2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

Minimum Violation Control Synthesis on Cyber-Physical Systems under Attacks

Luyao Niu, Jie Fu and Andrew Clark

Abstract— **Cyber-physical systems are conducting increasingly complex tasks, which are often modeled using formal languages such as temporal logic. The system's ability to perform the required tasks can be curtailed by malicious adversaries that mount intelligent attacks. At present, however, synthesis in the presence of such attacks has received limited research attention. In particular, the problem of synthesizing a controller when the required specifications cannot be satisfied completely due to adversarial attacks has not been studied. In this paper, we focus on the minimum violation control synthesis problem under linear temporal logic constraints of a stochastic finite state discrete-time system with the presence of an adversary. A minimum violation control strategy is one that satisfies the most important tasks defined by the user while violating the less important ones. We model the interaction between the controller and adversary using a concurrent Stackelberg game and present a nonlinear programming problem to formulate and solve for the optimal control policy. To reduce the computation effort, we develop a heuristic algorithm that solves the problem efficiently and demonstrate our proposed approach using a numerical case study.**

I. INTRODUCTION

Cyber-physical systems have been identified to play important roles in multiple application domains such as health care systems, cloud computing, and smart homes. To model the increasingly complex tasks and corresponding desired system behaviors consistently, rigorously and compactly, temporal logics such as linear temporal logic (LTL) and computation tree logic (CTL) are adopted in recent literature. Typical system properties that can be modeled using LTL, whose syntax and semantics have been well developed, include liveness (e.g., 'always eventually A'), reactivity (e.g., 'if A, then B'), safety (e.g., 'always not A') and so on.

Formal methods provide a class of theory and methods for controller design to satisfy given specifications modeled using temporal logics. Such control synthesis problems have been investigated in different applications such as robotic motion planning [1], [2] and optimal control [3], [4]. However, these works normally explicitly or implicitly assume the existence of the controller, which is not always the case.

In [5], unsynthesizable controllers are characterized as either unsatisfiability or unrealizability. Unsatisfiability is caused by the incompatibility of the specifications given to the system, while unrealizability is caused by uncertainties and stochastic errors. Different from uncertainties and stochastic errors, malicious attacks can also cause unsynthesizable controllers in CPS. Malicious attacks on CPS raise the concern on CPS security since they can lead to misbehaviors and failures. For instance, power outage caused by attackers on power system [6], a false data injection (FDI) based attack CarShark on automobiles [7] and widely known Stuxnet on industrial control system (ICS) all caused significant economic losses and/or safety risks.

The approaches proposed for analyzing uncertainties and stochastic errors are not applicable for analyzing malicious attacks on CPS. Moreover, uncertainties and stochastic errors are often viewed as identically and independently distributed random variables, which is not the case for malicious and strategic attacks. In the worst case, stochastic elements such as environment behavior are interpreted as malicious attacks on the system. Zero-sum game provides a good model for worst case analysis [8]. Meanwhile, failures returned by control synthesis framework could also be caused by malicious and strategic attacks such as jamming attack and Denial-of-Service (DoS) attack which are subject to different information pattern comparing to zero-sum game. In security domain, Stackelberg game is a more reasonable model [9], [10], where player 1 (always denoted as leader in the game) commits to its strategy first and player 2 (always denoted as follower in the game) observes leader's strategy and then plays its best response. Stackelberg game can capture the information asymmetry and model the value of information.

In this paper, we consider a stochastic discrete-time system with the presence of an adversary, which is abstracted as a stochastic game (SG). The system is given a set of specifications modeled in LTL co-safe (scLTL). We focus on the scenario where no controller can be synthesized to satisfy the specifications simultaneously due to either incompatibility between specifications or the presence of the adversary. Thus we aim at the minimum violation control strategy synthesis problem, i.e., compute a control strategy that violates the less important specifications and satisfies the most important specifications based on user's preference [11]. To the best of our knowledge, this is the first attempt to analyze minimum violation control synthesis on stochastic system in the presence of adversary. To summarize, we make the following contributions. We formulate a stochastic game to model the interaction between the controller and adversary. We give examples for typical attacks in CPS that can be incorporated into our proposed framework. To model limited observation capability of human adversary, anchoring bias is considered. We present the completion procedure to augment each automaton associated with each specification given to the system. We calculate the product SG using the completed automaton and SG. We formulate a nonlinear programming problem on the product SG to calculate the

L. Niu, J. Fu and A. Clark are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA. {lniu,jfu2,aclark}@wpi.edu

This work was supported by NSF grant CNS-1656981.

optimal control policy. A heuristic algorithm is proposed to compute an approximate solution. The proposed algorithm significantly saves computation cost and memory cost. The convergence of the algorithm is proved. A numerical case study is used to demonstrate the proposed approach. By using the proposed approach, more specifications can be satisfied when considering the presence of the adversary. Finally, we show the relationship between the controller's expected utility and the anchoring bias parameter of adversary.

The remainder of this paper is organized as follows. Related work is presented in Section II and preliminary backgrounds are presented in Section III. Section IV presents problem formulation. We give solution method in Section V. A numerical case study is given in Section VI. We conclude this paper in Section VII.

II. Related Work

Control synthesis under temporal logic constraints normally assumes the specifications can be satisfied. Contributions on the cases when the specifications cannot be fulfilled can be classified into four categories. First, the minimum violation problem for deterministic system has been studied in [11]–[14]. Violations caused by confliction between specifications have been studied in [11]–[13], and a control strategy that satisfies the most important specifications is synthesized. In [14], a two-player concurrent Stackelberg differential game is formulated. Quantitative preference over satisfactions of scLTL is investigated in [15]. However, contributions [11]–[15] focus on deterministic systems and hence the proposed approaches are not applicable to stochastic systems. Second, unsynthesizable specifications are analyzed in [5]. Third, model repair problem is investigated so that satisfaction on specifications is guaranteed [16], [17]. Finally, specification revision problem is investigated in [18]. Planning revision under temporal logic specification is investigated in [19]. However, none of the aforementioned papers consider the presence of adversary. Furthermore, non-deterministic automata are used in the aforementioned papers while deterministic automata are used in this paper.

Secure control in adversarial environment has been investigated using both control theoretic based approach [20] and game theoretic methods [10], [21]. When game theory meets temporal logic, turn-based two-player SG has been used to construct model checker [22] and model checking framework [23], [24]. The difference is that a general sum concurrent SG is considered in this paper. Secure control under LTL formula specification modeling liveness and safety constraints is considered in [25]. The proposed approach in [25] focuses on liveness and safety constraints, while this paper considers specifications modeled using scLTL.

III. Preliminaries

In this section, we present backgrounds on linear temporal logic and stochastic games.

A. Linear Temporal Logic (LTL)

An LTL formula consists of a set of atomic propositions Π, boolean operators including negation ($\neg$), conjunction ($\wedge$) and disjunction ($\vee$) and temporal operators including next (X) and until ($\mathcal{U}$) [26]. An LTL formula is defined inductively as

$$\phi = True \mid \pi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid X\phi \mid \phi_1 \; \mathcal{U} \; \phi_2.$$

Other operators such as implication ($\implies$), eventually (F) and always (G) can be defined using operators above. In particular, $\phi \implies \psi$ is equivalent to $\neg\phi \vee \psi$, $F\phi$ is equivalent to $True \; \mathcal{U} \; \phi$, and $G\phi$ is equivalent to $\neg F \neg \phi$.

The semantics of LTL formulas are defined over infinite words in 2^{Π}. Informally speaking, $G\phi$ is true if and only if ϕ is true for the current time step and all future time. $F\phi$ is true if and only if ϕ is true at some future time. $X\phi$ is true if and only if ϕ is true in the next time step. A word η satisfying an LTL formula ϕ is denoted as $\eta \models \phi$.

In this paper, we focus on syntactically co-safe LTL (scLTL) formulas.

Definition 1. (scLTL [27]): *Any string that satisfies a scLTL formula consists of a finite string (a good prefix) followed by any infinite continuation. This continuation does not affect the formula's truth value.*

By Definition 1, a word η satisfies an scLTL formula ϕ if it contains a good prefix $\eta_0\eta_1 \cdots \eta_n$ such that $\eta_0\eta_1 \cdots \eta_n \eta_{n+1}\eta_{n+2} \cdots \models \phi$ for any suffix $\eta_{n+1}\eta_{n+2} \cdots$.

For each scLTL formula, a deterministic finite automaton (DFA) can be obtained. A DFA is defined as follows.

Definition 2. (Deterministic finite automaton): *A DFA $\mathcal{A}$ is a tuple $\mathcal{A} = (Q, q_0, \Sigma, \delta, F)$, where Q is a finite set of states, $q_0 \in Q$ is the initial state, Σ is alphabet, $\delta : Q \times \Sigma \to Q$ is the set of transitions and $F \subseteq Q$ is the set of accepting states.*

A run on a DFA $\mathcal{A}$ over a finite input word $\sigma = \sigma_0\sigma_1 \cdots \sigma_n$ is a sequence of states $Q^* = q_0 q_1 \cdots q_n$ such that $\delta(q_{k-1}, \sigma_k) = q_k$ for all $0 \leq k \leq n$. A run is accepting if $q_n \in F$. The satisfaction of a formula ϕ by a run σ is denoted as $\sigma \models \phi$. To enable violations on specifications, we assume any DFA $\mathcal{A}$ is complete, i.e., for any $q \in Q$ and $\sigma \in \Sigma$, $\delta(q, \sigma)$ is defined. The completion procedure can be achieved by adding an additional $sink$ state and let $\delta(q, \sigma) = sink$ if $\delta(q, \sigma)$ is undefined.

B. Stochastic Game (SG)

A Stochastic Game (SG) is defined as follows.

Definition 3. (Stochastic Game): *A stochastic game $\mathcal{G}$ is a tuple $\mathcal{G} = (S, U_C, U_A, Pr, \Pi, \mathcal{L})$, where S is a finite set of states, U_C is a set of actions of the controller, U_A is a set of actions of an adversary, $Pr : S \times U_C \times U_A \times S \to [0, 1]$ is a transition function where $Pr(s, u_C, u_A, s')$ is the probability of a transition from state s to state s' when the controller takes action u_C and the adversary takes action u_A. Π is a* *set of atomic propositions. $\mathcal{L} : S \to 2^{\Pi}$ is a labeling function mapping each state to a subset of propositions in Π.*

Denote the admissible actions as the set of actions available to the controller (resp. adversary) at each state $s \in S$ as $U_C(s)$ (resp. $U_A(s)$). A finite (resp. infinite) path on SG $\mathcal{G}$ is a finite (resp. infinite) sequence of states denoted as $Path_{fin} = s_0 s_1 \cdots s_n$(resp. $Path_{inf} = s_0 s_1 s_2 \cdots$). Let $Path$ be the set of finite paths. A control policy $\mu : Path \times U_C \to \mathbb{R}$ (resp. adversary policy $\lambda : Path \times U_A \to \mathbb{R}$) is a function specifying the probability distribution over control (resp. attack) actions given historical trajectory $Path_{fin}$. An admissible policy is the policy whose support is the set of admissible actions at each state. In particular, we consider a *memoryless* control(resp. adversary) policy in this paper, i.e., μ (resp. λ) depends only on the current state.

Stackelberg SG is a widely adopted model in security domain [9]. In the Stackelberg setting, one player is the leader and another player is the follower. The leader first commits to a strategy μ. The follower then observes the strategy μ and play its best response λ. Given any control policy μ, the best response from the adversary is represented as $\mathcal{BR}(\mu) = \{\lambda | \lambda = \text{argmax}_\lambda \mathcal{T}_A(\mu, \lambda)\}$, where $\mathcal{T}_A(\mu, \lambda)$ is the adversary's utility given a pair of leader-follower strategies. The Stackelberg equilibrium is defined as follows.

Definition 4. (Stackelberg Equilibrium (SE)): *Denote the utility that the leader (resp. follower) gains in a stochastic game $\mathcal{G}$ under leader follower strategy pair (μ, λ) as $\mathcal{T}_C(\mu, \tau)$ (resp. $\mathcal{T}_A(\mu, \tau)$). A pair of leader follower strategy (μ, λ) is an SE if leader's strategy μ is optimal given that the follower observes its strategy and plays its best response, i.e., $\mu \in \text{argmax}_{\mu' \in \boldsymbol{\mu}} \mathcal{T}_C(\mu', \mathcal{BR}(\mu'))$, where $\boldsymbol{\mu}$ is the set of all admissible policies of the controller and $\lambda \in \mathcal{BR}(\mu')$ denotes the best response to the leader's strategy μ' from the follower.*

In Stackelberg games with human adversaries, *anchoring bias* is used to model the confidence of the adversary in its observations on μ [28]. When considering anchoring bias, the response λ might not be the best response to control policy μ. Human adversaries normally assign uniform probability to the control action at each state [28]. When more information is obtained via observation, adversaries slowly update the distributions. In this paper a linear model is adopted to represent the estimated probability. In this model, the estimated probability of human adversary that the controller takes action u_C at each state is calculated as

$$\tilde{\mu}(s, u_C) = \alpha \frac{1}{|U_C(s)|} + (1 - \alpha)\mu(s, u_C), \ \forall s, u_C \qquad (1)$$

where $\alpha \in [0, 1]$ is a parameter to tune the balance between the original and true probability. When $\alpha = 0$, the estimated probability becomes the true probability and thus the adversary plays its best response. When $\alpha = 1$, then the estimated probability becomes the uniform distribution, implying the adversary has no capability to observe or infer the control policy based on his observation.

IV. Problem Formulation

In this section, we first present the problem formulation. Then we show that several typical CPS security problems can be analyzed using the proposed framework. We consider a finite-state discrete-time system in the presence of an adversary, which can be abstracted using a SG $\mathcal{G}_0 = (S, U_C, U_A, Pr, \Pi, \mathcal{L})$ as defined in Definition 3.

We adopt the concurrent Stackelberg setting. In particular, the controller acts as the leader and the adversary is the follower. The controller first commits to its strategy (or control policy) μ. Then the adversary, who observes the historical behavior of the controller, plays its response λ to the control policy μ. We assume that both the controller and adversary can observe current state s. At each system state s, both the controller and adversary have to take actions simultaneously and the system evolves to state s' following transition function defined in Definition 3.

The system is assigned a set of specifications $\Phi = \{\phi_1, \phi_2, \cdots, \phi_n\}$ modeled using scLTL [12], [13]. By satisfying each specification $\phi_i \in \Phi$, the controller gains a reward $r(\phi_i)$. The objective of the controller is to maximize the total reward obtained via satisfying specifications. In the worst case, the adversary attempts to deviate system behavior and drive the system to violate specifications in Φ so as to minimize the total reward obtained by system. Hence, the specifications cannot be satisfied simultaneously due to either incompatibility of specifications or the presence of adversary. Thus we investigate the minimum violation problem on such a system as follows.

Problem 1. *Given an SG $\mathcal{G}_0$ abstracted from the system in the presence of an adversary and a set of specifications $\Phi = \{\phi_1, \cdots, \phi_n\}$ that potentially cannot be satisfied by system simultaneously, with each $\phi_i \in \Phi$ associated with a reward function $r(\phi_i)$, compute a control policy μ such that μ and the best response from adversary $\lambda \in \mathcal{BR}(\tilde{\mu})$ constitutes SE defined in Definition 4.*

In the following, we show several problems in security domain can be formulated using our proposed framework.

1) Patrolling Security Game with single type of adversary [29]: The states S are set as locations in PSG. The actions U_C and U_A are the actions available to the patrol unit and adversary, respectively. In particular, U_C includes the actions that transit the patrol unit among the locations, while U_A are the intrusion actions modeling which location is targeted by the adversary. The transition probability captures the transition uncertainty. The actions taken by both players jointly determine their utilities. For instance, the adversary wins if the target region is under attack without protection and the patrol unit wins otherwise.

The interaction between the patrol unit and adversary is modeled as a Stackelberg game. The security force is the leader while the adversary is the follower. The adversary can observe the schedule of security force (by waiting outside the environment indefinitely) and play its best response.

By using our proposed framework, task dependent rewards can be defined and thus more complex behaviors of the patrolling unit can be considered. For example, the patrolling unit can be given the following tasks: visit areas in sequence (e.g., 'First region A then region B then region C': $F\ (A \wedge (F\ B \wedge F\ C))$) and reactivity (e.g., 'if some passenger enters prohibited region, stop them': $prohibited \implies stop$).

2) Jamming Attacks on CPS: Applications such as SCADA networks and remotely controlled UAVs can be modeled as CPS where the controller communicates with the plant via a wireless network corrupted by a strategic jamming attacker.

Let the state of the plant evolves following a finite state discrete-time dynamics $x(k+1) = Ax(k) + Bu(k) + \omega(k)$, $k = 0, 1\cdots$, where $x(k)$ is the system state, $u(k) = \Gamma(u_C(k), u_A(k))$ is the system input jointly determined by the control signal $u_C(k)$ and adversary signal $u_A(k)$ for all k and $\omega(k)$ is stochastic disturbance. Function $\Gamma(u_C(k), u_A(k))$ can be formulated as: (i) $\Gamma(u_C(k), u_A(k)) = u_A(k) \cdot u_C(k),\ u_A(k) \in \{0, 1\},\ \forall k$ [30], or (ii) $\Gamma(u_C(k), u_A(k)) = u_A(k) + u_C(k),\ \forall k$ [31]. The formulation of (i) models scenario where the adversary can cause collision at the receiver equipped on the plant and result in denial-of service (DoS) attack. The formulation in (ii) models the scenario where the adversary can flip several bits in the packet and result in false information at the plant. Note that when the adversary launches DoS attack, the actuator can generate no input for the plant as $u(k) = 0$ when $u_A(k) = 0$ [30], or $u(k) = u(k-1)$ when $u_A(k) = 0$.

Consider the example of an autonomous UAV. The reachability specification can be given to the UAV as 'eventually reach target region and avoid obstacles', i.e, $G\neg\ obstacle\ \wedge\ F\ target$.

V. Proposed Solution for Problem 1

In this section, we first present a mixed integer nonlinear programming (MINLP) formulation. Then we propose a heuristic solution to compute a *proper* stationary control policy, which will be defined later.

For each specification ϕ_i, a complete DFA $\mathcal{A}_i = (Q^i, q_0^i, \Sigma, \delta^i, F^i)$ can be constructed. Given the set of complete automata $\mathcal{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_n\}$ with each $\mathcal{A}_i$ associated with ϕ_i, we can construct a product automaton using the following definition [26].

Definition 5. (Product automaton): *A product automaton obtained from $\mathcal{A}$ is a tuple $\mathcal{A}_\mathcal{P} = (Q_\mathcal{P}, q_{0,\mathcal{P}}, \Sigma, \delta_\mathcal{P}, F_\mathcal{P})$, where $Q_\mathcal{P} = Q^1 \times \cdots \times Q^n$ is a finite set of states, $q_{0,\mathcal{P}} = (q_0^1, \cdots, q_0^n)$ is the initial state, Σ is the alphabet inherited from $\mathcal{A}$, $\delta_\mathcal{P} = \Big((q^1, \cdots, q^n), \sigma, (q^{1\prime}, \cdots, q^{n\prime})\Big)$ if $\delta^i(q^i, \sigma) = q^{i\prime}$ for all $i\}$ and $F_\mathcal{P} = \big\{(q^1, \cdots, q^n) | q^i \in F^i,\ \forall i\big\}$ is the set of accepting states.*

Given the SG $\mathcal{G}_0$ and product automaton $\mathcal{A}_\mathcal{P}$, we can construct a product SG $\mathcal{G}$ defined as follows.

Definition 6. (Product SG): *Given SG $\mathcal{G}_0 = (S, U_C, U_A, Pr, \mathcal{L}, \Pi)$ and product automaton $\mathcal{A}_\mathcal{P} = (Q_\mathcal{P}, q_{0,\mathcal{P}}, \Sigma, \delta_\mathcal{P}, F_\mathcal{P})$, a (weighted and labeled) product SG is a tuple $\mathcal{G} = (S_\mathcal{P}, U_C, U_A, Pr_\mathcal{P}, Acc, W)$, where $S_\mathcal{P} = S \times Q_\mathcal{P}$ is a finite set of states, U_C (resp. U_A) is a finite set of control inputs (resp. attack signals), $Pr_\mathcal{P}((s, q^1, \cdots, q^n), u_C, u_A, (s', q^{1\prime}, \cdots, q^{n\prime})) = Pr(s, u_C, u_A, s')$ if $((q^1, \cdots, q^n), \mathcal{L}(s'), (q^{1\prime}, \cdots, q^{n\prime})) \in \delta_\mathcal{P}$, $Acc = S \times F_\mathcal{P}$, and W is a weight function assigning each transition a reward.*

The weight function of product SG $\mathcal{G}$ is defined as

$$W((s, q^1, \cdots, q^n), u_C, u_A, (s', q^{1\prime}, \cdots, q^{n\prime})) = \sum_{i=1}^{n} I_{ii'} r(\phi_i), \quad (2)$$

where the indicator $I_{ii'} = 1$ if $q^i \notin F^i$ and $q^{i\prime} \in F^i$ and $I_{ii'} = 0$ otherwise. By the definition (2), we have that a trace τ collects reward by satisfying specifications if a specification is satisfied at first time. We index the states in the product SG $\mathcal{G}$ as $s_\mathcal{P}$.

A proper control policy on product SG is defined as follows.

Definition 7. (Proper Policies): *A stationary control policy μ is proper if under μ, regardless of the policy chosen by the adversary, the set of destination states can eventually be reached with positive probability, where a destination state $s_\mathcal{P} = (s, q^1, \cdots, q^n)$ is a state such that q^i is an absorbing state in automaton $\mathcal{A}_i$ for all i.*

If a control policy μ' is improper, then under policy μ', there exists some state $s_\mathcal{P}$ that has zero probability to reach the set of destination states.

A. MINLP Formulation

For the controller's strategy, since randomized stationary strategies are considered in Problem 1, we have that

$$\mu(s_\mathcal{P}, u_C) \geq 0,\ \forall s_\mathcal{P} \in S_\mathcal{P}, u_C \in U_C, \quad (3)$$

$$\sum_{u_C \in U_C(s_\mathcal{P})} \mu(s_\mathcal{P}, u_C) = 1,\ \forall s_\mathcal{P} \in S_\mathcal{P}, \quad (4)$$

$$\lambda(s_\mathcal{P}, u_A) \in \{0, 1\},\ \forall s_\mathcal{P} \in S_\mathcal{P}, u_A \in U_A, \quad (5)$$

$$\sum_{u_A \in U_A(s_\mathcal{P})} \lambda(s_\mathcal{P}, u_A) = 1,\ \forall s_\mathcal{P} \in S_\mathcal{P}, \quad (6)$$

where (4) and (6) guarantees that the probability distribution sums to one. Eq. (5) holds since in Stackelberg games, it is sufficient to consider pure strategies for the follower [32].

The value function for the controller $V_C(s_\mathcal{P})$ (resp. adversary $V_A(s_\mathcal{P})$) is defined as the expected reward for the controller (resp. adversary) starting from state $s_\mathcal{P}$. The value functions can be characterized using the following lemma.

Lemma 1. *The expected reward of the controller and adversary induced by policy μ and λ can be represented as*

$$V_C(s_\mathcal{P}) = \sum_{u_C \in U_C} \Big[\mu(s_\mathcal{P}, u_C) \sum_{u_A \in U_A} \lambda(s_\mathcal{P}, u_A) \sum_{s'_\mathcal{P}} Pr_\mathcal{P}(s_\mathcal{P}, u_C, u_A, s'_\mathcal{P})(W(s_\mathcal{P}, u_C, u_A, s'_\mathcal{P}) + V_C(s'_\mathcal{P}))\Big],$$

$$V_A(s_{\mathcal{P}}) = \sum_{u_C \in U_C} \big[\tilde{\mu}(s_{\mathcal{P}}, u_C) \sum_{u_A \in U_A} \lambda(s_{\mathcal{P}}, u_A) \sum_{s'_{\mathcal{P}}} Pr_{\mathcal{P}}(s_{\mathcal{P}}, u_C, u_A, s'_{\mathcal{P}})(-W(s_{\mathcal{P}}, u_C, u_A, s'_{\mathcal{P}}) + V_A(s'_{\mathcal{P}}))\big].$$

Moreover, given a pair of policies μ and λ, the expected reward of the controller and adversary are the unique solutions to the linear equations above.

Based on Lemma 1, we have the following proposition which gives the sufficiency of considering proper policies.

Proposition 1. *If a proper control policy μ' is associated with the highest expected reward for the controller among all proper policies, then it associates with the highest expected reward among all stationary policies.*

We omit the proofs for Lemma 1 and Proposition 1 due to space limit. See [33] for detailed proofs. By Proposition 1, we can restrict the search space of control policy to the set of proper control policies. Denote the expected reward obtained by the controller starting from state $s_{\mathcal{P}}$ when the controller commits to strategy μ and adversary takes action u_A as $B_C(s_{\mathcal{P}}, \mu, u_A)$. Define $B_A(s_{\mathcal{P}}, \tilde{\mu}, u_A)$ for the adversary analogously. Then for all $s_{\mathcal{P}} \in S_{\mathcal{P}}, u_A \in U_A$, the expected reward for the controller (resp. adversary) can be represented as

$$B_C(s_{\mathcal{P}}, \mu, u_A) = \sum_{u_C \in U_C} \mu(s_{\mathcal{P}}, u_C)\big[\sum_{s'_{\mathcal{P}}} Pr_{\mathcal{P}}(s_{\mathcal{P}}, u_C, u_A, s'_{\mathcal{P}}) (W(s_{\mathcal{P}}, u_C, u_A, s'_{\mathcal{P}}) + V_C(s'_{\mathcal{P}}))\big], \tag{7}$$

$$B_A(s_{\mathcal{P}}, \tilde{\mu}, u_A) = \sum_{u_C \in U_C} \tilde{\mu}(s_{\mathcal{P}}, u_C)\big[\sum_{s'_{\mathcal{P}}} Pr_{\mathcal{P}}(s_{\mathcal{P}}, u_C, u_A, s'_{\mathcal{P}}) (-W(s_{\mathcal{P}}, u_C, u_A, s'_{\mathcal{P}}) + V_A(s'_{\mathcal{P}}))\big], \tag{8}$$

which are the expected utility of the controller and adversary, respectively. Note that the adversary's expected reward depends on its observation over the control policy $\tilde{\mu}$ defined in (1). Since λ is binary, we can bound the values for the adversary and controller using the big M method [32], respectively, for all $s_{\mathcal{P}}$ and u_A as follows:

$$B_A(s_{\mathcal{P}}, \tilde{\mu}, u_A) \le V_A(s_{\mathcal{P}}) \le B_A(s_{\mathcal{P}}, \tilde{\mu}, u_A) + (1 - \lambda(s_{\mathcal{P}}, u_A))Z, \tag{9}$$

$$V_C(s_{\mathcal{P}}) \le B_C(s_{\mathcal{P}}, \mu, u_A) + (1 - \lambda(s_{\mathcal{P}}, u_A))Z, \tag{10}$$

where Z is a sufficiently large positive number. Inequality (9) and (10) give bounds for $V_A(s_{\mathcal{P}})$ and $V_C(s_{\mathcal{P}})$. Depending on the value of λ, the upper bounds for $V_C(s_{\mathcal{P}})$ (resp. $V_A(s_{\mathcal{P}})$) can be either infinity ($\lambda(s_{\mathcal{P}}, u_A) = 1$) or $B_C(s_{\mathcal{P}}, \mu, u_A)$ (resp. $B_A(s_{\mathcal{P}}, \mu, u_A)$).

To compute the control policy that maximizes the expected utility of controller, the following optimization problem can be formulated [32].

$$\max_{\mu, \lambda, V_C, V_A} \quad \gamma^T V_C \tag{11}$$
$$\text{s.t.} \quad (1)\ (3)\ (4)\ (5)\ (6)\ (9) \text{ and } (10)$$

where γ is the initial distribution over state space $S_{\mathcal{P}}$. Since constraints (9) and (10) introduce nonlinearity and λ is binary, the optimization problem (11) is an MINLP.

B. Heuristic Solution

The MINLP (11) is nonconvex and solving it is NP-hard. In the following, we present a value iteration based heuristic solution to the MINLP (11).

Algorithm 1 Algorithm for computing a control strategy μ that maximizes the expected reward V_C.

```
1: Let H ← {V_{C,1}, ··· , V_{C,t}, ··· , V_{C,|H|}}, V ← ∅,
2: for V_{C,t} ∈ H do
3:     k ← 0
4:     repeat
5:         Solve MILP (12) to obtain expected reward V_C^k.
6:         k ← k + 1
7:     until γ^T V_C^k − γ^T V_C^{k−1} ≤ ε or MILP (12) is infeasible.
8:     if γ^T V_C^k − γ^T V_C^{k−1} ≤ ε then
9:         V ← V ∪ {γ^T V_C^k}
10:    end if
11: end for
12: if V = ∅ then
13:     Return to step 2
14: else
15:     t* ← argmax{V_{C,t} : t = 1, 2, ··· , |H|}
16:     μ ← policy obtained from V_{C,t*}
17:     return μ
18: end if
```

As shown in Algorithm 1, we first initialize an arbitrary set of initial policies using sampling approach, where the sample space is the product of $|S_{\mathcal{P}}|$ probability simplices in $\mathbb{R}^{|U_C|}$. Then by solving the optimal control problem from the perspective of adversary on the MDP induced by each control policy [1]–[4], we can solve for a set of expected rewards for the controller $\mathcal{H} = \{V_{C,1}, \cdots, V_{C,t}, \cdots, V_{C,|\mathcal{H}|}\}$ associated with the initial policies.

For each initial expected reward $V_{C,t} \in \mathcal{H}$, value iteration (line 3 to line 7) is used to find a control policy such that the objective function $\gamma^T V_C$ is maximized. In particular, at iteration $k+1$, given the expected reward obtained from previous iteration V_C^k, the following mixed integer linear programming (MILP) is solved to calculate the proper control policy μ^{k+1}.

$$\max_{\mu, \lambda, V_C, V_A} \quad \gamma^T V_C \tag{12}$$
$$\begin{aligned} \text{s.t.} \quad & V_C(s_{\mathcal{P}}) \le B_C^k(s_{\mathcal{P}}, \mu, u_A) \\ & \qquad + (1 - \lambda(s_{\mathcal{P}}, u_A)Z, \ \forall s_{\mathcal{P}}, u_A \\ & B_A^k(s_{\mathcal{P}}, \mu, u_A) \le V_A(s_{\mathcal{P}}) \\ & \quad \le B_A^k(s_{\mathcal{P}}, \mu, u_A) + (1 - \lambda(s_{\mathcal{P}}, u_A)Z, \ \forall s_{\mathcal{P}}, u_A \\ & (1)\ (3)\ (4)\ (5)\ (6) \end{aligned}$$

where $B_C^k(s_{\mathcal{P}}, \mu, u_A)$ and $B_A^k(s_{\mathcal{P}}, \mu, u_A)$ are obtained by (7) and (8) using $V_C^k(s_{\mathcal{P}})$ and $V_A^k(s_{\mathcal{P}})$, respectively. Note that when solving the MILP (12), the policy chosen by the adversary is the best response to $\tilde{\mu}^k$ obtained from (1). The algorithm terminates when either $V_C^k - V_C^{k-1} \leq \epsilon$ or the MILP (12) is infeasible. The first termination condition focuses on the scenario where an optimal V_C can be found by solving the optimization problem. Since the initial guess is given arbitrarily while V_C is bounded within $[0, \sum_{\phi\in\Phi} r(\phi)]$, thus MILP (12) might be infeasible. In this case, such an initial guess should be skipped and the value iteration module terminates. After a feasible V_C is found at some iteration t, we store V_C in vector $\mathcal{V}$. Then the control policy returned by Algorithm 1 is the control policy corresponding to the maximum value in $\mathcal{V}$.

The convergence of Algorithm 1 is presented in the following theorem.

Theorem 1. *Algorithm 1 converges in finite time.*

Before presenting the proof of Theorem 1, we first introduce two operators denoted as $T_\mu : \mathbb{R}^{|S_\mathcal{P}|} \to \mathbb{R}^{|S_\mathcal{P}|}$ and $T : \mathbb{R}^{|S_\mathcal{P}|} \to \mathbb{R}^{|S_\mathcal{P}|}$ as follows:

$$T_\mu V_C(s_\mathcal{P}) = \min_{\lambda\in\mathcal{BR}(\tilde{\mu})} \sum_{u_C\in U_C} \mu(s_\mathcal{P}, u_C) \sum_{u_A\in U_A} \lambda(s_\mathcal{P}, u_A) \sum_{s'_\mathcal{P}} \left[Pr_\mathcal{P}(s_\mathcal{P}, u_C, u_A, s'_\mathcal{P})(W(s_\mathcal{P}, u_C, u_A, s'_\mathcal{P}) + V_C(s'_\mathcal{P}))\right], \tag{13}$$

$$TV_C(s_\mathcal{P}) = \max_{\mu} \min_{\lambda\in\mathcal{BR}(\tilde{\mu})} \sum_{u_C\in U_C} \mu(s_\mathcal{P}, u_C) \sum_{u_A\in U_A} \lambda(s_\mathcal{P}, u_A) \sum_{s'_\mathcal{P}} \left[Pr_\mathcal{P}(s_\mathcal{P}, u_C, u_A, s'_\mathcal{P})(W(s_\mathcal{P}, u_C, u_A, s'_\mathcal{P}) + V_C(s'_\mathcal{P}))\right], \tag{14}$$

The following lemmas characterizes the operator T_μ.

Lemma 2. *For any vectors V and V' such that $V \leq V'$, we have $T_\mu^k V \leq T_\mu^k V'$ for all policies μ and k, where $T_\mu^k(\cdot)$ iteratively applying T_μ operator k times.*

We omit the proof for Lemma 2 due to space limit. See [33] for detailed proofs.

Lemma 3. *Denote the expected reward induced by proper control policy μ and adversary policy $\lambda \in \mathcal{BR}(\tilde{\mu})$ as $V_C^{\mu,\lambda}$. Then $V_C^{\mu,\lambda}$ satisfies $\lim_{M\to\infty}(T_\mu^M V_C) = V_C^{\mu,\lambda}$.*

Proof. Since we focus on stationary policies, then by inducting Lemma 1, $T_\mu^M V_C$ can be represented as

$$T_\mu^M V_C = Pr^M V_C + \sum_{m=0}^{M-1} Pr^m \tilde{W}, \tag{15}$$

where Pr is the transition matrix of the Markov chain induced by control policy μ and adversary policy λ. Since the control policy μ is proper, we can eventually reach the set of destination states with probability 1. By definition (2), no reward can be collected when starting from destination states. Therefore, we have $\lim_{M\to\infty} Pr^M V_C = 0$. Then, by taking limit on both sides of (15) as M tends to infinity, we have $\lim_{m\to\infty} T_\mu^M V_C = \lim_{M\to\infty} \sum_{m=0}^{M-1} Pr^m \tilde{W}$. By the definition of $V_C^{\mu,\lambda}$, we have $\lim_{M\to\infty}(T_\mu^M V_C) = V_C^{\mu,\lambda}$, and hence Lemma 3 is proved. □

Finally, we have the following proposition.

Proposition 2. *The optimal expected total reward for the controller at each iteration k satisfies $V_C^k = TV_C^{k-1}$.*

Proof. Suppose the expected reward for the controller is $\bar{V}_C^k$ at some iteration k such that $\bar{V}_C^k \neq TV_C^{k-1}$. If $\bar{V}_C^k > TV_C^{k-1}$, we have that $\bar{V}_C^k$ is not a feasible solution to MILP (12). If $\bar{V}_C^k < TV_C^{k-1}$, then starting from $\bar{V}_C^k$, we can always search along some direction in the feasible region of (12) until we reach the boundary of the feasible region to find some $\hat{V}_C^k \geq \bar{V}_C^k$. Hence, $\bar{V}_C^k$ is not the optimal solution to (12). Therefore, we have $V_C^k = TV_C^{k-1}$ holds. □

In the following, we present the proof of Theorem 1.

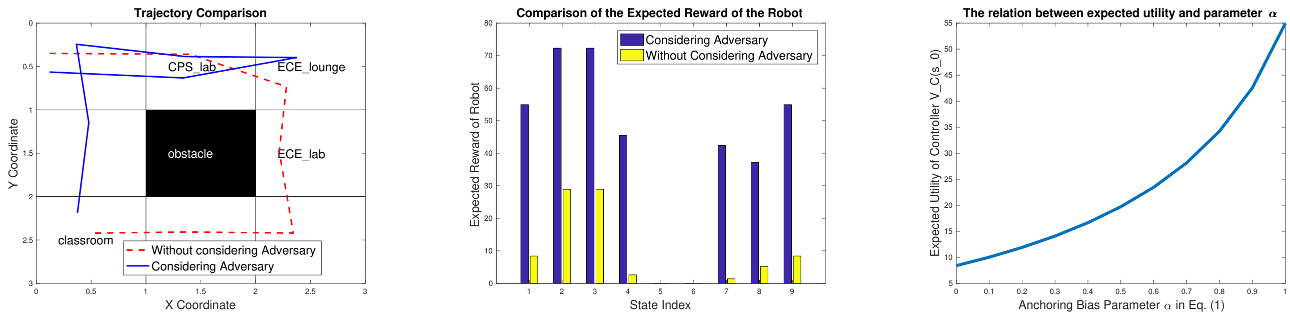
Proof. (Proof of Theorem 1.) We show that Algorithm 1 terminates within finite iterations because both outer and inner loops terminate within finite iterations.

First, the outer loop executes exactly $|\mathcal{H}|$ times and thus the outer loop terminates within finite iterations.

Next, we show at each outer loop iteration t, the value iteration module converges within finite time. It is obvious that the inner loop terminates when the initial guess on V_C is not feasible. In the following we focus on the feasible case. Let k be the iteration index of value iteration (line 3 to line 7). Let us denote the expected reward of the controller induced by control policy μ^k and adversary policy $\lambda^k \in \mathcal{BR}(\mu^k)$ at each iteration k as V_C^k. Let the expected reward of each transition starting from state $s_\mathcal{P}$ and the transition matrix under control policy μ^k and adversary policy λ^k be $\tilde{W}^k(s_\mathcal{P}) = \sum_{u_C}\sum_{u_A}\sum_{s'_\mathcal{P}} \mu^k(s_\mathcal{P}, u_C)\lambda^k(s_\mathcal{P}, u_A)W(s_\mathcal{P}, u_C, u_A, s'_\mathcal{P})$ and $Pr^k(s_\mathcal{P}, s'_\mathcal{P}) = \sum_{u_C} \mu^k(s_\mathcal{P}, u_C)\sum_{u_A}\lambda^k(s_\mathcal{P}, u_A)$ $Pr(s_\mathcal{P}, u_C, u_A, s'_\mathcal{P})$, respectively. By Lemma 1 and Proposition 2, we observe that $V_C^{k+1} = TV_C^k$ is equivalent to find a control policy μ^{k+1} such that $T_{\mu^{k+1}}V_C^k = TV_C^k$. Therefore $V_C^k = T_{\mu^k}V_C^k = \tilde{W}^k + Pr^k V_C^k \leq \tilde{W}^{k+1} + Pr^{k+1}V_C^k = T_{\mu^{k+1}}V_C^k$, where the inequality holds by definition (13) and (14), i.e., $T_{\mu^k}V_C^k \leq TV_C^k$. View V_C^k as $T_{\mu^{k+1}}^0 V_C^k$. Then composing $T_{\mu^{k+1}}$ m times and taking the limit as $m \to \infty$, by Lemma 2, we can construct a sequence of inequalities $V_C^k \leq T_{\mu^{k+1}}V_C^k, T_{\mu^{k+1}}V_C^k \leq T_{\mu^{k+1}}^2 V_C^k, \cdots, T_{\mu^{k+1}}^{m-1}V_C^k \leq T_{\mu^{k+1}}^m V_C^k$. Therefore, we have $V_C^k \leq \lim_{m\to\infty} T_{\mu^{k+1}}^m V_C^k = V_C^{k+1}$, where the convergence of T_μ^m follows from Lemma 3. Hence, the expected reward increases with respect to the number of iterations k. Since V_C is upper bounded by $\sum_{\phi\in\Phi} r(\phi)$, we claim that the value iteration module converges within finite time. □

Furthermore, we characterize the value function returned by Algorithm 1 using the following proposition.

Proposition 3. *The expected reward of the controller returned by Algorithm 1 is the value function obtained by committing to control strategy μ returned by Algorithm 1.*



(a) (b) (c)

Fig. 1: Fig. 1a shows the comparison of the trajectories. The trajectory in solid line is generated using the proposed approach. The trajectory in dotted line is generated without considering the presence of adversary. Fig. 1b shows the comparison of expected rewards obtained using different approaches. The blue bars are generated using the proposed approach. The yellow bars are generated without considering the presence of adversary. Fig. 1c shows the relationship between controller's expected reward and anchoring bias parameter.

The advantage of Algorithm 1 is that it significantly reduces computation and memory cost comparing to global optimization techniques [34] and discretization-based approximate algorithms [32]. Global optimization techniques, for example, spatial branch and bound has been demonstrated non-efficient comparing to MILP. The approximate solution proposed in [32] introduces extra binary variables and constraints, whose sizes are linear to the discretization resolution. The introduction of extra variables and constraints weakens its scalability, especially for the large state space in product SG. In contrast, Algorithm 1 introduces no additional variables when solving the MILP. Therefore, Algorithm 1 significantly saves memory and model construction time for commercial solvers. Algorithm 1 does not guarantee that a global optimal solution will be found. Hence, executing Algorithm 1 from different initial points can improve the performance of Algorithm 1.

VI. Case Study

In this section, we present a numerical case study to demonstrate the proposed approach.

A. Case Study Settings

Suppose a robot is performing tasks modeled in scLTL in a bounded environment. We consider the robot following standard discrete time model $x(t+1) = x(t) + (u_C(t) + u_A(t) + \vartheta(t))\,\Delta t$, where $x(t) \subset \mathbb{R}^2$ is the location of the robot at time t, $u_C(t) \in \mathcal{U} \subset \mathbb{R}^2$ is the control input from the controller, $u_A(t) \in \mathcal{A} \subset \mathbb{R}^3$ is the input signal from the adversary and $\vartheta(t) \subset \mathbb{R}^2$ is the stochastic disturbance, $\Delta t = t_{k+1} - t_k$ is the time interval. Therefore, we have that the control signal of the robot is compromised by the adversary. Here we let $\mathcal{A} \subset \mathcal{U}$.

We divide the region into 9 sub-regions with each size is $1m \times 1m$. We abstract the stochastic game as follows [25]. Let each sub-region be a state in the stochastic game. Hence, the stochastic game has 9 states and we will refer to state and sub-region interchangeably in the following. Each state can be mapped to a subset of atomic propositions by labeling function $\mathcal{L}$ as shown in Fig. 1a. The action sets for the controller and adversary are defined as $U_C = U_A = \{N, S, W, E\}$, implying moving towards the adjacent subregion. When the adversary compromises the control input, the probability that the robot transits to its intended state is 0.6. Moreover, when the robot is at ECE_lab, the adversary can block all the transitions of the robot (e.g., close the door of the room).

Suppose the robot is given 4 specifications $\Phi = \{\phi_1, \phi_2, \phi_3, \phi_4\}$ as shown in Table I. The robot is required to visit the CPS_lab or ECE_lab before visiting $classroom$. Moreover, they are required to be visited in this particular order if possible. In the meantime, the robot should avoid $obstacle$ during the visit to guarantee safety property. Finally, the robot is required to eventually visit ECE_lounge once it has visited CPS_lab.

B. Case Study Results

Let the upper left state in Fig. 1a be the initial state. Fig. 1a shows two trajectories generated using the proposed approach and the control policy synthesized without considering the presence of the adversary. Without considering the adversary, the control policy attempts to satisfy all the specifications in Φ. However, the adversary is capable to block all the transitions at state marked as ECE_lab. Therefore, following this policy can only satisfy specification ϕ_4. Our proposed approach takes the potential impacts from the adversary into consideration. By using the proposed approach, specifications ϕ_2 and ϕ_4 are satisfied. Hence, the robot can obtain higher reward by using the proposed

formula	$r(\phi_i)$
$G\neg obstacle \wedge F(CPS\_lab \wedge (FECE\_lab \wedge Fclassroom))$	50
$G\neg obstacle \wedge F(CPS\_lab \wedge Fclassroom)$	20
$G\neg obstacle \wedge F(ECE\_lab \wedge Fclassroom)$	20
$CPS\_lab \implies FECE\_lounge$	10

TABLE I: Specifications given to the robot. The specifications are indexed from ϕ_1 to ϕ_4 from top to bottom.

approach. The increment of the robot's expected reward achieved using our proposed approach is shown in Fig. 1b.

In Fig. 1c, we investigate the relationship between observation capability of the adversary and expected reward of the controller. We vary α in (1) from 0 to 1. When $\alpha = 0$, the adversary has unlimited observation capability and it has perfect knowledge of the controller's strategy. When $\alpha = 1$, the adversary makes no observation over the controller's strategy and it assumes the adversary plays uniform strategy. From Fig. 1c, we observe that the expected reward of the controller increases with respect to the reduction of adversary's observation capability. Hence the more observations the adversary makes, the lower expected reward the controller obtains.

VII. CONCLUSION

In this paper, we have investigated minimum violation problem on stochastic system in the presence of an adversary. The system is given a set of specifications modeled in scLTL. We model the interaction between the controller and adversary using a stochastic Stackelberg game. Moreover, to model the behavior of human adversaries, we consider anchoring bias. We rely on the concept of Stackelberg equilibrium to synthesize a control strategy. An efficient heristic algorithm is proposed to compute the control policy. We show the proposed algorithm converges in finite time and demonstrate the proposed approach using a numerical case study.

REFERENCES

[1] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," in *the Proc. of Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 2689–2696.

[2] M. Lahijanian, S. B. Andersson, and C. Belta, "Temporal logic motion planning and control with probabilistic satisfaction guarantees," *Transactions on Robotics*, vol. 28, no. 2, pp. 396–409, 2012.

[3] E. M. Wolff, U. Topcu, and R. M. Murray, "Robust control of uncertain markov decision processes with temporal logic specifications," in *the Proc. of Intl. Conf. on Decision and Control (CDC)*. IEEE, 2012, pp. 3372–3379.

[4] X. Ding, S. L. Smith, C. Belta, and D. Rus, "Optimal control of markov decision processes with linear temporal logic constraints," *Transactions on Automatic Control*, vol. 59, no. 5, pp. 1244–1257, 2014.

[5] V. Raman and H. Kress-Gazit, "Automated feedback for unachievable high-level robot behaviors," in *the Proc. of Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 5156–5162.

[6] K. O'Connell, "CIA Report: Cyber extortionists attacked foreign power grid, disrupting delivery," http://www.ibls.com/internet_law_news_portal_view.aspx?id=1963&s=latestnews.

[7] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, *et al.*, "Experimental security analysis of a modern automobile," in *Symp. on Security and Privacy (SP)*. IEEE, 2010, pp. 447–462.

[8] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.

[9] M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

[10] M. Zhu and S. Martinez, "Stackelberg-game analysis of correlated attacks in cyber-physical systems," in *the Proc. of American Control Conference (ACC)*. IEEE, 2011, pp. 4063–4068.

[11] J. Tumová, L. I. R. Castro, S. Karaman, E. Frazzoli, and D. Rus, "Minimum-violation ltl planning with conflicting specifications," in *the Proc. of American Control Conference (ACC)*. IEEE, 2013, pp. 200–205.

[12] J. Tumova, S. Karaman, C. Belta, and D. Rus, "Least-violating planning in road networks from temporal logic specifications," in *the Proc. of Intl. Conf. on Cyber-Physical Systems (ICCPS)*. IEEE, 2016, pp. 1–9.

[13] C.-I. Vasile, J. Tumova, S. Karaman, C. Belta, and D. Rus, "Minimum-violation scLTL motion planning for mobility-on-demand," in *theProc. of Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1481–1488.

[14] P. Chaudhari, T. Wongpiromsarny, and E. Frazzoli, "Incremental minimum-violation control synthesis for robots interacting with external agents," in *the Proc. of American Control Conference (ACC)*. IEEE, 2014, pp. 1761–1768.

[15] M. Lahijanian, S. Almagor, D. Fried, L. E. Kavraki, and M. Y. Vardi, "This time the robot settles for a cost: A quantitative approach to temporal logic planning with partial satisfaction." in *AAAI*, 2015, pp. 3664–3671.

[16] F. Buccafurri, T. Eiter, G. Gottlob, and N. Leone, "Enhancing model checking in verification by AI techniques," *Artificial Intelligence*, vol. 112, no. 1, pp. 57–104, 1999.

[17] E. Bartocci, R. Grosu, P. Katsaros, C. Ramakrishnan, and S. Smolka, "Model repair for probabilistic systems," *Tools and Algorithms for the Construction and Analysis of Systems*, pp. 326–340, 2011.

[18] K. Kim, G. Fainekos, and S. Sankaranarayanan, "On the minimal revision problem of specification automata," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1515–1535, 2015.

[19] M. Guo and D. V. Dimarogonas, "Reconfiguration in motion planning of single-and multi-agent systems under infeasible local LTL specifications," in *the Proc. of Intl. Conf. on Decision and Control (CDC)*. IEEE, 2013, pp. 2758–2763.

[20] Y. Shoukry, P. Nuzzo, A. Puggelli, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada, "Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach," *Transactions on Automatic Control*, vol. 62, no. 10, pp. 4917–4932, 2017.

[21] Q. Zhu and T. Basar, "Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: games-in-games principle for optimal cross-layer resilient control systems," *Control Systems*, vol. 35, no. 1, pp. 46–65, 2015.

[22] T. Chen, V. Forejt, M. Z. Kwiatkowska, D. Parker, and A. Simaitis, "Prism-games: A model checker for stochastic multi-player games." in *the Proc. of Intl. Conf. on TACAS*. Springer, 2013, pp. 185–191.

[23] T. Quatmann, C. Dehnert, N. Jansen, S. Junges, and J.-P. Katoen, "Parameter synthesis for markov models: Faster than ever," in *Intl. Symp. on Automated Technology for Verification and Analysis*. Springer, 2016, pp. 50–67.

[24] M. Kattenbelt, M. Kwiatkowska, G. Norman, and D. Parker, "A game-based abstraction-refinement framework for markov decision processes," *Formal Methods in System Design*, vol. 36, no. 3, pp. 246–280, 2010.

[25] L. Niu and A. Clark, "Secure control under linear temporal logic constraints," in *the Proc. of American Control Conference (ACC)*. IEEE, 2018, pp. 3544–3551.

[26] C. Baier, J.-P. Katoen, and K. G. Larsen, *Principles of Model Checking*. MIT Press, 2008.

[27] O. Kupferman and M. Y. Vardi, "Model checking of safety properties," *Formal Methods in System Design*, vol. 19, no. 3, pp. 291–314, 2001.

[28] C. R. Fox and Y. Rottenstreich, "Partition priming in judgment under uncertainty," *Psychological Science*, vol. 14, no. 3, pp. 195–200, 2003.

[29] N. Basilico, N. Gatti, and F. Amigoni, "Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder," *Artificial Intelligence*, vol. 184, pp. 78–123, 2012.

[30] A. Gupta, C. Langbort, and T. Başar, "Optimal control in the presence of an intelligent jammer with limited actions," in *the Proc. of Intl. Conf. on Decision and Control (CDC)*. IEEE, 2010, pp. 1096–1101.

[31] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal jamming attacks and network defense policies in wireless sensor networks," in *the Proc. of Intl. Conf. on Computer Communications*. IEEE, 2007, pp. 1307–1315.

[32] Y. Vorobeychik and S. P. Singh, "Computing stackelberg equilibria in discounted stochastic games." in *AAAI*, 2012.

[33] L. Niu, J. Fu, and A. Clark, "Minimum Violation Control Synthesis on Cyber-Physical Systems under Attacks," *ArXiv e-prints,* `arXiv: 1809.00975v1[cs.SY]`, Aug. 2018, https://arxiv.org/abs/1809.00975v1.

[34] S. Burer and A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surveys in Operations Research and Management Science*, vol. 17, no. 2, pp. 97–106, 2012.