Bounded Synthesis and Reinforcement Learning of Supervisors for Stochastic Discrete Event Systems With LTL Specifications

Ryohi Oura, Toshimitsu Ushio, Member, IEEE, and Ami Sakakibara, Member, IEEE

Abstract—In this article, we consider supervisory control of stochastic discrete event systems (SDESs) under linear temporal logic specifications. Applying the bounded synthesis, we reduce the supervisory synthesis to the problem of satisfying a safety condition. First, we consider a directed controller that allows at most one controllable event to be enabled. We assign a negative reward to the unsafe states and introduce an expected return with a state-dependent discount factor. We compute a winning region and a directed controller with the maximum satisfaction probability using a dynamic programming method, where the expected return is used as a value function. Next, we construct a permissive supervisor via the optimal value function. We show that the supervisor accomplishes the maximum satisfaction probability and maximizes the reachable set within the winning region. Finally, for an unknown SDES, we propose a two-stage model-free reinforcement learning method for efficient learning of the winning region and the directed controllers with the maximum satisfaction probability. We also demonstrate the effectiveness of the proposed method by simulation.

Index Terms—Bounded synthesis, linear temporal logic (LTL), reinforcement learning (RL), stochastic discrete event systems (SDESs).

I. INTRODUCTION

A DISCRETE event system (DES) is a discrete-state event-driven system whose state changes by the occurrence of events [1]. A DES captures the characteristics of manufacturing systems [2], robot systems [3], and so on. The supervisory control theory was initially developed by Ramadge and Wonham [4]. In their framework, the DES is modeled as an automaton and the control objective is specified by a formal language. The supervisor dynamically restricts the behavior of the DES by disabling some controllable events to ensure that the controlled DES generates the specified language. Infinite games such as mean-payoff and safety games have been utilized to synthesize a supervisor that accomplishes specified objectives. The authors in [5], [6], and [7] formulated some infinite horizon objectives as mean-payoff games. A safety game was used to symbolically synthesize a permissive supervisor for timed DESs in [8] and [9].

In general, more than one event can be enabled at a state in the DES. Then, the nondeterministic occurrences of the events can be modeled by a stochastic process quantitatively. Various stochastic discrete event system (SDES) models have been advocated. Particularly, from the perspective of supervisory control, SDESs are modeled by probabilistic automata, probabilistic languages, and so on [10], [11], [12], [13], [14]. The authors in [10] and [11] modeled the SDES as a probabilistic automaton and initially investigated the probabilistic supervisory control problem. They provided a necessary and sufficient condition for the existence of a probabilistic supervisor by which the controlled SDES satisfies a given probabilistic specification. They also developed a concrete algorithm for the synthesis of a probabilistic supervisor that achieves the probabilistic specification. Kumar and Garg [12] used a probabilistic language for modeling the SDES and considered a range control problem. The synthesized probabilistic supervisor restricts the SDES so that the generated probabilistic language lies between the lower and upper bound constraints. Pantelic and Lawford [13] addressed the case where the language specification is not achievable on the SDES and investigated an optimal supervisory control problem. The optimal supervisor is synthesized by minimizing the pseudometric between the unachievable specification and its achievable approximation. Recently, Deng et al. [14] considered the probabilistic supervisory control for the SDES under partial observation. They defined the notions of probabilistic controllability and observability and provided the polynomial verification algorithm for the notions. They demonstrated a necessary and sufficient condition for the existence of probabilistic supervisors. Besides, the optimal control problem was addressed for the case where the specification is not achievable. In other directions and extensions, some fault localization and decentralized supervisory control for SDESs were investigated [15], [16]. Recently, Lin and Ying [17] proposed a probabilistic quantification of fuzzy DESs [18].

In the aforementioned literature, desired properties were specified using formal languages. However, in general, it is difficult to precisely convert the desired property into a formal language. To overcome this issue, temporal logic has been leveraged [19].
An advantage of temporal logic is the resemblance to natural languages, and thus, it has been widely used in several engineering fields. Particularly, complicated mission or behavior in controlled systems such as robot motion planning can be specified by temporal logic precisely and many synthesis methods of a controller or a planner that satisfy the specifications have been proposed [25], [26], [27], [28]. Linear temporal logic (LTL) is often used as a specification language due to its rich expressivity. It can describe many important \(\omega\)-regular properties such as liveness, safety, and persistence [23]. It is known that any LTL formula can be converted into an \(\omega\)-automaton with the Büchi or the Rabin acceptance condition [23], [29]. Recently, the bounded synthesis approach for LTL specifications was proposed [30], [31]. Intuitively, its main idea is converting the LTL formula into a safety automaton such that all words recognized by it satisfy the LTL formula. The procedure of obtaining the safety automaton is summarized as follows. First, a given LTL specification is converted into a (universal) co-Büchi automaton (cBA). Next, for a nonnegative integer \(K\), the cBA is restricted to a \(K\)-co-Büchi automaton (\(K\)cBA) that rejects any word visiting the set of accepting states more than \(K\) times. Finally, the \(K\)cBA is determined by the usual subset construction with counters, resulting in a safety automaton. The approach has some advantages over the other automata-theoretic synthesis methods. For example, the obtained automaton is deterministic and has a smaller state space than the corresponding Rabin automaton in general. Moreover, safety automata are tractable and suitable for the synthesis of a permissive supervisor.

On the other hand, the model uncertainty is an important issue. Recently, reinforcement learning (RL) [32] has been paid much attention to as a useful approach to controller synthesis problems for systems whose dynamics are unknown. In general, an RL method learns an optimal control policy by trial and error on a controlled stochastic system such as a Markov decision process (MDP). The authors in [33] and [34] proposed an RL-based method to learn an optimal supervisor for an SDES, where rewards depend on selected control patterns. Hence, the search space becomes exponential for the size of the event set. Furthermore, they did not consider temporal logic specifications.

Recently, automata-guided RL methods for the synthesis of a controller or planner to achieve a given LTL specification have been proposed by numerous researchers [35], [36], [37], [38], [39], [40], [41], [42], [43]. This direction was initially suggested in [35]. In the literature, the authors converted the given LTL specification into a deterministic Rabin automaton (DRA) and defined a reward function based on the Rabin acceptance condition. The RL of stochastic games for LTL specifications was investigated using Rabin automata in [36].

However, such RL-based methods cannot be directly applied to the problem of synthesizing supervisors for SDESs under LTL specifications. More specifically, in general, supervisors directly constructed by simply combining optimal control policies obtained from [35], [36], [37], [38], [39], [40], [41], [42], and [43] fail to accomplish the desired performance. This is because the stochastic dynamics of the Markov chain (MC) induced by the supervisor differs from those of the MCs induced by the optimal control policies. See [44, Example 1], for instance. Moreover, most existing automata-guided RL methods reduced the satisfaction problem of an LTL formula into a repeated-reachability problem using the Büchi or the Rabin acceptance conditions. Thus, it is hard to obtain the winning region and all winning policies.

In this article, we introduce a novel value-based RL method for the synthesis of a permissive supervisor for the unknown SDES constrained by LTL specifications. Our contributions are as follows.

1) Inspired by the bounded synthesis [30], [31], we reduce the supervisor synthesis problem into the satisfaction problem of a safety condition by converting the given LTL specification into a safety automaton. We define the reward function via the acceptance condition of the product of the SDES and the automaton.

2) We apply a dynamic programming (DP) method to the computation of both the winning region and an optimal directed controller [45], [46] with the maximum satisfaction probability, where we leverage the expected return as the value function. Using the obtained optimal state value function, we construct a permissive supervisor with the maximum satisfaction probability.

3) We propose a two-stage model-free RL method for the case where the dynamics of the SDES is unknown. For the first stage, by RL, we estimate the exact winning region and simultaneously compute all winning directed controllers. For the second stage, we compute a directed controller that maximizes the probability of reaching the estimated winning region (EWR) by relearning the state value function for the states outside the winning region. Using the learned value function, we obtain the same permissive supervisor as the case using the DP-based method with probability 1.

The advantages of our method using the bounded synthesis and RL are threefold.

1) We reduce the synthesis of the supervisor for the SDES under the LTL specification into a value-based RL for a safety condition. So, we determine the winning region by the state-value function instead of dealing directly with the state space. Furthermore, the learning problem of a desired supervisor is decomposed into the learning optimal behaviors in and outside the winning region.

2) As the parameter \(K\) of the \(K\)cBA for the LTL specification decreases, the size of its state space is smaller but the satisfaction of the specification is more conservative.

3) The synthesized supervisor is guaranteed to be an optimal one under mild conditions.

The rest of this article is organized as follows. Section II reviews an MDP, LTL, and automata. Section III formulates a supervisory control problem. Section IV proposes a synthesis method based on the bounded synthesis and DP for directed controllers. Section V proposes the construction of permissive supervisors via the optimal value function. Section VI proposes a two-stage RL method. Section VII gives a numerical example to demonstrate the effectiveness of our proposed method. Finally, Section VIII concludes this article.
II. PRELIMINARIES

Notations: \( \mathbb{N} \) is the set of positive integers. \( \mathbb{N}_0 \) is the set of nonnegative integers. \( \mathbb{R} \) is the set of real numbers. \( \mathbb{R}_{\geq 0} \) is the set of nonnegative real numbers. We denote the cardinality of a set \( T \) by \(|T|\). Denoted by \( T^w \) and \( T^* \) are the sets of finite and infinite sequences obtained from a finite set \( T \), respectively. For sets \( L_1 \subseteq T^w \) and \( L_2 \subseteq T^* \), we denote by \( L_1L_2 \) the concatenation of \( L_1 \) and \( L_2 \). Let \( \varepsilon \) be the empty string.

A. Stochastic Discrete Event Systems

We consider an SDES \( D = (S,E,P_T,P_E,s^0,AP,L) \), where \( S \) is the finite set of states, \( E \) is the finite set of events, \( P_T : S \times S \times E \rightarrow [0,1] \) is the transition probability, \( P_E : E \times S \times 2^E \rightarrow [0,1] \) is the controlled event occurrence probability, \( s^0 \in S \) is the initial state; \( AP \) is the finite set of atomic propositions, and \( L : S \rightarrow 2^{AP} \) is the labeling function that assigns a set of atomic propositions to each state \( s \in S \).

We assume that \( E \) is partitioned into the controllable event set \( E_c \) and the uncontrollable event set \( E uc \) that is, \( E = E_c \cup E uc \). We assume that, for any state \( s \in S \) and any event \( e \in E \), \( \sum_{s \in S} P_T(s'|s,e) \in \{0,1\} \). Then it is said that the event \( e \) is enabled at the state \( s \) if \( \sum_{s' \in S} P_T(s'|s,e) = 1 \).

Denoted by \( E(s) \) is the sets of events enabled at the state \( s \)

\[
E(s) = \left\{ e \mid \sum_{s' \in S} P_T(s'|s,e) = 1 \right\}.
\]

Let \( E_c(s) \) and \( E uc(s) \) be the sets of controllable and uncontrollable events enabled at the state \( s \), respectively

\[
E_c(s) = E(s) \cap E_c
\]

\[
E uc(s) = E(s) \cap E uc.
\]

A nonempty subset \( \xi \) of \( E(s) \) satisfying \( E uc(s) \subseteq \xi \) is called a control pattern at the state \( s \). Note that \( E uc(s) \) is a control pattern at \( s \) if and only if \( E uc(s) \neq \emptyset \). Let \( \Xi(s) \subseteq 2^E \) be the set of control patterns at the state \( s \). Let \( \Xi = \bigcup_{s \in S} \Xi(s) \). We assume that all events in \( E \) \( \xi \) are disabled, that is, for any \( s \in S \), any \( \xi \in \Xi(s) \), and any \( e \in E \)

\[
P_E(e|s,\xi) \in \begin{cases} (0,1], & \text{if } e \in \xi \\ \{0\}, & \text{if } e \notin \xi \end{cases}
\]

\[
\sum_{e \in \xi} P_E(e|s,\xi) = 1.
\]

Note that, by (4), \( P_E(e|s,\xi) > 0 \) for any \( s \in S \), any \( \xi \in \Xi(s) \), and any \( e \in E uc(s) \).

The controlled transition function \( P : S \times \{(s,\xi)|s \in S, \xi \in \Xi(s)\} \rightarrow [0,1] \) is defined as follows: for any \( s, s' \in S \) and any \( \xi \in \Xi(s) \)

\[
P(s'|s,\xi) = \sum_{e \in \xi} P_E(e|s,\xi)P_T(s'|s,e).
\]

Note that, for any \( s \in S \) and any \( \xi \in \Xi(s) \), we have

\[
\sum_{s' \in S} P(s'|s,\xi) = 1.
\]

Note that, in this article, we do not choose an event but a control pattern deterministically. An event \( \sigma \) is said to be control-enabled (with respect to a control pattern \( \xi \)) if \( \sigma \in \xi \) [4]. A control pattern \( \xi \) at the state \( s \) is called a directed control pattern if \( |\xi \cap E uc(s)| \leq 1 \) [45, 46]. For convenience, we will write \( \xi e \) for the directed control pattern including the controllable event \( e \). Let \( \xi 0(s) = E uc(s) \). Denoted by \( \Xi dir(s) \) is the set of directed control patterns at the state \( s \). Note that \( \xi 0(s) \in \Xi dir(s) \) if and only if \( E uc(s) \) is nonempty. Moreover, let \( \Xi dir = \bigcup_{s \in S} \Xi dir(s) \).

Remark 1: Our definition of the SDES generalizes the definition of the probabilistic generator and the automaton in [10] and [12], respectively. When we assume that the state transition is deterministic and each event in a given \( \xi \) occurs with the probability in accordance with the prior occurrence probability under the control pattern by which all events are control-enabled, our definition coincides with the defined SDEs in [10] and [12].

In the SDES \( D \), an infinite path starting from a state \( s_0 \in S \) is defined as a sequence \( \rho = s_0s_1s_2 \ldots \in S(ES)\) such that \( e_i \in E(s_i) \) and \( P_T(s_{i+1}|s_i,e_i) > 0 \) for any \( i \in \mathbb{N}_0 \). A finite path is a finite sequence in \( S(ES)^* \). For a path \( \rho = s_0s_1s_2 \ldots \), we define the corresponding labeled path \( L(\rho) = L(s_0)L(s_1) \ldots \in (2^{AP})^\omega \). InfPath\(^D(\xi) \) (resp., FinPath\(^D(\xi) \)) is defined as the set of infinite (resp., finite) paths starting from \( s_0 = s \) in the MDP \( D \) and, for simplicity, InfPath\(^D(s') \) (resp., FinPath\(^D(s') \)) is denoted by InfPath\(^D \) (resp., FinPath\(^D \)). For each finite path \( \rho \), last(\( \rho \) denotes its last state. For the SDES \( D \), the smallest \( \sigma \)-algebra over all possible infinite paths is constructed with a usual way and the unique probability measure \( P^D \) on it is defined [23]. For \( P \subseteq \text{InfPath}^D \) whose all infinite paths contain a finite path \( s_0s_1s_2 \ldots s_\ell \in \text{FinPath}^D \) as their prefix, its probability is given by \( P^D(P) = \sum_{\ell=0}^{\infty} P_T(s_{\ell+1}|s_\ell, e_\ell)P_E(e_\ell|s_\ell, E(s_\ell)) \).

Example 1: We consider an example of the transition probabilities of an SDES \( (S,E,P_T,P_E,s^0,AP,L) \), where \( S = \{s_0,s_1,s_2\} \), \( s^0 = s_1 \), \( E_c = \emptyset \), \( E uc = \{err\} \), \( AP = \{r\} \), and the labeling function \( L \) is defined as \( L(s_2) = \{r\} \) and \( L(s) = \emptyset \) for \( s = s_0, s_1 \). Only the transitions from \( s_0 \) and \( s_2 \) with the event \( b \) and \( c \) are probabilistic. Each transition from \( s_0 \) to \( s_1 \) and \( s_2 \) with \( b \) occurs with probability 0.5. The transition from \( s_2 \) to \( s_1 \) with \( c \) occurs with probability 0.7 and, with probability 0.3, the state stays \( s_2 \). Other than the state \( s_0 \), the event occurrence probabilities is equal to the inverse of the number of control-enabled events. That is, for each state \( s \in S \setminus \{s_0\} \), \( P_E(e|s,\xi) = \frac{1}{|\xi|} \) for each \( \xi \in \Xi(s) \) and any \( e \in \xi \). We show the event occurrences under some control patterns and the state transitions at \( s_0 \) in Fig. 1. Let \( \xi_0 = \{err\} \), \( \xi_1 = \{a,err\} \), \( \xi_2 = \{b,err\} \), and \( \xi_3 = \{a,b,err\} \). The values at the edges between control patterns and events are the event occurrence probabilities. The values at the edges between events and states are the transition probabilities.

B. LTL and Automata

We use LTL formulas to describe various temporal constraints or properties specified to the SDES. LTL formulas are constructed from a set of atomic propositions, Boolean operators,
and temporal operators. We use the standard notations for the Boolean operators: \( \top \) (true), \( \neg \) (negation), and \( \land \) (conjunction).

LTL formulas over a set of atomic propositions \( AP \) are defined as
\[
\varphi ::= \top | \alpha \in AP | \varphi_1 \land \varphi_2 | \neg \varphi | X\varphi | \varphi_1 U \varphi_2
\]
where \( \varphi, \varphi_1, \) and \( \varphi_2 \) are LTL formulas. Additional Boolean operators are defined as \( \perp ::= \neg \top, \varphi_1 \lor \varphi_2 ::= \neg \neg \varphi_1 \land \neg \varphi_2, \) and \( \varphi_1 \Rightarrow \varphi_2 ::= \neg \varphi_1 \lor \varphi_2. \) The operators \( X \) and \( U \) are called “next” and “until,” respectively.

Let \( \rho[i] \) be the \( i \)th suffix \( \rho[i] = s_0e_is_{i+1} \ldots = s_i \) of \( \rho \) be the \( i \)th state \( \rho[i] = s_i. \)

**Definition 1:** For an LTL formula \( \varphi \) and an infinite path \( \rho = s_0a_0s_1 \ldots \) of a DES \( D \) with \( s_0 \in S, \) the satisfaction relation \( D, \rho \models \varphi \) is recursively defined as follows:
\[
D, \rho \models \top, \\
D, \rho \models \alpha \in AP \iff \alpha \in L(\rho[0]), \\
D, \rho \models \varphi_1 \land \varphi_2 \iff D, \rho \models \varphi_1 \land D, \rho \models \varphi_2, \\
D, \rho \models \neg \varphi \iff D, \rho \not\models \varphi, \\
D, \rho \models X\varphi \iff D, \rho[1] \models \varphi, \\
D, \rho \models \varphi_1 U \varphi_2 \iff \exists j \geq 0, D, \rho[j] \models \varphi_2 \\
\land \forall i, 0 \leq i < j, D, \rho[i] \models \varphi_1.
\]

The next operator \( X \) requires that \( \varphi \) is satisfied by the next state suffix of \( \rho. \) The until operator \( U \) requires that \( \varphi_1 \) holds true until \( \varphi_2 \) becomes true over the path \( \rho. \) Using the operator \( U, \) we define two temporal operators: 1) eventually, \( F \varphi ::= \top U \varphi; \) and 2) always, \( G \varphi ::= \neg F \neg \varphi. \) In the following, we write \( \rho \models \varphi \) without referring to \( D \) for simplicity.

We define an \( \omega \)-automaton.

**Definition 2 (\( \omega \)-automaton):** An \( \omega \)-automaton is a tuple \( A = (X, \Sigma, \delta, x^I, \text{Acc}), \) where \( X \) is the finite set of states, \( \Sigma \) is the input alphabet including \( \varepsilon, \delta : X \times \Sigma \rightarrow 2^X \) is the transition function, \( x^I \in X \) is the initial state, and \( \text{Acc} \) is the accepting set, namely, the set of accepting states.

An infinite sequence \( w \in \Sigma^\omega \) is called a word. An infinite sequence \( r = x_0\sigma_0x_1 \ldots \in X(\Sigma X)^\omega \) is called a run on \( A \) generated by a word \( w = \sigma_0x_0 \ldots \Sigma^\omega \) if \( x_{i+1} \in \delta(x_i, \sigma_i) \) for any \( i \in \mathbb{N}_0. \)

For an \( \omega \)-automaton \( A \) and a word \( w, \) we denote by \( \text{Runs}(w; A) \) the set of runs on \( A \) generated by \( w. \) Moreover, for a state \( x \) and a run \( r, \) we denote by \( \text{Visits}(x; r) \) the number of times \( r \) visits \( x. \)

We define the universal co-Büchi and the universal \( K \)-co-Büchi automaton.

**Definition 3 (universal \((K)\)-co-Büchi automaton):** Let \( w \in \Sigma^\omega \) be a word. An \( \omega \)-automaton \( A \) with the following universal co-Büchi acceptance condition is called a universal co-Büchi automaton (cBA).

1) The universal co-Büchi acceptance condition: \( w \) is accepted by \( A \) if and only if, for any \( r \in \text{Runs}(w; A), \) \( \sum_{x \in \text{Acc}} \text{Visits}(x; r) < \infty. \)

For a nonnegative integer \( K, \) an \( \omega \)-automaton \( A \) with the following universal \( K \)-co-Büchi acceptance condition is called the universal \( K \)-co-Büchi automaton (KCBA).

1) The universal \( K \)-co-Büchi acceptance condition: \( w \) is accepted by \( A \) if and only if, for any \( r \in \text{Runs}(w; A), \)
\[
\sum_{x \in \text{Acc}} \text{Visits}(x; r) \leq K.
\]

Denoted by \( L_c(A) \) and \( L_{c,K}(A) \) are the sets of words accepted by the \( \omega \)-automaton with the universal co-Büchi and the universal \( K \)-co-Büchi acceptance condition, respectively. Moreover, to clarify which acceptance condition is adopted, the cBA and the KCBA are denoted by \( B \) and \( (B, K), \) respectively.

Note that, for any \( \omega \)-automaton \( A \) and any nonnegative integers \( K_1, K_2 \in \mathbb{N}, \) if \( K_1 \leq K_2 \) then we have \( L_{c,K_1}(A) \subseteq L_{c,K_2}(A) \subseteq L_c(A) \) \([31]\).

We determine the KCBA by a normal subset construction with counters \([31]\).

**Definition 4 (Determination of KCBA):** For a KCBA \((B, K) = (X, \Sigma, \delta, x^I, \text{Acc}), \) its determination (dKCBA) is a tuple \( \text{det}(B, K) = (F, \Sigma, \Delta, F^I, \text{Acc}_d), \) where
\[
1) F = \{ F \mid F \text{ is a mapping from } X \text{ to } \{-1, \ldots, K + 1\} \}; \\
2) F^I : X \rightarrow \{-1, \ldots, K + 1\} \text{ is a mapping such that } \Delta(F, \sigma)(x') = \max\{\min(K + 1, F(x) + (x' \in \text{Acc}))|x' \in \delta(x, \sigma), F(x) \neq -1\},
\]
\[
\text{where } \max \emptyset = -1;
\]
\[
3) F^I : X \rightarrow \{-1, 0, 1\} \text{ is a mapping such that }
F^I(x) = \begin{cases} 
-1, & \text{if } x \neq x^I \\
(x \in \text{Acc}), & \text{otherwise}
\end{cases}
\]
\[
\text{where } (x \in \text{Acc}) = 1 \text{ if } x \text{ is in } \text{Acc}, \text{ otherwise } (x \in \text{Acc}) = 0;
\]
\[
4) \text{Acc}_d = \{ F \in F \mid \exists x \in X \text{ s.t. } F(x) > K \}.
\]

Intuitively, the determination is constructed by, for all \( x \in X, \) keeping track of the maximal number of accepting states that have been visited by runs ending up \( x \) using the counters \( F \in F. \) For each \( x \in X, \) the count for \( x \) is set to \(-1 \) if no run starting from \( x^I \) ends up in \( x. \) The accepting states are represented by the set of \( F \in F \) such that an original state \( x \) has a count greater than \( K, \) i.e., \( F(x) > K. \)

We say that an \( \omega \)-automaton \( A \) is deterministic if \( |\delta(x, \sigma)| \leq 1 \) for any \( x \in X \) and any \( \sigma \in \Sigma. \) \( A \) is complete if, for any \( w \in \Sigma^\omega, \) \( \text{Runs}(w; A) \neq \emptyset \) holds.

The following proposition is shown in \([31]\).

**Proposition 1:** Let \( B \) be a complete cBA. The corresponding \( \text{det}(B, K) \) is deterministic, complete, and \( L_{c,K}(B) = L_{c,0}(\text{det}(B, K)). \)
It is known that, for any LTL formula \( \varphi \), there exists a complete cBA that accepts all words satisfying \( \varphi \) [31]. In particular, we represent a complete cBA recognizing an LTL formula \( \varphi \) as \( B_\varphi \), whose input alphabet is given by \( \Sigma = 2^\mathbb{A} \). By Proposition 1, there exists a dKcBA constructed as a 0-co-Büchi automaton from a given LTL formula such that all words recognized by the dKcBA satisfy the LTL formula.

Note that, for a dKcBA, the size of \( \mathcal{F} \) is about \( K^{\lvert \text{Acc} \rvert} \) times larger than the set of states \( X \) of the original cBA in the worst case.

**Definition 5 (Sink set):** Let \( A = (X, \Sigma, \delta, x^i, \text{Acc}) \) be an \( \omega \)-automaton. For the subset of states \( X_{\text{sub}} \subseteq X \), we say that \( X_{\text{sub}} \) is a sink set if there is no outgoing transition from \( X_{\text{sub}} \) to \( X \setminus X_{\text{sub}} \), that is \( \delta(x, \sigma) \subseteq X_{\text{sub}} \) for any \( x \in X_{\text{sub}} \) and any \( \sigma \in \Sigma \).

Note that the set of accepting states Acc\(_d\) can be constructed as a sink set for any dKcBA \( \det(B, K) \). This is because, for any run \( r \in X^\omega \), once \( r \) enters Acc\(_d\), it never satisfies the acceptance condition of \( \det(B, K) \). Practically, Acc\(_d\) can be constructed as a singleton.

We consider an LTL specification \( \varphi = \mathcal{G}\mathcal{F}r \) that represents the specification “the state labeled by \{r\} is visited infinitely often.” Then, shown in Fig. 2(a) are a complete cBA and its corresponding complete dKcBAs with \( K = 1, 2, 3 \) converted from \( \varphi \). We represent their accepting states with red circles. Note that states of the automaton are represented by the pair of the state of the original \( K \)-co-Büchi automaton and the number counted by the counters \( F \) of the dKcBA. We omit the states of the automaton that are not reachable from the initial state.

**C. Product Stochastic Discrete Event Systems**

We introduce the product of an SDES and a complete dKcBA.

**Definition 6 (Product SDES):** For a complete dKcBA \( \det(B, K) = (\mathcal{F}, \Sigma, \Delta, F^I, \text{Acc}_d) \) and an SDES \( D = (S, E, P_T, P_E, s^I, AP, L) \), the product SDES is a tuple \( D^\omega = D \otimes \det(B_\varphi, K) = (S^\omega, E^\omega, P_T^\omega, P_E^\omega, s^I \otimes \text{Acc}_d^\omega) \), where \( S^\omega = S \times F \) is the finite set of states; \( E^\omega = E, E^e = E_c, \) and \( E^\omega_{ec} = E_{uc} \); \( P_T^\omega : S^\omega \times S^\omega \times E^\omega \rightarrow [0, 1] \) is the transition probability defined as, for any \( s^\omega = (s, F) \) and any \( e \in E^\omega(s) \)

\[
P_T^\omega(s^\omega | s^\omega, e) = \begin{cases} P_T(s^I | s, e), & \text{if } F = \Delta(F, L(s)) \vphantom{\prod} \\ 0, & \text{otherwise} \end{cases}
\]

(8)

where \( s^\omega = (s', F') \); \( P_E^\omega : E^\omega \times S^\omega \times S^\omega \rightarrow [0, 1] \) is the event occurrence probability defined as \( P_E^\omega(e | s^\omega, \xi) = P_E(e | s, \xi) \) for any \( s^\omega = (s, F) \in S^\omega \) and any \( \xi \in \Xi(s) \); \( s^\omega I = (s^I, F^I) \) is the initial state, where \( F^I = \Delta(F^I, L(s^I)) \), and \( \text{Acc}_d^\omega = S \times \text{Acc}_d \).

Intuitively, the product SDES is a synchronized structure between the SDES and the dKcBA. In other words, the product SDES represents simultaneously the transition of the SDES and the associated transition of the dKcBA.

Note that \( \text{Acc}_d \) is nonempty since \( \text{Acc}_d \) is nonempty. For simplicity, the product SDES \( D^\omega \) of a given SDES and a dKcBA converted from an LTL formula \( \varphi \) will be called a product SDES associated with \( \varphi \).

For each \( s^\omega = (s, F) \in S^\omega \), let \( E^\omega(s^\omega) = E(s), E^\omega_c(s^\omega) = E_c(s), \) and \( E^\omega_{uc}(s^\omega) = E_{uc}(s) \). Likewise, let \( \Xi^\omega_{dir}(s) = \Xi^\omega_{dir}(s) \) and \( \Xi^\omega_{dir}(s) = \Xi^\omega_{dir}(s) \) for each \( s^\omega = (s, F) \in S^\omega \). The controlled transition probability \( P^\omega : S^\omega \times \Xi(s) \rightarrow [0, 1] \) is defined as, for any \( s, s' \in S^\omega \) and any \( \xi \in \Xi(s) \), \( P^\omega(s' | s, \xi) = \sum_{e \in E^\omega} P_E^\omega(e | s, \xi) \). For convenience, we sometimes omit the superscripts \( \omega \).

Note that \( \text{Acc}_d \) has no outgoing transition to \( S^\omega \setminus \text{Acc}_d \) since \( \text{Acc}_d \) is a sink set. Thus, once a path of the product SDES enters \( \text{Acc}_d \), from then on, its suffix always stays in \( \text{Acc}_d \).

For any product SDES \( D^\omega \), its acceptance condition is a safety condition since it is satisfied when any path generated on \( D^\omega \) always stays in \( S^\omega \setminus \text{Acc}_d \). In the following, we call \( s \in S^\omega \setminus \text{Acc}_d \) a safe state and \( s \in \text{Acc}_d \) an unsafe state.

For a subset \( S^\omega_{sub} \) of \( S^\omega \), we introduce an atomic proposition “This state belongs to \( S^\omega_{sub} \),” which denotes \( S^\omega_{sub} \) by abuse of notation, that is, we say that a state \( s \in S^\omega \) satisfies \( S^\omega_{sub} \) if \( s \in S^\omega_{sub} \). Then, the acceptance condition of \( D^\omega \) is represented by

\[
\varphi_B = \mathcal{G} \neg \text{Acc}_d^\omega
\]

(9)

that is, an infinite path \( \rho \in \text{InfPath}^\omega \) is accepted by the SDES \( D^\omega \) if and only if \( \rho \models \varphi_B \).

The **determined logical part** of \( D^\omega = (S^\omega, E^\omega, P_T^\omega, P_E^\omega, s^I \otimes \text{Acc}_d^\omega) \) is a tuple logic \( (\Delta^\omega, \mathcal{E} \times S^\omega, \delta, s^\omega I) \), where, \( \delta : S^\omega \times (\mathcal{E} \times S^\omega) \rightarrow S^\omega \) is the transition function defined as, for each \( (s^\omega, (e, s^\omega)) \in S^\omega \times (\mathcal{E} \times S^\omega) \), \( \delta(s^\omega, (e, s^\omega)) = s^\omega I \) if \( P_T^\omega(s^\omega | s^\omega, e) > 0 \); otherwise \( \delta(s^\omega, (e, s^\omega)) \) is undefined.

In Fig. 3, we show the event occurrences and the state transitions from the initial state of the product SDES constructed from the SDES in Example 1 and the dKcBA with \( K = 1 \) depicted in Fig. 2(b).
III. SUPERVISED STOCHASTIC DISCRETE EVENT SYSTEMS

We consider a supervisory control problem of an SDES $D$ with an LTL formula $\varphi$. We synthesize a controller, called a supervisor, to restrict the behavior to satisfy the specification. For $D$ and $\varphi$, without loss of generality, the supervisor is given by the pair of the determinized logical part of the corresponding product SDES $\hat{D}$ and a mapping from its state to a control pattern $[4]$. Formally, for $D$ and $\varphi$, we define the supervisor as $SV = (\text{logic}(D^\circ), SV)$ where and $SV : Q \to \Xi$ is the mapping from a state to a control pattern.

We define the behavior of the SDES controlled by a supervisor. Let $SV = (\text{logic}(D^\circ), SV)$ be the supervisor for the SDES $D = (S, E, P_T, P_E, s^l, AP, L)$ under $\varphi$. The SDES $D$ controlled by $SV$ is defined as $SV / D = (\hat{S}, E, \hat{P}_T, \hat{P}_E, s^l, AP, L)$, where $\hat{S} = S \times S^\circ, \hat{P}_T : \hat{S} \times \hat{S} \times E \to [0, 1]$ is defined as, for any $(s,s^o, (\tilde{s}, \tilde{F})) \in S \times S^\circ$,

$$P_T((s', s^o), (\tilde{s}, \tilde{F})) = \begin{cases} P_T(s'|s, e), & \text{if } \tilde{s} = s', \rho_{s^o} = \delta(s', (e, s^o)) \\ 0, & \text{otherwise} \end{cases}$$

and any state $s \in S$, the probability measure of all paths starting from $s$ that satisfies an LTL formula $\varphi$ on the SDES $D$ under $SV$ is defined as follows:

$$Pr^D_{SV}(s \models \varphi) := Pr^D_{SV} \left( \{ \rho \in \text{InfPath}^D_{SV}(s) \mid \rho \models \varphi \} \right).$$

(11)

Similarly, we define the probability of paths starting from the initial state $s$ and the initial directed control pattern $\xi_e$ satisfies $\varphi$ on $D$ under $SV$ as $Pr^D_{SV}(s, \xi_e \models \varphi)$. We call $Pr^D_{SV}(s^l \models \varphi)$ the satisfaction probability of $\varphi$ on the SDES $D$ under $SV$. Intuitively, the satisfaction probability of $\varphi$ means that the probability of satisfying $\varphi$ from the initial state $s^l$ by $SV$ for $D$.

An MC induced by the product SDES $D^\circ$ with a supervisor $SV$ is a tuple $MC^\circ_{SV} = (S^\circ_{SV}, P^\circ_{SV}, s^l_{SV})$, where $S^\circ_{SV} = S^\circ, P^\circ_{SV}(s^l|s, SV(s))$ for $s, s^l \in S^\circ$. The state set $S^\circ_{SV}$ of $MC^\circ_{SV}$ can be represented as a disjoint union of a set of transient states $T^\circ_{SV}$ and closed irreducible sets of recurrent states $R^\circ_{SV}$ with $j \in \{1, \ldots, h\}$, i.e., $S^\circ_{SV} = T^\circ_{SV} \cup R^\circ_{SV} \cup \ldots \cup R^\circ_{SV}$ [47]. In the following, we say a “recurrent class” instead of a “closed irreducible set of recurrent states” for simplicity.

Definition 7 (Reachable set): For the set of states $S$ of an SDES $D$, the reachable set from a state $s \in S$ is defined as follows:

$$Re(S; s) = \{ \last(\rho) \in S \mid \rho \in \text{FinPaths}^D(s) \}.$$  (12)

Moreover, we define the reachable set from $s \in S$ under a supervisor $SV$ as follows:

$$Re_{SV}(S; s) = \{ \last(\rho) \in S \mid \rho \in \text{FinPaths}^D_{SV}(s) \}.$$  (13)

For simplicity, we denote the reachable set from the initial state $s^l$ by $Re(S)$ instead of $Re(S; s^l)$. Analogously, for a supervisor $SV$, we write $Re_{SV}(S)$ instead of $Re_{SV}(S; s^l)$. The reachable sets for a controlled SDES and a product SDES are defined in the same way.

Remark 2: For any controlled SDES $SV / D$ and any reachable state $s = (s, s^o) \in Re(S), s = \tilde{s}$ holds.

For the product SDES $D^\circ$, let

$$W = \{ s \in S^\circ \mid \exists SV \text{ s.t. } Pr^D_{SV}(s \models \varphi_B) = 1 \}.$$  (14)

$W$ is called the winning region of $D^\circ$. Then, a supervisor SV such that $Pr^D_{SV}(s \models \varphi_B) = 1$ is called a winning supervisor at $s$. Moreover, it is called a winning directed controller if it is a directed controller. Furthermore, let

$$W_p = \{ (s, \xi) \in S^\circ \times \Xi_{\text{dir}} \mid \exists SV \text{ s.t. } Pr^D_{SV}(s, \xi \models \varphi_B) = 1 \}.$$  (15)

$W_p$ is called the winning pair set of $D^\circ$.

We define the inclusion relations between supervisors.

Definition 8 (Inclusion relation): Let $D^\circ$ be a product SDES. For any supervisor SV and $SV'$, we say that SV includes $SV'$ if $SV'(s) \subseteq SV(s)$ for any $s \in S^\circ$ and denote $SV' \subseteq SV$ if $SV$ includes $SV'$.

We omit the subscript $D^\circ$ for simplicity.

Definition 9 (Sure satisfaction of safety): For a product SDES $D^\circ$ and a supervisor $SV$, we say that SV forces $D^\circ$ to be safe at

1 Since the transitions by the occurrence of the event in SDES is nondeterministic, we need the state after the transition to construct the supervisor.
We denote by $SV^\circ_{\text{sure}}(s)$ the set of supervisors that force $D^\circ$ to be safe surely at $s \in S^\circ$. For simplicity, we denote $SV^\circ_{\text{sure}}(s^I)$ by $SV^\circ_{\text{sure}}$. It is said that a supervisor SV satisfies the acceptance condition of the product SDES $D^\circ$ surely if $SV \in SV^\circ_{\text{sure}}$.

The following lemma shows an important property of supervisors for the product SDES $D^\circ$.

**Lemma 1:** Given a product SDES $D^\circ$ of an SDES $D$ and a dKcBA $\det(B,K)$ converted from a given LTL formula $\varphi$, the following three conditions are equivalent for any state $s \in S^\circ$ and any supervisor SV.

1. $SV \in SV^\circ_{\text{sure}}(s)$.
2. $Pr_{SV}^{D^\circ}(s | \varphi_B) = 1$.
3. $Pr_{SV}^{D^\circ}(s | \varphi_B) = 1$ for any directed controller $SV^d \subseteq SV$.

**Proof:** We fix a state $s \in S^\circ$ arbitrarily.

1. $\Rightarrow$ 2) There is no path $\rho \in \text{InfPath}_{SV}^{D^\circ}(s)$ such that $\rho \not\models \varphi_B$.

2. $\Rightarrow$ 3) Suppose that there exists a directed controller $SV^d \subseteq SV$ such that $Pr_{SV}^{D^\circ}(s | \varphi) > 0$. Then, there exists $\rho = s_0e_0s_1 \ldots \in \text{InfPath}_{SV}^{D^\circ}(s)$ such that $\rho \not\models \varphi_B$, and hence, there exists $l \in \mathbb{N}$ such that $s_{l-1} \not\in \text{Acc}^\circ$ and $s_l \in \text{Acc}^\circ$. Note that $\text{InfPath}_{SV}^{D^\circ}(s) \subseteq \text{InfPath}_{SV}^{D^\circ}(s)$. Thus, we have $\rho \in \text{InfPath}_{SV}^{D^\circ}(s)$ and hence $Pr_{SV}^{D^\circ}(s | \varphi_B) \geq \prod_{i=0}^{l-1} P_T(e_i | s_i, SV(s_i)) > 0$. This contradicts $Pr_{SV}^{D^\circ}(s | \varphi_B) = 1$.

3. $\Rightarrow$ 1) Suppose that $SV \not\in SV^\circ_{\text{sure}}(s)$. Then, there exists a path $\rho = s_0e_0s_1 \ldots \in \text{InfPath}_{SV}^{D^\circ}(s)$ such that $\rho \not\models \varphi_B$. Then without loss of generality, we can assume that $\rho = s_0e_0s_1 \ldots s_{i-1}e_i^{-1}e_i s_{i+1} s_{i+2} \ldots$ such that $s_i \in \text{Acc}^\circ$ and, for any $i$, with $0 \leq i < j \leq l$, $s_i \not\models s_j$. Hence, there exists $SV^d \subseteq SV$ such that $\rho \in \text{InfPath}_{SV^d}(s)$. Thus, we have $Pr_{SV}^{D^\circ}(s | \varphi_B) \geq \prod_{i=0}^{l-1} P_T(e_i | s_i, SV^d(s_i)) > 0$. However, this contradicts the assumption that $Pr_{SV}^{D^\circ}(s | \varphi_B) = 1$ for any $SV^d \subseteq SV$.

**IV. Bounded Synthesis of Directed Controllers**

In this section, for a given SDES $D$, a given LTL formula $\varphi$, and a nonnegative integer $K$, we aim at the synthesis of directed controllers $SV^d$ for the product SDES $D^\circ = D \otimes \det(B_e,K)$ such that $Pr_{SV^d}^{D^\circ}(s^I | \varphi_B)$ is maximized, where $s^I$ and $\text{Acc}^\circ$ are the initial state and the accepting set of $D^\circ$. We propose a synthesis method based on the bounded synthesis [30], [31]. In the following, we first define a reward function and an expected return. Next, we show some important properties between the expected return and directed controllers for the product SDES. Finally, we give a DP-based method to obtain an optimal directed controller.

---

### A. Reward Function and Expected Returns

We define a reward function based on the $K$-co-Büchi acceptance condition of the product SDES.

**Definition 10 (Reward function):** The reward function $R: S^\circ \rightarrow \mathbb{R}$ is defined by

$$R(s) = \begin{cases} (1 - \gamma_{\text{acc}})r_n, & \text{if } s \in \text{Acc}^\circ \\ 0, & \text{otherwise} \end{cases}$$

where $r_n$ is a negative value and $\gamma_{\text{acc}} \in [0,1)$.

We introduce a discount factor function and value functions for the product SDES inspired by [39].

**Definition 11 (Discount factor function):** A discount factor function $\Gamma: S^\circ \rightarrow \{\gamma_{\text{acc}}, \gamma\}$, where $\gamma \in (0,1)$, is defined by

$$\Gamma(s) = \begin{cases} \gamma_{\text{acc}}, & \text{if } s \in \text{Acc}^\circ \\ \gamma, & \text{otherwise} \end{cases}$$

**Definition 12 (Expected return):** For a directed controller $SV^d$ for the product SDES $D^\circ$, the reward function $R$ as defined in (17), and the discount factor function $\Gamma$ as defined in (18), we define the expected return, or the state-value function $V_{SV^d}: S^\circ \rightarrow \mathbb{R}$ as follows. For each $s \in S^\circ$,

$$V_{SV^d}(s) = E_{SV^d} \left[ \sum_{t=0}^{\infty} R(s_t+1) \prod_{k=0}^{t-1} \Gamma(s_{k+1}) \bigg| s_0 = s \right]$$

where $E_{SV^d}$ denotes the expected value of accumulated rewards from the state $s$ under the directed controller $SV^d$ and $\prod_{k=0}^{t-1} \Gamma(s_{k+1}) = 1$.

An upper and a lower bound of the state-value function $V_{SV^d}$ are $0$ and $r_n$, respectively, and $V_{SV^d}(s) = r_n$ for any unsafe state $s \in \text{Acc}^\circ$. We define a function $Q_{SV^d}: \{(s,\xi_e)|s \in S^\circ, \xi_e \in \Xi^\circ_{\text{dir}}(s)\} \rightarrow \mathbb{R}$ as follows. For any $s \in S^\circ$ and any $\xi_e \in \Xi^\circ_{\text{dir}}(s)$,

$$Q_{SV^d}(s,\xi_e) = E_{SV^d} \left[ \sum_{t=0}^{\infty} R(s_t+1) \prod_{k=0}^{t-1} \Gamma(s_{k+1}) \bigg| s_0 = s, \xi_0 = \xi_e \right]$$

where $R$ and $\Gamma$ are defined as (17) and (18), respectively. We call $Q_{SV^d}$ a state directed control pattern (state-DCP for short) value function under $SV^d$.

The optimal Bellman operator $T$ is introduced as follows:

$$TV(s) = \max_{\xi_e \in \Xi^\circ_{\text{dir}}(s)} \sum_{s' \in S^\circ} P^\circ(s'|s,\xi_e) \{R(s') + \Gamma(s) V(s')\}$$

and the $\infty$-norm $\|\cdot\|$ is defined by $\|V\| = \max_{s \in S^\circ} |V(s)|$. Then, for any function $V_i(t = 1,2)$ from $S^\circ$ to $\mathbb{R}$, any state $s \in S^\circ$, and any directed control pattern $\xi_e \in \Xi^\circ_{\text{dir}}(s)$,

$$\sum_{s' \in S^\circ} P^\circ(s'|s,\xi_e) \{R(s') + \Gamma(s) V_1(s')\} = \sum_{s' \in S^\circ} P^\circ(s'|s,\xi_e) \{R(s') + \Gamma(s) V_2(s')\}$$

---

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
\begin{align}
&+ \Gamma(s) \sum_{s' \in S^O} \mathcal{P}^O(s'|s, \xi_e) \{V_1(s') - V_2(s')\} \\
&\leq TV_2(s) + \Gamma(s) \parallel V_1 - V_2 \parallel.
\end{align}

Thus
\begin{equation}
TV_1(s) - TV_2(s) \leq \Gamma(s) \parallel V_1 - V_2 \parallel. \tag{22}
\end{equation}

Let \( \gamma_m = \max(\gamma_{acc}, \gamma) < 1 \). Then, by (22), we have
\begin{equation}
\parallel TV_1 - TV_2 \parallel \leq \gamma_m \parallel V_1 - V_2 \parallel. \tag{23}
\end{equation}

Therefore, by Banach’s fixed point theorem, the operator \( \mathcal{T} \) has the unique fixed point \( V^* \), which satisfies the following:
\begin{equation}
V^*(s) = \max_{\xi_e \in \Xi_d(s)} \sum_{s' \in S^O} \mathcal{P}^O(s'|s, \xi_e) \{\mathcal{R}(s') + \Gamma(s)V^*(s')\}. \tag{24}
\end{equation}

We consider a directed controller \( SV^{ds} \) such that \( V^*(s) = \sum_{s' \in S^O} \mathcal{P}^O(s'|s, SV^{ds}(s)) \{\mathcal{R}(s') + \Gamma(s)V^*(s')\} \) for each \( s \in S^o \). Then
\begin{equation}
SV^{ds}(s) = V^* = \max_{SV^{ds}} V_{SV^{ds}}(s). \tag{25}
\end{equation}

Therefore, \( V^* \) is the optimal state value function and \( SV^{ds} \) is called an optimal directed controller.

Similarly, the following has the unique solution \( Q^*\):
\begin{equation}
Q^*(s, \xi_e) = \max_{s' \in S^O} \mathcal{P}^O(s'|s, \xi_e) \{\mathcal{R}(s') + \Gamma(s)Q^*(s', \xi_e')\}. \tag{26}
\end{equation}

Then, for any \( s \in S^O \) and \( \xi_e \in \Xi_d(s) \)
\begin{equation}
Q^*(s, \xi_e) = \max_{SV^{ds}} Q_{SV^{ds}}(s, \xi_e). \tag{27}
\end{equation}

\( Q^* \) is called the optimal state-DCP value function.

\section*{B. Properties Between Directed Controllers and Expected Return}

For any product SDES \( D^O \) associated with an LTL formula \( \varphi \) and any supervisor \( SV \), let \( S^O_{SV} = \{s \in S^O | \exists \rho \in \text{InfPath}_{SV}^D(s) \text{ s.t. } \rho \models \varphi_B\} \). Note that \( S^O_{SV} \) is nonempty since it contains all unsafe states. Moreover, let \( N_\theta = \min\{n \in N_0 | \rho[n] \in \text{Acc}^O\} \) for each \( \rho \in \text{InfPath}_{SV}^D(s) \). We define \( \min \emptyset = \infty \). Then, we show the following two lemmas.

\textbf{Lemma 2:} Given a product SDES \( D^O \) associated with an LTL formula \( \varphi \), for any supervisor \( SV \), \( \max_{s \in S^O_{SV}} \mathbb{E}_{SV}[\rho[0] = s, \rho \not\models \varphi_B] < \infty \).

\textbf{Proof:} For any supervisor \( SV \), any state \( s \in S^O_{SV} \), and any path \( \rho \in \text{InfPath}_{SV}^D(s) \) with \( \rho \models \text{FAcc}^O \), we have \( \mathbb{E}_{SV}[\rho[0] = s, \rho \not\models \varphi_B] < \infty \). Therefore, we have \( \max_{s \in S^O_{SV}} \mathbb{E}_{SV}[\rho[0] = s, \rho \not\models \varphi_B] < \infty \).

\textbf{Lemma 3:} Given a product SDES \( D^O \) associated with an LTL formula \( \varphi \), a directed controller \( SV^d \), and the reward function \( R \) defined as (17), the following two conditions hold for any discount factor \( \gamma \in (0, 1) \).
1) For any \( s \in S^O \), \( V_{SV^d}(s) = 0 \) if and only if \( Pr_{SV^d}^D(s \models \varphi_B) = 1 \).
2) There exists a constant \( N \) such that, for any \( s \in S^O \),
\begin{equation}
r_n Pr_{SV^d}^D(s \not\models \varphi_B) \leq V_{SV^d}(s) < r_n Pr_{SV^d}^D(s \not\models \varphi_B) \gamma^N \tag{28}
\end{equation}
if and only if \( Pr_{SV^d}^D(s \models \varphi_B) < 1 \).

\textbf{Proof:} For any \( s \in S^O \) and any \( \gamma \in (0, 1) \), we have
\begin{equation}
V_{SV^d}(s) = \mathbb{E}_{SV^d} \left[ \sum_{t=0}^{\infty} \mathcal{R}(s_{t+1}) \prod_{k=0}^{t-1} \mathcal{G}(s_{k+1})|s_0 = s \right]
= \mathbb{E}_{SV^d} \left[ \sum_{t=0}^{\infty} \mathcal{R}(s_{t+1}) \prod_{k=0}^{t-1} \mathcal{G}(s_{k+1})|\rho[0] = s, \rho \not\models \varphi_B \right] \\
\times Pr_{SV^d}^D(s \not\models \varphi_B) \\
= \mathbb{E}_{SV^d} \left[ \gamma^{N_n-1} \sum_{t=0}^{\infty} \gamma^{t_{acc}} \mathcal{R}(s_{n,t+1})|\rho[0] = s, \rho \not\models \varphi_B \right] \\
\times Pr_{SV^d}(s \not\models \varphi_B).
\end{equation}

Thus, we have
\begin{equation}
V_{SV^d}(s) = r_n Pr_{SV^d}^D(s \not\models \varphi_B) \mathbb{E}_{SV^d}[\gamma^{N_n-1}|\rho[0] = s, \rho \not\models \varphi_B]. \tag{29}
\end{equation}

We now prove 1).

(\implies) By (29), we have \( V_{SV^d}(s) = 0 \).

(\therefore) Suppose \( Pr_{SV^d}^D(s \not\models \varphi_B) > 0 \). Then, by Lemma 2, \( \mathbb{E}[\rho[0] = s, \rho \not\models \varphi_B] < \infty \). Recall that the reward \( r_n < 0 \). By Jensen’s inequality, \( V_{SV^d}(s) \leq r_n Pr_{SV^d}^D(s \models \varphi_B) \mathbb{E}[\rho[0] = s, \rho \not\models \varphi_B] < 0 \). This contradicts \( V_{SV^d}(s) = 0 \).

Next, we prove 2).

(\implies) Let \( N = \max_{s \in S^O_{SV}} \mathbb{E}[\rho[0] = s, \rho \not\models \varphi_B] \). Recall that \( r_n < 0 \). Note that \( N < \infty \) by Lemma 2 and it is independent of the states. Moreover, \( \gamma^{N_n-1}|\rho[0] = s, \rho \not\models \varphi_B| > \gamma^N > 0 \) since \( \gamma \in (0, 1) \). By Jensen’s inequality, (28) holds.

(\therefore) Suppose \( Pr_{SV^d}^D(s \not\models \varphi_B) = 0 \). Then, we have \( V_{SV^d}(s) = Pr_{SV^d}^D(s \not\models \varphi_B) = 0 \) by (29). This contradicts the existence of the constant \( N \) that satisfies (28).

We now show that an optimal directed controller is winning at any state in the winning region \( W \).

\textbf{Theorem 1:} Given a product SDES \( D^O \) associated with an LTL formula \( \varphi \) and the reward function \( R \) defined as (17), for the winning region \( W \) of \( D^O \) and any optimal directed controller \( SV^{ds} \), we have
\begin{equation}
Pr_{SV^{ds}}^D(s \models \varphi_B) = 1 \quad \text{for any } s \in W. \tag{30}
\end{equation}

\textbf{Proof:} By (14), (2) and (3) of Lemma 1, and (1) of Lemma 3, for any \( s \in W \), there exists a directed controller \( SV^d \) such that \( V_{SV^d}(s) = 0 \). Thus, by (25), we have \( V^*(s) = 0 \) for any \( s \in W \). Hence, by (1) of Lemma 3, for any optimal directed controller \( SV^{ds} \), (30) holds.
The following theorem shows that, for each state, we can achieve the given LTL formula almost surely if there exists a directed control pattern whose state-DCP value is 0.

**Theorem 2:** Given a product SDES $D^\otimes$ associated with an LTL formula $\varphi$ and its state $s \in S^\otimes$, there exists a supervisor $SV$ such that $P^D_{SV}(s \models \varphi_B) = 1$ if and only if there exists $\xi_e \in \Xi^\otimes_{\text{dir}}(s)$ such that $Q^\ast(s, \xi_e) = 0$.

**Proof:**

($\Leftarrow$) By 1) of Lemma 3, we have $P^D_{SV}(s \models \varphi_B) = 1$, where $SV^\ast$ is an optimal directed controller with $SV^\ast(s) = \xi_e$.

($\Rightarrow$) By 2) and 3) of Lemma 1, for any directed controller $SV^d \subseteq SV$, we have $P^D_{SV}(s \models \varphi_B) = 1$. Thus, by 1) of Lemma 3, we have $Q_{SV}(s, SV^d(s)) = 0$, which implies that $Q^\ast(s, SV^d(s)) = 0$.

**Remark 3:** Theorem 2 implies that, if $Q^\ast(s^\prime, \xi_e) < 0$ for any $\xi_e \in \Xi^\otimes_{\text{dir}}(s^\prime)$, then there is no supervisor that satisfies the acceptance condition of the product SDES with probability $1$. In general, supervisors constructed via the value function based on the directed control patterns can fail to accomplish the maximum satisfaction probability in such cases, for instance, see [44, Example 1]. Furthermore, the occurrence probabilities of events can change depending on given control patterns. Taking these issues into account, we will construct a supervisor in Section V.

In the following lemma, we show that, for any state of a given product SDES, its state value converges to the negative reward $r_n$ multiplied by the probability of reaching an unsafe state from the state as the discount factor $\gamma$ goes to 1. The lemma follows from a similar proof as [39, Th. 1].

**Lemma 4:** Given a product SDES $D^\otimes$ associated with an LTL formula $\varphi$, and the reward function $R$ defined as (17), for any $\varepsilon > 0$, there exists $\gamma' > 0$ such that, for any $\gamma > \gamma'$, any state $s \in S^\otimes$, and any directed controller $SV^d$, the following holds:

$$0 \leq V_{SV^d}(s) - r_nP^D_{SV^d}(s \models \varphi_B) < \varepsilon.$$  

**Proof:** For any state $s \in S^\otimes$ and any directed controller $SV^d$, consider the following two cases 1) $P^D_{SV^d}(s \models \varphi_B) = 0$ and 2) $P^D_{SV^d}(s \models \varphi_B) > 0$. For the case 1), we have $V_{SV^d}(s) = 0$ by 1) of Lemma 3. For the case 2), by 2) of Lemma 3, we have

$$0 \leq V_{SV^d}(s) - r_nP^D_{SV^d}(s \models \varphi) \leq r_nP^D_{SV^d}(s \models \varphi) \gamma^N - r_nP^D_{SV^d}(s \models \varphi)$$

$$= |r_n|P^D_{SV^d}(s \models \varphi)(1 - \gamma^N)$$

$$\leq |r_n|(1 - \gamma^N)$$

where $N$ is a constant that satisfies 2) of Lemma 3. Thus, for any $\varepsilon > 0$, there exists $\gamma' \in (0, 1)$ such that, for any $\gamma > \gamma'$, $|r_n|(1 - \gamma^N) < \varepsilon$. Note that $\gamma'$ is independent of $s$ and $SV^d$.

Therefore, (31) holds for any $\gamma > \gamma'$, any state $s \in S^\otimes$, and any directed controller $SV^d$.

For any product SDES $D^\otimes$ associated with an LTL formula $\varphi$, any $s \in S^\otimes$, and any directed controller $SV^d$, we define

$$P^D_{\text{max}}(s \models \varphi_B) = \max_{SV^d} P^D_{SV^d}(s \models \varphi_B).$$ (32)

Similarly, we define

$$P^D_{\text{max}}(s, \xi_e \models \varphi_B) = \max_{SV^d} P^D_{SV^d}(s, \xi_e \models \varphi_B).$$ (33)

By Lemma 4, we now show that the satisfaction probability of an optimal directed controller is exactly equal to the maximum satisfaction probability when the discount factor $\gamma$ is sufficiently close to 1. This plays an important role to assure that the supervisor constructed in Section V achieves the maximum satisfaction probability.

**Theorem 3:** Given a product SDES associated with an LTL formula $\varphi$ and the reward function $R$ defined as (17), there exists a discount factor $\gamma' \in (0, 1)$ such that, for any $\gamma > \gamma'$ and any $s \in S^\otimes$, the following holds:

$$P^D_{SV^d}(s \models \varphi_B) = P^D_{\text{max}}(s \models \varphi_B).$$ (34)

**Proof:** Note that the number of directed controllers is finite since $|E_c| < \infty$ and $|S^\otimes| < \infty$. Thus, there exists $\varepsilon > 0$ such that, for any nonoptimal directed controller $SV^d$, there exists a state $s \in S^\otimes$, we have

$$V_{SV^d}(s) + \varepsilon < V^\ast(s).$$ (35)

Then, by Lemma 4, there exists $\gamma' \in (0, 1)$ such that, for any $\gamma > \gamma'$

$$V^\ast(s) - r_nP^D_{SV^d}(s \models \varphi_B) < \varepsilon.$$ (36)

By (35) and (36), we have

$$V_{SV^d}(s) < r_nP^D_{SV^d}(s \models \varphi_B).$$

Note that $r_n$ is negative. Hence, by (28), we have

$$P^D_{SV^d}(s \models \varphi_B) > P^D_{SV^d}(s \models \varphi_B).$$

Therefore, for any $SV^d$, we have

$$P^D_{SV^d}(s \models \varphi_B) \leq P^D_{SV^d}(s \models \varphi_B)$$ (37)

which implies together with (32) that (34) holds.

By Theorem 3, we obtain a directed controller with the maximum satisfaction probability by maximizing the state value function when the discount factor $\gamma$ exceeds a certain value $\gamma'$ that is less than 1.

When we know the full information about the dynamics of the SDES, to compute the directed controllers achieving the maximum satisfaction probability, we can use a DP method such as the value iteration. We show the value iteration with directed controllers in Algorithm 1.\footnote{We use the value iteration here but any DP method can be employed to our proposed framework.} Note that $\Gamma(s) < 1$ for any state $s \in S^\otimes$, and thus, the Bellman operator corresponding to (26) is a contraction mapping. This implies that, for any product SDES $D^\otimes$ and any state $s \in S^\otimes$, initializing the state-DCP value function $Q$ with 0, if there exists a winning directed controller $SV^d$, then $Q(s, SV^d(s))$ is always 0 during the value iteration. Moreover, Algorithm 1 converges exponentially.

**Example 2:** We consider the SDES in Example 1 and the dKcBA shown in Fig. 2(b), i.e., the dKcBA with $K = 1$ for $\varphi = GFr$. The initial state of the corresponding product

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Algorithm 1: Value Iteration With Directed Control.

**Input:** LTL formula \( \varphi \) and SDES \( D \).

**Output:** An optimal supervisor \( SV \).

1: Convert \( \varphi \) to dKdrBA det(\( B_{d}, K \)).
2: Construct the product SDES \( D^{\circ} \) of \( D \) and det(\( B_{d}, K \)).
3: Initialize \( Q : S^{\circ} \times E^{\circ} \rightarrow \mathbb{R} \) with 0.
4: while \( Q \) does not converge do
5:   for all state \( s \in S^{\circ} \) and \( \xi_{e} \in \Xi_{d}^{\circ}(s) \) do
6:      \[ Q(s^{\circ}, \xi_{e}) \leftarrow \sum_{s' \in S^{\circ}} \frac{P_{T}^{d}(s | s^{\circ}, \xi_{e}) \sum_{\xi_{e}'} P_{E}^{d}(\xi_{e} | s, \xi_{e}')}{\xi_{e}} \times \left\{ \mathcal{R}(s') + \Gamma(s') \max_{\xi_{e} \in \Xi_{d}^{\circ}(s')} Q(s', \xi_{e}') \right\} \]
7:   end for
8: end while

**TABLE I**

| State-DCP Values for Example 2 Obtained by Algorithm 1 |
|-----------------|-----------------|-----------------|-----------------|
| State \( s \)   | Value \( a \)   | Value \( b \)   | Value \( c \)   |
| \( (s_1, (x_0,0), (x_1,1)) \) | -0.9999 | 0 | - | - |
| \( (s_1, (x_0,0), (x_2,1)) \) | -1 | -1 | 0 | - |
| \( (s_2, (x_0,0)) \) | 0 | -0.4999 | 0 | - |
| \( (s_0, (x_0,0), (x_1,1)) \) | -0.49995 | -0.74992 | - | -0.9999 |
| \( (s_0, (x_0,0), (x_2,1)) \) | -1 | -1 | -1 | - |

where \( W \) is the winning region of \( D^{\circ} \).

Proof: We choose an optimal directed controller \( SV^{d*} \) arbitrarily. To establish (38), it is sufficient to show that, for any \( s \in S^{\circ} \) and any \( \rho \in \text{InfPath}_{SV^{d*}}(s) \), \( \rho \models FW \) implies \( \varphi_B \land FW \). Fix a state \( s \in S^{\circ} \) and a path \( \rho \in \text{InfPath}_{SV^{d*}}(s) \) arbitrarily. Suppose that \( \varphi_B \models FW \). Then, by Theorem 1 and 1) and 2) of Lemma 1, we have \( \rho \models \varphi_B \). Thus, for any optimal directed controller \( SV^{d*} \) and any \( s \in S^{\circ} \), (38) holds.

Let \( MC^{\circ}_{SV^{d*}} = (S^{\circ}, P^{\circ}_{SV^{d*}}, s^{\circ}, d, AP, L) \) be the MC induced by \( D^{\circ} \) and \( SV^{d*} \). \( S^{\circ} \) is partitioned into a transient states \( T_{SV^{d*}}, \) and \( h \) recurrent classes \( \{ R_{SV^{d*}}^{j} \}_{j=1}^{h} \), that is, \( S^{\circ} = T_{SV^{d*}} \cup R_{SV^{d*}}^{1} \cup \ldots \cup R_{SV^{d*}}^{h} \). Suppose that there exists a state \( s \notin \text{Acc} \cup W \), then, we have \( Pr_{SV^{d*}}^{*}(s \models \varphi_B) < 1 \) by the definition of \( W \). Thus, there exists a path that eventually reaches \( \text{Acc} \) from \( s \) on \( MC^{\circ}_{SV^{d*}} \). Recall that \( \text{Acc} \) has no outgoing transition to \( S^{\circ} \setminus \text{Acc} \). Hence, we have that the probabilities of reaching \( \text{Acc} \) from \( s \) and returning to \( s \) from \( \text{Acc} \) are positive and 0, respectively, which implies \( s \in T_{SV^{d*}} \). Therefore, by the property of transient states [47], (39) holds.

**Proposition 2:** Given a product SDES \( D^{\circ} \) associated with an LTL formula \( \varphi \) and the reward function \( R \) defined as (17), for any optimal directed controller \( SV^{d*} \) and any state \( s \in S^{\circ} \), the following holds:

\[ Pr_{SV^{d*}}^{*}(s \models \varphi_B) = Pr_{SV^{d*}}^{*}(s \models FW) \] (40)

Proof: For any \( s \in S^{\circ} \) and any optimal directed controller \( SV^{d*} \), we have

\[ Pr_{SV^{d*}}^{*}(s \models \varphi_B) = Pr_{SV^{d*}}^{\circ}(s \models \varphi_B \land (FW \lor G \neg W)) \]

By (38) and (39), we have

\[ Pr_{SV^{d*}}^{\circ}(s \models \varphi_B \land (FW \lor G \neg W)) = Pr_{SV^{d*}}^{d}(s \models FW) + Pr_{SV^{d*}}^{d}(s \models \varphi_B \land G \neg W) = Pr_{SV^{d*}}^{d}(s \models FW) \]

Thus, (40) holds.

**V. Construction of Permissive Supervisor**

By the discussions in Sections III and IV, an optimal directed controller is winning at any state in the winning region \( W \). Thus, a supervisor that includes only optimal directed controllers surely satisfies \( \varphi_B \) at any state in \( W \). Besides, an optimal directed controller achieves the maximum satisfaction probability of directed controllers when the discount factor is sufficiently close to 1. So, in this section, we construct a supervisor via the optimal state-DCP value function and show that the supervisor achieves the maximum satisfaction probability and maximizes the number of the reachable states within the winning region.

For the product SDES \( D^{\circ} \), we construct the following supervisor \( SV^* : S^{\circ} \rightarrow \Xi \). For each \( s \in S^{\circ} \),

\[ SV^*(s) \]
where $\xi \in \arg \max_{\xi \in \Xi^{s}_T}(s)Q^*(s, \xi)$, Note that $W$ is determined as $\{s \in S^0 \mid V^1(s) = 0\}$ by (14) and 1) of Lemma 3.

We will show that the supervisor $SV^*$ achieves the maximum satisfaction probability of directed controllers when $T$ is sufficiently close to 1.

**Lemma 6:** Given a product SDES $D^\circ$ associated with an LTL formula $\varphi$ and the reward function $\mathcal{R}$ defined as (17), for any state $s \in S^0$, the supervisor $SV^*$ defined as (41) satisfies

$$Pr^D_{SV^*}(s \models \varphi_B \land \mathcal{R}W) = Pr^D_{SV^*}(s \models \mathcal{R}W). \quad (42)$$

**Proof:** It is sufficient to show that, for any $s \in S^0$ and any $\rho \in \text{InfPath}_{SV^*}^D(s)$, $\rho \models \mathcal{R} \land W$ implies $\rho \models \varphi_B \land \mathcal{R} \land W$. We fix a state $s \in S^0$ and a path $\rho \in \text{InfPath}_{SV^*}^D(s)$ arbitrarily. Suppose that $\rho \models \mathcal{R} \land W$. Then, by Theorem 1 and 1) of Lemma 1, we have $\rho \models \varphi_B$. Thus, for any $s \in S^0$, (42) holds. $\Box$

**Lemma 7:** Given a product SDES $D^\circ$ associated with an LTL formula $\varphi$, for any state $s \in S^0$, the supervisor $SV^*$ defined as (41) satisfies

$$Pr^D_{SV^*}(s \models \varphi_B) = Pr^D_{SV^*}(s \models \mathcal{R}W). \quad (43)$$

**Proof:** For any $s \in S^0$, by (42), we have

$$Pr^D_{SV^*}(s \models \varphi_B) = Pr^D_{SV^*}(s \models \mathcal{R}W) = Pr^D_{SV^*}(s \models \mathcal{R}W) + Pr^D_{SV^*}(s \models \varphi_B \land \mathcal{R} \land W).$$

By (41), there exists an optimal directed controller $SV^d$ such that $SV^d(s) = SV^d(s)$ holds for any $s \notin W$. Thus, (43) holds by (39).

**Lemma 8:** Given a product SDES $D^\circ$ associated with an LTL formula $\varphi$ and the reward function $\mathcal{R}$ defined as (17), there exists an optimal directed controller $SV^d$ such that, for any $s \in S^0$

$$Pr^D_{SV^d}(s \models \varphi_B) = Pr^D_{SV^d}(s \models \varphi_B) \quad (44)$$

where the supervisor $SV^*$ is defined as (41).

**Proof:** By (41), there exists an optimal directed controller $SV^d$ such that $SV^d(s) = SV^*(s)$ holds for any $s \notin W$, and thus, the following holds:

$$Pr^D_{SV^d}(s \models \mathcal{R}W) = Pr^D_{SV^d}(s \models \mathcal{R}W).$$

Moreover, for any $s \in W$, clearly we have

$$Pr^D_{SV^d}(s \models \mathcal{R}W) = Pr^D_{SV^d}(s \models \mathcal{R}W).$$

Hence, by Proposition 2 and Lemma 7, (44) holds for any $s \in S^0$. $\Box$

**Theorem 4:** Given a product SDES $D^\circ$ associated with an LTL formula $\varphi$ and the reward function $\mathcal{R}$ defined as (17), there exists $\gamma' \in (0, 1)$ such that, for any $\gamma > \gamma'$ and any $s \in S^0$, the following holds:

$$Pr^D_{SV^*}(s \models \varphi_B) = Pr^D_{SV^d}(s \models \varphi_B) \quad (45)$$

where the supervisor $SV^*$ is defined as (41).

**Proof:** This immediately follows from Theorem 3 and Lemma 8.

By Theorem 4, $SV^*$ defined as (41) accomplishes the maximum satisfaction probability of directed controllers by taking the discount factor $\gamma$ sufficiently close to 1.

We define the notion of maximal permissiveness.

**Definition 13 (Maximal permissiveness):** We say that a supervisor $SV^* \in SV^D_{\text{sure}}$ is maximally permissive if, for any supervisor $SV \in SV^D_{\text{sure}}$, the following holds:

$$|\text{Res}_{SV}(S^0)| \geq |\text{Res}_{SV^*}(S^0)|. \quad (46)$$

**Theorem 5:** Given a product SDES $D^\circ$ associated with an LTL formula $\varphi$, the reward function $\mathcal{R}$ defined as (17), and the supervisor $SV^*$ defined as (41), for any state $s \in W$, we have $SV^* \in SV^D_{\text{sure}}(s)$ and $|\text{Res}_{SV}(S^0; s)| \geq |\text{Res}_{SV^*}(S^0; s)|$ for any $SV \in SV^D_{\text{sure}}(s)$.

**Proof:** For any $s \in W$, by Lemma 7, we have $Pr^D_{SV^*}(s \models \varphi_B) = 1$. Thus, by 1) and 2) of Lemma 1, $SV^*$ is maximally permissive. $\Box$

**Remark 4:** In synthesizing a supervisor that maximizes the size of the reachable set under the maximization of the satisfaction probability of $\varphi_B$, the directed control-based synthesis reduces computational complexity with respect to the size of the event set compared to dealing directly with control patterns. In detail, the exploration spaces for the directed control-based method and the straightforward one dealing with control patterns are proportion to $E_c$ and $2E_c$, respectively.

**Corollary 1:** Given a product SDES $D^\circ$ associated with an LTL formula $\varphi$ and the reward function $\mathcal{R}$ defined as (17), if the initial state $s^{0\mathcal{R}} \in W$, then the supervisor $SV^*$ is maximally permissive.

**Proof:** By Theorem 5, for any state $s$ in the winning region $W$, the number of the reachable states from $s$ is maximized while satisfying $\varphi_B$ surely under the supervisor $SV^*$ defined as (41). Moreover, if $s^{0\mathcal{R}} \in W$ then $SV^*$ is maximally permissive. $\Box$

**Remark 5:** By Corollary 1, it is sufficient to check whether the maximal value of the state-DCP function at the initial state is 0 in order to confirm the maximal permissiveness of $SV^*$.

In Example 2, the maximum state-DCP value of the initial state is 0. Thus, we construct the maximally permissive supervisor $SV^*$ as $SV^*(s_1, \{(x_0, 0), (x_1, 1)\}) = \emptyset$ and $SV^*(s_2, \{(x_0, 0)\}) = \{a, c\}$.

VI. 2-STAGE RL FOR OPTIMAL SUPERVISORS

Practically, the dynamics of the SDES such as the transition probability is often unknown. Then, we use an RL method to obtain an optimal directed controller.

For the product SDES associated with a given LTL formula, in order to learn the directed controllers that accomplish the maximum satisfaction probability, we decompose the problem
of learning the desired directed controllers into the following two stages. The two-stage RL is partially based on Q-learning.

Stage 1) We estimate the winning pair set \( W_p \) by learning the state-DCP value function. Subsequently, we synthesize the winning directed controllers at any state in the winning region \( W \) using the estimate of \( W_p \).

Stage 2) When the initial state does not belong to the winning region, we compute a directed controller that forces the paths on the product SDES to reach the winning region with the maximum probability by relearning the state value function for the states outside the winning region.

Note that the decomposition is based on Proposition 2. In other words, learning of the desired supervisor is decomposed into 1) learning \( W_p \) (the behavior in \( W \)); and 2) learning a directed controller that maximizes the probability of reaching \( W \) (the behavior outside \( W \)).

### A. Learning of Winning Pair Set and Winning Directed Controllers

For Stage 1), we propose Algorithm 2 based on Q-learning so as to obtain the winning pair set.

For a product SDES \( D^\otimes \), let \( Q^K \) be a state-DCP value function learned up to the episode \( k \in \mathbb{N} \) by Algorithm 2.

We call the following set of pairs of states and directed control patterns an estimated winning pair set learned up to the episode \( k \in \mathbb{N}_0 \):

\[
W^K_p = \{(s, \xi) \mid s \in S^\otimes, \xi \in \Xi^\otimes_{\text{dir}}(s), Q^K(s, \xi) = 0\}. \tag{47}
\]

We call the following subset of \( S^\otimes \setminus \text{Acc}^\otimes \) an EWR learned up to the episode \( k \in \mathbb{N}_0 \):

\[
W^K = \{s \in S^\otimes \mid \exists \xi \in \Xi^\otimes_{\text{dir}}(s) \text{ s.t. } Q^K(s, \xi) = 0\}. \tag{48}
\]

We now describe Algorithm 2. Recall that the state values of unsafe states and states in the winning region are \( r_n \) and 0, respectively. Thus, at Line 3, we initialize the state-DCP value function with 0 and \( r_n \) for the safe states and the unsafe states, respectively. \( W^0 \) and \( W_p^0 \) are initialized with \( S^\otimes \setminus \text{Acc}^\otimes \) and \( \{(s, \xi) \mid s \in S^\otimes \setminus \text{Acc}^\otimes, \xi \in \Xi^\otimes_{\text{dir}}(s)\} \), respectively. At Line 6, Algorithm 2 continues until the estimated winning pair set converges. At Line 7, for a current state \( s \in W^K \), we choose a directed control pattern \( \xi \) from \( \{\xi \in \Xi^\otimes_{\text{dir}}(s) \mid Q^K(s, \xi) = 0\} \). An event \( e_{oc} \in \xi \) occurs and observes the next state \( s' \). If \( s' \notin W^K \), we terminate the exploration in the episode. In Line 15, if \( e_{oc} \) is uncontrollable, i.e., the state leaves from the current EWR by an uncontrollable event, we update the state-DCP values for all directed control patterns. Other than that, we update the state-DCP value of \( s \) and \( \xi \) in Line 20. From Line 22 to 24, we update \( W^K \) and \( W_p^K \), pick up \( s \) from \( W^K \), and continue the learning.

We will show that Algorithm 2 estimates the winning pair set and synthesizes all winning directed controllers for the winning region.

**Lemma 9:** Given a product SDES \( D^\otimes \) associated with an LTL formula \( \varphi \) and the reward function \( R \) defined as (17), when conducting Algorithm 2, for any state \( s \in S^\otimes \setminus \text{Acc}^\otimes \) and any \( \xi \in \Xi^\otimes_{\text{dir}}(s) \), if there exists \( k' \in \mathbb{N}_0 \) such that \( Q^{k'}(s, \xi) < 0 \), then, for any \( k \geq k' \), we have \( Q^K(s, \xi) < 0 \).

**Proof:** We fix a state \( s \in S^\otimes \) and \( \xi \in \Xi^\otimes_{\text{dir}}(s) \) arbitrarily. Suppose that there exists \( k' \in \mathbb{N}_0 \) such that \( Q^{k'}(s, \xi) < 0 \). Note that the reward \( R(s) \) is 0 or negative. Thus, we have \( Q^K(s, \xi) < 0 \) for any \( k \geq k' \).

Lemma 9 implies that \( W^K_p \) and \( W^K \) are monotonically decreasing with respect to \( k \):

\[
W^{k+1}_p \subseteq W^K_p \tag{49}
\]

\[
W^{k+1} \subseteq W^K. \tag{50}
\]

---

**Algorithm 2:** Learning of \( W_p \) and \( W \).

**Input:** LTL formula \( \varphi \) and SDES \( D \).

**Output:** The state-DCP value function \( Q^K \) such that, for any \( s \in S^\otimes \) and \( \xi \in \Xi^\otimes_{\text{dir}}(s) \), \( Q^K(s, \xi) = 0 \) if and only if \( Q^K(s, \xi) = 0 \).

1. Convert \( \varphi \) to dFscBA det(\( B_\varphi, K \)).
2. Construct the product SDES \( D^\otimes \) of \( D \) and det(\( B_\varphi, K \)).
3. Initialize \( Q^K(s, \xi) \) with 0 for any \( s \in S^\otimes \setminus \text{Acc}^\otimes \) and any \( \xi \in \Xi^\otimes_{\text{dir}}(s) \), and \( r_n \) for any \( s \in \text{Acc}^\otimes \) and any \( \xi \in \Xi^\otimes_{\text{dir}}(s) \).
4. Compute \( W^K_p \) and \( W^0 \).
5. Pick up \( s \in W^0 \) at random and set \( k = 0 \).
6. While \( W^K_p \) does not converge do
7. While True do
8. Choose \( \xi \) from \( \{\xi \in \Xi^\otimes_{\text{dir}}(s) \mid Q^K(s, \xi) = 0\} \) uniformly and at random.
9. Observe an event \( e_{oc} \in \xi \) and the next state \( s' \).
10. If \( s' \notin W^K \) then
11. Go to Line 14.
12. Else
13. Set \( s \leftarrow s' \).
14. End while
15. If \( e_{oc} \) is uncontrollable, then
16. For all events \( \xi \in \Xi^\otimes_{\text{dir}}(s) \) do
17. \[
Q^{k+1}(s, \xi) \leftarrow (1 - \alpha)Q^K(s, \xi) + \alpha \{R(s') + \Gamma(s') \max_{\xi' \in \Xi^\otimes_{\text{dir}}(s')} Q^K(s', \xi')\}.
\]
18. End for
19. Else
20. \[
Q^{k+1}(s, \xi) \leftarrow (1 - \alpha)Q^K(s, \xi) + \alpha \{R(s') + \Gamma(s') \max_{\xi \in \Xi^\otimes_{\text{dir}}(s')} Q^K(s', \xi)\}.
\]
21. End if
22. Set \( k \leftarrow k + 1 \).
23. Compute \( W^K_p \) and \( W^K \) and pick up \( s \in W^K \).
24. End while

---

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Hence, there exist $W_p^\infty = \lim_{k \to \infty} W_p^k$ and $W^\infty = \lim_{k \to \infty} W^k$.

**Lemma 10:** Given a product SDES $D^\phi$ associated with an LTL formula $\phi$ and the reward function $R$ defined as (17), for any $s \in S^D \setminus \text{Acc}^D$ and any $\xi \in \Xi_{\text{dir}}(s)$, if there exists $k \in \mathbb{N}$ such that $Q^k(s, \xi) = 0$ and $Q^{k+1}(s, \xi) < 0$, then there exists a state $s' \in S^D \setminus W^k$ such that $P^\infty(s'|s, \xi) > 0$.

**Proof:** We fix a state $s \in S^D$ and $\xi \in \Xi_{\text{dir}}(s)$ arbitrarily. Suppose that there exists $k \in \mathbb{N}_0$ such that $Q^k(s, \xi) = 0$ and $Q^{k+1}(s, \xi) < 0$. Then, by the update rule of the state-DCP value function in Algorithm 2, there exists an event $e' \in \xi$ such that it triggers an outgoing transition from $s$ to $s' \in S^D \setminus W^k$ with a positive probability. Thus, there exists a state $s' \in S^D \setminus W^k$ such that $P^\infty(s'|s, \xi) > 0$.

**Proposition 3:** Given a product SDES $D^\phi$ associated with an LTL formula $\phi$ and the reward function $R$ defined as (17), when conducting Algorithm 2, for any state $s \in S^D$, any $\xi \in \Xi_{\text{dir}}(s)$, and any $k \in \mathbb{N}_0$, if $Q^k(s, \xi) < 0$ holds, then we have $P_{\max}^D(s, \xi, t = \varphi_B) < 1$.

**Proof:** We prove by induction with respect to $k \in \mathbb{N}_0$. Consider the case where $k = 0$. Then, for any $s \in S^D$ and any $\xi \in \Xi_{\text{dir}}(s)$, if $Q^0(s, \xi) < 0$ holds, then we have $s \in \text{Acc}^D$, and thus, $P_{\max}^D(s, \xi, t = \varphi_B) < 1$. Let $k \in \mathbb{N}_0$ be given. For any $s \in S^D \setminus \text{Acc}^D$ and any $\xi \in \Xi_{\text{dir}}(s)$, suppose that $Q^k(s, \xi) < 0$ implies $P_{\max}^D(s, \xi, t = \varphi_B) < 1$. Then, for $k+1$, any $s \in S^D \setminus \text{Acc}^D$ and any $\xi \in \Xi_{\text{dir}}(s)$, suppose that $Q^{k+1}(s, \xi) < 0$. Consider the two cases 1) $Q^k(s, \xi) < 0$ and 2) $Q^k(s, \xi) = 0$. For the case 1), we immediately have $P_{\max}^D(s, \xi, t = \varphi_B) < 1$ by the induction hypothesis. For the case 2), by Lemma 10, there exists a state $s' \in W^k$ such that $P^\infty(s'|s, \xi) > 0$. Thus, we have the same result as the case 1) by applying the induction hypothesis to $s'$ and all $\xi \in \Xi_{\text{dir}}(s')$. Thus, Proposition 3 is proved.

Proposition 3 implies that, for the winning region $W^\infty$, the winning pair set $W_p^\infty$, and any $k \in \mathbb{N}_0$, the following relations hold:

$$W_p \subseteq W_p^k$$

$$W \subseteq W^k.$$  \hspace{1cm} (51)

$$W \subseteq W^k.$$  \hspace{1cm} (52)

Due to (49), (50), (51), and (52), for each $k \in \mathbb{N}_0$ and each $s \in W^k$, we pick up a state from $W^k$ at the first step of each episode, restrict the candidates of directed control pattern to be chosen to $\{\xi \in \Xi^D \mid Q^k(s, \xi) = 0\}$, and terminate the current episode when the state leaves from $W^k$ in Algorithm 2.

**Assumption 1:** Given a product SDES $D^\phi$ associated with an LTL formula $\phi$, when conducting Algorithm 2, the following assumptions hold.

1) The learning ratio $\alpha \in (0, 1)$ is a constant.
2) For each $s \in W^\infty$, $s$ is observed infinitely often with probability 1.

To show that $W^\infty = W_p^\infty$ holds with probability 1, we define the following supervisor $SV^\infty$ constructed via $W_p^\infty$. For any $s \in S^D$

$$SV^\infty(s) = \begin{cases} \bigcup \{\{s, \xi) \in W_p^\infty \mid \xi, \text{ if } s \in W^\infty \} \xi, & \text{if } s \in W^\infty \\ \Xi_{\text{dir}}(s), & \text{otherwise.} \end{cases}$$  \hspace{1cm} (53)

**Lemma 11:** Given a product SDES $D^\phi$ associated with an LTL formula $\phi$ and the reward function $R$ defined as (17), if $R_{SV^\infty}(S^D; s) \subseteq W^\infty$ holds for any $s \in W^\infty$, then $W^\infty \subseteq W_p^\infty$ holds.

**Proof:** Let $SV^\infty$ be the supervisor defined as (53). Suppose that $R_{SV^\infty}(S^D; s) \subseteq W^\infty$ holds for any $s \in W^\infty$. Then, we have $s' \notin \text{Acc}^D$ for any $s' \in R_{SV^\infty}(S^D; s)$ since $W^\infty \cap \text{Acc}^D = \emptyset$. This implies together with 1) and 3) of Lemma 1 that, for any directed control pattern $\xi \subseteq SV^\infty(s)$, we have $(s, \xi) \in W_p^\infty$. Thus, $W_p^\infty \subseteq W_p^\infty$ holds.

**Proposition 4:** Given a product SDES $D^\phi$ associated with an LTL formula $\phi$ and the reward function $R$ defined as (17), we have $W_p^\infty = W_p^\infty$ with probability 1 under Assumption 1.

**Proof:** Let $SV^\infty$ be the supervisor defined as (53). For simplicity, we abbreviate “with probability 1” as “w.p.1.” By (51) and Lemma 11, it is sufficient to show that, for any $s \in W^\infty$, $SV^\infty(S^D; s) \subseteq W^\infty$ holds w.p.1. Suppose that there exists $s' \in R_{SV^\infty}(S^D; s)$ such that $s' \in \text{Acc}^D$ for any $s' \in R_{SV^\infty}(S^D; s)$ since $W^\infty \cap \text{Acc}^D = \emptyset$. Suppose that there exists $k \in \mathbb{N}_0$ such that $s_n \in W^k$ and $s_n \notin W^\infty$. Thus, there exists $k \in \mathbb{N}_0$ such that $s_n \in W^k$ and $s_n \notin W^\infty$. By 2) of Assumption 1, $s_n$ is visited infinitely often w.p.1. Thus, w.p.1, for any $e \in SV^\infty(s_n)$, the transition $(s_n, e, s_n)$ occurs infinitely often. Hence, w.p.1, there exists $k > l$ such that $(s_n, e, s_n)$ is observed at the step $l$. Thus, by 1) of Assumption 1, w.p.1, for any $\xi \subseteq SV^\infty(s_n)$, if $e \in \xi$ and $Q^l(s_n, \xi) < 0$, then this contradicts that $\xi \subseteq SV^\infty(s_n)$, which implies that $R_{SV^\infty}(S^D; s) \subseteq W^\infty$ holds w.p.1.

**Proposition 4** implies $W^\infty = W$ with probability 1. Moreover, by 1) of Lemma 3, we have that, for any $s \in S^D$ and any $\xi \in \Xi_{\text{dir}}(s)$, $Q^\infty(s, \xi) = 0$ if and only if $Q^\infty(s, \xi) = 0$.

**B. Learning of Directed Controllers With Maximum Satisfaction Probability**

After conducting Algorithm 2, if the initial state does not belong to $W^\infty$, we have to compute a directed controller that forces the paths on the controlled product SDES to reach the winning region with the maximum probability. For Stage 2, we propose Algorithm 3 based on Q-learning.3

We conduct Algorithm 3 on the same product SDES given to Algorithm 2 with the state-DCP value function learned by it. At Line 4, if the current state is in $W^\infty$, we terminate the episode. In other words, we compute the state values of states outside the EWR during Algorithm 3. Note that, in practice, we terminate an episode in Algorithm 3 after entering $\text{Acc}^D$.

We denote by $Q^\infty$ the state-DCP value function obtained from Algorithm 3.

**Proposition 5:** Given a product SDES $D^\phi$ associated with an LTL formula $\phi$ and the reward function $R$ defined as (17), for any state $s \in S^D$ and any $\xi \in \Xi_{\text{dir}}(s)$, we have $Q^\infty(s, \xi) = Q^\infty(s, \xi)$.}

\footnote{When the initial state is in the winning region, we do not have to conduct Algorithm 3.}
Algorithm 3: Q-Learning of SV\(^d\): Maximizing the Satisfaction Probability of \(\varphi_B\).

**Input:** The product SDES \(D^\oplus\) given to Algorithm 2 and the state-DCP value function learned by Algorithm 2.

**Output:** The optimal state-DCP value function \(Q^*\).

1. Let \(k = 0\).
2. while \(Q^k\) does not converge do
3. \( s \leftarrow (s^f, F^f) \).
4. for \(s \not\in W^\infty\) do
5. Choose \(\xi_e \in \Xi_{dir}(s)\).
6. An event \(e_{oc} \in \xi_e\) occurs.
7. Observe the next state \(s'\).
8. Obtain the reward \(R(s')\).
9. \[
Q^{k+1}(s, \xi_e) \leftarrow (1 - \alpha_k)Q^k(s, \xi_e) + \alpha_k \{R(s') + \Gamma(s') \max_{\xi_{e'} \in \Xi_{dir}(s')} Q^k(s', \xi_{e'})\}.
\]
10. \( s \leftarrow s' \).
11. end for
12. \( k \leftarrow k + 1 \).
13. end while

with probability 1 under the assumption that \(\sum_{k=0}^{\infty} \alpha_k = \infty\) and \(\sum_{k=0}^{\infty} \alpha_k \beta_k < \infty\) hold in Algorithm 3.

Proof: Note that the correct optimal state value of any state in \(W^\infty\) is 0 with probability 1 by Proposition 3. The proof follows from [48, Prop. 4.5].

By Propositions 4 and 5, using Algorithms 2 and 3, we obtain the same permissive supervisor \(SV^d\) defined as (41) by replacing \(Q^*\) and \(W\) with \(Q^*\) and \(W^\infty\), respectively, with probability 1.

**VII. EXAMPLE**

We apply the proposed method to a motion planning problem with two robots. They move in an inner environment that consists of 7 rooms shown in Fig. 4(a). The state space of the SDES is \(S = \{(s_1, s_2)\}; s_j \in \{R_i\}_{i=1}^7\), where \(s_1\) and \(s_2\) represent the rooms in which the respective robot is and \(R_i\) with \(i = 0, 1, \ldots, 6\) represents a room. The two robots survey on the environment. Let \(E = \{e_1^1, e_1^2, e_1^3, e_2^1, e_2^2, e_2^3, e_2^4, e_2^5, e_2^6\}\) be the set of events that indicate movements of the robots. The event \(e_2^k\) represents that the robot \(k\) tries to move to the next room \(R_i\) from the current room. For example, when the event \(e_1^1\) occurs at \((s_1, s_2) = (R_2, R_3)\), the robot 1 moves to the next room \(R_i\) from \(R_2\) while the robot 2 stays at the current room. The event \(e_2^1\) is an uncontrollable and it can occur when the robot 2 is in \(R_0\) or \(R_4\), which means the event \(e_2^1\) trigger unintended moves of the robot 2. The transitions from \(R_0\) to \(R_2\), \(R_3\), and \(R_4\) by the event \(e_2^1\) occur with probabilities 0.1, 0.2, and 0.7, respectively. The both transitions from \(R_4\) to \(R_0\) and \(R_6\) by \(e_2^1\) occur with probability 0.5. Similarly, the controllable event \(e_1^1\) triggers the probabilistic transition when the robot 1 is in \(R_3\). Associated with \(e_1^1\), the transitions from \(R_3\) to \(R_0\) and \(R_{i_1}\) occur with the probability 0.5.

The other transitions are deterministic, that is, the robots move to the intended directions. We require both robots to return their recharge rooms (rooms 1 and 3) infinitely often while avoiding being in the same room at the same time. The specification is represented formally as \(G F r_1 \land G F r_2 \land G \neg \bar{u}\), where \(r_1\) means that the robot 1 is in \(R_3\), \(r_2\) means that the robot 2 is in \(R_1\), and \(u\) means that both robots are in the same room. A cBA converted from \(\varphi\) is shown in Fig. 4(b). We use the dKcBA obtained from the cBA with \(K = 10\).

We set \(\gamma = 0.9999\), \(\gamma_{acc} = 0.9\), \(r_n = -1\), and \(s^f = (R_0, R_3)\). We conduct the proposed 2-stage RL method. We train directed controllers with 4000 and 200 000 episodes for Algorithms 2 and 3, respectively. We iterate \(T_{ep}\) 5000 steps per episode and repeat 100 learning sessions. For the stage 1, when sampling an initial state from the EWR, we use the probability distribution \(p^k: W^k \rightarrow [0, 1]\) at each step \(k \in \mathbb{N}_0\) defined as

\[
p^k(s) = \frac{\sum_{t=0}^{k} 1_{s_t}(s) - 1}{\sum_{s \in W^k} \left(\sum_{t=0}^{k} 1_{s_t}(s)\right)^{-1}}.
\]

where \(s_t\) is the state at the step \(t\). Intuitively, for any \(k \in \mathbb{N}_0\) and any \(s \in W^k\), the smaller the times of visiting \(s\) are in the learning, the greater the probability of sampling \(s\) is.

**A. Results for Stage 1 (Algorithm 2)**

Shown in Fig. 5 are the average reward and the average steps to leave the EWR per episode obtained by Algorithm 2. They converge to 0 and 5000, respectively, by 2500 episodes. Moreover, we observe a lot of episodes where the obtained reward is 0 before the 2500th episode. This is because the robots tend to move only within the EWR during Algorithm 2, that is, the robots try to avoid reaching an unsafe state. To show that the learned supervisor maximizes the reachable set within the winning region, we introduce the following index \(\text{Ind}_{\text{dir}}\) for each
is the frequency of each value of \( \text{Ind}^k \) for the last episode of each learning session. Note that \( \text{Ind}^k \in \{0, 1/9, \ldots, 8/9, 1\} \). \( \text{Ind}^k \) is 7/9 or more in all sessions and moreover \( \text{Ind}^k = 1 \) in 52 sessions. This implies that the supervisor obtained by Algorithm 3 approaches a supervisor that achieves the maximum satisfaction probability.

**VIII. CONCLUSION**

This article proposed a novel value-based synthesis of supervisors for SDESs. For a given LTL formula and a design parameter \( K \), we constructed a universal \( K \)-co-Büchi automaton and synthesized optimal directed controllers such that the probability of the acceptance of controlled behaviors by the automaton is maximized. Then, we considered the synthesis of an optimal supervisor and showed that it maximizes the reachability set under the maximization of the probability of the acceptance. Moreover, we considered the case where the system is unknown and proposed a 2-stage RL-based method for an efficient synthesis of the desired supervisor and showed that it synthesizes an optimal supervisor with probability 1 under some mild conditions.

We used the tabular-based RL. It is future work that we use deep neural networks in order to apply to large scale systems. Moreover, it is also important future work to show the bounded synthesis approach is practically useful.

**REFERENCES**

[1] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Berlin, Germany: Springer, 2009.

[2] J. Campos, C. Seatzu, and X. Xie, *Formal Methods in Manufacturing*. Boca Raton, FL, USA: CRC Press, 2014.

[3] C. Mahulea, M. Kloetzer, and R. González, *Path Planning of Cooperative Mobile Robots Using Discrete Event Models*. Piscataway, NJ, USA: IEEE Press, 2020.

[4] P. J. Ramadge and W. M. Wonham, ”Supervisory control of a class of discrete event processes,” *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, 1987.

[5] Y. Ji, X. Yin, and S. Lafortune, ”Supervisory control under local mean payoff constraints,” in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 1043–1049.

[6] Y. Ji, X. Yin, and S. Lafortune, ”Optimal supervisory control with mean payoff objectives and under partial observation,” *Automatica*, vol. 123, 2021, Art. no. 109359.

[7] S. Pruekprasert, T. Ushio, and T. Kanazawa, ”Quantitative supervisory control game for discrete event systems,” *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 2987–3000, Oct. 2016.

[8] E. Asarin, O. Maler, and A. Pnueli, ”Symbolic controller synthesis for discrete and timed systems,” in *Proc. Int. Hybrid Syst. Workshop*, 1994, pp. 1–20.

[9] E. Asarin, O. Maler, A. Pnueli, and J. Sifakis, ”Controller synthesis for timed automata,” *HAC Proc. Volumes*, vol. 31, no. 18, pp. 447–452, 1998.

[10] M. Lawford and W. M. Wonham, ”Supervisory control of probabilistic discrete event systems,” in *Proc. 36th Midwest Symp. Circuits Syst.*, 1993, pp. 327–331.
Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.