PAPER Special Section on Formal Approach

# Optimal Stabilizing Controller for the Region of Weak Attraction under the Influence of Disturbances

**Sasinee PRUEKPRASERT**[†a)], ***Nonmember and*** **Toshimitsu USHIO**[†b)], ***Fellow***

**SUMMARY** This paper considers an optimal stabilization problem of quantitative discrete event systems (DESs) under the influence of disturbances. We model a DES by a deterministic weighted automaton. The control cost is concerned with the sum of the weights along the generated trajectories reaching the target state. The region of weak attraction is the set of states of the system such that all trajectories starting from them can be controlled to reach a specified set of target states and stay there indefinitely. An optimal stabilizing controller is a controller that drives the states in this region to the set of target states with minimum control cost and keeps them there. We consider two control objectives: to minimize the worst-case control cost (1) subject to all enabled trajectories and (2) subject to the enabled trajectories starting by controllable events. Moreover, we consider the disturbances which are uncontrollable events that rarely occur in the real system but may degrade the control performance when they occur. We propose a linearithmic time algorithm for the synthesis of an optimal stabilizing controller which is robust to disturbances.
***key words:*** *stabilization, state attraction, quantitative discrete event systems, state feedback controllers, optimal control*

## 1. Introduction

Supervisory control initiated by Ramadge and Wonham [1], [2] is a formal approach to design of a controller, which is called a supervisor, for discrete event system(DESs) modeled by automata. In their framework, a DES spontaneously and asynchronously generates events which are partitioned into controllable and uncontrollable ones. The supervisory control problem is to design a supervisor that determines enabling or disabling of each controllable event from the observation of system's behavior so that the supervised DES satisfies a control specification given by a language or a predicate. The supervisory control is applied to software adaptation and automatic software development [3], [4], and the formal methods in manufacturing systems [5].

The stabilization problem of DESs, which was introduced in [6], is to synthesize a controller that drives a system from arbitrary initial states to a given set of target states and keeps it there indefinitely. A linear complexity algorithm for computing the region of weak attraction, which is the set of all stabilizable states, was presented in [7]. A slightly different stabilization problem was proposed independently in [8]. In the latter work, the objective is to control the trajectories from arbitrary states to revisit the target states infinitely often. Another notion of stability called language-stability was proposed in [7], where stability of the system is defined in term of its behavior. The stabilization problem under partial observation was studied in [9]. The concept of stabilization can be applied to many fields, including fault-tolerant control [10], the control of reconfigurable manufacturing systems [11], [12], and the control of gene regulatory networks in systems biology [9].

A system may have uncertainty in the sense that some of the events rarely occur. Software and hardware interrupts, and emergency preemptions in embedded systems are examples of the uncertainty. A specification for such an uncertain system is formally described by a modal transition system, where transitions are partitioned into two types: *may* and *must* transitions [13], [14]. A must transition must be allowed by the system while a may transition may or may not be. As a motivational example of the modal transition system, an e-mail system shown in Fig. 1 is considered in [14], where *check* is a may transition and the others are must transitions. Namely, after the email is received, the system may or may not check the email (e.g., for virus) before delivering it to the receiver. A may transition corresponds to a disturbance, which is an uncertain and uncontrollable event, in directed control framework [15]–[18]. We consider the case where the event *deliver* in Fig. 1 is controllable and *check* is a disturbance. Then, the system halts at state $q_1$ if the controller disabled *deliver* and *check* never occurs. Therefore, the controller should enable at least one controllable event at any state such that all uncontrollable events are disturbances.

The above discussion emphasises the control of qualitative DESs where all possible behaviors of the DES are partitioned only into legal and illegal ones. This paper, in contrary, focuses on quantitative DESs which is a more general setting. In a quantitative DES, a measure or a control cost function is assigned on the set of generated trajectories. As a result, the behavior of a DES can be controlled based on a preference and the specification can be formalized more accurately. Optimal control of qualitative DESs has been widely investigated in many literatures under different objectives and assumptions [16]–[25]. The optimal

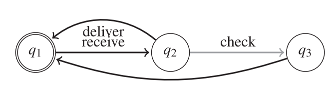



**Fig. 1** A simple email system.

Manuscript received July 14, 2015.
Manuscript revised October 27, 2015.
Manuscript publicized May 2, 2016.
[†]The authors are with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560–8531 Japan.
a) E-mail: sasinee@hopf.sys.es.osaka-u.ac.jp
b) E-mail: ushio@sys.es.osaka-u.ac.jp
DOI: 10.1587/transinf.2015FOP0004

controller problems of qualitative DESs are to synthesize the controller under which the quantitative evaluation of the controlled behaviors is optimized. A quadratic-time complexity algorithm for computing an optimal and minimally restrictive stabilizing controller was proposed in [20]. However, the previous researches on optimal stabilization mainly aim to minimize the worst-case controlled performances.

This paper considers a novel optimal stabilization problem of quantitative discrete event systems under the influence of disturbances. We model a DES by a deterministic quantitative automaton where a cost is assigned to each transition. The control cost is evaluated by the sum of the costs of the generated trajectory reaching a given set of target states. An optimal stabilizing controller is a controller that drives as many states as possible to reach the set of target states with minimum control cost and stay there indefinitely. We consider two control objectives. The first objective is to minimize the worst-case control cost subject to all trajectories. The second one is to minimize the worst-case control cost subject to the enabled trajectories starting by controllable events. In other words, the worst-case control cost of the trajectories following the enabled controllable events is less than or equal to that of any other controllable event. Furthermore, we study the optimal control under the influence of the disturbances.

The remaining of the paper is organized as follows. Section 2 provides the preliminaries. Section 3 formulates the optimal stabilizing control problem. Section 4 proposes an algorithm for the problem. Section 5 presents illustrative examples. Section 6 considers the time complexity and shows the experimental result. Finally, Sect. 7 draws the conclusions.

## 2. Preliminaries

### 2.1 Quantitative Discrete Event Systems

We consider a quantitative discrete event system (DES) modeled by a deterministic weighted automaton $G = (Q, \Sigma = \Sigma_c \cup \Sigma_u, \delta = \delta_c \cup \delta_u, w)$, where $Q$ is a finite set of states, $\Sigma$ is a finite set of events, $\delta \subseteq Q \times \Sigma \times Q$ is a deterministic transition relation, $w : Q \times \Sigma \times Q \to [0, \infty)$ is a weight function which assigns a non-negative cost for each transition. Let $\Sigma(q) = \{\sigma \in \Sigma | \exists q' \in Q, (q, \sigma, q') \in \delta\}$, i.e., the set of active events at the state $q$ in the DES $G$. A state $q$ is said to be *dead* (in $G$) if $\Sigma(q) = \emptyset$. The set of events $\Sigma$ is partitioned into the set of uncontrollable events $\Sigma_u$ and the set of controllable events $\Sigma_c$. The set of transitions is likewise partitioned into the set of uncontrollable transitions $\delta_u \subseteq Q \times \Sigma_u \times Q$, and set of controllable ones $\delta_c \subseteq Q \times \Sigma_c \times Q$.

$r = q_0\sigma_1 q_1 \ldots q_n \in Q(\Sigma Q)^*$ is a *run* generated by $G$ starting from $q \in Q$ if $q_0 = q$ and $(q_i, \sigma_{i+1}, q_{i+1}) \in \delta$ for each $i \in \{0, 1, \ldots, n-1\}$. We say $r$ *visits* $q_i$ for each $i \in \{0, 1, \ldots, n\}$. We also say that $r$ visits, or enters, a set of states $Q' \subseteq Q$ if $r$ visits at least one of the states in $Q'$. $r$ is called a *cyclic run* if it (re)visits a state more than once, i.e., there exist $i, j \in \{0, 1, \ldots, n\}$ such that $i \neq j$ and $q_i = q_j$. $Ru(G, q)$ represents the set of all possible runs generated in $G$ starting from $q \in Q$. For a given set of states $Q' \subseteq Q$, $Ru_{Q'}(G, q)$ is the set of all runs in $G$ that start from $q$ and end right after entering the set $Q'$. Formally, $Ru_{Q'}(G, q) = \{q_0\sigma_1 q_1 \ldots q_n \in Ru(G, q) | q_n \in Q' \text{ and } q_i \notin Q', \forall i \in \{0, 1, \ldots, n-1\}\}$. Moreover, let $Ru^c_{Q'}(G, q) = \{q_0\sigma_1 q_1 \ldots q_n \in Ru_{Q'}(G, q) | \sigma_1 \in \Sigma_c\}$, and $Ru^u_{Q'}(G, q) = \{q_0\sigma_1 q_1 \ldots q_n \in Ru_{Q'}(G, q) | \sigma_1 \in \Sigma_u\}$. In other words, $Ru^c_{Q'}(G, q)$ and $Ru^u_{Q'}(G, q)$ are the sets of runs in $Ru_{Q'}(G, q)$ whose first transition is controllable and uncontrollable, respectively.

$V : \bigcup_{q \in Q} Ru(G, q) \to [0, \infty)$ is a function that represents the control cost for each generated run and is defined as follows: for each run $r = q_0\sigma_1 q_1 \ldots q_n$,

$$V(r) = \begin{cases} 0 & \text{if } r = q_0 \in Q, \\ \sum_{i=0}^{n-1} w(q_i, \sigma_{i+1}, q_{i+1}) & \text{otherwise.} \end{cases}$$

### 2.2 Region of Weak Attraction

Let $E \subseteq Q$ be the set of target states. For a given $\delta' \subseteq \delta$, $E$ is *$\delta'$-invariant* if there does not exist $(q, \sigma, q') \in \delta'$ such that $q \in E$ and $q' \in Q \setminus E$ (i.e., no transition goes out of $E$.). A state $q$ is *strongly attractable* (to $E$) if the following three conditions hold [6].

(a1) $E$ is $\delta$-invariant.
(a2) For any state $q'$ visited by a run in $Ru(G, q)$, $Ru_E(G, q') \neq \emptyset$.
(a3) There is no cyclic run in $Ru_E(G, q)$.

In other words, a state $q$ is strongly attractable (to $E$) if all runs starting from $q$ eventually enter $E$ within a finite number of transitions and indefinitely remain in $E$. A set of states $Q' \subseteq Q$ is strongly attractable to $E$ if all states in $Q'$ are strongly attractable to $E$.

A feedback controller $K$ is a mapping from $Q$ to $2^{\Sigma_c}$ that assigns a set of enabled controllable events $K(q)$ at each state $q$. For each $q \in Q$, $K(q) \subseteq \Sigma_c \cap \Sigma(q)$ is the set of all enabled controllable events at the state $q$. Let $G_K = (Q, \Sigma = \Sigma_c \cup \Sigma_u, \delta_K, w)$ be the DES controlled by the controller $K$ where $\delta_K = \{(q, \sigma, q') \in \delta | \sigma \in K(q) \cup \Sigma_u\}$. No controller can disable any uncontrollable event in $\Sigma_u$.

A state $q \in Q$ is *weakly attractable* (to $E$) if there exists a controller $K$ such that $q$ is strongly attractable to $E$ in $G_K$. We also say that the controller $K$ *stabilizes* the state $q$, and $q$ is stabilized by $K$. The set of all weakly attractable states to $E$ is called the *region of weak attraction* [6]. Note that if the condition (a1) is omitted, the definitions of strongly and weakly attractable states are the same as those of prestable and prestabilizable states proposed by [8], respectively. From the above definitions, we have the following lemma.

**Lemma 1.** *A state $q$ is strongly attractable to $E$ in $G_K$ if and only if $q$ and $G_K$ satisfy the following two conditions:*

1. *$q \in E$ or $q$ is not dead in $G_K$;*
2. *for any transition $(q, \sigma, q') \in \delta_K$, $q'$ is strongly attractable to $E$ in $G_K$.*

## 3. Formulation

### 3.1 Disturbances

In this paper, we consider the control under the influence of disturbances. Disturbances, which was introduced in directed control framework [15]–[18], are uncontrollable events whose occurrences are uncertain in the sense that they rarely occur in the real system. From the practical point of view, a state where all enabled events are disturbances can be considered as a dead state. In other words, we design a controller in such a way that it enables at least one controllable event at any state where all enabled uncontrollable events are disturbances. Moreover, we must consider the effect of the disturbances on the control performance when they occur.

Let $\Sigma_d \subseteq \Sigma_u$ be the set of disturbances. A state $q$ is weakly attractable *under disturbances* if there exists a controller $K$ that satisfies the following two conditions:

1. $K$ stabilizes $q$;
2. for each state $q' \in Q \setminus E$ visited by any run in $Ru_E(G_K, q)$, $K(q') \cap \Sigma(q') \neq \emptyset$ if $\Sigma(q') \cap \Sigma_u \subseteq \Sigma_d$.

We also say the controller $K$ stabilizes the state $q$ under disturbances. The *region of weak attraction under disturbances*, denoted by $\Omega(E)$, is the set of all states that are weakly attractable to $E$ under disturbances. Obviously, if $\Sigma_d = \emptyset$, $\Omega(E)$ is also the region of weak attraction. Let $\mathbb{S}$ be the set of all controllers that stabilize $\Omega(E)$ under disturbances. Note that $\mathbb{S}$ is finite because $Q$ and $\Sigma$ are finite and the number of controllers of $G$ is bounded by $|Q| \cdot 2^{|\Sigma_c|}$.

An optimal control problem for attraction is considered in [20]. A controller achieves optimal attraction if the worst-case cost of the generated runs is minimized. An optimal control problem under the influence of disturbances was also studied in the directed control framework in [16]–[18].

### 3.2 Optimal Attraction under Disturbances

This paper studies non-negative-cost DESs and enriches the optimal control framework for attraction in such a way that, aside from the worst-case-cost runs, the controller also enables some of the other runs that yield a less cost. However, considering the best-case-cost runs is perhaps overly optimistic and not suitable for real systems. Thus we define an optimal stabilizing controller as follows.

For a controller $K$ and a state $q$, let

$$\lambda_K(q) = \begin{cases} \max\limits_{r \in Ru_E(G_K, q)} V(r) & \text{if } Ru_E(G_K, q) \neq \emptyset, \\ \infty & \text{otherwise.} \end{cases}$$

That is, $\lambda_K(q)$ is the worst-case value of runs starting from $q$ to $E$ in the DES controlled by $K$. Similarly, let

$$\lambda_K^c(q) = \begin{cases} \max\limits_{r \in Ru_E^c(G_K, q)} V(r) & \text{if } Ru_E^c(G_K, q) \neq \emptyset, \\ \infty & \text{otherwise,} \end{cases}$$

and

$$\lambda_K^u(q) = \begin{cases} \max\limits_{r \in Ru_E^u(G_K, q)} V(r) & \text{if } Ru_E^u(G_K, q) \neq \emptyset, \\ \infty & \text{otherwise.} \end{cases}$$

An *optimal stabilizing controller* is a controller $K \in \mathbb{S}$ such that for any $K' \in \mathbb{S}$ and $q \in \Omega(E) \setminus E$, the following two conditions hold.

(b1) $\lambda_K(q) \leq \lambda_{K'}(q)$.
(b2) If $K(q) \neq \emptyset$, then $\lambda_K^c(q) \leq \lambda_{K'}^c(q)$.

The *optimal stabilizing control problem* is to design an optimal stabilizing controller.

The condition (b2) can be interpreted as minimizing the worst-case control cost subject to the runs starting from each state $q$ by the occurrence of an enabled controllable event. At the state $q$, an optimal stabilizing controller $K$ may enable only controllable transitions such that the value $\lambda_K^c(q)$ is minimized and lower than or equal to $\lambda_K(q)$. Note that the (conventional) optimal stabilization problem in [20] requires only condition (b1) and does not consider the notion of disturbances. The algorithm proposed in [20] computes the region of weak attraction and a stabilizing controller within $O(|Q|^2)$.

Obviously, if all events are controllable ($\Sigma_u = \Sigma_d = \emptyset$), the optimal stabilizing control problem is the single-source shortest path problem in directed graphs with non-negative costs which can be solved by several algorithms, including Dijkstra's algorithm [26]. Based on this concept, we propose an $O(|Q| \cdot log|Q| + |\delta|)$ algorithm to solve the optimal stabilizing control problem in the next section. For the system where the number of events is insignificantly small, the time complexity of the proposed algorithm is $O(|Q| \cdot log|Q|)$.

## 4. An Algorithm for the Computation of an Optimal Stabilizing Controller

If the target set $E$ is $\delta_u$-invariant, a controller $K$ may disable all active controllable events at the states in $E$ so that $E$ becomes $\delta_K$-invariant in $G_K$. If the target set $E$ is empty or not $\delta_u$-invariant, all states are obviously neither strongly nor weakly attractable to $E$, and $\Omega(E)$ is empty. We therefore assume that $q_e$ is the only target state and dead, i.e., $E = \{q_e\}$ and $\Sigma(q_e) = \emptyset$. In such cases where there are more than one target states or the target states are not dead (but $\delta_u$-invariant), we can replace all target states by $q_e$ and replace each incoming transition to target states by a transition to $q_e$ with the same event within $O(|Q| + |\delta|)$. Since $|\delta| \leq |Q| \cdot |\Sigma|$, this replacement can be done in $O(|Q| \cdot |\Sigma|)$. Based on these assumptions, we propose Algorithm 1 which computes the region of weak attraction under disturbances along with an optimal controller.

**Algorithm 1** optimal stabilizing controller $K^h$ , $\Omega(E)$

**Require:** $G = (Q, \Sigma = \Sigma_u \cup \Sigma_c, \delta = \delta_u \cup \delta_c, w), E = \{q_e\}$

1: $V_c(q_e) \leftarrow V_u(q_e) \leftarrow 0, K(q_e) \leftarrow \emptyset, deg_u(q_e) \leftarrow 0, chk_c(q_e) \leftarrow false$

2: $Q_d \leftarrow \{q \in Q | \Sigma_u \cap \Sigma(q) \neq \emptyset$ and $\Sigma_u \cap \Sigma(q) \subseteq \Sigma_d\}$

3: **for all** $q \in Q \setminus \{q_e\}$ **do**

4: $\quad V_c(q) \leftarrow V_u(q) \leftarrow \infty, K(q) \leftarrow \emptyset$

5: $\quad deg_u(q) \leftarrow |\Sigma(q) \cap \Sigma_u|, chk_c(q) \leftarrow false$

6: **end for**

7: $H^0 \leftarrow \{q_e\}, N^0 \leftarrow Q \setminus \{q_e\}, h \leftarrow 0$

8: **while** $H^h \neq \emptyset$ **do**

9: $\quad U \leftarrow \emptyset, h \leftarrow h + 1$

10: $\quad$ select $q^h \in \arg\min_{q \in H^{h-1}}$ MAX-REAL$(V_u(q), V_c(q))$

11: $\quad V_\lambda(q^h) \leftarrow$ MAX-REAL$(V_u(q), V_c(q))$

12: $\quad$ **for all** $(q, \sigma_u, q^h) \in \delta_u$ such that $q \in N^{h-1} \cup H^{h-1} \setminus \{q^h\}$ **do**

13: $\quad\quad deg_u(q) \leftarrow deg_u(q) - 1$

14: $\quad\quad$ **if** $deg_u(q) = 0$ and ($q \notin Q_d$ or $chk_c(q)$) **then**

15: $\quad\quad\quad U \leftarrow U \cup \{q\}$.

16: $\quad\quad$ **end if**

17: $\quad\quad V_u(q) \leftarrow$ MAX-REAL$(V_u(q), V_\lambda(q^h) + w(q, \sigma_u, q^h))$

18: $\quad\quad$ **if** $V_c(q) > V_u(q)$ and $q \notin Q_d$ **then**

19: $\quad\quad\quad V_c(q) \leftarrow V_u(q), K(q) \leftarrow \emptyset$

20: $\quad\quad$ **end if**

21: $\quad$ **end for**

22: $\quad$ **for all** $(q, \sigma_c, q^h) \in \delta_c$ such that $q \in N^{h-1} \cup H^{h-1} \setminus \{q^h\}$ **do**

23: $\quad\quad$ **if** $q \in Q_d$ and $\neg chk_c(q)$ **then**

24: $\quad\quad\quad V_c(q) \leftarrow V_\lambda(q^h) + w(q, \sigma_c, q^h)$

25: $\quad\quad\quad K(q) \leftarrow \{\sigma_c\}, chk_c(q) \leftarrow true$

26: $\quad\quad$ **else if** $V_c(q) > V_\lambda(q^h) + w(q, \sigma_c, q^h)$ **then**

27: $\quad\quad\quad V_c(q) \leftarrow V_\lambda(q^h) + w(q, \sigma_c, q^h), K(q) \leftarrow \{\sigma_c\}$

28: $\quad\quad$ **else if** $V_c(q) = V_\lambda(q^h) + w(q, \sigma_c, q^h)$ **then**

29: $\quad\quad\quad K(q) \leftarrow K(q) \cup \{\sigma_c\}$

30: $\quad\quad$ **end if**

31: $\quad\quad$ **if** $deg_u(q) = 0$ and ($q \notin Q_d$ or $chk_c(q)$) **then**

32: $\quad\quad\quad U \leftarrow U \cup \{q\}$.

33: $\quad\quad$ **end if**

34: $\quad$ **end for**

35: $\quad H^h \leftarrow H^{h-1} \setminus \{q^h\} \cup U$

36: $\quad N^h \leftarrow N^{h-1} \setminus U$

37: **end while**

38: **return** $K, Q \setminus N^h$

39: **function** MAX-REAL$(V_1, V_2)$

40: $\quad$ **if** $V_1 < V_2 < \infty$ or $V_1 = \infty$ **then return** $V_2$

41: $\quad$ **else return** $V_1$ **end if**

42: **end function**

Algorithm 1 first considers three sets of states: $Q_d$, $H^0$, and $N^0$. $Q_d$ is the set of all states from which all outgoing uncontrollable events are disturbances. We thus need to ensure that the controller enables at least one controllable event at each state in this set. $H^0$ is the target set that is certainly weakly attractable to itself. $N^0 = Q \setminus E$ is the set of the rest of the states in the system. Then, in the $h$-th iteration of the while loop between lines 8 and 37, the algorithm computes the following sets of states.

- $N^h$ is a set of states that are not yet determined to be weakly attractable to $E$ under disturbances.
- $H^h$ is a set of weakly attractable states where the control action has not been determined.
- $Q \setminus (N^h \cup H^h) = \{q^1, q^2, \ldots, q^h\}$ is a set of weakly attractable states where the control action has been determined by the end of the $h$-th iteration.
- $U$ is a temporary set for transferring the weakly attractable states from $N^h$ to $H^h$.

In the $h$-th iteration of the while loop between lines 8 and 37, $q^h$ is selected from $H^{h-1}$, and only states in $N^{h-1}$ can be transferred to $H^h$. At the end of the $h$-th iteration of the while loop, $H^h \cap N^h = \emptyset$, $\{q^1, q^2, \ldots, q^h\} = Q \setminus (H^h \cup N^h)$, and $Q = \{q^1, q^2, \ldots, q^h\} \cup H^h \cup N^h$. The number of the states is finite and the algorithm exits the while loop when $H^h = \emptyset$. Hence, the algorithm surely terminates after a finite number of iterations of the while loop. We suppose that the algorithm terminates after the $\ell$-th iteration. For each $h \in \{1, 2, \ldots, \ell\}$, we have

$$Q \setminus \{q^1, q^2, \ldots, q^h\} = H^h \cup N^h \text{ and } H^h \cap N^h = \emptyset. \quad (1)$$

In the $h$-th iteration of the while loop, the for loop between lines 12 and 21 (*resp.* lines 22 and 34) considers each transition $(q, \sigma, q^h) \in \delta_u$ (*resp.* $\delta_c$) such that $\sigma \in \Sigma$ and $q \in N^{h-1} \cup H^{h-1} \setminus \{q^h\}$. The algorithm then computes $V_c(q)$, $V_u(q)$, $K(q)$, $deg_u(q)$, and $chk_c(q)$ for the state $q$. $V_u(q)$ is used to calculate $\min_{K \in \mathbb{S}} \lambda^u_K(q)$. $V_c(q)$ is used to calculate $\min_{K \in \mathbb{S}} \lambda^c_K(q)$ if $q \in Q_d$; and to calculate $\min_{K \in \mathbb{S}} \min\{\lambda^u_K(q), \lambda^c_K(q)\}$ if $q \notin Q_d$. $K(q)$ is used to compute an optimal stabilizing controller. $deg_u(q)$ is the number of outgoing uncontrollable transitions from $q$ to $Q \setminus \{q^1, q^2, \ldots, q^h\}$. If $deg_u(q) = 0$, all outgoing uncontrollable transitions from $q$ lead to $\{q^1, q^2, \ldots, q^h\}$. If $q \in Q_d$, $chk_c(q)$ indicates whether or not the controller $K$ has already enabled at least one outgoing controllable transition from $q$ to a weakly attractable state in $Q \setminus (N^h \cup H^h)$. The algorithm need to ensure that $chk_c(q)$ is true if $q \in Q_d$.

Let $V^h_c(q)$, $V^h_u(q)$, $K^h(q)$, $deg^h_u(q)$, and $chk^h_c(q)$ represent the corresponding variables at the end of the $h$-th round of the while loop. Then, we have the following lemmas.

**Lemma 2.** *For any $h \in \{2, \ldots, \ell\}$ and any $i \in \{h, h+1, \ldots, \ell\}$, $V^i_c(q^h) = V^{h-1}_c(q^h)$, $V^i_u(q^h) = V^{h-1}_u(q^h)$, $K^i(q^h) = K^{h-1}(q^h)$, $deg^i_u(q^h) = deg^{h-1}_u(q^h)$ and $chk^i_c(q^h) = chk^{h-1}_c(q^h)$.*

*Proof.* In the $h$-th iteration of the while loop, the for loop between lines 12 and 21 (*resp.* lines 22 and 34) considers each transition $(q, \sigma, q^h) \in \delta_u$ (*resp.* $\delta_c$) such that $\sigma \in \Sigma$ and $q \in N^{h-1} \cup H^{h-1} \setminus \{q^h\}$. From Eq. (1), $N^{h-1} \cup H^{h-1} \setminus \{q^h\} = (N^{h-1} \cup H^{h-1}) \setminus \{q^1, q^2, \ldots, q^h\}$. Thus, for each $i \in \{h, h+1, \ldots, \ell\}$, the variables $V_c(q^h)$, $V_u(q^h)$, $K(q^h)$, $deg_u(q^h)$, and $chk_c(q^h)$ are not updated in the $i$-th iteration of the while loop between lines 8 and 37. □

**Lemma 3.** *Consider each $h \in \{1, 2, \ldots, \ell\}$ and $q \in Q \setminus \{q^1, q^2, \ldots, q^h\}$. The following conditions hold at the end of the h-th round of the while loop between lines 8 and 37.*

1. *If there exists $i \leq h$ such that $(q, \sigma, q^i) \in \delta_u$,*

$$V^h_u(q) = \max_{\substack{(q,\sigma,q^i) \in \delta_u \\ i \in \{1,2,\ldots,h\}}} (w(q, \sigma, q^i) + V_\lambda(q^i)).$$

*Otherwise, $V^h_u(q) = \infty$.*

2. *If $q \in Q_d$ and there exists $i \leq h$ such that $(q, \sigma, q^i) \in \delta_c$,*

$$V^h_c(q) = \min_{\substack{(q,\sigma,q^i) \in \delta_c \\ i \in \{1,2,\ldots,h\}}} (w(q, \sigma, q^i) + V_\lambda(q^i)).$$

*If $q \notin Q_d$ and there exists $i \leq h$ such that $(q, \sigma, q^i) \in \delta_c$,*

$$V_c^h(q) = \min \begin{cases} \max\limits_{\substack{(q,\sigma,q^i)\in\delta_u \\ i\in\{1,2,\ldots,h\}}} (w(q,\sigma,q^i) + V_\lambda(q^i)) \\ \min\limits_{\substack{(q,\sigma,q^i)\in\delta_c \\ i\in\{1,2,\ldots,h\}}} (w(q,\sigma,q^i) + V_\lambda(q^i)). \end{cases}$$

*Otherwise, $V_c^h(q) = \infty$.*

3. *$\sigma \in K^h(q)$ if and only if there exists $(q, \sigma, q') \in \delta_c$ such that $q' \in \{q^1, q^2, \ldots q^h\}$ and $w(q, \sigma, q') + V_\lambda(q') = V_c^h(q)$.*
4. *$deg_u^h(q)$ is the number of outgoing uncontrollable transitions from $q$ to states in $Q \setminus \{q^1, q^2, \ldots, q^h\}$.*
5. *If $q \in Q_d$, $chk_c^h(q)$ is true if and only if the controller $K^h$ enables at least one controllable transition from $q$ to a state in $\{q^1, q^2, \ldots, q^h\}$.*

*Proof. Base Case.* Obviously, $q^1 = q_e$ and $V_c(q^1) = V_u(q^1) = 0$. For each $q \in Q \setminus \{q^1\}$, the algorithm initially set $V_c(q) = V_u(q) = \infty$, $K(q) = \emptyset$, $deg_u(q) = |\Sigma(q) \cap \Sigma_u|$, and $chk_c(q) = false$. In the first iteration of the while loop, the for loop between lines 12 and 21 (*resp.* lines 22 and 34) considers each transition $(q, \sigma, q^1) \in \delta_u$ (*resp.* $\delta_c$) such that $\sigma \in \Sigma$ and $q \in Q \setminus \{q^1\}$. It can be easily shown that conditions 1 - 5 hold for any state $q \in Q \setminus \{q^1\}$.

*Inductive Hypothesis.* Consider $h \in \{1, 2, \ldots, \ell - 1\}$. Assume that conditions 1 - 5 hold for any $q \in Q \setminus \{q^1, q^2, \ldots, q^h\}$ at the end of the $h$-th iteration of the while loop.

*Inductive Step.* From Eq. (1), $Q \setminus \{q^1, q^2, \ldots, q^h\} = H^h \cup N^h$ and $H^h \cap N^h = \emptyset$. The state $q^{h+1}$ is selected from $H^h$. The for loop between lines 12 and 21 considers each transition $(q, \sigma, q^{h+1}) \in \delta_u$ such that $q \in Q \setminus \{q^1, q^2, \ldots, q^{h+1}\}$. $deg_u(q)$ and $V_u(q)$ can be updated at lines 13 and 17, respectively. From the inductive hypothesis, conditions 1 and 4 hold for $q$ after the $(h+1)$-st iteration of the while loop.

Then, we consider conditions 2, 3, and 5. First, consider the case where $q \notin Q_d$. Since $q \notin Q_d$, condition 5 holds. If $V_c(q) > V_u(q)$, the algorithm sets $V_c(q) = V_u(q) = \max\limits_{\substack{(q,\sigma,q^i)\in\delta_u \\ i\in\{1,2,\ldots,h+1\}}} (w(q, \sigma, q^i) + V_\lambda(q^i))$ and $K(q) = \emptyset$ at line 19. Then, the for loop between lines 22 and 34 considers each $(q, \sigma, q^{h+1}) \in \delta_c$ such that $q \in Q \setminus \{q^1, q^2, \ldots, q^{h+1}\}$ and updates the values $V_c(q)$ and $K(q)$. From the inductive hypothesis, conditions 2 and 3 hold for $q$ after the $(h+1)$-st iteration of the while loop.

Next, consider the case where $q \in Q_d$. The for loop between lines 22 and 34 considers each $(q, \sigma, q^{h+1}) \in \delta_c$ such that $q \in Q \setminus \{q^1, q^2, \ldots, q^{h+1}\}$ and updates the variables $V_c(q)$, $K(q)$, and $chk_c(q)$. If $V_c(q)$ is updated at line 24, $chk_c(q) = false$ at line 23. From the inductive hypothesis, $(q, \sigma, q^{h+1})$ is the first controllable transition from $q$ to $\{q^1, q^2, \ldots, q^{h+1}\}$ that is considered by Algorithm 1. Hence, $K$ enables $(q, \sigma, q^{h+1})$ and $chk_c(q)$ is set to be true at line 25. Otherwise, $V_c(q)$ can be updated at line 27, and $K(q)$ can be updated at lines 27 and 29. Then, from the inductive hypothesis, conditions 2, 3, and 5 hold for $q$ after the $(h+1)$-st iteration of the while loop. □

Next, we show that all states in $Q \setminus N^\ell$ are weakly attractable to $E$ and are stabilized by $K^\ell$ under disturbances.

**Lemma 4.** *$Q \setminus N^\ell$ is strongly attractable to $E$ in $G_{K^\ell}$.*

*Proof. Base Case.* The target state $q^1 = q_e$ is strongly attractable to $E$ in $G_{K^\ell}$.

*Inductive Hypothesis.* Consider $h \in \{1, 2, \ldots, \ell - 1\}$. Assume all states in $\{q^1, q^2, \ldots, q^h\}$ are strongly attractable to $E$ in $G_{K^\ell}$.

*Inductive Step.* Since $q^{h+1}$ is selected from $H^h$, there exist $j \leq h$ and $\sigma \in \Sigma$ such that $(q^{h+1}, \sigma, q^j) \in \delta_{K^j}$ where $q^{h+1}$ is added in $H^j$ in the $j$-th iteration of the while loop. From condition 3 of Lemma 3, there is no transition in $\delta_{K^h} \cap \delta_c$ from $q^{h+1}$ to $Q \setminus \{q^1, q^2, \ldots, q^h\}$. Since $q^{h+1}$ is added in $H^j$, $deg_u^j(q^{h+1}) = 0$. From Lemma 2, $\mathrm{deg}_u^h(q^{h+1}) = 0$. From condition 4 of Lemma 3, there is no transition in $\delta_{K^h} \cap \delta_u$ from $q^{h+1}$ to $Q \setminus \{q^1, q^2, \ldots, q^h\}$. Therefore, there is no transition in $\delta_{K^h}$ from $q^{h+1}$ to $Q \setminus \{q^1, q^2, \ldots, q^h\}$.

From conditions 1 and 2 of Lemma 3, $V_u^h(q^{h+1}) < \infty$ or $V_c^h(q^{h+1}) < \infty$. If $V_u^h(q^{h+1}) < \infty$, there exists an uncontrollable transition from $q^{h+1}$ to $\{q^1, q^2, \ldots, q^h\}$ which cannot be disabled by $K^h$. If $V_c^h(q^{h+1}) < \infty$, from condition 3 of Lemma 3, $K^h$ enables a controllable transition from $q^{h+1}$ to $\{q^1, q^2, \ldots, q^h\}$. Thus, for both cases, there is at least one transition in $\delta_{K^h}$ from $q^{h+1}$ to $\{q^1, q^2, \ldots, q^h\}$ and there is no transition from $q^{h+1}$ to $Q \setminus \{q^1, q^2, \ldots, q^h\}$. From Lemma 2, the transition from $q^{h+1}$ to $\{q^1, q^2, \ldots, q^h\}$ is also included in $\delta_{K^\ell}$. From the inductive hypothesis, all states in $\{q^1, q^2, \ldots, q^h\}$ are strongly attractable to $E$ in $G_{K^\ell}$. Thus, $q^{h+1}$ is strongly attractable to $E$ in $G_{K^\ell}$ by Lemma 1. □

**Lemma 5.** *$K^\ell$ stabilizes $Q \setminus N^\ell$ under disturbances.*

*Proof.* From Lemma 4, $K^\ell$ stabilizes the set $Q \setminus N^\ell = \{q^1, q^2, \ldots, q^\ell\}$. For each $i \in \{1, 2, \ldots, \ell\}$, since $q^i$ is selected from $H^{i-1}$, there exists $j \leq i - 1$ such that $q^i$ is added in $H^j$. Hence, $deg_u^j(q^i) = 0$. From Lemma 2, and conditions 2 and 3 of Lemma 3, there is no run in $G_{K^\ell}$ starting from a state $q \in Q \setminus N^\ell$ that visits $N^\ell$.

As a result, it is sufficient to show that $K^\ell(q^i) \neq \emptyset$ for each $i \in \{2, 3, \ldots, \ell\}$ such that $q^i \in Q_d$. Recall that there exists $j \leq i - 1$ where $q^i$ is added in $H^j$. Hence, $chk_c^j(q^i)$ is true. From Lemma 2 and conditions 2, 3, and 5 of Lemma 3, $chk_c^\ell(q^i)$ is true and $K^\ell(q^i) \neq \emptyset$. Thus, this lemma holds. □

The following theorem shows that $Q \setminus N^\ell$ is the region of weak attraction under disturbances.

**Theorem 6.** *$Q \setminus N^\ell = \Omega(E)$, and is stabilized by $K^\ell$ under disturbances.*

*Proof.* From Lemma 5, $Q \setminus N^\ell \subseteq \Omega(E)$ and $Q \setminus N^\ell$ is stabilized by $K^\ell$. Suppose that $\Omega(E) \nsubseteq Q \setminus N^\ell$, that is, there exists a state $q \in \Omega(E) \cap N^\ell$ which is stabilized by a controller $K' \in \mathbb{S}$ under disturbances. For each $i \in \{1, 2, \ldots, \ell\}$, $q$ is not included in $H^i$. Therefore, the state $q$ satisfies any of the following two conditions.

1. For any $\sigma \in \Sigma(q)$ and any $i \in \{1, 2, \ldots, \ell\}$, $(q, \sigma, q^i) \notin \delta$.
2. $deg_u^\ell(q) > 0$ or ($q \in Q_d$ and $chk_c^\ell(q)$ is false).

From Lemma 3, if $deg_u^\ell(q) > 0$, there is at least one uncontrollable transition from $q$ to a state in $N^\ell$ which cannot be disabled by the controller $K'$. If $q \in Q_d$ and $chk_c^\ell(q)$ is false, from condition 5 of Lemma 3, there is no controllable transition from $q$ to $Q \setminus N^\ell$. For this case, the controller $K'$ must enable at least one transition from $q \in Q_d$ to a state in $N^\ell$. Recall that $E = \{q^1\} \subseteq Q \setminus N^\ell$. Thus, for each $q \in \Omega(E) \cap N^\ell$, either $Ru_E(G_{K'}, q) = \emptyset$ or there exists a cyclic run in $Ru_E(G_{K'}, q)$ that visits at least one state in $N^\ell$ more than once. Therefore, $K'$ does not stabilize $q$ which is a contradiction. □

Then, we show that $K^\ell$ is an optimal stabilizing controller.

**Lemma 7.** *For each* $q^h \in \{q^2, q^3, \ldots, q^\ell\}$, $V_\lambda(q^h) = \min_{K \in \mathbb{S}} \lambda_K(q^h) = \lambda_{K^\ell}(q^h)$ *and* $V_\lambda(q^h) \geq V_\lambda(q^{h-1})$.

*Proof. Base Case.* Since $q^1 = q_e \in E$, $\lambda_K(q^1)$ is zero for any controller $K$. The algorithm accordingly assigns $V_\lambda(q^1) = \text{MAX-REAL}(V_u(q^1), V_c(q^1)) = 0$. In the 2nd iteration of the while loop, we have $V_\lambda(q^2) = \text{MAX-REAL}(V_u^h(q^2), V_c^h(q^2))$ and $q^2 \in \arg\min_{q' \in H^1} \text{MAX-REAL}(V_u^2(q'), V_c^2(q'))$. Since the costs of all transitions are non-negative, from Lemma 2 and conditions 1-3 of Lemma 3, this lemma holds for $q^2$.

*Inductive Hypothesis.* Consider $h \in \{2, 3, \ldots, \ell-1\}$. Suppose that for each $i \in \{2, \ldots, h\}$, this lemma holds for $q^i$.

*Inductive Step.* From Algorithm 1, we have $V_\lambda(q^{h+1}) = \text{MAX-REAL}(V_u^h(q^{h+1}), V_c^h(q^{h+1}))$ and $q^{h+1} \in \arg\min_{q' \in H^h} \text{MAX-REAL}(V_u^h(q'), V_c^h(q'))$. Since the costs of all transitions are non-negative, from the inductive hypothesis, Lemma 2, and conditions 1-3 of Lemma 3, this lemma holds for $q^{h+1}$. □

Lemma 7 implies the condition (b1) of an optimal stabilizing controller. $V_\lambda(q^h) = \lambda_{K^\ell}(q^h)$ represents the worst-case cost from any state $q^h$ under the controller $K^\ell$. Then, we show the condition (b2).

**Lemma 8.** *For each* $K' \in \mathbb{S}$ *and each* $q^h \in \{q^2, \ldots, q^\ell\}$ *such that* $K^\ell(q^h) \neq \emptyset$, $\lambda_{K^\ell}^c(q) \leq \lambda_{K'}^c(q)$.

*Proof.* From Theorem 6 and Lemma 7, we have $V_\lambda(q^1) = \min_{K \in \mathbb{S}} \lambda_K(q^1) \leq V_\lambda(q^2) = \min_{K \in \mathbb{S}} \lambda_K(q^2) \leq \ldots \leq V_\lambda(q^\ell) = \min_{K \in \mathbb{S}} \lambda_K(q^\ell)$. Since the costs of all transitions are non-negative, this Lemma holds by condition 2 of Lemma 3. □

From Lemmas 7 and 8, we have the following theorem.

**Theorem 9.** *Algorithm 1 computes the region of weak attraction under disturbances* $Q \setminus N^\ell$ *and an optimal stabilizing controller* $K^\ell$.

## 5. Illustrative Examples

### 5.1 A Simple DES

We show the result from applying Algorithm 1 to the DES in Fig. 2. Initially, $Q_d = \{q^4, q^6, q^8\}$, $H^0 = \{q_e\}$, and $N^0 = \{q^2, q^3, \ldots, q^8\}$. Then, the result after each iteration of the while loop is as follows.

- the 1st iteration
  - $V_\lambda(q_e) = 0$, $H^1 = \{q^2, q^3\}$
  - $V_u^1(q^2) = V_u^1(q^3) = V_u^1(q^4) = 1$
  - $V_c^1(q^3) = 1$, $K^1(q^3) = \{c3\}$
  - $V_c^1(q^6) = 2$, $K^1(q^6) = \{c6\}$, $chk_c^1(q^6) = true$
- the 2nd iteration
  - $V_\lambda(q^2) = 1$, $H^2 = \{q^3, q^4\}$
  - $V_c^2(q^4) = 2$ , $K^2(q^4) = \{c4\}$, $chk_c^2(q^4) = true$
  - $V_c^2(q^3) = 1$ , $K^2(q^3) = \{c3, c3'\}$
- the 3rd iteration
  - $V_\lambda(q^3) = 1$, $H^3 = \{q^4\}$
  - $V_u^3(q^8) = 1$, $V_c^3(q^6) = 1$, $K^3(q^6) = \{c6'\}$
  - $V_c^3(q^7) = 4$, $K^3(q^7) = \{c7\}$
- the 4th iteration
  - $V_\lambda(q^4) = 2$, $H^4 = \{q^5, q^6\}$
  - $V_u^4(q^6) = 7$, $deg_u^4(q^6) = 0$
  - $V_c^4(q^5) = 7$, $K^4(q^5) = \{c5\}$
- the 5th iteration
  - $V_\lambda(q^5) = 7$, $H^5 = \{q^6\}$, $N^5 = \{q^7, q^8\}$
- the 6th iteration
  - $V_\lambda(q^6) = 7$, $H^6 = \emptyset$, $N^6 = \{q^7, q^8\}$

Algorithm 1 returns the set $Q \setminus N^6 = \{q_e, q^2, q^3, q^4, q^5, q^6\}$ and the controller $K^6$ given by $K^6(q_e) = K^6(q^2) = \emptyset$, $K^6(q^3) = \{c3, c3'\}$, $K^6(q^4) = \{c4\}$, $K^6(q^5) = \{c5\}$, $K^6(q^6) = \{c6'\}$, $K^6(q^7) = \{c7\}$. Notice that although disabling $c4$ at $q^4$ decreases $V_\lambda(q^4)$, $c4$ is enabled since $d4$ is a disturbance.

The result from our algorithm can be compared with ones of the previous works as follows. In the directed control framework [15]–[18], the controller enables at most one controllable event at $q^3$, i.e., either $c3$ or $c3'$, but not both. In the (conventional) optimal stabilization framework that does not restrict the condition (b2) [6], the controller $K^6$ may enable both $c6$ and $c6'$, because the worst-case value of the runs following from both events are still lower than that of $d6$. Besides, since disturbances are not considered, $c4$ and $c7$ are disabled in order to minimize the values $V_\lambda(q^4)$ and $V_\lambda(q^7)$. $q^7$ and $q^8$ would then become weakly attractable to $E$. In our setting, we need to disable $c6$ since the worst-case

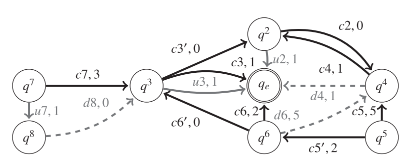



**Fig. 2** A DES modeled by an automaton. The labels of the transitions are events and their costs. $\Sigma_c = \{c2, c3, c3', c4, c5, c5' c6, c6', c7\}$, $\Sigma_u = \{u2, u3, d4, d6, u7, d8\}$, and $\Sigma_d = \{d4, d6, d8\}$.

value of runs after its occurrence is higher than that of $c6'$. Note that if the disturbance $d6$ does not occur at $q^6$, disabling $c6$ decreases the worst-case control cost. In addition, we may enable both $c3$ and $c3'$ since the worst-case value of the controlled runs following from them is minimal.

### 5.2 Automated Guided Vehicles (AGVs)

We consider a system with two automated guided vehicles (AGVs) in Fig. 3(a). The goal of the first and the second AGVs are to reach stations A and C without collision, respectively. The black transitions represent controllable routes that can be enable and disabled by the controller. The two gray transitions are uncontrollable routes. The dashed gray transition represents an emergency route at station E. This emergency route may be unpredictably enabled by the local controller at station E; therefore, it is a disturbance. The cost of each transition represents the transit time.

The automaton in Fig. 3(b) represents all possible routes of the system in Fig. 3(a) as follows: each state $ij$ of Fig. 3(b) represents when the first vehicle is in the station $i$ and the second vehicle is in the station $j$ of Fig. 3(a). The dead states AA, BB, CC, DD , and EE represent the collisions. The target state AC is also dead. Then, our objective is to find a minimally restrictive optimal controller that drives both vehicles to the state AC of Fig. 3(b). By applying Algorithm 1 to the automaton in Fig. 3(b), we obtain the controlled system as shown in Fig. 3(c).

For example, consider the controlled system starting from the state ED in Fig. 3(c). The controllable transition from ED to BD is enabled because the other transition is a disturbance. Both transitions from ED to BD allow the first vehicle to move from station E to station B in Fig. 3(a). Then, the first vehicle may move to station A directly, or move to station C first and then to station A. In either way, the system reaches the state AD with the same cost. Finally, the second vehicle moves to station C and the controlled system reaches the target state AC.

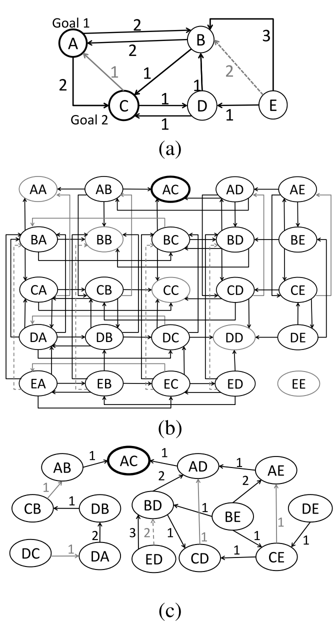



**Fig. 3** (a) An example of a traffic system. (b) An automaton representing possible routes of the system (a). (c) An optimal controlled system.

## 6. Computational Time Complexity

In this section, we discuss the computational time complexity of Algorithm 1. In the algorithm, line 1 initializes the variables in $O(1)$. Line 2 computes $Q_d$ in $O(|Q| + |\delta|)$. The for loop between lines 3 and 6 is also $O(|Q| + |\delta|)$. From Eq. (1), Algorithm 1 terminates within $\ell \leq |Q|$ iterations of the while loop between lines 8 and 37. In the $h$-th iteration of the while loop, $h \in \{1, 2, \ldots, \ell\}$, the state $q^h$ with minimal $\textsc{max-real}(V_u(q^h), V_c(q^h))$ is selected from $H^{h-1}$. Then, the for loops lines 12–21 and 22–34 consider each transition in $\delta$ at most once. For each $q \in H^h$, $V_c(q)$ is possibly updated at line 27. This update may decrease the value $\textsc{max-real}(V_u(q), V_c(q))$. At lines 35 and 36, $q^h$ is deleted from $H$ and the states in $U$ are moved from $N$ to $H$. Note that, in any iteration of the while loop, $|U|$ is less than or equal to the number of iterations considered in the for loops between lines 12–21 and 22–34.

We can implement the set $H$ using a priority queue which provides the following operations: select and delete a state in $\arg\min_{q \in H} \textsc{max-real}(V_u(q), V_c(q))$, insert a state $q$ in $H$, and decrease the values $V_c(q)$ and $\textsc{max-real}(V_u(q), V_c(q))$ for a state $q \in H$. Using strict Fibonacci heap [27], the selection, insertion, and decreasing the values are $O(1)$; and the deletion is $O(log|Q|)$. Thus, the over all complexity of the while loop, as well as Algorithm 1, is $O(|Q| \cdot log|Q| + |\delta|)$. Note that in the real considered system, the number of the states is often far larger than the number of the events. If $|\Sigma|$ is insignificantly small and dominated by $log|Q|$, the complexity of Algorithm 1 is $O(|Q| \cdot log|Q|)$.

We implement our algorithm to confirm its efficiency. The computation environment is Windows 7 Enterprise Service Pack 1, on AMD Phenom(tm) II X6 1090T Processor 3.20 GHz with 8.00 GB memory. We consider systems with 100,000 to 1,000,000 connected states. The numbers of controllable transitions, uncontrollable transitions, and disturbances are 5, 0.01, and 0.002 times the number of states, respectively. Shown in Fig. 4 is a relation between the average computation time of Algorithm 1 and the number of states. This result confirms that the time complexity is linearithmic.

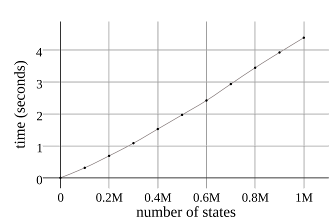



**Fig. 4** Relation between the average computation time of Algorithm 1 and the number of states.

## 7. Conclusions

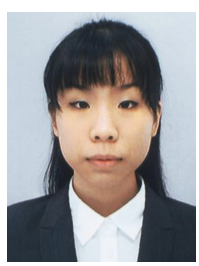
This paper formulates a novel optimal stabilization problem of quantitative discrete event systems under the influence of disturbances, which are uncontrollable events whose occurrences are uncertain in the sense that they rarely occur in the real system. The objective is to compute a region of weak attraction under disturbances, along with an optimal stabilizing controller that minimizes not only the worst-case control cost subject to all enabled runs but also the worst-case control cost subject to all enabled runs starting by controllable events. We propose an $O(|Q| \cdot log|Q| + |\delta|)$ algorithm for the problem. If $|\Sigma|$ is insignificantly small and dominated by $log|Q|$, the complexity is $O(|Q| \cdot log|Q|)$. It is future work to extend the proposed algorithms to the other control problems such as control under partial observation.

## Acknowledgments

This research was supported in part by JSPS KAKENHI Grant Number 24656262.

### References

[1] P.J. Ramadge and W.M. Wonham, "Supervisory control of a class of discrete-event process," SIAM J. Contr. Optimiz., vol.25, no.1, pp.475–498, 1987.
[2] P.J. Ramadge and W.M. Wonham, "The control of discrete event systems," Proc. IEEE, vol.77, no.1, pp.81–98, 1989.
[3] N. Khakpour, F. Arbab, and E. Rutten, "Supervisory controller synthesis for safe software adaptation," Proc. 12th International Workshop on Discrete Event Systems, pp.39–45, 2014.
[4] J. Baeten and J. Markovski, "The role of supervisory controller synthesis in automatic control software development," Science of Computer Programming, vol.97, no.1, pp.17–22, 2015.
[5] J. Campos, C. Seatzu, and X. Xie, ed., Formal methods in manufacturing, CRC Press, Boca Raton, Florida, 2014.
[6] Y. Brave and M. Heymann, "On stabilization of discrete-event processes," Proc. 28th IEEE Conf. on Decision and Control, pp.2737–2742, 1990.
[7] R. Kumar, V.K. Garg, and S.I. Marcus, "Language stability and stabilizability of discrete event dynamical systems," SIAM J. Control and Optimization, vol.31, no.5, pp.1294–1320, 1993.
[8] C.M. Özveren, A.S. Willsky, and P.J. Antsaklis, "Stability and stabilizability of discrete event dynamic systems," J. ACM, vol.38, no.3, pp.729–751, 1991.
[9] K.W. Schmidt and C. Breindl "A framework for state attraction of discrete event systems under partial observation," Information Sciences, vol.281, pp.265–280, 2014.
[10] Q. Wen, R. Kumar, and J. Huang, "Framework for optimal fault-tolerant control synthesis: maximize prefault while minimize post-fault behaviors," IEEE Trans. Systems, Man, and Cybernetics: Systems, vol.44, no.8, pp.1056–1066, 2014.
[11] H.M. Khalid, M.S. Kirik, and K.W. Schmidt, "Abstraction-based supervisory control for reconfigurable manufacturing systems," in: 4th IFAC Workshop on Dependable Control of Discrete Systems, vol.4, no.1, pp.157–162, 2013.
[12] K.W. Schmidt, "Optimal configuration changes for reconfigurable manufacturing systems," in: 52nd IEEE Conference on Decision and Control, pp.7621–7626, 2013.
[13] K.G. Larsen, "Modal specications," Lect. Notes Comput. Sci., vol.407, pp.232–246, 1990.
[14] U. Fahrenberg, K.G. Larsen, A. Legay, and L.M. Traonouez, "Parametric and quantitative extensions of modal transition systems," Lect. Notes Comput. Sci., vol.8415, pp.84–97, 2014.
[15] J. Huang and R. Kumar, "Nonblocking directed control of discrete event systems," in: 44th IEEE Conference on Decision and Control, pp.7627–7632, 2005.
[16] J. Huang and R. Kumar, "An optimal directed control framework for discrete event systems," IEEE Trans. Systems, Man and Cybernetics, vol.37, no.5, pp.780–791, 2007.
[17] J. Huang and R. Kumar, "Optimal nonblocking directed control of discrete event systems," IEEE Trans. Automatic Control, vol.53, no.7, pp.1592–1603, 2008.
[18] J. Huang and R. Kumar, "Directed control of discrete event systems for safety and nonblocking," IEEE Trans. Autom. Sci. Eng., vol.5, no.4, pp.620–629, 2008.
[19] R. Sengupta and S. Lafortune, "An optimal control theory for discrete event systems," SIAM J. Control and Optimization, vol.36, no.2, pp.488–541, 1998.
[20] Y. Brave and M. Heymann, "On optimal attraction in discrete-event processes," Information Sciences, vol.67, no.3, pp.245–276, 1993.
[21] R. Kumar and V.K. Garg, "Optimal supervisory control of discrete event dynamical systems," SIAM J. Control and Optimization, vol.33, no.2, pp.419–439, 1995.
[22] C.M. Lagoa, "On the design of optimal and robust supervisors for deterministic finite state automata," in: 44th IEEE Conference on Decision and Control, and the European Control Conference, pp.2745–2750, 2005.
[23] Q. Hu and W. Yue, "Optimal control for discrete event systems with arbitrary control pattern," Discrete and Continuous Dynamical Systems - Series B, vol.3, no.3, pp.535–558, 2006.
[24] L. Grigorov and K. Rudie, "Near-optimal online control of dynamic discrete-event systems," Discrete Event Dynamic Systems, vol.16, no.4, pp.419–449, 2006.
[25] R. Su, "On the complexity of synthesizing a minimum-weighted supervisor under partial observation," Automatica, vol.50, no.6, pp.1725–1729, 2014.
[26] E.W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol.1, no.1, pp.269–271, 1959.
[27] G.S. Brodal, G. Lagogiannis, and R.E. Tarjan, "Strict fibonacci heaps," Proc. 44th symposium on Theory of Computing, pp.1177–1184, 2012.



**Sasinee Pruekprasert** received the B. E. degree from Kasetsart University, Bangkok, Thailand, in 2010, and the M. E. degree from Osaka University, Osaka, Japan, in 2012. She is currently working toward the Ph.D. degree in the Department of Systems Innovation, Osaka University. Her research interests include control of discrete event systems and game theory applications.

**Toshimitsu Ushio** received his B. E., M. E., and Ph. D. degrees in 1980, 1982, and 1985, respectively, from Kobe University. He joined Osaka University as an Associate Professor in 1994, and is currently a Professor. His research interests include control and analysis of discrete event systems and scheduling of real-time systems. He is a member of IEEE, SICE, and ISCIE.