Supervisory Control of Partially Observed Quantitative Discrete Event Systems for Fixed-Initial-Credit Energy Problem

Sasinee PRUEKPRASERT††, Nonmember and Toshimitsu USHIO††, Fellow

SUMMARY This paper studies the supervisory control of partially observed quantitative discrete event systems (DESs) under the fixed-initial-credit energy objective. A quantitative DES is modeled by a weighted automaton whose event set is partitioned into a controllable event set and an uncontrollable event set. Partial observation is modeled by a mapping from each event and state of the DES to the corresponding masked event and masked state that are observed by a supervisor. The supervisor controls the DES by disabling or enabling any controllable event for the current state of the DES, based on the observed sequences of masked states and masked events. We model the control process as a two-player game played between the supervisor and the DES. The DES aims to execute the events so that its energy level drops below zero, while the supervisor aims to maintain the energy level above zero. We show that the proposed problem is reducible to finding a winning strategy in a turn-based reachability game.

key words: supervisory control, discrete event system, partial observation, optimal control, energy game

1. Introduction

The supervisory control, which was introduced by Ramadge and Wonham in [1], is a formal approach to the design of a controller for a discrete event systems (DES). In [1], a DES is modeled by an automaton that spontaneously generates events that are partitioned into controllable and uncontrollable events. A supervisor observes the sequence of events generated by the DES, and then control it by disabling any of the controllable events at the current state of the DES. The supervisory control problem is to design a supervisor such that the language generated by the controlled DES satisfies a control specification, i.e., a given target language. Controller design problems can be formulated as two-player games played between the controller and the system. In [2], the control problem under budget constraints was considered under a two-player game setting. In [3], the minimum attention controller synthesis for omega-regular objectives was studied using a two-player game automaton.

The supervisor may not be able to clearly observe some events or some states due to the lack of sensors. As a result, the supervisor may not be able to explicitly determine the current state of the system. The framework of supervisory control under partial observation was introduced in [4], where the set of events is partitioned into the set of observable and unobservable ones. In real world, sensors can be classified into 2 kinds: state-based sensors (i.e., location sensors) and event-based sensors (i.e., touch sensors). In [5], Takai et al. proposed another framework for partial observability by introducing mappings from each event and each state to the corresponding masked event and masked state, respectively. Then, the supervisor determines the set of possible current states of the DES and controls it based on the observed masked events and masked states.

Energy games are two-player games played on weighted graphs where player-1 aims to maintain the energy of the system in a given range, and player-2 aims to prevent player-1 from achieving her goal. This class of games is applicable in the design of resource-constrained reactive systems, for example, an automatic lawn mower system with rechargeable battery [9]. In [10], a polynomial-time algorithm for deciding the strategy of player-1 in energy games under full observation was proposed. The energy game under partial observation with fixed initial credit is shown to be in ACK-complete [11]. However, the game with unknown initial credit is undecidable [12].

In this paper, we study the supervisory control of partially observed quantitative DESs under the energy objective where the initial credit energy is fixed [12]. We model the control of the DES using a two-player game played between the supervisor and the DES on a weighted automaton. The DES aims to execute the events so that its energy level drops below zero after a finite number of events occur. On the other hand, the supervisor aims to maintain the energy level above zero. The fixed-initial-credit energy problem is to compute a supervisor under which the controlled DES contains no deadlock and the energy level of the DES never goes below zero. We show that the proposed problem is reducible to finding a winning strategy in a turn-based reachability game [12], [13]. The preliminary version of this paper was presented in [14].

The rest of the paper is organized as follows. Section 2 introduces quantitative DESs and provides the basic notations. Section 3 introduces supervisory control under partial observation based on two-player game setting. Section 4 formulates the control problem and proposes algorithms. Section 5 provides an application example of a path planning robot problem that is a fundamental problem in mobile robots. Finally, Sect. 6 presents the conclusions.

 Manuscript received August 29, 2016.
 Manuscript revised January 21, 2017.
 Manuscript publicized March 7, 2017.
 †The author is with the National Institute of Informatics, Tokyo, 101–8430 Japan.
 ††The author is with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560–8531 Japan.
 ’She was with the Graduate School of Engineering Science, Osaka University, when the current work was done.
 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.2016FOP0008
2. Quantitative Discrete Event Systems

In this paper, \( \mathbb{N} \) denotes the set of natural numbers including zero, and \( \mathbb{Z} \) denotes the set of integers.

We consider a quantitative DES modeled by a weighted automaton \( G = \langle X, \Sigma, \delta, x_G, \omega \rangle \), where \( X \) is a finite set of states, \( \Sigma \) is a finite set of events, \( \delta \subseteq X \times \Sigma \times X \) is a set of transition relations, \( x_G \in X \) is the initial state, and \( \omega : \delta \to \mathbb{Z} \) is a function that assigns a weight to each transition. The event set \( \Sigma \) is partitioned into two disjoint sets: the uncontrollable events set \( \Sigma_u \) and the controllable events set \( \Sigma_c \). Likewise, transitions in \( \delta \) is partitioned into a set of uncontrollable transitions \( \delta_c = \delta \cap (X \times \Sigma_c \times X) \) and a set of uncontrollable transitions \( \delta_u = \delta \setminus \delta_c \). At each state \( x \in X \), an event \( \sigma \in \Sigma \) is called an active event if there exists an outgoing transition \( (x, \sigma, x') \in \delta \). Moreover, the transition \( (x, \sigma, x') \in \delta \) is said to be an active transition at the state \( x \). Let \( \Sigma(x) = \{ \sigma \in \Sigma \mid \exists x' \in X, (x, \sigma, x') \in \delta \} \) be the set of active events at the state \( x \). A state \( x \in X \) such that \( \Sigma(x) = \emptyset \) is called a deadlock state.

The DES \( G \) repeatedly executes one of the active events at its current state. Starting from the initial state \( x_G \), the DES changes its states according to any of the active transitions that are labeled by the generated events. The behavior of the DES is represented by the generated sequence, called the run, of its executed events and system states. Formally, a run is a finite sequence \( r = x_0\sigma_1x_1 \ldots x_n \in X(\Sigma X)^\ast \) such that \( (x_i, \sigma_{i+1}, x_{i+1}) \in \delta \) for each \( i \in \{0, 1, \ldots, n-1\} \). Let \( last(r) \) denote the last state \( x_n \) of the run \( r \). Namely, \( x_n \) is the current state of the DES after the run \( r \) is generated. The run \( r \) is called a cycle if \( x_0 = x_n \) (i.e., the DES state returns the state \( x_0 \)), and is called a deadlock run if \( \Sigma(x_0) = \emptyset \) (i.e., the DES enters a state with no active events and halts).

For each state \( x \in X \), \( Run(G, x) = \{ x_0\sigma_1x_1 \ldots x_n | x_0 = x \text{ and } (x_i, \sigma_{i+1}, x_{i+1}) \in \delta, \forall i \in \{0, 1, \ldots, n-1\} \} \) is the set of all runs generated by the DES \( G \) starting from \( x \). Let \( Run(G) = Run(G, x_G) \).

3. The Control under Partial Observation

We consider partially observed DESs whose states or events cannot be completely observed by a supervisor. The partially observed DESs studied in this paper is based on the framework proposed in [5]. Since the supervisor may not be able to determine the current state of the DES, the supervisor controls the DES based on the set of possible candidates of the current state of the DES.

Let \( Y \) be the set of marked states and \( \Lambda \) be the set of masked events [5]. A surjective function \( M_Y : X \to Y \) (resp. \( M_\Lambda : \Sigma \to \Lambda \)) maps each state (resp. event) to its masked state (resp. masked event). A supervisor cannot detect the states of the DES and the executed events, but their masked states and masked events. We assume that the masked state of the initial state is \( M_Y(x_G) = y_G \). An observation function \( M_O : \bigcup_{x \in X} Run(G, x) \to Y(AY)^\ast \) is defined as follows: for each run \( r = x_0\sigma_1x_1 \ldots \sigma_n x_n \in \bigcup_{x \in X} Run(G, x) \),

\[
M_O(r) = \begin{cases} 
M_Y(x_0), & \text{if } r = x_0, \\
M_Y(x_0)\sigma_1 \ldots \sigma_{n-1} M_\Lambda(\sigma_n) M_Y(x_n), & \text{otherwise}.
\end{cases}
\]

The function \( M_O \) maps each generated run to the sequence of masked states and masked events, called the masked run, that is observed by the supervisor. For the DES in Fig. 1, we have \( M_O(x_0a_1x_1) = M_O(x_0a_2x_2) = y_0a_2y_1 \). Therefore, if the supervisor observes the sequence \( y_0a_2y_1 \), the set of possible current states of the DES is \( \{x_1, x_2\} \). For each set of runs \( R \subseteq Run(G) \), let \( M_O(R) = \{M_O(r) | r \in R\} \) be the set of all masked runs of \( R \).

A supervisor observes the generated masked runs, and controls the DES by disabling some controllable events at the current state of the system. A set of controllable events \( \gamma \subseteq \Sigma_c \) is called a control pattern. Let \( \Gamma = 2^\Sigma \) be the set of all control patterns. If the DES generates a run \( r \) and the supervisor observes the masked run \( M_O(r) \) and assigns a control pattern \( \gamma \in \Gamma \), then the events included in \( \gamma \) are disabled at the state \( last(r) \) in the controlled DES.

We consider the control process as a two-player game played between the supervisor and the DES. A strategy of the supervisor (resp. the DES) is a function \( \pi_S : M_O(Run(G)) \to \Gamma \) (resp. \( \pi_D : Run(G) \times \Gamma \to \delta \)). We impose the following conditions on the strategy \( \pi_D \): for each run \( r \in Run(G) \) and each control pattern \( \gamma \in \Gamma \) such that \( \Sigma(last(r)) \cap \gamma \neq \emptyset \), if \( \pi_D(r, \gamma) \neq \emptyset \), then \( x = last(r) \) and \( \sigma \in \Sigma \setminus \gamma \). In other words, the strategy \( \pi_D \) selects an active transition at the current state that is enabled by the supervisor if it exists.

For each generated run \( r \), the supervisor first selects the control pattern \( \gamma = \pi_S(M_O(r)) \), then the DES executes the transitions \( (last(r), \sigma, x') = \pi_D(r, \gamma) \). This process is repeated so that the set of runs generated under the pair of strategies \( (\pi_S, \pi_D) \) is \( \text{Play}(G, \pi_S, \pi_D) = \{x_0\sigma_1x_1 \ldots x_n \in Run(G) | \sigma_{i+1} \in \Sigma(last(x_0\sigma_1 \ldots x_i)) \setminus \pi_S(M_O(x_0\sigma_1 \ldots x_i)) \text{ and } (x_i, \sigma_{i+1}, x_{i+1}) \in \pi_D(M_O(x_0\sigma_1 \ldots x_i), \pi_S(M_O(x_0\sigma_1 \ldots x_i))) \} \) for each \( i \in \{0, 1, \ldots, n-1\} \). Notice that there may exist more than one run generated under \( (\pi_S, \pi_D) \) because the strategy \( \pi_S \) is a function defined on the set of masked runs \( M_O(Run(G)) \). Thereby, \( \text{Play}(G, \pi_S, \pi_D) \) is a set of runs. \( \Pi_S \) and \( \Pi_D \) denote the sets of strategies of the supervisor and the DES, respectively. Let \( \text{Play}(G, \pi_S, \pi_D) = \{r \in \text{Play}(G, \pi_S, \pi_D) | \pi_D \in \Pi_D \} \) (resp. \( \text{Dead}(G, \pi_S) = \{r \in \text{Play}(G, \pi_S) | \Sigma(last(r)) \setminus \pi_S(M_O(r)) = \emptyset\}) \) be the set of all runs (resp. deadlock runs) generated under the strategy \( \pi_S \in \Pi_S \).

4. Fixed-Initial-Credit Energy Problem

Let \( e_0 \in \mathbb{N} \) be the initial credit energy given at the initial
state \( x_{G0} \) of the DES. For each transition \((x, \sigma, x') \in \delta \), the weight \( w(x, \sigma, x') \) indicates the energy that the system gains (if \( w(x, \sigma, x') \geq 0 \)) or loses (if \( w(x, \sigma, x') < 0 \)) by executing the transition. Then, we consider the following problem.

**Problem 1.** For a given initial credit energy \( e_0 \in \mathbb{N} \), the fixed-initial-credit energy problem is to find a strategy \( \pi_t \in \Pi_S \) such that

1. \( \text{Dead}(G, \pi_t) = 0 \), and
2. \( EL(r) \geq 0 \) for all \( r = x_0 \sigma_1 x_1 \ldots \sigma_n x_n \in \text{Play}(G, \pi_t) \), where \( EL(r) = e_0 + \sum_{i=0}^{n-1} w(x_i, \sigma_{i+1}, x_{i+1}) \).

\( EL(r) \) is called an energy level of the system by the run \( r \).

In other words, the strategy \( \pi_t \) controls the DES in such a way that 1) there is no deadlock run and 2) the energy level of the system never drops below zero. Next, we show that the problem is reducible to computing a winning strategy in a turn-based reachability game [13].

In order to solve the problem, we introduce an observation function which indicates the set of possible current states of the DES and their energy levels. Formally, an observation function is a function \( o : X \rightarrow \mathbb{Z} \cup \{ \perp \} \). For each state \( x \in X \), if \( o(x) \in \mathbb{Z} \), then \( x \) is a possible current state and the energy level of the system is \( o(x) \). Otherwise, \( x \) is not the current state. Denoted by \( \text{supp}(o) = \{ x \in X | o(x) \neq \perp \} \) is the set of all possible current states of the DES according to the observation function \( o \). The function \( o \) is said to be non-negative if \( o(x) \geq 0 \) for all \( x \in \text{supp}(o) \). Let \( O \) be the set of all observation functions of the DES \( G \). We consider a relation \( \leq \) defined on \( O \) as follows: for each \( o_1, o_2 \in O \), we have \( o_1 \leq o_2 \) if (1) \( \text{supp}(o_1) = \text{supp}(o_2) \) and (2) \( o_1(x) \leq o_2(x) \) for each \( x \in \text{supp}(o_1) \).

For a control pattern \( \gamma \in \Gamma \), a masked event \( \lambda \in \Lambda \), and a masked state \( y \in Y \), the \((\gamma, \lambda, y)\)-successor of \( o_1 \) if the following conditions hold.

1. \( \text{supp}(o_2) = \{ x \in X | \exists (x_1, \sigma, x_2) \in \delta, x_1 \in \text{supp}(o_1), \sigma \in M^\Lambda(\lambda) \setminus \gamma, M^\Lambda(\gamma) \} \).
2. For each \( x_2 \in \text{supp}(o_2), \sigma \in M^\Lambda(\lambda) \setminus \gamma, M^\Lambda(\gamma) \), \( M^\Lambda(\gamma) \).
3. For each \( x_1 \in \text{supp}(o_1) \), there exists \( (x_1, \sigma, x_2) \in \delta \) such that \( \sigma \in M^\Lambda(\lambda) \setminus \gamma \).

Namely, the \((\gamma, \lambda, y)\)-successor of \( o_1 \) the set of possible current states changes from \( \text{supp}(o_1) \) to \( \text{supp}(o_2) \) if the supervisor selects the control pattern \( \gamma \), and observes the masked state \( y \) and the masked event \( \lambda \). The condition 3 guarantees that the selected control pattern \( \gamma \) enables at least one event at each state in \( \text{supp}(o_1) \). Therefore, assigning the control pattern \( \gamma \) for the observation \( \text{supp}(o_1) \) does not generate a deadlock run. Moreover, the observation \( o_2 \) indicates the worst-case energy of the DES at each state in \( \text{supp}(o_2) \). For any observation \( o \in O \), \( \text{succ}(o, \gamma, \lambda, y) \) denotes the \((\gamma, \lambda, y)\)-successor of \( o \).

Then, we construct a game automaton \( H = \langle Q_H = Q_S \cup Q_D, \Sigma_H = \Gamma \cup (\Lambda \times Y), \delta_H = \delta_S \cup \delta_D, o_{H0} \rangle \) with \( Q_S \subseteq O((\Lambda \times Y) \Omega) \) (resp. \( Q_D \subseteq O((\Lambda \times Y) \Omega) \)) which is the set of states of the supervisor (resp. the DES).\(^{1}\) \( \delta_S \subseteq Q_S \times (\Lambda \times Y) \times Q_S \) is the set of out-going transitions from the supervisor’s (resp. the DES’s) states, and \( o_{H0} \in Q_S \) is the initial state. Algorithm 1 computes the game automaton \( H \), and the set \( Q^* = \{ q = o_0(\lambda_1, y_1) \ldots (\lambda_n, y_n) \in Q_S | o_0 \text{ is non-negative, } \Sigma_H(q) = 0, \text{and there exists an integer } m \in \{ 0, 1, \ldots, n-1 \} \text{ such that } o_m \leq o_n \} \) for each \( q = o_0(\lambda_1, y_1) \ldots (\lambda_n, y_n) \in Q_S \). \( \text{pre}(q) = 0 \) is the index \( m \) such that \( o_m \leq o_n \). This algorithm is modified from the algorithm for solving the fixed initial credit energy problem for a two-player game played on graph proposed in [12].

Figure 2 shows the game automaton constructed from the DES in Fig. 1 using Algorithm 1 where the initial credit \( e_0 = 0 \). The deadlock states in this automaton are \( h_2, h_4, h_5, \) and \( h_6 \). Notice that \( h_2 = o_0(h_2, y_2) \), and \( \text{supp}(o_2) \) contains the deadlock state \( x_3 \). Moreover, \( h_4 = o_0(a_1, y_1) o_1(b, y_2) o_3 \), and the function \( o_2 \) is not non-negative because \( o_2(x_3) = -1 \).

\(^{1}\)In these concatenations, each element in \( O \) is regarded as a symbol representing the corresponding observation function.
Therefore, we have $Q^* = \{h_5, h_6\}$.

**Theorem 2.** The state set $Q_H$ is finite.

**Proof.** Let $O$ be regard as the set of symbols representing the observation functions. Consider an infinite sequence $o_0o_1 \ldots \in O^\omega$ such that for each $k \in \mathbb{N}$, $o_k$ represents a non-negative observation function. Since the state set $X$ is finite, $\{\text{supp}(o) : o \in O\} \subseteq 2^X$ is also finite. Then, there exist $i, j \in \{0, 1, \ldots\}$ such that $i < j$ and $o_i \leq o_j$ by Dickson’s lemma [15].

We prove this theorem using a contradiction. Suppose that $Q_H$ is infinite. Since $X$ and $\Sigma$ are finite, the number of outgoing transitions at each state in $Q_H$ is also finite. By König’s lemma [16], [17], there exists an infinite sequence $q_0^S \gamma_1 q_1^D (\lambda_1, y_1) q_1^S \ldots \gamma_m q_m^D (\lambda_m, y_m) q_m^S \in \text{Run}(H)$ such that $q_0^S \in Q^*$ and $q_m^S = \text{pre}(q_m^S)$, there exists runs $q_0^S \gamma_1 q_1^D (\lambda_1, y_1) q_1^S \ldots \gamma_m q_m^D (\lambda_m, y_m) q_m^S \in \text{Run}(H')$ that contains the corresponding non-negative cycle in $q_m^S \ldots q_0^S$. The modified automaton of the game automaton in Fig. 2 is illustrated in Fig. 3. Then, from a given strategy $\phi_s \in \Phi_s$ for the game $H$, we define a strategy $\pi_{\phi_s} \in \Pi_s$ for the game $G$ as follows. For each $r = y_0\lambda_1 y_1 \ldots \lambda_n y_n = M_G(\text{Run}(G))$, there exists $\pi_{\phi_s}(r) = \phi_s(r)$.

1. if there exists a run $r_H = q_0^S \gamma_1 q_1^D (\lambda_1, y_1) q_1^S \ldots \gamma_m q_m^D (\lambda_m, y_m) q_m^S \in \text{Run}(H')$ such that $\gamma_i = \phi_s(q_i)$ for each $i \in \{0, 1, \ldots, n - 1\}$, then $\pi_{\phi_s}(r) = \phi_s(r_H)$;
2. otherwise, $\pi_{\phi_s}(r) = \emptyset$.

**Theorem 4.** There exists $\pi_s \in \Pi_s$ that satisfies the fixed-initial-credit energy problem if and only if there exists $\phi_s \in \Phi_s$ that satisfies the $Q^*$-reachability problem. Moreover, for a given strategy $\phi_s \in \Phi_s$ that satisfies the $Q^*$-reachability problem, the strategy $\pi_{\phi_s} \in \Pi_s$ satisfies the fixed-initial-credit energy problem.

**Proof.** ($\rightarrow$) Let $\pi_s \in \Pi_s$ be a strategy of the supervisor for the game $G$ that satisfies the fixed-initial-credit energy problem. Let $\phi_s \in \Phi_s$ be a strategy of the supervisor for the game $H$ defined as follows.

1. $\phi_s(q_0) = \pi_s(M_s(x_0 \gamma_0))$.
2. For each $q = o_0(\lambda_1, y_1) o_1(\lambda_n, y_n) \in Q_S$, $\phi_s(q) = \pi_s(M_s(x_0 \gamma_0) \lambda_1 y_1 \ldots \lambda_n y_n) = \phi_s(q_0)$.

Then, we show that last($\text{Dead}(H, \phi_s)) \subseteq Q^*$ using a contradiction. From the construction of $H$, we have $\text{Dead}(H, \phi_s) \subseteq Q_s$. Suppose that there exists $r_H = \text{Dead}(H, \phi_s)$ where last($r_H) \notin Q^*$. Since Algorithm 1 does not add any outgoing transition from the state last($r_H$) in $\delta_s$, at least one of the following cases holds.

1. last($r_H$) is not non-negative.
2. There does not exist a transition $(y, \lambda, y, o) \in G \times X \times Y \times O$ such that $o$ is the $(y, \lambda, y)$-successor of $o_n$.

For case 1, from the construction of $H$, there exists $r_G = x_0\gamma_1 x_1 \ldots x_n \in \text{Play}(G, \pi_s)$ such that $M_G(r_G) = y_{G0} \lambda_1 y_1 \ldots \lambda_n y_n$ and $e_0 + E\text{L}(r_G) < 0$. For case 2, from the construction of $H$, there exists $r_G = x_0\gamma_1 x_1 \ldots x_n \in \text{Play}(G, \pi_s)$ such that $M_G(r_G) = y_{G0} \lambda_1 y_1 \ldots \lambda_n y_n$ and $\Sigma(x_n) = 0$. Therefore, $\text{Dead}(G, \pi_s) \neq \emptyset$. Both cases are contradictions to the definition of Problem 1.

($\leftarrow$) Let $\phi_s \in \Phi_s$ be a strategy of the supervisor for the game $H$ that satisfies the $Q^*$-reachability problem. We show...
that the strategy \( \pi_\delta \in \Pi \) satisfies the fixed-initial-credit energy problem using a contradiction. Suppose that \( \pi_\delta \) does not satisfy the problem. By the definition of Problem 1.1, we have \( \text{Dead}(G, \pi_\delta) \neq \emptyset \) or there exists \( r_G \in \text{Play}(G, \pi_\delta) \) with \( e_0 + EL(h) < 0 \).

First, consider the case where \( \text{Dead}(G, \pi_\delta) \neq \emptyset \). Let \( r_G = x_0 \sigma_1 x_1 \ldots \sigma_n x_n \in \text{Dead}(G, \pi_\delta) \). From the constructions of \( H \) and \( H' \), there exists \( r_H = q_0^G y_1^G q_1^0(\lambda_1, y_1) q_1^G \ldots y_n^G q_n^G(y_n, y_n) q_n^G \in \text{Play}(H', \phi_\delta) \) such that \( \Sigma_H(q_n^H) = \emptyset \) and \( M_0(r_G) = y_{G0} \lambda_1 y_1 \ldots \lambda_n y_n \). However, since \( q_n^G \in Q'_G \) and \( \Sigma_H(q_n^H) = \emptyset \), we have \( q_n^G \notin Q' \). Thus, \( r_H \in \text{Play}(H, \phi_\delta) \) is the run that visits a deadlock state which is not in \( Q' \). This is a contradiction to the definition of Problem 3.

Next, consider the case where there exists \( r_G = x_0 \sigma_1 x_1 \ldots \sigma_n x_n \in \text{Play}(G, \pi_\delta) \), \( e_0 + EL(h) < 0 \). From the constructions of \( H \) and \( H' \), there exists \( r_H = q_0^G y_1^G q_1^0(\lambda_1, y_1) q_1^G \ldots y_n^G q_n^G(y_n, y_n) q_n^G \in \text{Play}(H', \phi_\delta) \) such that \( o = \text{last}(q_n^H) \in O \) is not non-negative and \( M_0(r_G) = y_{G0} \lambda_1 y_1 \ldots \lambda_n y_n \). From the construction of \( H' \), \( r_H \) must also be included in \( \text{Play}(H, \phi_\delta) \). Obviously, \( q_n^G \in Q' \). This is a contradiction to the definition of Problem 3. \( \square \)

From Theorem 4, the fixed-initial-credit energy problem can be solved by algorithms for computing a winning positional strategy of the first player turn-based reachability games [13]. By the similar discussion in [11], [18], the upper bound of the size of game \( H \), as well as the upper bound of the time complexity of Algorithm 1, is bounded by the Ackermann function.

5. Application Example

We consider the synthesis of a supervisor for robot path planning for exploring the rooms shown in Fig. 4(a). The paths are represented by the weighted automaton shown in Fig. 4(b). The task of the robot is to explore an area while receiving the control commands sent via a wireless network. The initial state is \( x_0 \), the initial credit energy is \( e_0 = 0 \), and the transition from \( x_0 \) to \( x_1 \) represents the energy recharge. The negative weight of the other transitions represent the energies needed for traveling through the paths. The uncontrollable event \( u_0 \) represents a doorway that always open, while the controllable events \( c_0 \) and \( c_1 \) represent ones that are controlled by the supervisor. Since the doorways \( c_0 \) and \( u_0 \) at each state are close to each other, they are observed by the masked event \( \lambda_0 \). Moreover, the states in the same areas are mapped to the same masked state. The control objective is to allow the robot to explore the rooms provided that its energy never drops below zero.

By applying algorithm 1, we obtain the game automaton in Fig. 5. The square states are the states of the supervisor (in \( Q_S \)) and the circle states are ones of the DES (in \( Q_D \)). The square states with thick frames belong to the set \( Q' \). Due to limitations of space, we only show the last observation function at each state of the supervisor. Moreover, instead of showing the labels of the transitions in \( \delta_D \), we do not include them in the figure. The edges in the automaton indicate the transitions between states. Each edge is labeled with a transition function, which specifies the next state and the credit change associated with that transition.
show the disabled events at each state of the DES. That is, the circle states labeled by \( (c_0, c_1) \), \( c_1 \), \( c_0 \), and the no-labeled circle states represent the control patterns \{ \( c_0, c_1 \) \}. We only show the states of the DES where at least one controllable event is disabled. For example, at the initial state \( o_0 \), we only consider the control pattern \( \emptyset \) because there is no outgoing controllable transition at the state \( x_0 \) of the DES in Fig. 4 (b).

By using the algorithms for reachability games [13], we compute the strategies of the supervisor that can control the generated runs to reach the states in \( Q^+ \). The runs are shown as the black-line transitions in Fig. 5. Notice that there are two possible ways to control the DES for the masked run \( o_0(\lambda_0, y_1)\lambda_1 \): by disabling or enabling the event \( c_0 \). If we enable the event \( c_0 \), the corresponding strategy \( \pi \), of the supervisor for the DES in Fig. 4 (b) is as follow:

- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \emptyset \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \{ c_0, c_1 \} \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \{ \emptyset \} \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \emptyset \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \{ c_0, c_1 \} \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \emptyset \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \{ c_0, c_1 \} \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \emptyset \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \{ c_0, c_1 \} \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \emptyset \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \{ c_0, c_1 \} \),
- \( \pi(s_0(o_0(\lambda_0, y_1)\lambda_1)) = \emptyset \).

The runs generated by the supervised DES under \( \pi \) are shown in thick lines in Fig. 4. In the supervised DES, the energy of each run never drops below zero.

6. Conclusions

We studied the supervisory control of partially observed quantitative DESs under the fixed-initial-credit energy objective. Partial observation is modeled by a mapping from each event and state of the DES to the corresponding masked event and masked state that are observed by a supervisor. The fixed-initial-credit energy problem is to synthesize a supervisor under which the controlled DES does not contain a deadlock and the energy level of the system never goes below zero. Then, the proposed problem was reduced to computing a winning strategy in a turn-based reachability game whose size is bounded by the Ackermann function. It is future work to consider the lower bound of the size of reachability game.

References