted total reward of the system supervised under π when string s has occurred. Knowing the supremum and the infimum of the discounted total rewards may help us to resolve the system.

We define the optimal value functions, respectively by

$$I^*(s) = \sup_{\pi \in \Gamma^{L(G)}} I(\pi, s), \quad s \in L(G), \tag{3}$$

$$J^*(s) = \sup_{\pi \in \Gamma^{L(G)}} J(\pi, s), \quad s \in L(G). \tag{4}$$

$I^*(s)$ and $J^*(s)$ are respectively the best case and the worst case we have for the discounted total reward. We call a policy π^* I-optimal at string s if $I(\pi^*, s) = I^*(s)$ and call π^* I-optimal if it is optimal at all $s \in L(G)$. J-optimal policies are defined similarly.

3 Optimality

The optimality in the model is studied by using ideas presented in [10] for Markov decision processes (MDPs). Following [10], we define the following general condition.

Condition (GC): $v_s(t)$, the discounted total reward for occurring t after s is well defined for each finite string $s \in L(G)$ and each infinite string $t \in \Sigma^\omega$ with $st \in L^\omega(G)$.

We should point out that $v_s(t)$ is well defined as a series where t is infinite. Condition (GC) will be true, for example, when the reward function $c(\cdot,\cdot)$ is nonnegative, or is nonpositive, or is uniformly bounded and $\beta \in (0,1)$. Condition (GC) implies that both of the objective functions $I(\pi,s)$ and $J(\pi,s)$ are well defined for each policy π and string $s \in L(G)$. Surely, condition (GC) is the base for discussing the optimal control problem. Thus, the condition is assumed throughout this paper.

The following lemma is obvious from condition (GC).

Lemma 1. 1) For any infinity string $t = \sigma_0\sigma_1 \cdots \in L^\omega(G)$, there is no simultaneous positive infinity reward and negative infinity reward in t, i.e., there is no $k_1, k_2 \geq -1$ such that $c(t_{k_1}, \sigma_{k_1+1}) = +\infty$ but $c(t_{k_2}, \sigma_{k_2+1}) = -\infty$, where $t_k = \sigma_0\sigma_1 \cdots \sigma_k$ for $k \geq 0$ and $t_{-1} = \epsilon$.

2) For each infinite string $s \cdot t \in L^\omega(G)$, both $v_\epsilon(s)$ and $v_s(t)$ would not be simultaneously infinity with different symbols. □

Now we introduce some concepts. Suppose that $\Sigma'(s) \subset \Sigma(s)$ is an event subset for each $s \in L(G)$. We call $\Sigma' := \{\Sigma'(s), s \in L(G)\}$ to be a constrain on G. When we constrain the DES G by Σ', we mean that the event set $\Sigma(s)$ is replaced by $\Sigma'(s)$ at the string s, that is, the event that can occur at s is among $\Sigma'(s)$. Moreover, for any $s \in L(G)$ and $r = \sigma_0\sigma_1 \cdots \sigma_k \in \Sigma^*$, if $\sigma_l \in \Sigma'(s\sigma_0\sigma_1 \cdots \sigma_{l-1}\})$ for all $l = 0, 1, \cdots, k$, then we say that r can occur after the string s through Σ'. The set of such strings r is denoted by $\Sigma'^*(s)$.

Definition 1.
1) For two strings $s, t \in L(G)$, if there is $r \in \Sigma'^*(s)$ such that $t = sr$ then we say that t can be reached from s through Σ', or s can reach t through Σ', and is denoted by $s \xrightarrow{r}_{\Sigma'} t$ or simply $s \rightarrow_{\Sigma'} t$. It is clear that $t \xrightarrow{\epsilon}_{\Sigma'} t$ for each string $t \in L(G)$.
2) For a language $L \subset L(G)$ and a string $s \in L(G)$, if there is string $t \in$ L such that $s \rightarrow_{\Sigma'} t$ then we say that L can be reached from s through Σ' and is denoted by $s \rightarrow_{\Sigma'} L$. $L \rightarrow_{\Sigma'} s$ is similar.
3) For two languages L_1, $L_2 \subset L(G)$, if L_1 can reach some string in L_2 through Σ' then we say that L_2 can be reached from L_1 through Σ', or L_1 can reach L_2 through Σ', and is denoted by $L_1 \rightarrow_{\Sigma'} L_2$.

For a language $L \subset L(G)$, let $L^*_{\Sigma'} = \{t \mid L \rightarrow_{\Sigma'} t\}$ be the set of strings that can be reached from L through Σ', and $\overline{L}_{\Sigma'} = \{t \mid t \rightarrow_{\Sigma'} L\}$ be the set of strings that can reach L through Σ'. It is obvious that $\overline{L}_{\Sigma'} \rightarrow_{\Sigma'} L \rightarrow_{\Sigma'} L^*_{\Sigma'}$. Moreover, $L \subset L^*_{\Sigma'}$ and $L \subset \overline{L}_{\Sigma'}$. For the reverse inclusions, we introduce the following definition.

Definition 2. For a language $L \subset L(G)$, if $L^*_{\Sigma'} = L$ then we call L invariant under Σ', if $\overline{L}_{\Sigma'} =$ L then we call L closed under Σ'.

It is clear that when $\Sigma'(s) = \Sigma(s)$ for all $s \in L(G)$, the closed language defined above is identical to that defined in literature of the supervisory control [4]. Under the constrain of Σ', a language L is closed means that the past of L is included in L itself, while L is invariant means that L includes its future, that is, the system will remain in L whenever the system begins in L.

3.1 Maximum Discounted Total Reward

We discuss the objective $I(\boldsymbol{\pi}, s)$ in this subsection. It is easy to see that the policy $\boldsymbol{\pi}^*$ with $\boldsymbol{\pi}^*(s) = \Sigma$ for each $s \in L(G)$ is I-optimal. Hence,

$$I^*(s) = I(\boldsymbol{\pi}^*, s) = \max_{t \in L^\omega(G,s)} v_s(t), \quad s \in L(G). \tag{5}$$

Then the remaining problems are, in fact, to find strings with the maximal discounted total reward and the value of $I^*(s)$. We call a string $t \in L^\omega(G, s)$ a maximal string from string s if $v_s(t) = I^*(s)$. In the following, we discuss how to find a maximal string. Obviously, this problem is a dynamic programming problem, where the state space is $L(G)$, the action set at $s \in L(G)$ is the event set $\Sigma(s)$, the reward function is $c(s, \sigma)$ and the state transition is $(s, \sigma) \rightarrow s\sigma$. Since the reward function is unbounded and the number of horizons is infinite, we can not directly use the optimality principle from dynamic programming. We shall prove the optimality equation by applying the ideas used by Hu and Xu in [10] for Markov decision processes.

We let for $s \in L(G)$ an event subset to be

$$\Sigma_1(s) = \{\sigma \in \Sigma(s) \,|\, c(s,\sigma) > -\infty\}.$$

Lemma 2. I^* satisfies the following I-optimality equation

$$I(s) = \max_{\sigma \in \Sigma_1(s)} \{c(s,\sigma) + \beta I(s\sigma)\}, \quad s \in L(G). \tag{6}$$

Here, the maximum value is defined to be $-\infty$ when $\Sigma_1(s)$ is empty.

Proof. For each $s \in L(G)$, if $\Sigma_1(s) = \emptyset$, then $c(s,\sigma) = -\infty$ for each $\sigma \in \Sigma(s)$. Hence, for each $t \in L^\omega(G,s)$, $v_s(t) = -\infty$ and so $I^*(s) = -\infty$. Therefore, Eq. (6) is true for such string s.

Then we suppose that $\Sigma_1(s)$ is non-empty. In this case, for any string $t \in L^\omega(G,s)$, if $t = \sigma t'$ with $c(s,\sigma) = -\infty$ then $v_s(t) = -\infty$ and such string would not be maximal. Hence,

$$I^*(s) = \max_{\sigma t \in L^\omega(G,s), \sigma \in \Sigma_1(s)} v_s(t).$$

This results in that $\Sigma(s)$ can be sized down to $\Sigma_1(s)$ at string s. Moreover, if there is an event $\sigma \in \Sigma_1(s)$ such that $c(s,\sigma) = +\infty$, then it is easy to see that $I^*(s) = +\infty$ and so the I-optimality equation (6) is true for string s. Otherwise, $c(s,\sigma)$ is finite for all $\sigma \in \Sigma_1(s)$. Then

$$\begin{aligned}
I^*(s) &= \max_{\sigma_0 \sigma_1 \cdots \in L^\omega(G,s), \sigma_0 \in \Sigma_1(s)} \sum_{k=0}^{\infty} \beta^k c(s\sigma_0\sigma_1 \cdots \sigma_{k-1}, \sigma_k) \\
&= \max_{\sigma_0 \in \Sigma_1(s)} \max_{\sigma_1 \sigma_2 \cdots \in L^\omega(G,s\sigma_0)} \left\{ c(s,\sigma_0) + \beta \sum_{k=1}^{\infty} \beta^{k-1} c(s\sigma_0\sigma_1 \cdots \sigma_{k-1}, \sigma_k) \right\} \\
&= \max_{\sigma_0 \in \Sigma_1(s)} \left\{ c(s,\sigma_0) + \beta \max_{\sigma_1 \sigma_2 \cdots \in L^\omega(G,s\sigma_0)} \sum_{k=1}^{\infty} \beta^{k-1} c(s\sigma_0\sigma_1 \cdots \sigma_{k-1}, \sigma_k) \right\} \\
&= \max_{\sigma \in \Sigma_1(s)} \{c(s,\sigma) + \beta I^*(s\sigma)\}.
\end{aligned}$$

Hence, the I-optimality equation (6) is true. □

We separate the language $L(G)$ into several sub-languages. Let

$$\begin{aligned}
L^0(G) &= \{s \in L(G) \,|\, I^*(s) \in (-\infty, +\infty)\}, \\
L^+(G) &= \{s \in L(G) \,|\, I^*(s) = +\infty\}, \\
L^-(G) &= \{s \in L(G) \,|\, I^*(s) = -\infty\}
\end{aligned}$$

be the sets of strings with finite, positive infinite, and negative infinite optimal values, respectively. From the proof of Lemma 2, we know that $\{s \in L(G) \,|\, \Sigma_1(s) = \emptyset\} \subset L^-(G)$. We further let $L^{+\infty}(G) = \{s \in L(G) \,|\, \text{there is } \boldsymbol{\pi} \text{ such that } I(\boldsymbol{\pi}, s) = +\infty\} = \{s \in L(G) \,|\, \text{there is } t \in \Sigma^\omega$ such that $s \cdot t \in L^\omega(G)$ and $v_s(t) = +\infty\}$. It is clear that

$$\{s \mid c(s,\sigma)=+\infty \text{ for some } \sigma\} \subset L^{+\infty}(G) \subset L^{+}(G).$$

We have the following lemma on $L^+(G)$ and $L^-(G)$.

Lemma 3. Under $\Sigma_1 := \{\Sigma_1(s),\ s \in L(G)\}$, $L^+(G)$ is closed while $L^-(G)$ is invariant. Moreover, there exists maximal strings from $s \in L^{+\infty}(G)$, there is no maximal string from $s \in L^+(G)-L^{+\infty}(G)$ and each string from $s \in L^-(G)$ is maximal.

Proof. We prove the lemma by the following three steps.
1) For any string $s \in L(G)$ with $s \to_{\Sigma_1} L^+(G)$, there are strings $t \in L^+(G)$ and $r \in \Sigma_1^*(s)$ such that $t = sr$. Since $v_s(r) > -\infty$ and $I^*(t) = +\infty$, we have

$$\begin{aligned} I^*(s) &= \max_{x \in L^{\omega}(G,s)} v_s(x) \\ &\geq \max_{y: ry \in L^{\omega}(G,s)} v_s(ry) \\ &= \max_{y \in L(G,sr)} \{v_s(r) + \beta^{|r|} v_{s \cdot r}(y)\} \\ &= v_s(r) + \beta^{|r|} \max_{y \in L^{\omega}(G,t)} v_t(y) \\ &= v_s(r) + \beta^{|r|} I^*(t) \\ &= \infty. \end{aligned}$$

Hence, $I^*(s) = \infty$ and $s \in L^+(G)$. Thus $L^+(G)$ is closed under Σ_1.
2) Suppose that $L^-(G)$ is not invariant, then there is string $t \in L(G)-L^-(G)$ with $L^-(G) \to_{\Sigma_1} t$. Surely, there are strings $s \in L^-(G)$ and $r \in \Sigma_1^*(s)$ such that $t = sr$. Then from the proof for 1) we have that

$$I^*(s) \geq v_s(r) + \beta^{|r|} I^*(s \cdot r) > -\infty.$$

This contradicts $I^*(s) = -\infty$ for $s \in L^-(G)$. Hence, $L^-(G)$ is invariant under Σ_1.
3) The remaining results are obvious. □

From the above lemma, we know that under the constrain Σ_1, each string t that can reach some string $s \in L^+(G)$ belongs to $L^+(G)$ too, while each string t that can be reached from some string $s \in L^-(G)$ belongs to $L^-(G)$ too. For $L^0(G)$, we let

$$\Sigma_2(s) = \begin{cases} \{\sigma \in \Sigma(s) \mid c(s,\sigma) > -\infty,\ s\sigma \in L^0(G)\}, & s \in L^0(G), \\ \{\sigma \in \Sigma(s) \mid c(s,\sigma) > -\infty\}, & \text{otherwise}. \end{cases}$$

It is obvious from Lemma 3 that under Σ_2, $L^+(G)$ is still closed and $L^-(G)$ is still invariant. Summarizing the above results, we have the following theorem.

Theorem 1. Under either Σ_1 or Σ_2, $L^+(G)$ is closed and $L^-(G)$ is invariant. While under Σ_2, $L^0(G)$ is invariant and the I-optimality equation in $L^0(G)$ is equivalent to

$$I(s) = \max_{\sigma\in\Sigma_2(s)} \{c(s,\sigma)+\beta I(s\sigma)\}, \quad s\in L^0(G). \tag{7}$$

□

With the above theorem, we focus our attention to $\{I^*(s),\ s\in L^0(G)\}$. In the following, we further characterize solutions of the I-optimality equation (7) in $L^0(G)$.

Let $\Sigma_2^\omega(s)$ be the infinite language defined similarly as the language $\Sigma'^*(s)$.

Lemma 4. We have the following four statements.

1) I^* satisfies the following condition:

$$\limsup_{n\to\infty} \beta^n I(s\cdot t_n)\geq 0, \qquad \forall t\in\Sigma_2^\omega(s) \text{ and } \forall s\in L^0(G) \text{ with } v_s(t)\neq-\infty. \tag{8}$$

2) $I\geq I^*$ if I is a solution of the I-optimality equation (7) and satisfies condition Eq. (8).

3) $I\leq I^*$ if I is a solution of the I-optimality equation (7) and satisfies

$$\liminf_{n\to\infty} \beta^n I(s\cdot t_n)\leq 0, \qquad \forall t\in\Sigma_2^\omega(s) \text{ and } \forall s\in L^0(G) \text{ with } v_s(t)\neq-\infty. \tag{9}$$

4) $I=I^*$ if I is a solution of the I-optimality equation (7) and satisfies

$$\lim_{n\to\infty} \beta^n I(s\cdot t_n)= 0, \qquad \forall t\in\Sigma_2^\omega(s) \text{ and } \forall s\in L^0(G) \text{ with } v_s(t)\neq-\infty. \tag{10}$$

Proof.

1) We denote by $t=\sigma_0\sigma_1\cdots$. The result follows $\beta^n I^*(s\cdot t_n)\geq\beta^n\Sigma_{k=n}^{\infty} c(s\cdot t_k,\sigma_k)$ which tends to zero since $v_s(t)$ is finite by the condition and $s\in L^0(G)$.

2) Suppose that I satisfies the given conditions. Then for each $s\in L^0(G)$ and $t\in\Sigma_2^\omega(s)$ with $v_s(t)\neq\infty$, it follows the I-optimality equation (7) that

$$I(s)\geq v_s(t_n)+\beta^n I(s\cdot t_n), \qquad n\geq 0.$$

By taking $\limsup_{n\to\infty}$ in the above inequality we obtain that $I(s)\geq v_s(t)$ due to condition Eq. (8). Since $t\in\Sigma_2^\omega(s)$ is arbitrary, we know that $I(s)\geq I^*(s)$.

3) This can be proved similarly as in 2).

4) The result follows 2) and 3). □

From the above lemma, especially result 1), condition Eq. (9) is equivalent to condition Eq. (10) for $I=I^*$. So, the following theorem is obvious.

Theorem 2.

1) I^* is the smallest solution of the I-optimality equation (7) satisfying condition Eq. (8).

2) I^* is the unique solution of the I-optimality equation (7) satisfying condition Eq. (9) (or equivalently condition Eq. (10)), if and only if the I-optimality equation (7) has a solution satisfying condition Eq. (9) (or equivalently condition Eq. (10)). □

In the following, we characterize maximal strings and optimal supervisors. For this, we introduce a string function $h: L^0(G) \to \Sigma$ satisfying $h(s) \in \Sigma_2(s)$ for $s \in L^0(G)$. A string function h and a string $s \in L^0(G)$ inductively determine the unique infinite string: if $sr \in L^0(G)$ is determined for some $r \in \Sigma^*$ then the next event is $h(sr)$. We denote by $h^\omega(s)$ the infinite string and by $I(h,s) = v_s(h^\omega(s))$ the discounted total reward for $h^\omega(s)$ after s. The string function h is said to be optimal in $L^0(G)$ if $I(h,s) = I^*(s)$ for all $s \in L^0(G)$. h characterizes maximal strings. In fact, if h is optimal then $h(s)$ is a maximal string from string s. The following theorem can be proved similarly as that in Lemma 4.

Theorem 3.

1) For any given string function h, h is optimal in $L^0(G)$, if and only if $I(h)$ is finite and is a solution of the I-optimality equation (7).

2) If h attains the maximum of the I-optimality equation (7), then h is optimal in $L^0(G)$, if and only if $\lim_{n\to\infty} \beta^n I^*(s \cdot t_n) = 0$ with $t = h^\omega(s)$ for any $s \in L^0(G)$. □

The following corollary characterizes the relation among optimal policies with optimal string functions, and also characterizes the structure of the set of all optimal policies.

Corollary 1. In $L^0(G)$, if h is an optimal string function, then any policy $\boldsymbol{\pi}$ with $h(s) \in \boldsymbol{\pi}(s)$ for all $s \in L^0(G)$ is optimal. □

3.2 Minimum Discounted Total Reward

For the objective $J(\boldsymbol{\pi}, s)$, it is easy to see that the policy $\boldsymbol{\pi}_u$ with $\boldsymbol{\pi}_u(s) \equiv \Sigma_u$ for $s \in L(G)$ is optimal. We define a system $G_u = \{\Sigma_u, Q, \delta, q_0\}$ to be a sub-system of G by restricting the event set to Σ_u. Suppose that G_u is alive. Let $L(G_u, s)$ and $L^\omega(G_u, s)$ be respectively the finite and infinite languages generated by G_u with the initial state $\delta(s, q_0)$. Then

$$J^*(s) = \max_{t \in L^\omega(G_u, s)} v_s(t), \quad s \in L(G).$$

Hence, all the results in the above sub-section are true for J^* except that the event set Σ should be replaced by Σ_u. We omit the details here. We will add the term "G_u" when we cite the results, e.g., "Theorem 1 for G_u".

Corollary 1 for G_u can be replaced by the following corollary, where the policy π_m^* is defined by

$$\pi_m^*(s) = \Sigma_u \cup \{\sigma \in \Sigma_c \,|\, c(s,\sigma) + \beta J^*(s\sigma) = J^*(s)\}, \quad s \in L^0(G).$$

Corollary 2. Any policy π with $\pi_u \le \pi \le \pi_m^*$ is optimal. Hence, the minimal optimal policy is π_u while the maximal optimal policy is π_m^*. □

3.3 Stationary Case

In this subsection, we consider a special case, called the stationary case, where the reward function depends on string only through the current state. That is, there is an extended real-valued function $c(q,\sigma)$ defined in $Q\times\Sigma$ such that $c(s,\sigma) = c(\delta(s,q_0),\sigma)$ for $s \in L(G)$ and $\sigma \in \Sigma(s)$. Here, $c(q,\sigma)$ can be interpreted as the reward of the occurring event σ at the state q.

First, we say that a state $q \in Q$ can reach a state $q' \in Q$ (or q' can be reached from q) if there is string s such that $q' = \delta(s,q)$. As in Definition 1, a state q can reach a state subset P and other similar concepts can be similarly defined. We give the following definition which is similar to Definition 2.

Definition 3. A predicate $P \subset Q$ is said to be closed if any state that can reach P is in P, while P is said to be invariant if any state that can be reached from P is in P.

The closed property is for the system's past while the invariant property is for the system's future.

Let

$$v_q(t) = \sum_{k=0}^{\infty} \beta^k c(q'_k, \sigma_k)$$

be the discounted total reward of occurring string $t = \sigma_0\sigma_1\cdots$ from the state q, where $q'_{k+1} = \delta(q'_k, \sigma_k)$ with $q'_0 = q$ and $k = 0, 1, \cdots$. Then, we have that

$$v_s(t) = v_{\delta(s,q_0)}(t), \qquad s \in L(G),\ t \in L^{\omega}(G,s).$$

This together with Eq. (5) implies that $I^*(s) = \max_{t \in L^{\omega}(G,\ \delta(s,q_0))} v_q(t)$, which depends on string s only through the state $q = \delta(s,q_0)$. Let the optimal stationary value function be

$$I^*(q) = \max_{t \in L^{\omega}(G,q)} v_q(t), \quad q \in Q.$$

Then

$$I^*(s) = I^*(\delta(s,q_0)), \qquad s \in L(G). \tag{11}$$

Based on this equation, all results given in Theorems 1-3 can be simplified.

Let $Q^0 = \{q \in Q \,|\, I^*(q) \in (-\infty,+\infty)\}$, $Q^+ = \{q \in Q \,|\, I^*(q) = +\infty\}$, $Q^- = \{q \in Q \,|\, I^*(q) = -\infty\}$ be the state subsets of finite, positive infinite and negative infinite optimal values, respectively. Furthermore, let

$$\Sigma^*(q) = \begin{cases} \{\sigma \in \Sigma(q) | \, c(q,\sigma) > -\infty\}, & q \notin Q^0 \\ \{\sigma \in \Sigma(q) | \, c(q,\sigma) > -\infty, \, \delta(\sigma,q) \in Q^0\}, & q \in Q^0. \end{cases}$$

Then the following theorem follows Eq. (11) and Theorem 1.

Theorem 4. For the stationary case, $\Sigma(q)$ can be sized down to $\Sigma^*(q)$ for $q \in Q$. After this sizing, Q^+ is closed while Q^- and Q^0 are invariant. Moreover, $\{I^*(q), q \in Q^0\}$ satisfies the following stationary I-optimality equation:

$$I(q) = \max_{\sigma \in \Sigma^*(q)} \{c(q,\sigma) + \beta I(\delta(\sigma,q))\}, \quad q \in Q^0. \tag{12}$$

Similar results as those given in Theorems 2 and 3 and those in Subsection 3.2 are also true. But we omit them here.

The above discussion says that when the system is stationary, the optimal value function and the I-optimality equation are also stationary. In general, when the system has a special structure, we can often get better results. As in [12], the classical results in the supervisory control can also be obtained from our model with a special reward function.

4 An Example for Resource Allocation Systems

Reveliotis and Choi [13] studied the optimality of randomized deadlock avoidance policies for resource allocation systems (RAS) based on one example. In this section, we modify this example and consider it from another viewpoint, in where there are two machines, R_1 and R_2, and two job types, JT_1 and JT_2. Its DES model is given in Figure 1.

The state set is $S = \{(i,j), i,j = 0,1,2\} \cup \{(1^*,j), \, (i,2^*), \, (1^*,2^*) \, | \, i,j = 1,2\}$. In the state variable, the first component i represents that a job i is being processed in machine 1 and 1^* represents that a job 1 has been processed and is waiting in machine 1, while the second component j represents that a job j is being processed in machine 2 and 2^* represents that a job 2 has been processed and is waiting in machine 2. In state $(1^*, 2^*)$, the system is deadlocked and should be resolved artificially, i.e., the system should exchange the two blocked jobs in the two machines.

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

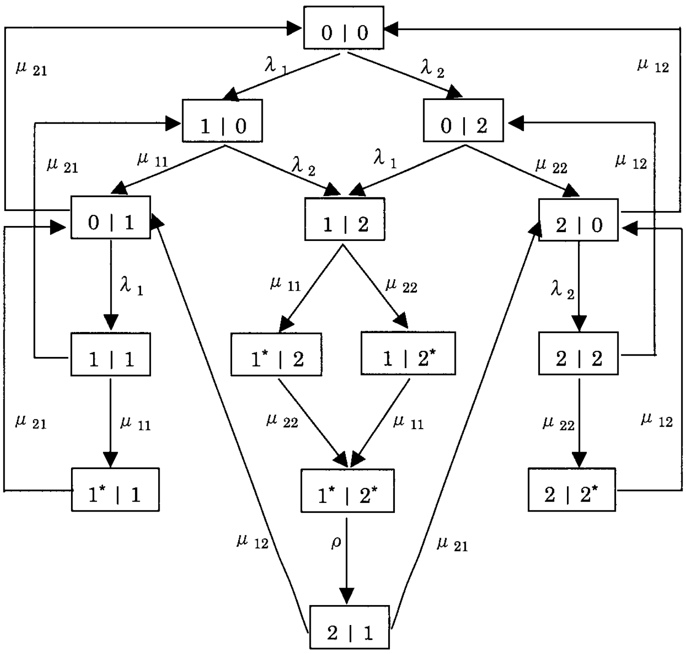

Figure 1: A resource allocation system: the DES model.

Reveliotis and Choi [13] considered randomness in the example where λ_i and μ_{ij} are rates of exponential distributions for respective processing times and introduced control probabilities ω_1 and ω_2 for respective transition $(1,0)\to(1,2)$ and $(0,2)\to(1,2)$, and a control ρ at state $(1,2)$ by swapping the two deadlocked jobs. Reveliotis and Choi discussed the optimal values of ω_1 and ω_2 to maximize the long-run average throughput of the system.

In order to avoid the deadlock, we can either prohibit event λ_1 from occurring at state $(0,2)$ and event λ_2 from occurring at state $(1,0)$ such that the system would not reach state $(1,2)$ or allow event ρ to occur (i.e., to exchange the two blocked jobs) at state $(1^*, 2^*)$. Certainly, each action has an adequate cost. Then the problem of which action is better arises. We will use the model and analysis discussed in the previous section to solve this problem.

Now, prohibiting event λ_1 at state $(0,2)$ is equivalent to the occurrence of event μ_{22} at state $(0,2)$ and prohibiting event λ_2 at state $(1,0)$ is equivalent to the occurrence of event μ_{11} at state $(1,0)$. So, we introduce a cost a_1 for occurring event μ_{22} at state $(0,2)$ and a cost a_2 for occurring event μ_{11} at state $(1,0)$. Moreover, suppose that ρ is the cost for exchanging the two blocked jobs at state $(1^*,2^*)$ (we use the same symbol ρ to represent the cost and the event of exchange). It is assumed that a_1, a_2 and ρ are nonnegative. We define the cost function as

$$c((1,0),\ \mu_{11})=a_2, \quad c((0,2),\ \mu_{22})=a_1, \quad c((1^*,2^*),\ \rho)=\rho$$

and all other $c(q,\sigma)=0$.

Here, we consider the maximal discounted total cost. Then the stationary I-optimality equation (12) is given as follows.

$$
\begin{aligned}
&V(0,0)=\max\{\beta V(1,0),\beta V(0,2)\},\\
&V(1,0)=\max\{\beta V(1,2),a_2+\beta V(0,1)\},\\
&V(0,2)=\max\{\beta V(1,2),a_1+\beta V(2,0)\},\\
&V(0,1)=\max\{\beta V(1,1),\beta V(0,0)\},\\
&V(2,0)=\max\{\beta V(2,2),\beta V(0,0)\},\\
&V(1,1)=\max\{\beta V(1^*,1),\beta V(1,0)\},\\
&V(2,2)=\max\{\beta V(2,2^*),\beta V(0,2)\},\\
&V(1,2)=\max\{\beta V(1^*,2),\beta V(1,2^*)\},\\
&V(2,1)=\max\{\beta V(0,1),\beta V(2,0)\},\\
&V(1^*,1)=\beta V(0,1),\\
&V(2,2^*)=\beta V(2,0),\\
&V(1^*,2)=\beta V(1^*,2^*),\\
&V(1,2^*)=\beta V(1^*,2^*),\\
&V(1^*,2^*)=\rho+\beta V(2,1).
\end{aligned}
$$

We shall simplify the above equations. By substituting the last equation for $V(1^*,2^*)$ into the equations for $V(1^*,2)$ and $V(1,2^*)$ we have that

$$V(1^*,2)=V(1,2^*)=\beta\rho+\beta^2 V(2,1).$$

Again by substituting these two equations together with the equations for $V(1^*,1)$ and $V(2,2^*)$ into the equations for $V(1,1)$, $V(2,2)$, and $V(1,2)$ we can obtain that

$$
\begin{aligned}
&V(1,1)=\max\{\beta^2 V(0,1),\beta V(1,0)\},\\
&V(2,2)=\max\{\beta^2 V(2,0),\beta V(0,2)\},\\
&V(1,2)=\beta^2\rho+\beta^3 V(2,1).
\end{aligned}
$$

Hence, to solve the stationary I-optimality equation, it suffices to solve first the following set of equations:

$$
\begin{cases}
V(0,0)=\max\{\beta V(1,0),\beta V(0,2)\},\\
V(1,0)=\max\{\beta^3\rho+\beta^4 V(2,1),a_2+\beta V(0,1)\},\\
V(0,2)=\max\{\beta^3\rho+\beta^4 V(2,1),a_1+\beta V(2,0)\},\\
V(0,1)=\max\{\beta V(1,1),\beta V(0,0)\},\\
V(2,0)=\max\{\beta V(2,2),\beta V(0,0)\},\\
V(1,1)=\max\{\beta^2 V(0,1),\beta V(1,0)\},\\
V(2,2)=\max\{\beta^2 V(2,0),\beta V(0,2)\},\\
V(2,1)=\max\{\beta V(0,1),\beta V(2,0)\}.
\end{cases}
\tag{13}
$$

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

$$\begin{cases} V_{n+1}(0,0)=\max\{\beta V_n(1,0),\beta V_n(0,2)\}, \\ V_{n+1}(1,0)=\max\{\beta^3\rho+\beta^4 V_n(2,1),a_2+\beta V_n(0,1)\}, \\ V_{n+1}(0,2)=\max\{\beta^3\rho+\beta^4 V_n(2,1),a_1+\beta V_n(2,0)\}, \\ V_{n+1}(0,1)=\max\{\beta V_n(1,1),\beta V_n(0,0)\}, \\ V_{n+1}(2,0)=\max\{\beta V_n(2,2),\beta V_n(0,0)\}, \\ V_{n+1}(1,1)=\max\{\beta^2 V_n(0,1),\beta V_n(1,0)\}, \\ V_{n+1}(2,2)=\max\{\beta^2 V_n(2,0),\beta V_n(0,2)\}, \\ V_{n+1}(2,1)=\max\{\beta V_n(0,1),\beta V_n(2,0)\}. \end{cases} \tag{14}$$

Similarly as in Markov decision processes [11], it can be proven that $\lim_{n\to\infty} V_n(i,j)=V(i,j)$ for each i,j. So, for a given small constant $\epsilon>0$, when $|V_{n+1}(i,j)-V_n(i,j)|<\epsilon$ for all i,j, we stop the above iterative computing procedure and take $V_{n+1}(i,j)$ as an approximate value of $V(i,j)$ for i,j. Moreover, by substituting these values into previous equations we can compute $V(1^*,2)$, $V(1,2^*)$, $V(1^*,2^*)$ and $V(1,2)$.

But fortunately, the above set of equations (13) can be solved directly.

Suppose first that $a_1\ge a_2$. Then it follows the successive approximation of Eq. (14) with $V_0(i,j)=0$ for all i,j that for each n$\ge$ 1 we have that

$$V_n(0,2)\ge V_n(1,0),\quad V_n(2,0)\ge V_n(0,1),\quad V_n(2,2)\ge V_n(1,1).$$

Hence,

$$V(0,2)\ge V(1,0),\quad V(2,0)\ge V(0,1),\quad V(2,2)\ge V(1,1).$$

This together with Eq. (13) implies that

$$V(0,0)=\beta V(0,2),\quad V(2,1)=\beta V(2,0).$$

In the following, we will solve the set of equations (13) in three cases.

Case 1. $\beta^3\rho+\beta^4 V(2,1)\le a_2+\beta V(0,1)$. In this case, the set of equations (13) becomes that

$$\begin{aligned} &V(0,0)=\beta V(0,2),\\ &V(1,0)=a_2+\beta V(0,1),\\ &V(0,2)=a_1+\beta V(2,0),\\ &V(1,1)=\max\{\beta^2 V(0,1),\beta a_2+\beta^2 V(0,1)\}=\beta a_2+\beta^2 V(0,1),\\ &V(2,2)=\max\{\beta^2 V(2,0),\beta a_1+\beta^2 V(2,0)\}=\beta a_1+\beta^2 V(2,0),\\ &V(0,1)=\max\{\beta^2 a_2+\beta^3 V(0,1),\beta^2 a_1+\beta^3 V(2,0)\}=\beta^2 a_1+\beta^3 V(2,0),\\ &V(2,0)=\max\{\beta^2 a_1+\beta^3 V(2,0),\beta^2 a_1+\beta^3 V(2,0)\}=\beta^2 a_1+\beta^3 V(2,0),\\ &V(2,1)=\beta V(2,0)=\beta^3 a_1+\beta^4 V(2,0). \end{aligned}$$

By solving it we obtain that

$$V(0,0)=V(2,2)=\tfrac{\beta}{1-\beta^3}a_1,\quad V(1,0)=a_2+\tfrac{\beta^3}{1-\beta^3}a_1,$$
$$V(0,2)=\tfrac{1}{1-\beta^3}a_1,\quad V(0,1)=V(2,0)=\tfrac{\beta^2}{1-\beta^3}a_1,$$
$$V(1,1)=\beta a_2+\tfrac{\beta^4}{1-\beta^3}a_1,\quad V(2,1)=\tfrac{\beta^3}{1-\beta^3}a_1.$$

Then

$$V(1,2)=\beta^2\rho+\tfrac{\beta^6}{1-\beta^3}a_1,\quad V(1^*,2)=V(1,2^*)=\beta\rho+\tfrac{\beta^5}{1-\beta^3}a_1,$$
$$V(1^*,1)=\tfrac{\beta^3}{1-\beta^3}a_1,\quad V(2,2^*)=\tfrac{\beta^3}{1-\beta^3}a_1,$$
$$V(1^*,2^*)=\rho+\tfrac{\beta^4}{1-\beta^3}a_1.$$

Moreover, the condition "$\beta^3\rho+\beta^4V(2,1)\le a_2+\beta V(0,1)$" is equivalent to "$\rho\le\frac{1}{\beta^3}a_2+\frac{1-\beta^4}{1-\beta^3}a_1$".

Case 2. $a_2+\beta V(0,1)\le\beta^3\rho+\beta^4V(2,1)\le a_1+\beta V(2,0)$. In this case, the set of equations (13) becomes that

$$\begin{aligned}
&V(0,0)=\beta V(0,2),\\
&V(2,1)=\beta V(2,0),\\
&V(1,0)=\beta^3\rho+\beta^4V(2,1)=\beta^3\rho+\beta^5V(2,0),\\
&V(0,2)=a_1+\beta V(2,0),\\
&V(0,1)=\max\{\beta V(1,1),\beta^2a_1+\beta^3V(2,0)\},\\
&V(2,0)=\max\{\beta V(2,2),\beta^2a_1+\beta^3V(2,0)\},\\
&V(1,1)=\max\{\beta^2V(0,1),\beta^4\rho+\beta^6V(2,0)\},\\
&V(2,2)=\max\{\beta^2V(2,0),\beta a_1+\beta^2V(2,0)\}=\beta a_1+\beta^2V(2,0).
\end{aligned}$$

By solving it we obtain that

$$V(0,0)=V(2,2)=\tfrac{\beta}{1-\beta^3}a_1,\quad V(1,0)=\beta^3\rho+\tfrac{\beta^7}{1-\beta^3}a_1,$$
$$V(0,2)=\tfrac{1}{1-\beta^3}a_1,\quad V(1,1)=\beta^4\rho+\tfrac{\beta^8}{1-\beta^3}a_1,$$
$$V(0,1)=V(2,0)=\tfrac{\beta^2}{1-\beta^3}a_1,\quad V(2,1)=\tfrac{\beta^3}{1-\beta^3}a_1.$$

Then

$$V(1,2)=\beta^2\rho+\tfrac{\beta^6}{1-\beta^3}a_1,\quad V(1^*,2)=V(1,2^*)=\beta\rho+\tfrac{\beta^5}{1-\beta^3}a_1,$$
$$V(1^*,1)=V(2,2^*)=\tfrac{\beta^3}{1-\beta^3}a_1,\quad V(1^*,2^*)=\rho+\tfrac{\beta^4}{1-\beta^3}a_1.$$

Moreover, the condition "$a_2+\beta V(0,1)\le\beta^3\rho+\beta^4V(2,1)\le a_1+\beta V(2,0)$" is equivalent to "$\frac{1}{\beta^3}a_2+\frac{1-\beta^4}{1-\beta^3}a_1\le\rho\le\frac{1-\beta^7}{\beta^3(1-\beta^3)}a_1$".

Case 3. $\beta^3\rho+\beta^4V(2,1)\geq a_1+\beta V(2,0)$. In this case, the set of equations (13) becomes that

$$
\begin{aligned}
&V(0,0)=\beta V(0,2),\\
&V(2,1)=\beta V(2,0),\\
&V(1,0)=V(0,2)=\beta^3\rho+\beta^5V(2,0),\\
&V(0,1)=\max\{\beta V(1,1),\beta^5\rho+\beta^7V(2,0)\},\\
&V(2,0)=\max\{\beta V(2,2),\beta^5\rho+\beta^7V(2,0)\},\\
&V(1,1)=\max\{\beta^2V(0,1),\beta^4\rho+\beta^6V(2,0)\},\\
&V(2,2)=\max\{\beta^2V(2,0),\beta^4\rho+\beta^6V(2,0)\}.
\end{aligned}
$$

By solving it we obtain that

$$
\begin{aligned}
&V(0,0)=V(2,2)=V(1,1)=\tfrac{\beta^4}{1-\beta^7}\rho, \quad V(1,0)=V(0,2)=\tfrac{\beta^3}{1-\beta^7}\rho,\\
&V(0,1)=V(2,0)=\tfrac{\beta^5}{1-\beta^7}\rho, \quad V(2,1)=\tfrac{\beta^6}{1-\beta^7}\rho.
\end{aligned}
$$

Then

$$
\begin{aligned}
&V(1,2)=\tfrac{\beta^2}{1-\beta^7}\rho, \quad V(1^*,2)=V(1,2^*)=\tfrac{\beta}{1-\beta^7}\rho,\\
&V(1^*,1)=V(2,2^*)=\tfrac{\beta^6}{1-\beta^7}\rho, \quad V(1^*,2^*)=\tfrac{1}{1-\beta^7}\rho.
\end{aligned}
$$

Moreover, the condition "$\beta^3\rho+\beta^4V(2,1)\geq a_1+\beta V(2,0)$" is equivalent to "$\rho\geq\frac{1-\beta^7}{\beta^3(1-\beta^3)}a_1$".

From the above three cases, we have the following proposition, which is obvious since each cost increases the discounted total cost.

Proposition 1. For each state (i,j), $V(i,j)$ is increasing respectively in the costs a_1, a_2 and ρ. □

Let

$$
\rho_1^*=\frac{1-\beta^7}{\beta^3(1-\beta^3)}a_1, \qquad \rho_2^*=\frac{1}{\beta^3}a_2+\frac{1-\beta^4}{1-\beta^3}a_1
$$

be two constants. The following proposition solves the comparison of prohibiting event λ_i with resolving the deadlock.

Proposition 2. When $a_1\geq a_2$, it is better to prohibit event λ_1 at state $(0,2)$ than to resolve the deadlock if and only if $\rho\geq\rho_1^*$ and is better to prohibit event λ_2 at state $(1,0)$ than to resolve the deadlock if and only if $\rho\geq\rho_2^*$.

Proof. From the optimality equation, it is better to prohibit event λ_1 at state $(0,2)$ than to resolve the deadlock if and only if the discounted total cost of a policy which prohibits λ_1 at $(0,2)$ is smaller than that of a policy which let λ_1 occur at $(0,2)$, that is,

$$
a_1+\beta V(2,0)\leq\beta V(1,2).
$$

Due to the previous three cases, we know that the above condition is true if and only if $\rho \geq \rho_1^*$, i.e., case 3 happens.

Similarly, it is better to prohibit event λ_2 at state $(1,0)$ than to resolve the deadlock if and only if

$$a_2 + \beta V(0,1) \leq \beta V(1,2),$$

which is equivalent to $\rho \geq \rho_2^*$, i.e., cases 2 and 3 happen. □

Similarly we can obtain the following proposition for $a_1 \leq a_2$, where the two constants are respectively

$$\rho_1^0 = \frac{1}{\beta^3} a_1 + \frac{1-\beta^4}{1-\beta^3} a_2, \qquad \rho_2^0 = \frac{1-\beta^7}{\beta^3(1-\beta^3)} a_2.$$

Proposition 3. When $a_1 \leq a_2$, it is better to prohibit event λ_1 at state $(0,2)$ than to resolve the deadlock if and only if $\rho \geq \rho_1^0$ and is better to prohibit event λ_2 at state $(1,0)$ than to resolve the deadlock if and only if $\rho \geq \rho_2^0$.

The above two propositions say that there are threshold values for comparison of prohibiting event λ_i with resolving the deadlock.

Remark. If there is a cost of the occurrence of any event, then the stationary I-optimality equation will be more complex and difficult to solve. But we can still prove the existence of the threshold values.

5 Conclusions

In this paper, we presented an optimal control problem for discrete event systems and solved it through the optimality equations by applying ideas for Markov decision processes under some necessary conditions. When the reward function is stationary, the optimality equations and the optimal policies are also stationary. We also used the model for optimal control to solve a resource allocation system.

Further research may include another optimal control problem for DESs where rewards occur when control inputs at strings are chosen. This will be more difficult to analyze than the model studied in this paper.

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] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. of the IEEE*, Vol. 77, No. 1, pp. 81-98, 1989.

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

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

[7] R. Kumar and V. K. Garg, "Optimal supervisory control of discrete event dynamic systems," *SIAM J. Control and Optimization*, Vol. 33, No. 2, pp. 419-439, 1995.

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

[9] Q. Hu and Y. Liu, "Markov decision processes methods in static stability of discrete event systems," *Acady Acta Mathemeticae Applicate Sinica* (in Chinese series), Vol. 24, No. 3, pp. 377-383, 2001.

[10] 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, No. 2, pp. 255-266, 1999.

[11] Q. Hu and W. Yue, "Analysis for some properties of discrete time Markov decision processes," *J. of Optimization*, Vol. 52, No. 4-5, pp. 495-505, 2003.

[12] Q. Hu and W. Yue, "Optimal control of discrete event systems with discounted criterion," *Memoirs of Konan University, Science Series*, Vol. 50, No. 2, pp. 77-98, 2003.

[13] S. A. Reveliotis and J. Y. Choi, "On the optimality of randomized deadlock avoidance policies," *Discrete Event Dynamic Systems: Theorey and Application*, Vol. 13, pp. 303-320, 2003.