

Article

Optimal Opacity-Enforcing Supervisory Control of Discrete Event Systems on Choosing Cost

Yinyin Dai, Fei Wang  and Jiliang Luo 

College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China;
crystle@hqu.edu.cn (Y.D.); jlluo@hqu.edu.cn (J.L.)

* Correspondence: feiw545@163.com

Abstract: To ensure opacity, it is optimal to retain as many as possible occurring event sequences. Contrary to this problem, the other optimal goal is to preserve the minimal occurring event sequences. Based on the choosing cost, an optimal opacity-enforcing problem with minimal discount choosing cost is presented under two constraints in this paper. The first constraint is the opacity of the controlled system. The second is the retention of the secret to the maximum. To solve the model, two scenarios on opacity are considered. For the two scenarios, some algorithms are presented to achieve the optimal solution for the model by using the method of dynamic programming. Then, the solutions produced by the algorithms are proved to be correct by theoretical proof. Finally, some illustrations and an application example on location privacy protection for the algorithms are given.

Keywords: opacity; optimal supervisors; dynamic programming; discrete event systems

1. Introduction

In modern society, when important information about enterprises is released online, it generally needs to be as complete as possible. In order to keep the security and opacity of important data transmission, it is necessary to use some unimportant data to confuse and make it difficult to distinguish from an adversary. The more information is used to confuse important data, the better it is. Therefore, some papers (e.g., [1–4]) discussed the issue of maximum opaque sublanguages. However, in reality, due to the cost involved in using data transmission, less unimportant data and lower transmission costs are preferred. Therefore, this paper proposes an optimization mathematical model to find the most cost-effective information to confuse the important data to be released and proposes an algorithm to obtain the optimal control strategy.

In 2004, opacity was first introduced to analyze cryptographic protocols in [5]. In 2005, the research modeled as Petri nets in [6] brought opacity to the field of discrete event systems (DES). Afterward, the research on opacity in DES boomed up. In DES models, the definition of opacity was divided into two cases: language-based opacity [1,7,8] (e.g., strong opacity [7], weak opacity [7], non-opacity [7]) and state-based opacity [6,9–13] (e.g., current-state opacity [6,9], initial-state opacity [6], initial-and-final-state opacity [13], K -step opacity [9–11], infinite-step opacity [12,14]). In ref. [15], the works of [13] were extended, and it was shown that the various notions of opacity can be transformable to each other. Then, in Ref. [16], the existing notions of opacity were unified and a general framework was provided. With the definition of opacity, the verification approach was investigated in the previous works. Once a system was not opaque, supervisory control theory [1,4,17] or the enforcement approach [18–22] were presented to ensure opacity in the system. In general, supervisory control theory restricts the behavior of the system to ensure opacity, whereas the enforcement approach does not restrict but modifies the output of the system to ensure opacity. For example, in Ref. [1], fix-point theory was used to check the opacity of the closed system at every iteration to achieve the maximal



Citation: Dai, Y.; Wang, F.; Luo, J. Optimal Opacity-Enforcing Supervisory Control of Discrete Event Systems on Choosing Cost. *Appl. Sci.* **2024**, *14*, 2532. <https://doi.org/10.3390/app14062532>

Received: 6 February 2024

Revised: 13 March 2024

Accepted: 15 March 2024

Published: 17 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

permissive supervisor on condensed state estimates. In Ref. [4], the maximal permissive supervisor was obtained using the refining of the plant and observer instead of condensed state estimates. In Ref. [17], strong infinite- and k -step opacity was transformed into a language and an algorithm was presented to enforce infinite- and k -step opacity by a supervisor. In Ref. [18], the synthesis of insertion function was extended from current-state opacity [23] to infinite-step opacity and K -step opacity. In Ref. [19], fictitious events were inserted at the output of the systems to enforce the opacity of the system. And some works [20,21] extended the method of [19], where the authors of Ref. [20] discussed the problem of opacity enforcement under energy constraints and the authors of Ref. [21] studied supervisory control under local mean payoff constraints. In Ref. [22], current-state opacity was verified and enforced based on an algebraic state space approach for partially observed finite automaton.

To ensure opacity, in Ref. [24], some algorithms were proposed to design a controller to control information released to the public. And then, in Ref. [25], the work of [24] was extended, and then an algorithm was presented for finding minimal information release policies for non-opacity. In Ref. [26], a reward was assigned for revealing the information of each transition and the maximum guaranteed payoff policy was found for the weighted DES. In Ref. [27], a dynamic information release mechanism was proposed to verify current-state opacity where information is partially released and state-dependent.

To ensure opacity, the secret is preserved by enabling/disabling some events to restrict the behavior of the system. If cost functions are defined in DES, two types of optimal supervisory control problems are developed: one is for event cost, and the other is for control cost. For example, in Ref. [28], the cost of events was defined to design a supervisor to minimize the total cost to reach the desired states. Then, in Ref. [29], the framework was extended to the partial observation of the system. Afterward, in Ref. [30], the mean payoff supervisory control problem was investigated for a system where each event has an integer associated with it. In [31,32], the costs of choosing control input and occurring event were defined and two optimal problems to minimize the maximal discounted total cost among all possible strings generated in the systems were solved using Markov decision processes.

In contrast to the supremal opaque sublanguage of the plant in [1–4,8], we want to find a ‘smallest’ closed controllable sublanguage of the plant, concerning which the secret is not only opaque but also ‘largest’. Since the class of opaque languages is not closed under intersection [8], ‘smallest’ does not refer to set inclusion, but to minimal discount total choosing cost. On the other hand, ‘largest’ refers to set inclusion, where it means the union of all the elements of the class of confused secret. To describe the optimal problem, a non-linear optimal supervisory control model is proposed by introducing the concept of choosing cost.

The paper is organized as follows. Section 2 establishes the background on supervisory control theory, opacity, and choosing cost of DES. In Section 3, we present an optimal supervisory control problem that is modeled by non-linear programming with two constraint conditions. In Section 4, we propose two scenarios to show the computational process of the optimal problem. The first scenario is divided into three cases in Section 5, where we suppose that the plant can ensure the secret is opaque, and some algorithms and theorems are put forward to the solution of the optimal problem. In Section 6, a generalized algorithm and theorem are proposed under the second scenario, where the plant cannot ensure the secret is opaque. Finally, the main contribution of the work is discussed in Section 7.

2. Preliminary

2.1. Supervisory Control Theory

We consider a DES modeled by a determined finite transition system, $G = (Q, \Sigma, \delta, q_0)$, where Q is a finite set of states, Σ is a finite set of events labeled in the transition, partial function $\delta : \Sigma \times Q \rightarrow Q$ is the transition function and $q_0 \in Q$ is the initial state. A run of G is a finite non-empty sequence, $q_0\sigma_1q_1\sigma_2 \cdots q_{n-1}\sigma_n$, where $q_i \in Q, \sigma_{i+1} \in \Sigma$ and $q_{i+1} = \delta(q_i, \sigma_i)$ for $i = 1 \cdots n - 1$. The trace $tr(\rho)$ of run ρ is $\sigma_1\sigma_2 \cdots \sigma_n$. The languages of G

are the set of all traces of runs of G , and they are denoted $L(G) = \{s \in \Sigma^* \mid \delta(s, q_0)!\}$. Event set Σ is assumed to be partitioned into controllable event set Σ_c and uncontrollable event set Σ_u , where $\Sigma = \Sigma_c \cup \Sigma_u$. Each subset of events is a control pattern, and the set of all control patterns is denoted by $\Gamma = \{\gamma \mid \Sigma_u \subseteq \gamma \subseteq \Sigma\}$. A supervisor for G is any map $f: L(G) \rightarrow \Gamma$. For system G , we denote $\Gamma^{L(G)}$ as the set of supervisors. The closed behavior of f/G , i.e., G under the supervision of f , is defined to be language $L(f/G) \subseteq L(G)$ described as follows:

- $\epsilon \in L(f/G)$;
- $s\sigma \in L(f/G) \Leftrightarrow s \in L(f/G), s\sigma \in L(G), \sigma \in f(s)$.

For language $K \subseteq L(G)$, notation $\bar{K}$ is said to be the prefix(-closure) of K . K is said to be (prefix-)closed if $\bar{K} = K$.

Definition 1 ([33]). We consider non-empty language K . Then, K is controllable if $\bar{K}\Sigma_u \cap L(G) \subseteq \bar{K}$.

A necessary and sufficient condition for the existence of a supervisor is given as follows:

Theorem 1 ([33]). We consider non-empty language $K \subseteq L(G)$. Then, there exists supervisor f such that $L(f/G) = K$ if and only if K is a controllable and closed language.

2.2. Supervisory Control for Opacity

Assuming that the adversary can be aware of any supervisor's control policy, a subset Σ_a of the events can be seen by the adversary through an observable function, $\theta: \Sigma^* \rightarrow \Sigma_a^*$. An adversary's observation of the system is denoted by $obs(G) = \{Q_a, \Sigma_a, \delta_a, q_{0a}\}$, which is an observer of system G . If an observer of the system is unable to determine some secret information, we offer the following equivalent definitions of opacity:

Definition 2 ([1]). We consider system G and non-empty language K . For any $s \in K \cap L(G)$, if there exists $s' \in L(G) - K$ such that $\theta(s) = \theta(s')$, we say K is (strongly) opaque with respect to $L(G)$ and Σ_a .

Definition 3 ([7]). We consider system G and non-empty language K . K is strongly opaque with respect to $L(G)$ and θ if $\theta(K) \subseteq \theta(L(G) - K)$.

If the condition of Definition 2 cannot be met, we say K is not (strongly) opaque with respect to $L(G)$ and Σ_a . For the definition of non-opacity, it is different from that of [7].

In [8], since opaque language has a desirable property that it is closed under union, there exist supremal controllable closed and opaque sublanguages of the system [1,4]. In [1,4,8], different works about supremal controllable closed and opaque sublanguage are computed.

2.3. The Definition of Choosing Cost

In [31,32], two optimal control models of DES are presented based on the costs of choosing a control input and an event occurring, respectively. In this paper, the choosing cost is obtained from the definition of the cost of choosing the control input after some string of [31,32]. For a given DES G , we let $c(s, \gamma)$ be the cost of choosing control input γ at string s , where $s \in L(G)$, $\gamma \in \Gamma$ and $c(s, \gamma)$ is nonnegative. For any supervisor $f: L(G) \rightarrow \Gamma$, we call $c(s, f) = c(s, f(s))$ the cost of choosing $f(s)$ after s under the supervision of f . All in all, we simply call $c(s, \gamma)$ (or $c(s, f)$) the choosing cost.

3. Optimal Supervisory Control Model on Opacity

In the section, an optimal supervisory control model is constructed to minimize the choosing cost of the controlled system which is opaque. Given system G and secret $K \subseteq L(G)$, as shown in [1], secret K is a regular language. For secret K , we suppose there exists a set of secret states $Q_s \subseteq Q$ which can recognize secret K , i.e., $s \in K$ iff $\delta(q_0, s) \in Q_s$.

To show the cost of information released, we introduce the cost of choosing control input in [31,32] and define the total choosing cost as follows:

Definition 4. Given closed-loop behavior $L(f/G)$, the discount total choosing cost of $L(f/G)$ is defined as $V(L(f/G), f) = \sum_{s \in L(f/G)} \beta^{|s|} c(s, f)$, where f is a supervisor controlling system G and $\beta > 0$ is a discount factor. For convenience, $V(L(f/G), f)$ is simplified as $V(L(f/G))$.

For system G , if there exist two supervisors f_1 and f_2 such that $L(f_1/G) \subseteq L(f_2/G)$, it is obvious that $V(L(f_1/G)) \leq V(L(f_2/G))$ by Definition 4.

To obtain the discount total choosing cost, we definite discount cost by the sum of choosing cost after s under f .

Definition 5. We consider $s, t_n = \sigma_1 \sigma_2 \cdots \sigma_n$ and $t_k = \sigma_1 \sigma_2 \cdots \sigma_k (0 \leq k \leq n, t_0 = \epsilon)$. Then, $V_s(t_n) = \sum_{k=0}^n \beta^{|s|+k} c(st_k, f_{st_n})$ is said to be the discount cost of choosing t_n after s under f , where $f_{st_n}(st_k) = f(st_k) \cap \Sigma_{st_n}(st_k)$ and $\Sigma_{st_n}(st_k) = \{\sigma | st_k \sigma \in \overline{st_n}\}$. Particularly, if $k = 0$, i.e., $t_0 = \epsilon$, then $V_s(\epsilon) = \beta^{|s|} c(s, \phi) = 0$; if $s = \epsilon$, then $st_n = t_n$ and $V_\epsilon(t_n) = \sum_{k=0}^n \beta^k c(t_k, f_{t_n})$, where $t_0 = \epsilon$ means that choosing $f(s)$ at s has nothing to do with Σ .

In Definition 5, if some string can be divided into two pieces, a formula is formulated to simplify the computation of discount cost in the following conclusions:

Proposition 1. We consider $t = st'$. Then, we have $V_\epsilon(t) = V_\epsilon(s) + V_s(t')$.

Proof. For $t = st'$, we proved the following formula, where $s = \sigma_1 \sigma_2 \cdots \sigma_{|s|}$ and $t' = \sigma_{|s|+1} \sigma_{|s|+2} \cdots \sigma_{|s|+n}$:

$$\begin{aligned} V_\epsilon(t) &= \sum_{k=0}^{|s|+n} \beta^k c(t_k, f_t) \\ &= c(\epsilon, f_s) + \beta c(\sigma_1, f_s) + \cdots + \beta^{|s|} c(s, f_s) + \sum_{k=1}^n \beta^{|s|+k} c(st'_k, f_{st'}) \\ &= \sum_{k=0}^{|s|} \beta^k c(\sigma_1 \cdots \sigma_k, f_s) + \sum_{k=0}^n \beta^{|s|+k} c(st'_k, f_{st'}) \\ &= V_\epsilon(s) + V_s(t') \end{aligned}$$

□

To generalize the above Proposition 1, we have the following conclusion as Proposition 2:

Proposition 2. We consider $s = s_1 s_2 \cdots s_k \in L(G)$. Then, we have $V_\epsilon(s) = V_\epsilon(s_1) + V_{s_1}(s_2) + \cdots + V_{s_1 s_2 \cdots s_{k-1}}(s_k)$.

Proof. The proof can proceed by induction.

Base case: If $k = 1$, then it holds that $V_\epsilon(s) = V_\epsilon(s_1)$.

Inductive hypothesis: Suppose that we have $V_\epsilon(s_1 \cdots s_{j-1} s_j) = V_\epsilon(s_1) + V_{s_1}(s_2) + \cdots + V_{s_1 s_2 \cdots s_{j-1}}(s_j)$ if $k = j$.

Inductive step: For $k = j + 1$, we prove that $V_\epsilon(s_1 \cdots s_{j-1} s_j s_{j+1}) = V_\epsilon(s_1) + V_{s_1}(s_2) + \cdots + V_{s_1 s_2 \cdots s_{j-1}}(s_j) + V_{s_1 s_2 \cdots s_j}(s_{j+1})$.

By Definition 5, we have the following, which completes the inductive step:

$$\begin{aligned}
 V_\epsilon(s) &= V_\epsilon(s_1 \cdots s_{j-1} s_j s_{j+1}) \\
 &= V_\epsilon(s_1 \cdots s_{j-1} s_j) + V_{s_1 \cdots s_{j-1} s_j}(s_{j+1}) \\
 &\quad (\because \text{Proposition 1}) \\
 &= V_\epsilon(s_1) + V_{s_1}(s_2) + \cdots + V_{s_1 s_2 \cdots s_{j-1}}(s_j) + V_{s_1 \cdots s_{j-1} s_j}(s_{j+1}) \\
 &\quad (\because \text{Hypothesis } k = j)
 \end{aligned}$$

□

To obtain the discount total choosing cost, we formulate Algorithm 1 to obtain $V(L(f/G))$ by the computation of $V_s(s')$.

Algorithm 1 Computing $V(L(f/G))$, the discount total choosing cost of $L(f/G)$

Input: $L(f/G)$;

Output: $V(L(f/G))$.

```

1: Let root =  $q_0$ .
2: Let  $Q_v = Q$ .
3: for  $q \in Q - \{q_0\}$  do
4:   Compute the in-degree of state  $q$ .
5:   if in-degree of state  $q$  is more than 1 then
6:     Copy state  $q$  and its successor states and traces based on its in-degree.
7:     Insert the states  $q$  and its successor into  $Q$  and  $Q_v$ .
8:   end if
9: end for
10: for  $q \in Q - \{q_0\}$  do
11:   Compute the out-degree of state  $q$ .
12:   if out-degree = 1 then
13:     Concatenate the transitions from  $q$ 's parent to  $q$ 's child via  $q$ , i.e., achieve a
        transition ( $q$ 's parent  $\xrightarrow{ab} q$ 's child) from the transitions ( $q$ 's parent  $\xrightarrow{a} q$  and
         $q \xrightarrow{b} q$ 's child).
14:      $Q = Q - \{q\}$ 
15:   end if
16: end for
17: At some node  $q$ , if there exists a transition  $s$  from root to  $q$  and a transition  $s'$  from  $q$  to
    its child (i.e., root  $\xrightarrow{s} q \xrightarrow{s'} q$ 's child), we have  $V_s(s')$ .
18: By breadth-first search, we traverse all the nodes and have  $V(L(f/G)) = \sum_{q \in Q_v} V_s(s')$ .
19: return  $V(L(f/G))$ 

```

As shown in Algorithm 1, closed-system f/G can be transformed into a tree automaton, where the closed-system is language-equivalent with the tree automaton. To obtain the longest common prefix of some strings, we need to compute the out-degree of each node. In the tree automaton, if the out-degree of some node is greater than one, the string from the root to the node must be the longest common prefix of some strings from the root to the leaves via the node.

To show the computational process of Algorithm 1, an example is given to obtain the discount total choosing cost.

Example 1. Suppose that $L(f/G) = \{\overline{s_1}, \overline{s_2}\}$ and t is the longest common prefix of s_1 and s_2 . To simplify the process of computation in Definition 4, we denote $t = \sigma_1 \cdots \sigma_l$, $t_1 = \sigma_{i_1} \cdots \sigma_{i_m}$ and $t_2 = \sigma_{j_1} \cdots \sigma_{j_n}$ such that $s_1 = tt_1$ and $s_2 = tt_2$. By Definition 4, we have the following equation:

$$\begin{aligned}
V(\{\bar{s}_1, \bar{s}_2\}) &= \sum_{k=0}^l \beta^k c(\sigma_1 \cdots \sigma_k, f_{\sigma_1 \cdots \sigma_l}) + \beta^l \sum_{k=0}^m \beta^k c(\sigma_1 \cdots \sigma_l \sigma_{i_1} \cdots \sigma_{i_k}, f_{\sigma_1 \cdots \sigma_l \sigma_{i_1} \cdots \sigma_{i_m}}) \\
&\quad + \beta^l \sum_{k=0}^n \beta^k c(\sigma_1 \cdots \sigma_l \sigma_{j_1} \cdots \sigma_{j_k}, f_{\sigma_1 \cdots \sigma_l \sigma_{j_1} \cdots \sigma_{j_n}}) \\
&= V_\epsilon(t) + V_t(t_1) + V_t(t_2) \\
&= V_\epsilon(tt_1) + V_t(t_2) \\
&= V_\epsilon(s_1) + V_t(t_2) \\
&= V_\epsilon(s_2) + V_t(t_1)
\end{aligned}$$

According to Algorithm 1, we transform $L(f/G)$ in Figure 1 into a tree automaton shown in Figure 2.

By Figure 2 and Algorithm 1, we have the following formula:

$$\begin{aligned}
V(\{\bar{s}_1, \bar{s}_2\}) &= V_\epsilon(t) + V_t(t_1) + V_t(t_2) \\
&= V_\epsilon(tt_1) + V_t(t_2) \\
&= V_\epsilon(s_1) + V_t(t_2) \\
&= V_\epsilon(s_2) + V_t(t_1)
\end{aligned}$$

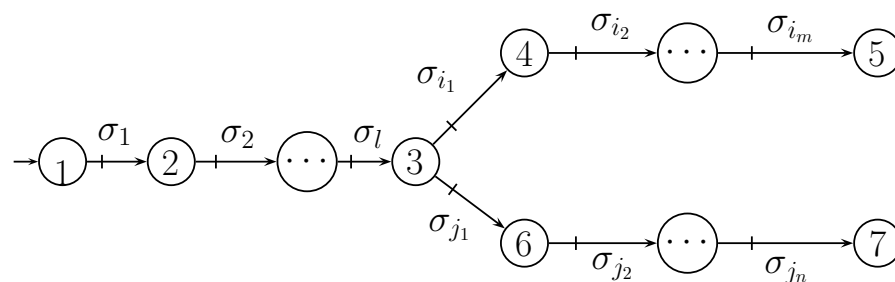


Figure 1. System f/G .

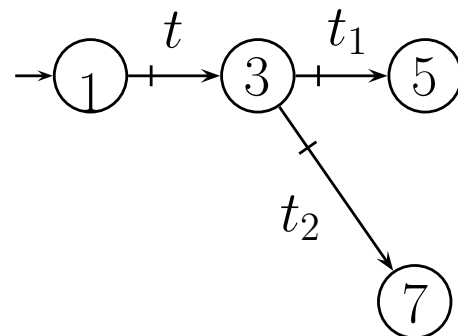


Figure 2. Tree automaton obtained from Figure 1, where $t = \sigma_1 \cdots \sigma_l$, $t_1 = \sigma_{i_1} \cdots \sigma_{i_m}$ and $t_2 = \sigma_{j_1} \cdots \sigma_{j_n}$.

If there exists language K such that $\bar{K} \subseteq L(f/G)$, the discount total choosing cost of $\bar{K}$ can be denoted by $V(\bar{K})$ and obtained as shown in Algorithm 1. If we denote $V(L(f/G) - \bar{K})$ by $V(L(f/g)) - V(\bar{K})$, we can also obtain the discount cost using Definition 5. $V(L(f/G) - \bar{K})$ means to compute the choosing cost outside of $\bar{K}$ in $L(f/G)$. Next, we continue the above Example 1.

Example 2. In Figure 1, we assume there exists state subset $Q_K = \{4, 6\}$ such that $K = t(\sigma_{i_1} + \sigma_{j_1})$. By Algorithm 1, we obtain a tree automaton based on $\bar{K}$ shown in Figure 3. So, it holds that $V(\bar{K}) = V_\epsilon(t) + V_t(\sigma_{i_1}) + V_t(\sigma_{j_1})$. According to the formulas of $V(L(f/G))$ and $V(\bar{K})$, we have the following equation:

$$\begin{aligned}
V(L(f/G) - \bar{K}) &= [V_e(t) + V_t(t_1) + V_t(t_2)] - [V_e(t) + V_t(\sigma_{i1}) + V_t(\sigma_{j1})] \\
&= V_t(t_1) - V_t(\sigma_{i1}) + V_t(t_2) - V_t(\sigma_{j1}) \\
&= V_{t\sigma_{i1}}(\sigma_{i2} \cdots \sigma_{im}) + V_{t\sigma_{j1}}(\sigma_{j2} \cdots \sigma_{jn})
\end{aligned}$$

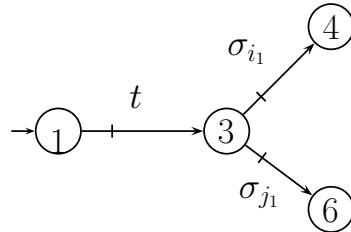


Figure 3. Tree automaton based on $\bar{K}$.

For the system and the secret, we propose an optimal problem to synthesize a supervisor such that the discount total choosing cost of the controlled system is minimal.

Optimal opacity-enforcing problem. Given system G and secret $K \subseteq L(G)$, we find supervisor f such that $L(f/G)$ satisfies the following conditions:

1. K is opaque with respect to $L(f/G)$ and Σ_a ;
2. All the secret information that has not been leaked in G remains in the closed-loop system f/G ;
3. For closed-loop behavior $L(f/G)$, discount total choosing cost $V(L(f/G))$ is minimal.

In Condition 1 of the above problem, if K is opaque with respect to $L(f/G)$ and Σ_a , we have $\theta(K \cap L(f/G)) \subseteq \theta(L(f/G) - K)$ by Definition 3.

In Condition 2, if the supremal controllable and closed sublanguage of $L(G)$ are denoted by $L(g/G)$ [4], the largest secret K permitted by supervisor f is $L(g/G) \cap K$. So, all the secret information that is not leaked in G is $L(g/G) \cap K$. Therefore, the second condition implies that $K \cap L(f/G) = K \cap L(g/G)$ holds.

In Condition 3, we hope that objective function $V(L(f/G))$ is minimal.

Based on the above optimal problem and its analysis, a non-linear optimal model is formulated as follows:

$$\begin{aligned}
&\min V(L(f/G)) \\
&s.t. \theta(K \cap L(f/G)) \subseteq \theta(L(f/G) - K) \\
&\quad K \cap L(f/G) = K \cap L(g/G) \\
&\quad f \in \Gamma^{L(G)}
\end{aligned} \tag{1}$$

For the above optimal Model (1), the objective function means that the supervised system's discount total choosing cost is minimal. And the first constraint condition means that K is opaque with respect to the controlled system. The latter implies that the biggest number of secrets not to be disclosed is in the controlled system.

4. Solution of Optimal Model on Choosing Cost

In this section, we first make the following assumption:

Assumption 1. If $s \in L(G)$ and $\gamma = \Sigma_u$, then we have $c(s, \gamma) = 0$.

For Assumption 1, we suppose that any uncontrollable event's choosing cost is 0.

Under the assumption, we consider the following two scenarios to solve optimal Model (1) in the following sections:

Scenario 1 Secret K is opaque with respect to $L(G)$ and Σ_a .

Scenario 2 Secret K is not opaque with respect to $L(G)$ and Σ_a .

Figure 4 illustrates the classification process of optimal Model (1) in the upcoming sections, i.e., Sections 5 and 6.

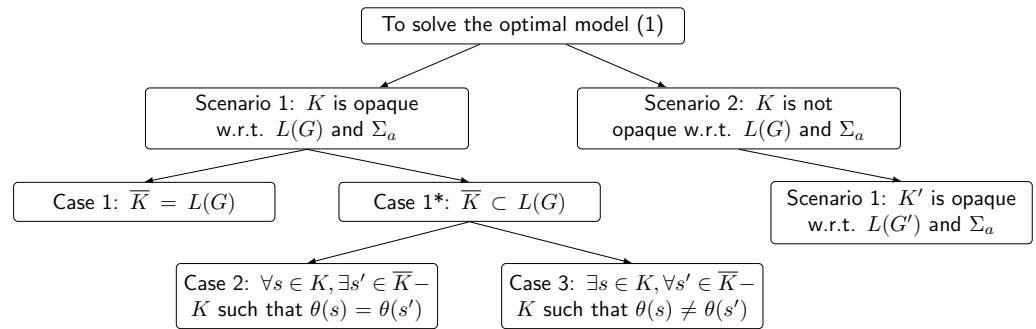


Figure 4. Flow chart to solve optimal Model (1), where G' is the closed-loop system under some supervisor and K' is the secret in G' .

In the process depicted in Figure 4, the main method used is verification of the opacity of some systems in Algorithm 2. The specific process is as follows:

Algorithm 2 How to know which scenario or case is available

```

1: if K is opaque with respect to L(G) and Σa then
2:   Scenario 1 (in Section 5) is available;
3:   if K̄ == L(G) then
4:     Case 1 (in Section 5.1) is available;
5:   else
6:     if K is opaque with respect to K̄ and Σa then
7:       Case 2 (in Section 5.2) is available;
8:     else
9:       Case 3 (in Section 5.3) is available;
10:    end if
11:  end if
12: else
13:   Scenario 2 (in Section 6) is available.
14: end if
15: return
  
```

5. Scenario 1: Secret K Is Opaque with Respect to $L(G)$ and Σ_a

We consider language $L(G)$ and regular language (secret) $K \subseteq L(G)$. By [4], we have $L(G) = L(g/G)$ for the reason that $L(g/G)$ is the supremal opaque sublanguage of $L(G)$. Therefore, the second condition of Model (1) is equivalent to formula $K \cap L(f/G) = K$, which implies that $K \subseteq L(f/G)$. Since $L(f/G)$ is prefix-closed, it is obvious that $K \subseteq L(f/G)$ is equivalent to formula $\bar{K} \subseteq L(f/G)$. So, in Model (1), the second constraint condition can be equivalently reduced to $\bar{K} \subseteq L(f/G)$. And then, optimal Model (1) can be displayed as the following Model (2):

$$\begin{aligned}
 & \min V(L(f/G)) \\
 & \text{s.t. } \theta(K \cap L(f/G)) \subseteq \theta(L(f/G) - K) \\
 & \quad \bar{K} \subseteq L(f/G) \\
 & \quad f \in \Gamma^{L(G)}
 \end{aligned} \tag{2}$$

To solve optimal Model (2), we consider the following three cases:

5.1. Case 1: $\bar{K} = L(G)$

We have $L(G) \subseteq L(f/G)$ in Condition 2 of Model (2). To ensure the feasible region is not empty, we can obtain supervisor f such that $L(G) = L(f/G)$. So, we have the following theorem:

Theorem 2. *We have system G and secret $K \subseteq L(G)$. In Case 1 of Scenario 1, $L(G) = L(f/G)$ is an optimal solution of optimal Model (2).*

Proof. From the above analysis, it is obvious that $L(G) = L(f/G)$ is the unique solution of the feasible set. So, $V(L(f/G))$ is minimal. $\square$

5.2. Case 2: $\bar{K} \subset L(G)$ and $\forall s \in K, \exists s' \in \bar{K} - K$ Such That $\theta(s) = \theta(s')$

In Case 2, the condition means that any secret cannot be distinguished from some non-secret in closure $\bar{K}$.

For secret K and its closure $\bar{K}$, we have $\theta(K) \subseteq \theta(\bar{K} - K) \subset \theta(L(G) - K)$. If there exists a closed-loop system f/G such that $\bar{K} \subseteq L(f/G)$, it obviously can ensure the opacity of secret K . So, the feasible region of Model (2) is not empty.

To find supervisor f , we infer whether $\bar{K}$ is controllable with respect to $L(G)$.

- If $\bar{K}$ is controllable with respect to $L(G)$, there exists supervisor f such that $L(f/G) = \bar{K}$. For closed-loop behavior $L(f/G)$, it satisfies the two constraint conditions of Model (2). So, the feasible region of Model (2) is not empty.
- If $\bar{K}$ is not controllable with respect to $L(G)$, we can find a controllable and closed superlanguage $\bar{K}^\downarrow$ of $\bar{K}$. Superlanguage $\bar{K}^\downarrow$ not only ensures the opacity of K (in Theorem A1 of Appendix A), but also maximizes secret K . So, the feasible region of Model (2) is not empty.

According to the above analysis, the following Theorem 3 states that an optimal solution of Model (2) can be obtained using the controllability of $\bar{K}$ or superlanguage $\bar{K}^\downarrow$.

Theorem 3. *We consider system G and secret $K \subseteq L(G)$. In Case 2 of Scenario 1, we have the following conclusions about the optimal solution of Model (2):*

- 1 If $\bar{K}$ is controllable with respect to $L(G)$, there exists supervisor f such that $L(f/G) = \bar{K}$, which is the optimal solution of Model (2).
- 2 If $\bar{K}$ is not controllable with respect to $L(G)$, there exists supervisor f such that $L(f/G) = \bar{K}^\downarrow$, which is the optimal solution of Model (2).

Proof. Firstly, we prove that closed-loop behavior $L(f/G)$ of Theorem 3 is a feasible solution of optimal Model (2).

If $\bar{K}$ is controllable with respect to $L(G)$, $\bar{K}$ is a closed and controllable sublanguage of $L(G)$. So, there exists supervisor f such that $L(f/G) = \bar{K}$. If $\bar{K}$ is not controllable with respect to $L(G)$, it is obvious that $\bar{K}^\downarrow$ is the infimal closed and controllable superlanguage of $\bar{K}$. So, there exists supervisor f such that $L(f/G) = \bar{K}^\downarrow$.

Therefore, we have $\bar{K} \subseteq L(f/G)$ which means constraint Condition 2 of Model (2) is true.

By Theorem A1 of Appendix A, we conclude that $L(f/G)$ can ensure the opacity of K under Case 2 of Scenario 1, which implies that constraint Condition 1 of Model (2) is true.

From the above points, it is true that $L(f/G)$ in Theorem 3 is a feasible solution of Model (2).

Next, we prove by contraction that the discount total choosing costs of $L(f/G)$ produced in Theorem 3 are minimal. We assume that there exists feasible solution $L(f'/G)$ ($\neq L(f/G)$) of Model (2) such that $V(L(f'/G)) < V(L(f/G))$.

According to constraint condition 2, we have $\bar{K} \subseteq L(f'/G)$. Afterward, we consider the controllability of $\bar{K}$.

If $\bar{K}$ is controllable, it holds that $L(f/G) = \bar{K}$ by Theorem 1, which means $L(f/G) \subseteq L(f'/G)$. So, we have $V(L(f/G)) \leq V(L(f'/G))$, which contracts with $V(L(f'/G)) < V(L(f/G))$.

If $\bar{K}$ is not controllable, it holds that $\bar{K} \subseteq L(f/G) = \bar{K}^\downarrow$ by Theorem 1. For any $s \in \bar{K}^\downarrow - \bar{K}$, there exists $t \leq s$ such that $t \in \bar{K}$ and $s = t\Sigma_u^*$ hold. As shown in Proposition 2 and Assumption 1, we have $V_\epsilon(s) = V_\epsilon(t) + V_t(\Sigma_u^*) = V_\epsilon(t)$. So, it holds that $V(\bar{K}) = V(\bar{K}^\downarrow) = V(L(f/G))$. According to formula $\bar{K} \subseteq L(f'/G)$, it is true that $V(L(f/G)) \leq V(L(f'/G))$, which contracts with $V(L(f'/G)) < V(L(f/G))$.

In summary, it is true that $V(L(f/G)) \leq V(L(f'/G))$, which means that the discount total choosing costs of $L(f/G)$ in Theorem 3 are minimal. $\square$

According to the proof of Theorem 3, we have the following corollaries:

Corollary 1. We consider language L . If new language L' is considered as the concatenation of any string of L with an uncontrollable string (i.e., Σ_u^*), then the discount total choosing costs of L and L' are the same, that is, $V(L) = V(L')$.

Corollary 2. We consider system G and secret $K \subseteq L(G)$. In Case 2 of Scenario 1, $V(\bar{K}) = V(\bar{K}^\downarrow) = V(L(f/G))$ holds, where $L(f/G)$ is the closed-loop system in Theorem 3.

Example 3. We consider finite transition system $G = (Q, \Sigma, \delta, q_0)$ shown in Figure 5, where $\Sigma_u = \{f, t\}$. Obviously, for system G , Assumption 1 is true. We suppose that secret $K = \{a, ab, aebgt\}$, which can be recognized by $Q_s = \{3, 6, 16\}$. To show choosing cost $c(s, \gamma)$, label $\textcircled{p} \xrightarrow{\cdot|n} \textcircled{q}$ means that if there is a transition from p to q by $\cdot$, notation n denotes choosing cost $c(s, \cdot)$. For control input Γ , the cost of choosing $\gamma \in \Gamma$ is defined as $c(s, \gamma) = \sum_{\sigma \in \gamma} c(s, \sigma)$.

We assume that the adversary has complete knowledge of the supervisor's control policy. From the adversary's view, the adversary can see a partial set of events, denoted by $\Sigma_a = \{a, b, d, f, g\}$. For secret K , it can be verified that K is opaque with respect to $L(G)$ and Σ_a (Scenario 1). To reduce the choosing cost, closed-loop system $L(f/G)$ can be obtained in Theorem 3, where $L(f/G) = \{\epsilon, t, a, ab, abt, abtf, ae, aeb, aebg, aebgt\}$ is shown in Figure 6.

By Definition 5 and Algorithm 1, $V(L(f/G)) = V_\epsilon(t) + V_\epsilon(a) + V_a(ebgt) + V_a(bt f) = 1.726$ is minimal.

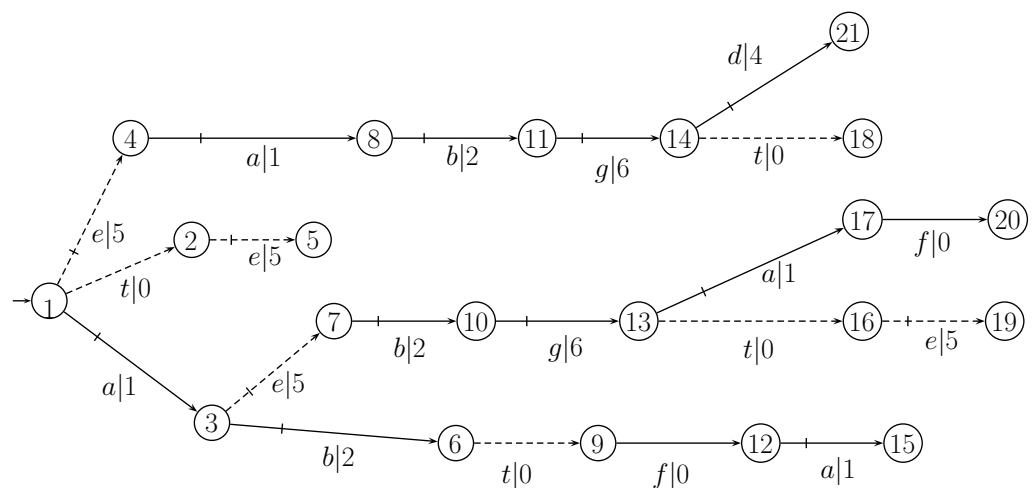


Figure 5. System G.

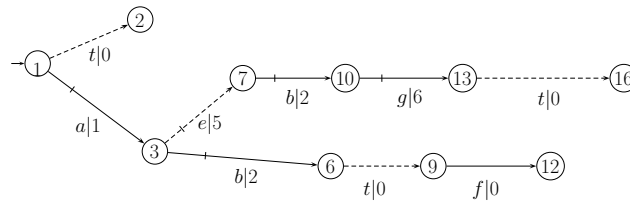


Figure 6. Closed-loop language $L(f/G)$.

5.3. Case 3: $\bar{K} \subset L(G)$ and $\exists s \in K, \forall s' \in \bar{K} - K$ Such That $\theta(s) \neq \theta(s')$

In Case 3, the condition means that there exists some secret in K such that all the non-secrets confused with them are outside of $\bar{K}$.

For $L(G)$, we let $\underline{K} = \{s \in K | \forall s' \in \bar{K} - K, \theta(s) \neq \theta(s')\}$ be a set of some secret which cannot be confused by any string in $\bar{K} - K$, and $L = \{s \in L(G) - K | \theta(s) \subseteq \theta(\underline{K})\}$ be a set of strings which can confuse the secret of $\underline{K}$. For language L , we call $[s] = \{s' \in L | \theta(s) = \theta(s')\}$ the coset (or equivalence class) of s with respect to L and θ , where s' is said to be the equivalent string of s . And $L/[\cdot]$ is defined as the quotient set of L with respect to coset $[\cdot]$. For a determined finite transition system, the number of strings in L is finite and the length of each string of L is finite, too. Obviously, coset $[\cdot]$ and quotient set $L/[\cdot]$ are also finite.

To solve Model (2), Algorithm 3 shown as follows is proposed by referring to Function 1 (seen in Algorithm 4) and Function 2 (seen in Algorithm 5).

Algorithm 3 Optimal supervisory control I

Input: Automaton G , secret K and choosing cost $c(s, \gamma)$;

Output: Closed-loop language $L(f/G)$.

```

1: Construct language  $L = \{s \in L(G) - K | \theta(s) \subseteq \theta(\underline{K})\}$ 
2:  $j = 1$ 
3: for  $s \in \underline{K}$  do
4:    $[s] = \{s' \in L | \theta(s) = \theta(s')\}$ 
5:   if  $[s] = \emptyset$  then
6:     Continue
7:   end if
8:    $H_j = [s]$ 
9:    $L = L - [s]$ 
10:   $j = j + 1$ 
11: end for
12:  $H = \{H_j | j \in \{1, 2, \dots, j-1\}\}$ 
13:  $V = \text{function1}(L/[\cdot], K)$ 
14:  $\text{Label} = \text{function2}(H, V)$ 
15: In Label, find and sort from  $t_s$  to  $t_t$ , i.e.,  $t_s \rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_{j-1} \rightarrow t_t$ 
16:  $\bar{K}' = \bar{K} \cup \{s_1\} \cup \dots \cup \{s_{j-1}\}$ 
17: By Theorem 3,  $L(f/G)$  can be got from  $\bar{K}'$ .
18: return  $L(f/G)$ 

```

In Line 13 of Algorithm 3, Function 1 (in Algorithm 4) shows how to compute the choosing cost outside of the closure of secret K . For a quotient set, we take any string of a coset and obtain a prefix with a maximal length in the closure of the secret. And then, we compute the discount cost of choosing the remaining string after the prefix. The specific process is shown in Algorithm 4.

In Line 14 of Algorithm 3, Function 2 constructs a weighted directed diagram with multi-stages and produces a path with minimal discount total choosing cost. For the diagram, the elements of a set H are regarded as the stages, and the elements of H_i are defined as the nodes of each stage. Based on dynamic programming, the optimal weight between different nodes of adjacent stages is obtained in Function 3 (in Algorithm 6). Then, the weighted directed diagram is obtained. For every node of the diagram, an ordered pair is obtained by employing Function 3 (in Algorithm 6). For the ordered pair, the first

element is the set of shortest paths with minimal discount total choosing cost from the starting node to the current node, and the second is the discount total choosing cost of the path. When the current node is the ending node, the path with minimal discount total choosing cost is obtained. The specific processes are shown in Algorithms 5 and 6.

Algorithm 4 Calculation of the choosing cost $V = \text{function1}(R, K)$ outside of $\bar{K}$

Input: Quotient set R and secret K ;

Output: Choosing cost V outside of $\bar{K}$.

```

1: for  $[s] \in R$  do
2:   for  $s \in [s]$  do
3:     let  $n$  be the length of  $s$ , and  $s$  is denoted by  $s = \sigma_1\sigma_2 \cdots \sigma_n$ 
4:      $i = n - 1$ 
5:     take  $s = s_it$ , and suppose that  $s_i = \sigma_1 \cdots \sigma_i, t = \sigma_{i+1} \cdots \sigma_n$ 
6:     while  $i < n$  do
7:       if  $i == -1$  then
8:          $s_i = \epsilon$ 
9:         Break
10:      end if
11:      if  $s_i \in \bar{K}$  then
12:        Break
13:      end if
14:       $i = i - 1$ 
15:    end while
16:    compute the cost  $t$  after  $s_i$ , that is  $V_{s_i}(t)$ 
17:  end for
18: end for
19: return  $V = \{V_{s_i}(t) | s' = s_it, s' \in [s], s_i \in \bar{K}, s_i\sigma_{i+1} \in \bar{s} - \bar{K}\}$ 

```

Algorithm 5 Finding the set of strings $\text{Label}_{t_t} = \text{function2}(H, V)$ whose choosing cost of the path from the starting node is minimal

Input: $H = \{H_1, H_2, \cdots, H_{|H|}\}$ and V , where $|H|$ is the number of the elements of H ;

Output: the strings set Label_{t_t} of the shorted path from the starting node to the ending node.

```

1:  $H_0 = \{t_s\}$ 
2:  $\text{Label}_{t_s} = \{t_s\}$ 
3: for  $j = 1 : 1 : |H|$  do
4:   for  $s \in H_j$  do
5:     if  $j = 1$  then
6:        $a_{t_s, s} = V_{s_i}(t)$ 
7:        $\text{Label}_s = \text{Label}_{t_s} \cup \{s\}$ 
8:        $V_{\min}(s) = a_{t_s, s}$ 
9:     else
10:      for  $s' \in H_{j-1}$  do
11:         $a_{s', s} = \text{function3}(s', s, \text{Label}_{s'}, V_{\min}(s'))$ 
12:         $V_{s'}(s) = V_{\min}(s') + a_{s', s}$ 
13:      end for
14:       $s' = \arg \min \{V_{s'}(s), s' \in H_{j-1}\}$ 
15:       $\text{Label}_s = \text{Label}_{s'} \cup \{s\}$ 
16:       $V_{\min}(s) = V_{s'}(s)$ 
17:    end if
18:  end for
19: end for
20:  $H_{j+1} = \{t_t\}$ 
21: for  $s \in H_j$  do
22:    $a_{s, t_t} = 0$ 
23: end for
24:  $s' = \arg \min \{V_{\min}(s) | s \in H_{|H|}\}$ 
25:  $\text{Label}_{t_t} = \text{Label}_{s'} \cup \{t_t\}$ 
26: return  $\text{Label}_{t_t}$ 

```

Algorithm 6 Computing optimal weight value $a_{s',s} = \text{function3}(s', s, \text{Label}_{s'}, V_{\min}(s'))$ between s' and s

Input: $s', s, \text{Label}_{s'}, V_{\min}(s')$;

Output: Optimal weight value $a_{s',s}$ from node s' to node s .

```

1:  $a_{s',s} = \infty$ 
2: for  $s'' \in \text{Label}_{s'}$  do
3:   compute the longest common prefix  $s_1$  of  $s$  and  $s''$  such that  $s = s_1 t_1$  and  $s'' = s_1 t'_1$ ,
   and  $t_1 \neq t'_1$ 
4:   if  $t'_1 = \epsilon$  then
5:     compute the cost  $t_1$  after  $s_1$ ,  $V_{s_1}(t_1)$ 
6:      $a_{s',s} = \min\{a_{s',s}, V_{s_1}(t_1)\}$ 
7:   else
8:     if  $t_1 = \epsilon$  then
9:        $a_{s',s} = 0$ 
10:      Break
11:    else
12:       $a_{s',s} = \min\{a_{s',s}, V_{s_1}(t_1)\}$ 
13:    end if
14:  end if
15: end for
16: return  $a_{s',s}$ 

```

According to the calculation process of Algorithm 3, we have the following theorem to show the solution of Model (2):

Theorem 4. We consider system G and secret $K \subseteq L(G)$. In Case 3 of Scenario 1, closed-loop behavior $L(f/G)$ produced in Algorithm 3 is an optimal solution of Model (2).

Proof. We first show that closed-loop behavior $L(f/G)$ produced in Algorithm 3 is a feasible solution of optimal Model (2).

1. Proof of the opacity (the first constraint condition).
As shown in Case 3, the secret of $K - \underline{K}$ can be confused by the non-secret strings of $\overline{K}$. For $\underline{K}$, all the non-secrets in $L(G)$ which cannot be distinguished from the secret in $\underline{K}$ are in $\cup_j H_j$ of Line 12. At Lines 14 and 15, string s_i is from H_i , where $i = \{1, 2, \dots, j-1\}$. At Lines 15 and 16, we have $\overline{K} \subseteq L(f/G)$ and $\{s_1, s_2, \dots, s_{j-1}\} \subseteq L(f/G)$. So, closed-loop behavior $L(f/G)$ produced in Algorithm 3 can ensure the opacity of secret K .
2. Proof of the secret remaining in the closed-loop system being maximal (the second constraint condition).
According to Lines 15 and 16, it holds that $\overline{K} \subseteq L(f/G)$. So, the second constraint condition is true.

To sum up, the closed-loop behavior $L(f/G)$ obtained in Algorithm 3 is a feasible solution of optimal Model (2).

Secondly, we show that the discount total choosing cost of closed-loop behavior $L(f/G)$ produced by Algorithm 3 is minimal.

Since it holds that $\overline{K} \subseteq L(f/G)$, the discount total choosing cost of $L(f/G)$ can be computed as follows:

$$V(L(f/G)) = V(\overline{K}) + V(L(f/G) - \overline{K}). \quad (3)$$

$V(\overline{K})$ is finite. To minimize the discount total choosing cost of $L(f/G)$, we need to show $V(L(f/G) - \overline{K})$ is minimal using Formula (3). As shown in Line 1 of Algorithm 3, it is obvious that language L contains all the non-secrets in $L(G)$, which cannot be distinguished

from all the secrets of $\underline{K}$. So, if we want to make $V(L(f/G) - \bar{K})$ minimal, all the strings s in $L(f/G) - \bar{K}$ must come from L . And then $L(f/G) - \bar{K} \subseteq L$ holds. According to Lines 3–11 of Algorithm 3, we have $L = \cup_j H_j$. And all the strings in H_j can confuse one secret of $\underline{K}$ and its equivalent secret. Only one string is chosen in each set, H_j of H , which is the necessary condition to minimize $V(L(f/G) - \bar{K})$.

At Line 13 of Algorithm 3 (i.e., function 1 of Algorithm 4), all the strings $s = s_i t$ in L are traversed and choosing cost $V_{s_i}(t)$ after $s_i \in \bar{K}$ can be obtained, where $t = \sigma_{i+1} \cdots \sigma_n$ and $s_i \sigma_{i+1} \notin \bar{K}$.

At Line 14 of Algorithm 3, a diagram with multi-stages is constructed in $\cup_j H_j \cup \{t_s\} \cup \{t_t\}$, where initial node t_s and ending node t_t are virtual, H_j is the set of nodes in the j th stage. To only pick a string in each H_j , we find a path from t_s to t_t . And then, Algorithm 6 is proposed to optimize weight $a_{s',s}$ of transition from node s' to node s between adjacent stages. For optimal weight, it is obvious that the discount total choosing cost of each node (i.e., string) of the path is equal to the total weight of the path (at Line 12 of Algorithm 5). At Lines 3–19 of Algorithm 5, the shortest path Label_s and its minimal discount total choosing cost $V_{\min}(s)$ of the j th stage can be obtained by $\text{Label}_{s'}$ and $V_{\min}(s')$ of the $(j-1)$ th stage based on dynamical programming. As shown in the above analysis about Line 14 of Algorithm 3 (i.e., Function 2 in Algorithm 5), the first element $\text{Label}_{s'}$ of the ordered pair $(\text{Label}_{s'}, V_{\min}(s'))$ is the shorted path (i.e., the set of strings) with minimal discount total choosing cost from starting node t_s to current node s' , and the second $V_{\min}(s')$ is the discount total choosing cost of the path. When the current node is t_t (i.e., Line 20 of Algorithm 5), Label_{t_t} is the shorted path from the initial to the ending node and $V_{\min}(t_t)$ is the minimal discount total choosing cost of the path (see Lines 21–25 of Algorithm 5). So, Label_{t_t} is the subset of L , whose discount total choosing cost is minimal and whose strings can confuse all the secrets of $\underline{K}$.

At Lines 16 and 17 of Algorithm 3, closed-loop behavior $L(f/G)$ can be ensured to be controllable and closed by Corollary 2. The discount total choosing $L(f/G)$ is minimal as shown in the above analysis.

All in all, closed-loop behavior $L(f/G)$ produced in Algorithm 3 is an optimal solution of Model (2). $\square$

Example 4. We consider finite transition system $G = (Q, \Sigma, \delta, q_0)$ and secret $K = \{a, ab, aebg, aebgt\}$ shown in Figure 7, where $\Sigma_u = \{f, t\}$ and K can be recognized by $Q_s = \{3, 6, 13, 16\}$. We suppose that the adversary has complete knowledge of the supervisor's control policy, and the observed set of events by the adversary is $\Sigma_a = \{a, d, f, g, t\}$. It is verified that K is opaque with respect to $L(G)$ and Σ_a . But K is not opaque with respect to $\bar{K}$ and Σ_a , i.e., secret $aebg, aebgt$ cannot be confused by any non-secret of $\bar{K}$. Case 3 of Scenario 1 is fulfilled and Assumption 1 is true. Next, we construct closed-loop behavior $L(f/G)$ using Algorithm 3.

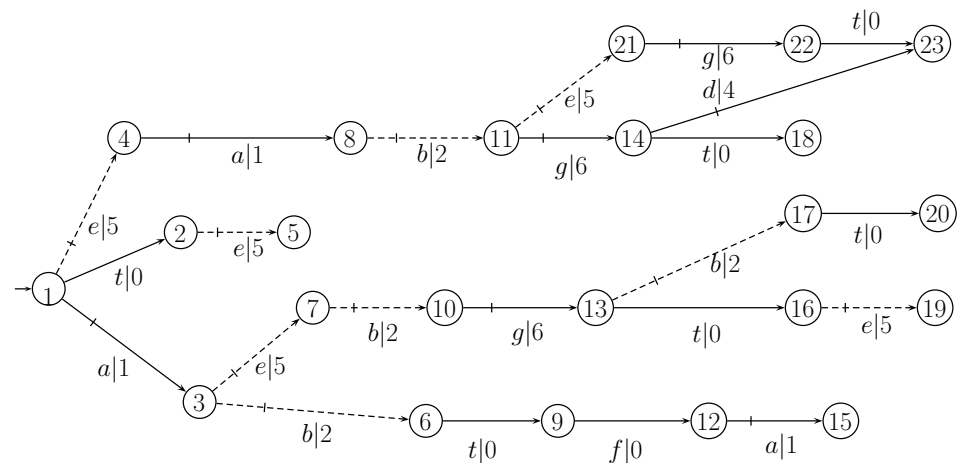


Figure 7. System G.

For system G and secret K , we obtain language $\underline{K} = \{aebg, aebgt\}$, where the strings in $\underline{K}$ cannot be confused by any strings of $\overline{K}$. From the opacity of $L(G)$, we can find sub-language $L = \{aebgte, aebgbt, aebgb, eabgt, eabg, eabeg, eabegt\}$, whose strings cannot be distinguished with the secret in $\underline{K}$. For language L , we offer the following computational process:

We take $aebg \in \underline{K}$, and then we have $\theta(aebg) = ag$ and $H_1 = [aebg] = \{eabg, eabeg, aebgb\}$.

We take $aebgt \in \underline{K}$, and then we have $\theta(aebgt) = agt$ and $H_2[aebgt] = \{eabgt, eabegt, aebgte, aebgbt\}$.

So, quotient set $L/[\cdot] = \{\{eabg, eabeg, aebgb\}, \{eabgt, eabegt, aebgte, aebgbt\}\}$ is a partition of L .

For $[aebg]$, we can compute the choosing cost of the suffix of the non-secret string out of $\overline{K}$ in the first stage (seen in Function 1 of Algorithm 4).

If $s = eabg$, we have $\epsilon \in \overline{K}$ and $V_\epsilon(eabg) = 5.126$.

If $s = eabeg$, we have $\epsilon \in \overline{K}$ and $V_\epsilon(eabeg) = 5.1256$.

If $s = aebgb$, we have $aebg \in \overline{K}$ and $V_{aebg}(b) = 0.0002$.

For coset $[aebgt]$, we can similarly obtain the following in the second stage:

If $s = eabgt$, we have $\epsilon \in \overline{K}$ and $V_\epsilon(eabgt) = 5.126$.

If $s = eabegt$, we have $\epsilon \in \overline{K}$ and $V_\epsilon(eabegt) = 5.1256$.

If $s = aebgte$, we have $aebgt \in \overline{K}$ and $V_{aebgt}(e) = 0.00005$.

If $s = aebgbt$, we have $aebg \in \overline{K}$ and $V_{aebg}(bt) = 0.0002$.

Based on H_1, H_2 and the choosing cost out of $\overline{K}$ above, a weighted directed diagram shown in Figure 8 is constructed using Algorithm 5 calling Algorithm 6. In the diagram, every node denoted by $\odot$ is shown as a fraction. For the fraction, its numerator is non-secret string $s = s_i\sigma_{i+1} \cdots \sigma_{i+k}$ in $\cup_{j=1}^2 H_j$, and its denominator is $V_{s_i}(t)$, where $s_i \in \overline{K}$ and $s_i\sigma_{i+1} \notin \overline{K}$.

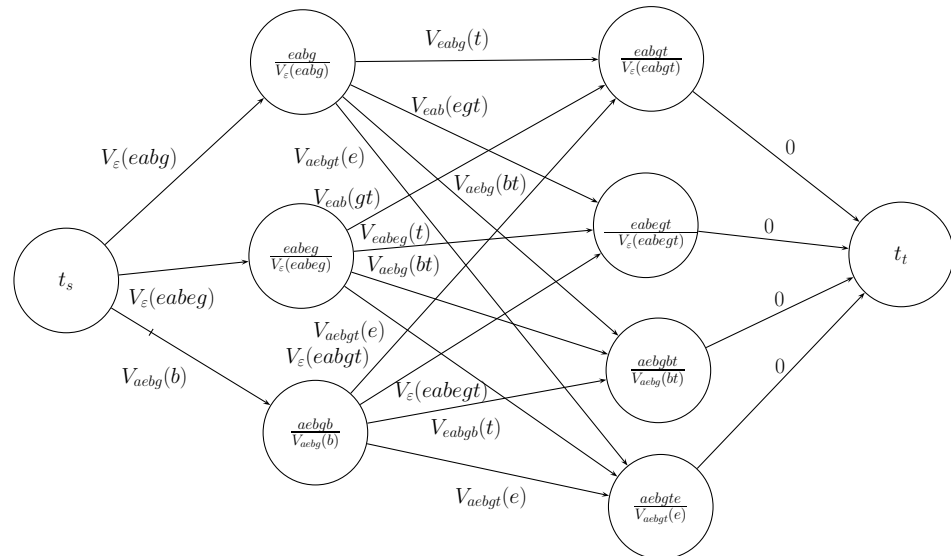


Figure 8. A weighted directed diagram.

To show the weight between nodes of adjacent stages, some weight is given as follows by Definition 5:

$V_{eabg}(t) = 0$, $V_{eab}(egt) = 0.0056$, $V_{eabeg}(t) = 0$, $V_{eab}(gt) = 0.006$, $V_{aebgb}(t) = 0$.

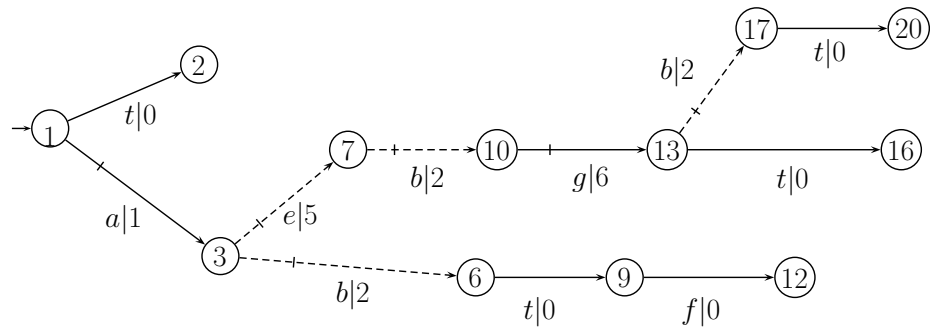
By Algorithm 5, label $Label_s$ and minimal choosing cost $V_{\min}(s)$ of a path from initial node t_s to current node s are computed in Table 1.

In Table 1, we have $Label_{t_t} = \{t_s, aebgb, aebgbt, t_t\}$ for ending node t_t . From Line 16 of Algorithm 3, we know that the shortest path is $t_s \rightarrow aebgb \rightarrow aebgbt \rightarrow t_t$. So, it holds that $\min V(L(f/G) - \overline{K}) = 0.0002$. Since $V(\overline{K}) = 1.726$ (by Algorithm 1) is finite, $V(L(f/G)) = 1.726 + 0.0002 = 1.7262$. So, in Line 17, we have $L(f/G) = \{\overline{t}, abtf, aebgt, aebgbt\}$ shown in Figure 9.

In Figure 9, it is verified that $V(L(f/G)) = 1.7262$ by Algorithm 1.

Table 1. The set $Label_s$ of shortest path and its minimal discount total choosing cost $V_{\min}(s)$ at every node s of the diagram.

s	$j = 0$ $Label_s, V_{\min}(s)$	$j = 1$ $Label_s, V_{\min}(s)$	$j = 2$ $Label_s, V_{\min}(s)$	$j = 3$ $Label_s, V_{\min}(s)$
t_s	$\{t_s\}, 0$			
$eabg$		$\{t_s, eabg\}, 5.126$		
$eabeg$		$\{t_s, eabeg\}, 5.1256$		
$aebgb$		$\{t_s, aebgb\}, 0.0002$		
$eabgt$			$\{t_s, eabg, eabgt\}, 5.126$	
$eabegt$			$\{t_s, eabeg, eabegt\}, 5.1256$	
$aebgbt$			$\{t_s, aebgb, aebgbt\}, 0.0002$	
$aebgte$			$\{t_s, aebgb, aebgte\}, 0.00025$	
t_t				$\{t_s, aebgb, aebgbt, t_t\}, 0.0002$

**Figure 9.** Closed-loop language $L(f/G)$.

6. Scenario 2: Secret K Is Not Opaque with Respect to $L(G)$ and Σ_a

For system G and secret $K \subseteq L(G)$, if K is not opaque with respect to $L(G)$ and Σ_a ; we need to design a supervisor to prohibit all the secrets disclosed. To obtain the supervisor, we can use the method of [1,4] to end up with the maximal permission sublanguage of $L(G)$ which can ensure the opacity of K . Then, Scenario 1 is fulfilled. Next, we propose Algorithm 7 to solve Model (1).

Algorithm 7 Optimal supervisory control II

Input: Automaton G , secret K , and choosing cost $c(s, \gamma)$;

Output: Closed-loop behavior $L(f/G)$.

- 1: Compute the observer $obs(G)$ of G
- 2: Compute the parallel composition $G \parallel obs(G)$
- 3: For the parallel composition $G \parallel obs(G)$, find supervisor g such that $L(g/G)$ is the maximal permissive closed-loop behavior which can ensure K opaque [1,4].
- 4: $G' = g/G$
- 5: $K' = K \cap L(g/G)$
- 6: **if** $\overline{K'} = L(G')$ **then**
- 7: Design supervisor f' such that $L(f'/G') = L(G')$
- 8: **else**
- 9: **if** $\forall s \in K', \exists \overline{K'} - K' \text{ s.t. } \theta(s) = \theta(s') \text{ then}$
- 10: Design supervisor f' from $\overline{K'}$ by Theorem 3, and then output closed-loop behavior $L(f'/G')$
- 11: **else**
- 12: Perform Algorithm 3 to design supervisor f' , and then output closed-loop behavior $L(f'/G')$
- 13: **end if**
- 14: **end if**
- 15: $L(f/G) = L(f'/G')$
- 16: **return** $L(f/G)$

According to the above algorithm, we first construct maximal permissive supervisor g to enforce the opacity of K . And then, it is verified that closed-loop behavior $L(g/G)$ and restricted secret $K \cap L(g/G)$ meet the requirements of Scenario 1. As shown in Theorem 2, Theorem 3, and Theorem 4, we can conclude that Algorithm 7 can produce an optimal solution of Model (1).

Theorem 5. We consider system G and secret $K \subseteq L(G)$. In Scenario 2, closed-loop behavior $L(f/G)$ obtained in Algorithm 7 is an optimal solution of Model (1).

Proof. Firstly, we prove that $L(f'/G')$ is a controllable and closed sublanguage of $L(G)$. As shown in Lines 7, 10, and 12, $L(f'/G')$ is a closed sublanguage of $L(G)$. Next, we prove $L(f'/G')$ is controllable with respect to $L(G)$.

$$\begin{aligned}
 & \forall s \in L(f'/G'), \sigma \in \Sigma_u, s\sigma \in L(G) \\
 \Rightarrow & s \in L(g/G), \sigma \in \Sigma_u, s\sigma \in L(G) \\
 & (\because L(f'/G') \subseteq L(G') = L(g/G)) \\
 \Rightarrow & s\sigma \in L(g/G) \\
 & (\because L(g/G) \text{ is controllable with respect to } L(G)) \\
 \Rightarrow & s\sigma \in L(f'/G') \\
 & (\because L(f'/G') \text{ is controllable with respect to } L(g/G))
 \end{aligned}$$

$L(f'/G')$ is a controllable and closed sublanguage of $L(G)$. At Line 15, there exists supervisor f such that $L(f/G) = L(f'/G')$.

Secondly, we show that closed-loop behavior $L(f/G)$ produced by Algorithm 7 is a feasible solution of Model (1).

1. Showing the opacity of $L(f/G)$.
In Lines 3–5, it is obvious that K' is opaque with respect to $L(G')$ and Σ_a . At Lines 6–14, by Theorems 2–4, it holds that K' is opaque with respect to $L(f'/G')$ and Σ_a , which implies that $\theta(K' \cap L(f'/G')) \subseteq \theta(L(f'/G') - K')$. According to Lines 4, 5, and 15, we have $\theta(K \cap L(f/G)) \subseteq \theta(L(f/G) - K \cap L(g/G))$. Since $L(f/G) \subseteq L(g/G)$, we have $\theta(L(f/G) - K \cap L(g/G)) \subseteq \theta(L(f/G) - K \cap L(f/G))$. So, $\theta(K \cap L(f/G)) \subseteq \theta(L(f/G) - K \cap L(f/G))$ holds, which means $\theta(K \cap L(f/G)) \subseteq \theta(L(f/G) - K)$ is true. Therefore, K is opaque with respect to $L(f/G)$ and Σ_a .
2. Showing that closed-loop behavior $L(f/G)$ can preserve the maximal secret information.
For system G' and secret K' , at Lines 4–14, it is obvious that $L(f'/G')$ is a feasible solution of Model (2), which implies that $\overline{K'} \subseteq L(f'/G')$. Since $K \cap L(g/G) = K'$ at Line 5 and $L(f/G) = L(f'/G')$ at Line 15, it holds that $K \cap L(g/G) \subseteq L(f/G)$, which implies that $K \cap L(g/G) \subseteq K \cap L(f/G)$. Since $L(f'/G') \subseteq L(G')$ holds, it is true that $L(f/G) \subseteq L(g/G)$. So, we have $K \cap L(f/G) \subseteq K \cap L(g/G)$. Therefore, we have $K \cap L(f/G) = K \cap L(g/G)$.

To conclude, closed-loop behavior $L(f/G)$ produced by Algorithm 7 is a feasible solution of Model (1).

Finally, we show that the discount total choosing cost of $L(f/G)$ produced by Algorithm 7 is minimal for Model (1).

We assume that closed-loop behavior $L(f/G)$ produced by Algorithm 7 is not the optimal solution of Model (1). So, there exists supervisor f_1 such that $L(f_1/G)$ is a feasible solution of Model (1) and $V(L(f_1/G)) < V(L(f/G))$ holds. For Model (1), the two constrain conditions are satisfied for $L(f_1/G)$. The two conditions mean that $K \cap L(f_1/G)$ is opaque with respect to $L(f_1/G)$ and Σ_a , and $K \cap L(f_1/G) = K \cap L(g/G)$ holds. As shown in Line 5, we have $K' = K \cap L(f_1/G)$. Taking $G_1 = f_1/G$, it holds that $\overline{K'} \subseteq L(G_1)$. Based on the assumption about $L(f_1/G)$ and $L(g/G)$, we have $L(f_1/G) \subseteq L(g/G)$. Then, we consider the following two cases:

Case 1 If $L(f_1/G) = L(g/G)$, we discuss the relation between $\overline{K'}$ and $L(G_1)$.

- 1.1 If $\overline{K'} = L(G_1)$, it holds that $L(f_1/G) = L(f/G)$ at Lines 6 and 7, which contracts with the assumption that $V(L(f_1/G)) < V(L(f/G))$.
- 1.2 If $\overline{K'} \subset L(G_1)$, it holds that $V(L(f/G)) \leq V(L(f_1/G))$ by Theorems 3 and 4 at Lines 9–13, which contracts with the assumption that $V(L(f_1/G)) < V(L(f/G))$.

Case 2 If $L(f_1/G) \subset L(g/G)$, it is true that $s \notin \overline{K'}$ for any $s \in L(g/G) - L(f_1/G)$ because of formulas $\overline{K'} \subseteq L(G_1)$ and $L(f_1/G) = L(G_1)$, which implies that s is out of $\overline{K'}$. Then, we discuss the relation between $\overline{K'}$ and $L(G_1)$ again.

- 2.1 If $\overline{K'} = L(G_1)$, there exists $s' \in \overline{K'} - K'$ such that $\theta(s) = \theta(s')$ for any $s \in K'$, which means that all the secret strings in K' can be confused by the non-secret string in $\overline{K'}$. By Corollary 2, it holds that $V(L(f_1/G)) = V(\overline{K'})$. In Lines 9 and 10, it is true that $V(L(f/G)) = V(\overline{K'})$. So, it holds that $V(L(f/G)) = V(L(f_1/G))$, which contracts with assumption that $V(L(f_1/G)) < V(L(f/G))$.
- 2.2 If $\overline{K'} \subset L(G_1)$, we discuss the following two sub-cases:
 - 2.2.1 If there exists $s' \in \overline{K'} - K'$ such that $\theta(s) = \theta(s')$ for any $s \in K'$, it is obvious that $V(L(f_1/G)) = V(\overline{K'})$ by Corollary 2. At Lines 9, 10 and 15 of Algorithm 7, it holds that $V(L(f/G)) = V(\overline{K'})$. So, it is true that $V(L(f/G)) = V(L(f_1/G))$, which contracts with assumption that $V(L(f_1/G)) < V(L(f/G))$.
 - 2.2.2 If there exists $s \in K'$ such that $\theta(s) \neq \theta(s')$ for any $s' \in \overline{K'} - K'$, we have the following formulas:

$$V(L(f_1/G)) = V(\overline{K'}, f_1) + V(L(f_1/G) - \overline{K'}) \quad (4)$$

$$V(L(f/G)) = V(\overline{K'}, f) + V(L(f/G) - \overline{K'}). \quad (5)$$

Owing to the definition of feasible solution, we have $\overline{K'} \subseteq L(f_1/G)$ and $\overline{K'} \subseteq L(f/G)$. So, it holds that $V(\overline{K'}, f_1) = V(\overline{K'}, f)$. For the remaining part of Formulas (4) and (5), we construct two weight directed diagrams T_1 and T , where T_1 (or T) is produced in Algorithm 5 (e.g., Line 14 of Algorithm 3) if $(f_1/G, K', c(s, \gamma))$ (or $(g/G, K', c(s, \gamma))$) is inputted in Algorithm 3. By the constructions of L and H in Algorithm 3, it is obvious that $T_1 \subseteq T$ (i.e., T_1 is a sub-diagram of T) and that $a_{s',s}^{T_1} \geq a_{s',s}^T$, where $a_{s',s}^{T_1}$ (and $a_{s',s}^T$) is the weight of arc (s, s') in diagram T_1 (and T , respectively). So, the sum of the weight of the shortest path of T is less than that of T_1 , which implies that $V(L(f/G) - \overline{K'}) \leq V(L(f_1/G) - \overline{K'})$. According to Formulas (4) and (5), we have $V(L(f/G)) \leq V(L(f_1/G))$, which contracts with assumption that $V(L(f_1/G)) < V(L(f/G))$.

To conclude, it is true that $V(L(f/G)) \leq V(L(f_1/G))$, which implies that the discount total choosing cost of closed-loop behavior $L(f/G)$ produced by Algorithm 7 is minimal for optimal Model (1). $\square$

To show the effectiveness of Algorithm 7, we introduce the model of [1] to compute the optimal choosing control strategy.

Example 5. We consider transition system G [1] shown in Figure 10, which models all sequences of possible moves of an agent in a three-storey building with a south wing and a north wing, both equipped with lifts and both connected by a corridor at each floor. Moreover, there is a staircase that leads from the first floor in the south wing to the third floor in the north wing. The agent starts from the first floor in the south wing. They can walk up the stairs (s) or walk through the corridors (c) from south to north without any control. The lifts can be used several times one floor upwards (u) and at most once on floor downwards (d) altogether. The moves of the lifts are controllable. Thus, $\Sigma_c = \{u, d\}$. The secret is that the agent is either on the second floor in the south wing or on the third floor in the north wing, i.e., $Q_s = \{1, 5, 7, 11\}$ marked by a double circle. The adversary may

gather the exact subsequence of moves in $\Sigma_a = \{u, c, s\}$ from sensors, but they cannot observe the downwards moves of the lifts.

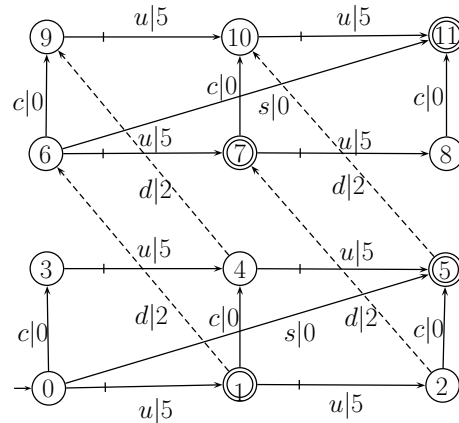


Figure 10. System G .

For every transition of system G , the choosing cost is shown in Figure 10. In [1], there are a unique supremal prefix-closed and controllable sublanguage $L(f/G)$ (shown in Figure 11) of $L(G)$ such that secret S is opaque with respect to $L(f/G)$ and Σ_a . So, $V(\bar{K}') = V_e(s) + V_e(cuu) = 0 + 0.55 = 0.55$.

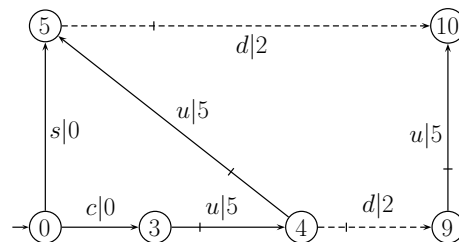


Figure 11. Closed-Loop System g/G .

We suppose that choosing cost $c(s, \gamma)$ is inserted in G and g/G , shown in Figures 10 and 11, respectively. According to Line 12 of Algorithm 7 (i.e., Algorithm 3), $\bar{K}' = \{s, cuu\}$ and $L = \{sd, cuud\}$. By the process of Algorithm 3, we have $H_1 = \{sd\}$ and $H_2 = \{cuud\}$, which means there exists only one path (shown in Figure 12) from the starting node to the ending node. So, $\min_{f \in \Gamma^{L(G)}} V(L(f/G) - \bar{K}') = V_s(d) + V_{cuu}(d) = 0.2 + 0.002 = 0.202$.

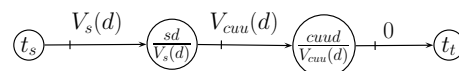


Figure 12. A weighted directed diagram.

At Lines 16 and 17 of Algorithm 3, it is obvious that $L(f/G)$ is shown in Figure 13, which has the minimal discount total choosing cost $V(f/G) = V(\bar{K}', f) + V(L(f/G) - \bar{K}') = 0.55 + 0.202 = 0.752$.

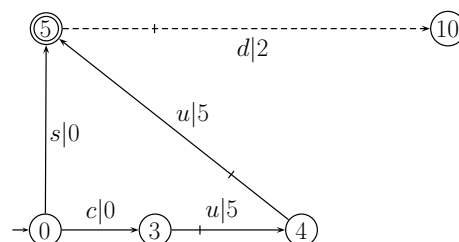


Figure 13. Closed-loop behavior $L(f/G)$ with the minimal discount total choosing cost.

The optimal supervisory control defined by $L(f/G)$ prevents the agent from using the lift of the south wing and the lift of the north wing from the second floor to the third floor at any time after they used this lift downwards, as well as the lift of the north wing downwards on the second floor.

7. Conclusions

When opacity-enforcing supervisory control is considered in Discrete Event Systems, we have to face another problem, i.e., cost. In reality, we hope we can reduce the cost while preserving the opacity of the supervised system. So, an optimal supervisory control model is formulated to enforce opacity by a supervisor with minimal discount total choosing cost. In the model, the objective function is to minimize the discount total choosing cost of closed-loop behavior, and the two constraint conditions are given: one is to enforce the opacity of closed-loop behavior, the other is to preserve the maximal part of secret information for closed-loop behavior. To solve the above optimal model, some algorithms and theorems are formulated from simplicity to complexity.

In this paper, the plant is modeled by the Finite Transition System, because coset $[\cdot]$ and quotient set $L/[\cdot]$ in Algorithms 3 and 4 may be infinite in Finite State Machine. To break the restriction, we will adopt a finite state to replace infinite event string and introduce the optimal control problem to Finite State Machine, which is our future work.

Author Contributions: Conceptualization, methodology, Y.D. and F.W.; supervision, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by National Natural Science Foundation of China grant number 61203040, Natural Science Foundation of Fujian Province grant number 2022J01295 and Science and Technology Association Project of Quanzhou.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Theorem A1. We consider system G and languages L, K satisfied with $\bar{K} \subseteq L$ and $\bar{K} \subseteq L(G)$. If K is opaque with respect to $\bar{K}$ and Σ_a , then K is opaque with respect to L and Σ_a .

Proof.

$$\begin{aligned}
 \forall s \in K &\Rightarrow \exists s' \in \bar{K} - K, s.t. \theta(s) = \theta(s') \\
 &(\because K \text{ is opaque with respect to } \bar{K} \text{ and } \Sigma_a) \\
 &\Rightarrow s' \in L - K, s.t. \theta(s) = \theta(s') \\
 &(\because \bar{K} \subseteq L) \\
 &\Rightarrow K \text{ is opaque with respect to } L \text{ and } \Sigma_a \\
 &(\because \text{Definition 2})
 \end{aligned} \tag{A1}$$

□

References

1. Dubreil, J.; Darondeau, P.; Marchand, H. Supervisory control for opacity. *IEEE Trans. Autom. Control* **2010**, *55*, 1089–1100. [\[CrossRef\]](#)
2. Takai, S.; Oka, Y. A formula for the supremal controllable and opaque sublanguage arising in supervisory control. *SICE J. Control. Meas. Syst. Integr.* **2008**, *1*, 307–311. [\[CrossRef\]](#)

3. Takai, S.; Watanabe, Y. Modular synthesis of maximally permissive opacity-enforcing supervisors for discrete event systems. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2011**, *E94A*, 1041–1044. [\[CrossRef\]](#)
4. Moulton, R.; Hamgini, B.; Khouzani, Z.; Meira-Goes, R.; Wang, F.; Rudie, K. Using subobservers to synthesize opacity enforcing supervisors. *Discret. Event Dyn. Syst.* **2022**, *32*, 611–640. [\[CrossRef\]](#)
5. Mazare, L. Using unification for opacity properties. In Proceedings of the Workshop on Information Technology&Systems, Las Vegas, LV, USA, 16–18 June 2004; pp. 165–176.
6. Bryans, J.W.; Koutny, M.; Ryan, P.Y. Modelling opacity using petri nets. *Electron. Notes Theor. Comput. Sci.* **2005**, *121*, 101–115. [\[CrossRef\]](#)
7. Lin, F. Opacity of discrete event systems and its applications. *Automatica* **2011**, *47*, 496–503. [\[CrossRef\]](#)
8. Ben-Kalefa, M.; Lin, F. Opaque superlanguages and sublanguagues in discrete event systems. *Cybern. Syst.* **2016**, *47*, 392–426. [\[CrossRef\]](#)
9. Saboori, A.; Hadjicostis, C.N. Notions of security and opacity in discrete event systems. In Proceedings of the IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 5056–5061.
10. Saboori, A.; Hadjicostis, C.N. Verification of k-step opacity and analysis of its complexity. *IEEE Trans. Autom. Sci. Eng.* **2011**, *8*, 549–559. [\[CrossRef\]](#)
11. Falcone, Y.; Marchand, H. Enforcement and validation (at runtime) of various notions of opacity. *Discret. Event Dyn. Syst.* **2015**, *25*, 531–570. [\[CrossRef\]](#)
12. Saboori, A.; Hadjicostis, C.N. Verification of infinite-step opacity and analysis of its complexity. In Proceedings of the IFAC Workshop on Dependable Control of Discrete Systems, Bari, Italy, 10–12 June 2009; pp. 46–51.
13. Wu, Y.C.; Lafortune, S. Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discret. Event Dyn. Syst.* **2013**, *23*, 307–339. [\[CrossRef\]](#)
14. Saboori, A.; Hadjicostis, C.N. Verification of initial-state opacity in security applications of DES. In Proceedings of the International Workshop on Discrete Event Systems, Gothenburg, Sweden, 28–30 May 2008; pp. 328–333.
15. Balun, J.; Masopust, T. Comparing the notions of opacity for discrete-event systems. *Discret. Event Dyn. Syst.* **2021**, *31*, 553–582. [\[CrossRef\]](#)
16. Wintenberg, A.; Blischke, M.; Lafortune, S.; Ozay, N. A general language-based framework for specifying and verifying notions of opacity. *Discret. Event Dyn. Syst.* **2022**, *32*, 253–289. [\[CrossRef\]](#)
17. Ma, Z.; Yin, X.; Li, Z. Verification and enforcement of strong infinite- and k-step opacity using state recognizers. *Automatica* **2021**, *133*, 109838. [\[CrossRef\]](#)
18. Liu, R.; Lu, J. Enforcement for infinite-step opacity and K-step opacity via insertion mechanism. *Automatica* **2022**, *140*, 110212. [\[CrossRef\]](#)
19. Ji, Y.; Wu, Y.C.; Lafortune, S. Enforcement of opacity by public and private insertion functions. *Automatica* **2018**, *93*, 369–378. [\[CrossRef\]](#)
20. Ji, Y.; Yin, X.; Lafortune, S. Enforcing opacity by insertion functions under multiple energy constraints. *Automatica* **2019**, *108*, 108476. [\[CrossRef\]](#)
21. Ji, Y.; Yin, X.; Lafortune, S. Opacity enforcement using nondeterministic publicly-known edit functions. *IEEE Trans. Autom. Control* **2019**, *64*, 4369–4376. [\[CrossRef\]](#)
22. Zhou, Y.; Chen, Z.; Liu, Z.X. Verification and enforcement of current-state opacity based on a state space approach. *Eur. J. Control* **2023**, *71*, 100795. [\[CrossRef\]](#)
23. Wu, Y.C.; Lafortune, S. Synthesis of insertion functions for enforcement of opacity security properties. *Automatica* **2014**, *50*, 1336–1348. [\[CrossRef\]](#)
24. Zhang, B.; Shu, S.L.; Lin, F. Maximum information release while ensuring opacity in discrete event systems. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 1067–1079. [\[CrossRef\]](#)
25. Behinaein, B.; Lin, F.; Rudie, K. Optimal information release for mixed opacity in discrete-event systems. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1960–1970. [\[CrossRef\]](#)
26. Khouzani, Z.A. Optimal Payoff to Ensure Opacity in Discrete-Event Systems. Master's Thesis, Queen's University, Kingston, ON, Canada, 2019.
27. Hou, J.; Yin, X.; Li, S. A framework for current-state opacity under dynamic information release mechanism. *Automatica* **2022**, *140*, 110238. [\[CrossRef\]](#)
28. Sengupta, R.; Lafortune, S. An optimal control theory for discrete event systems. *SIAM J. Control Optim.* **1998**, *36*, 488–541. [\[CrossRef\]](#)
29. Pruekprasert, S.; Ushio, T. Optimal stabilizing supervisor of quantitative discrete event systems under partial observation. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2016**, *99*, 475–482. [\[CrossRef\]](#)
30. Ji, X.; Lafortune, S. Optimal supervisory control with mean payoff objectives and under partial observation. *Automatica* **2021**, *123*, 109359. [\[CrossRef\]](#)
31. Hu, Q.; Yue, W. Two new optimal models for controlling discrete event systems. *J. Ind. Manag. Optim.* **2017**, *1*, 65–80. [\[CrossRef\]](#)

32. Yue, W.; Hu, Q. Optimal control for discrete event systems with arbitrary control pattern. *Discret. Contin. Dyn. Syst. Ser. B (DCDS-B)* **2012**, *6*, 535–558.
33. Cassandras, C.; Lafortune, S. *Introduction to Discrete Event Systems*; Springer: Berlin/Heidelberg, Germany, 2008.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.