2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Optimal Path Planning With Opacity-Preserving Temporal Logic Specifications Using Bipartite Synthesizers

Yiwei Zheng, Aiwen Lai, Weiyao Lan, and Xiao Yu

Abstract— **This paper investigates an optimal planning problem with the requirement of preventing high-level mission specifications from being revealed to the intruder. We assume that the behavior of the robotic system at some specific locations is partially observable, and the intruder is modeled as a passive observer for the observable behavior and the corresponding overall transition cost of the trajectories. We first use the transition system to model the robot system. Then, we say that the transition system is *LTL-based opaque* with respect to high-level mission specifications if the intruder cannot infer the exact behavior of the system via the observable sequence of the system. We design a synthesizer for the product automaton of the transition system and the Büchi automaton to find the evolutions of possible reachable states. Based on the synthesizer, the corresponding run can be synthesized in which the optimality, correctness, and opacity can be guaranteed.**

I. Introduction

Path planning, aimed at guiding robotic systems to reach a particular objective via the synthesized trajectories, is one of the important research problems and can be implemented with ground, aerial, and surface robotic systems [1]–[3]. It contributes to reducing labor costs and increasing reliability and scalability [4], [5].

Formal method provides a user-friendly and mathematically precise manner to solve path planning with temporal logic specifications, synthesizing optimal runs for the abstracted robotic systems. Currently, research mainly focuses on the optimal path planning problem satisfying particular mission specifications as well as some certain subtask requirements. Given a mission specification in the form of temporal logic, then the optimal runs and corresponding control strategies will be automatically synthesized with formal correctness [6]–[8]. In [6] and [7], the optimal path planning problem with temporal logic specifications was transformed into the shortest path problem on a directed weighted graph. In [9], a heuristic search method was proposed to synthesize end-to-end trajectories for the system. Although these methods are effective in synthesizing correct-by-construction trajectories under the temporal logic specifications, the requirement of information-flow security is not considered in these methods.

However, in order to achieve complex global tasks, the robotic system may need to acquire or transmit information with a centralized controller. In this process, malicious cyberattacks may occur such that critical information can be leaked [10]. For discrete-event systems, the notion of opacity is introduced to model the plausible deniability of the secret of a system. Along the research line of opacity, the secret of a system is modeled as a set of secret states, and the intruder is modeled as an eavesdropper which passively observes the observable behaviors of the system [11], [12]. A system is said to be opaque if the intruder cannot determine the visited secret states based on the observation. Various existing methods have been developed for the enforcement of opacity for non-opaque systems [13]–[16]. In [17], the structure of all-inclusive controller for opacity (AIC-O) was constructed for the enforcement of current-state opacity. In [18], publicly known nondeterministic edit functions were applied for the enforcement of opacity. In [19], the dynamic mask was synthesized for infinite-step opacity. Although these methods are effective in converting non-opaque systems to opaque, yet the method for enforcement of opacity is proposed for a set of static secrets, rather than preserving the intention of the system.

This paper focuses on designing a path planning method for a robotic system with temporal logic specifications, in which the mission specifications will not be revealed to the intruder via synthesized trajectories. To ensure the information security of the robotic system, we extend the notion of opacity and propose the definition of *LTL-based opacity* for the synthesized run of the transition systems. We say that the synthesized run is LTL-opaque if the passive intruder cannot infer the LTL specifications via the partial observation of the run. Then, to synthesize the optimal opaque run for the system, a new structure named synthesizer is constructed with respect to the given mission specifications. With the aid of the synthesizer, we find all sequences which guarantee the requirement of opacity. Finally, the optimal run can be found from the set of sequences.

The main contributions of this paper are summarized as follows. First, we propose a new type of information flow security named *LTL-based opacity*. Compared to [20], the objective is to synthesize the run in which the high-level mission specifications of the synthesized run are indistinguishable from the intruder. In this new notation of *LTL-based opacity*, the intruder is modeled as a passive observer and infers the given LTL specifications, and the run is LTL-based opaque if the intruder cannot determine the high-level mission specifications via the observable traces and the overall transition costs. Second, we propose a correct-by-construction method to synthesize the optimal opaque run for

This work was supported in part by the National Key R&D Program of China under Grant 2021ZD0112600, and in part by the National Natural Science Foundation of China under Grants 62173283, 62103344, and 62273285. (*Corresponding author: Xiao Yu.*)

The authors are with the Department of Automation, Xiamen University, Xiamen 361005, China. (e-mail: ywzheng1201@stu.xmu.edu.cn, aiwenlai@xmu.edu.cn, wylan@xmu.edu.cn, xiaoyu@xmu.edu.cn).

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

a robotic system and a general type of LTL specifications. We design a new synthesizer for the product system to estimate possible reachable states under the proposition of the LTL formula. Then, all sequences of states satisfying the requirements of correctness and opacity can be extracted, and the optimal run can be found within these cycles. Compared to [21] and [22], the goal is to synthesize the optimal run such that the intruder cannot identify whether the fixed secret locations have been visited, or which agent has visited the secret locations. In the proposed method, a more general type of secret is taken into consideration, which is a dynamic set of states under the mission specifications.

The remainder of this paper is organized as follows. Section II introduces the preliminaries of the transition system and LTL specifications. Section III proposes the definition of *LTL-based opacity* and the problem formulation for the synthesized runs. Section IV constructs the synthesizer and develops the algorithm that finds the optimal opaque run based on the synthesizer. Section V demonstrates a case study to illustrate the effectiveness of the synthesis algorithm. Section VI makes the conclusion and presents the future research directions.

II. Preliminaries and Problem Formulation

Consider a mobile robot deployed in a road network, which can be modeled as a *weighted transition system* (TS)

$$\mathbf{T} := (\mathcal{Q}, q^0, \delta, \Pi, L, w)$$

where

(i) $\mathcal{Q}$ is a finite set of states,
(ii) $q_0 \in \mathcal{Q}$ is the initial states,
(iii) $\delta \subseteq \mathcal{Q} \times \mathcal{Q}$ is the transition relation,
(iv) Π is a finite set of atomic propositions (APs),
(v) $L : \mathcal{Q} \to 2^{\Pi}$ is a map giving the set of APs satisfied in a state,
(vi) $w : \delta \to \mathbb{R}^+$ is a map assigning a positive integer weight to each transition.

For TS $\mathbf{T}$ abstracted from a robotic system and the corresponding environment, the transition relation δ and states $\mathcal{Q}$ are the road network and the intersections, respectively, the set of APs is defined based on the behavior of the robot on specific states, and the map of transition weight w corresponds to the traveling times between intersections. The full set of the runs of the TS $\mathbf{T}$ is a set of infinite sequences $R_{\mathbf{T},i} = \{q_0 q_1 \dots q_k \dots \in \mathcal{Q}^*\}$, where $k \in \mathbb{N}$, and $\mathcal{Q}^* = \cup_{j\in\mathbb{N}} q = q \cup qq \cup qqq \cup \dots, \forall q \in \mathcal{Q}$ denotes the Kleene closure of the set of states $\mathcal{Q}$. Then, we denote the run of TS $\mathbf{T}$ as $r \in R_{\mathbf{T},i}$.

For TS $\mathbf{T}$, the high-level mission specifications are described by linear temporal logic (LTL) formulas. An LTL formula consisting of a set of APs Π, boolean operators, and temporal operators, is formed according to the following syntax,

$$\phi := \text{TRUE} \mid \alpha \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \bigcirc\phi \mid \phi_1 \,\mathcal{U}\, \phi_2,$$

where $\alpha \in \Pi$ is an AP, and temporal operators $\bigcirc$ and $\mathcal{U}$ mean "next" and "until", respectively. Based on the aforementioned definitions, the other temporal operators, such as $\vee$ (conjunction), $\diamond$ (eventually), $\Box$ (always), and $\to$ (implication), are also induced in which $\phi_1 \vee \phi_2 := \neg(\neg\phi_1 \wedge \neg\phi_2)$, $\diamond\phi :=$ TRUE$\,\mathcal{U}\,\phi$, $\Box\phi := \neg\diamond\neg\phi$, and $\phi_1 \to \phi_2 = \neg\phi_1 \,\mathcal{U}\, \phi_2$. Given a run r of TS $\mathbf{T}$, we say r satisfies an LTL formula ϕ if the *trace* $\mathcal{L}(r) = L(q_0)L(q_1)\dots L(q_i)\dots, i \in \mathbb{N}$ satisfies ϕ.

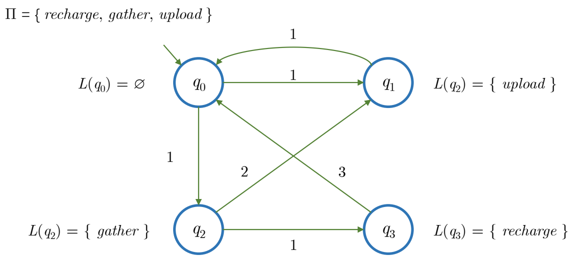

Π = { $recharge$, $gather$, $upload$ }
$L(q_0) = \varnothing$
q_0
1
1
q_1
$L(q_2) = \{\ upload\ \}$
1
2
3
$L(q_2) = \{\ gather\ \}$
q_2
1
q_3
$L(q_3) = \{\ recharge\ \}$

Fig. 1. TS $\mathbf{T}$ abstracted from persistent surveillance mission.

An LTL formula can be represented in an automata-theoretic setting as Büchi automaton (NBA)

$$\mathbf{B} := (S, S_0, \Sigma, \delta_{\mathbf{B}}, F)$$

where S is a finite set of states, $S_0 \subseteq S$ is a set of initial states, Σ is an input alphabet, $\delta_{\mathbf{B}} \subseteq S \times \Sigma \times S$ is a non-deterministic transition relation, and $F \subseteq S$ is a set of accepting states.

The product automaton $\mathbf{P} = \mathbf{T} \times \mathbf{B}$ between TS $\mathbf{T}$ and NBA $\mathbf{B}$ is the tuple $\mathbf{P} := (S_{\mathbf{P}}, S_{\mathbf{P},0}, \delta_{\mathbf{P}}, F_{\mathbf{P}}, w_{\mathbf{P}}, S_{\mathbf{P},\pi})$ where

(i) $S_{\mathbf{P}} = \mathcal{Q} \times S$ is a finite set of states,
(ii) $S_{\mathbf{P},0} = \{q_0\} \times S_0$ is a set of initial states,
(iii) $\delta_{\mathbf{P}} \subseteq S_{\mathbf{P}} \times S_{\mathbf{P}}$ is the transition relation and for states (q, s) and $((\bar{q}, \bar{s})$, $((q, s), (\bar{q}, \bar{s})) \in \delta_{\mathbf{P}}$ if and only if $(q, s) \in \delta_{\mathbf{T}}$ and $(s, \mathcal{L}(q), \bar{s}) \in \delta_{\mathbf{B}}$,
(iv) $F_{\mathbf{P}} = \mathcal{Q} \times F$ is a set of accepting (final) states,
(v) $w_{\mathbf{P}} : \delta_{\mathcal{P}} \to \mathbb{R}_{\geq 0}$, and $w_{\mathbf{P}}(((\bar{q}, \bar{s}), ((q, s), (\bar{q}, \bar{s}))) = w_{\mathbf{P}}((\bar{q}, \bar{s})), \forall((\bar{q}, \bar{s}), ((q, s), (\bar{q}, \bar{s})) \in \delta_{\mathcal{P}}$,

which can be considered as an NBA.

III. Problem Formulation

In this section, we focus on the notion of LTL-based opacity for the abstracted system. Suppose the high-level mission specification of the robot $\mathbf{T}$ is given in the form of

$$\phi := \varphi \wedge (\Box \diamond \pi), \tag{1}$$

where φ is an arbitrary proposition, and π is an AP, and the cost function

$$J(\mathbb{T}^{\pi}) = \limsup_{k \to +\infty} (\mathbb{T}^{\pi}(k+1) - \mathbb{T}^{\pi}(k)) \tag{2}$$

must be minimized. In (1) and (2), AP π must be satisfied infinitely often, and the maximal time between successive satisfactions of AP π must be minimized.

Meanwhile, there is an intruder modeled as a passive observer which knows the behavior of the robotic system and observes the satisfaction of the observable APs and the corresponding overall transition cost. The intruder can then infer the high-level mission specification based on the available information flow. Assume that the behavior of the TS $\mathbf{T}$ is partially observed and that the set of APs can be partitioned into the following two disjoint sets

$$\Pi_i = \Pi_{i,o} \cup \Pi_{i,uo}, \tag{3}$$

where Π_o is the set of observable APs and Π_{uo} is the set of unobservable APs. The behavior visible to an observer is defined by an *event* operator $\mathcal{E} : \mathcal{Q}^* \to 2^{\Pi}_o$ which is defined as follows

$$\mathcal{E}(r) = \cup_{i=0}^{n}\{L(q_i)\} \cap \Pi_o, \tag{4}$$

where $i, n \in \mathbb{N}$, and $r = q_0q_1 \dots q_n \dots \in r_{\mathbf{T}}$ is a run of TS $\mathbf{T}$. Inductively, the event operator $\mathcal{E}(r)$ output the set of observable APs within the run r. Furthermore, the intruder can calculate the overall transition cost $J(\mathbb{T}^{\pi})$ based on the set of observable APs. Then, if for a synthesized run ρ for TS $\mathbf{T}$, the intruder can never infer the mission specification π, then we say that the run ρ is *LTL-based opaque*. To this end, we propose the notion of *LTL-based opacity* which are formally defined as follows.

Definition 1 (LTL-Based Opacity): Let $\mathbf{T} := (\mathcal{Q}, q^0, \delta, \Pi, L, w)$ be a TS with $\Pi_o \subseteq \Pi$ as the set of observable APs and the LTL formula $\phi := \varphi \wedge (\Box \diamond \pi)$. A run ρ of TS $\mathbf{T}$ is said to be *LTL-based opaque* with respect to Π_o and π if

(i)

$$\rho \models \varphi \wedge (\Box \diamond \pi),$$
$$\exists \xi \in R_{\mathbf{T}},\ \xi \models \varphi \wedge (\Box \diamond \pi \wedge \Box \diamond \gamma), \mathcal{E}(\rho) = \mathcal{E}(\xi), \text{and}$$

(ii)

$$J_{\xi}(\mathbb{T}^{\pi}) = k_{\pi,\gamma}J_{\xi}(\mathbb{T}^{\gamma}).$$

where $k_{\pi,\gamma} \in \mathbb{N}_{>0}$, γ is an unobservable AP, and

$$J_{\xi}(\mathbb{T}^{\pi}) = \limsup_{k \to +\infty}(\mathbb{T}^{\pi}_{\xi}(k+1) - \mathbb{T}^{\pi}_{\xi}(k))$$

and

$$J_{\xi}(\mathbb{T}^{\gamma}) = \limsup_{k \to +\infty}(\mathbb{T}^{\gamma}_{\xi}(k+1) - \mathbb{T}^{\gamma}_{\xi}(k))$$

are cost functions with respect to run ξ and APs π and γ, respectively.

Inductively, for condition *(i)* of Definition 1, according to [6], the optimal run ρ satisfying the LTL specification $\phi = \varphi \wedge (\Box \diamond \pi)$ must be in the *prefix-suffix* form, and in the suffix cycle of the run ρ, AP π must be satisfied infinitely many times. If there exists a run ξ in which the suffix cycle satisfies the APs π and γ infinitely often, then the intruder cannot determine the goal AP by the trace of the run. Besides, for condition *(ii)* of Definition 1, the maximal time between successive satisfactions of AP π is in integral multiples of the time of AP γ. Therefore, the intruder cannot determine the mission specification π based on overall transition cost of the run. Now, we have the following proposition to illustrate the effectiveness of LTL-based opacity in security preservation.

Proposition 1: Given a TS $\mathbf{T}$, an LTL specification $\phi := \varphi \wedge (\Box \diamond \pi)$, a set of observable APs $\Pi_o \in \Pi$, and a run ρ, the intruder cannot infer the intention π of TS $\mathbf{T}$ if the run is *LTL-based opaque* with respect to $\mathbf{T}$, Π_o, and π.

Our goal is to synthesize the run which is *LTL-based opaque* for the robotic system. To this end, the cost function (2) should be minimized. Now, we formally give the problem formulations as follows.

Problem 1: Given a robotic system modeled as TS $\mathbf{T} := (\mathcal{Q}, q^0, \delta, \Pi, L, w)$, an LTL formula $\phi := \varphi \wedge (\Box \diamond \pi)$ over the set APs Π, if the set of APs can be partitioned into two disjoint sets as (3), and the behavior of the intruder can be defined by *event* operator in (4), synthesize individual runs ρ for TS $\mathbf{T}$ which are *LTL-based opaque* and satisfy the following conditions,

(i)

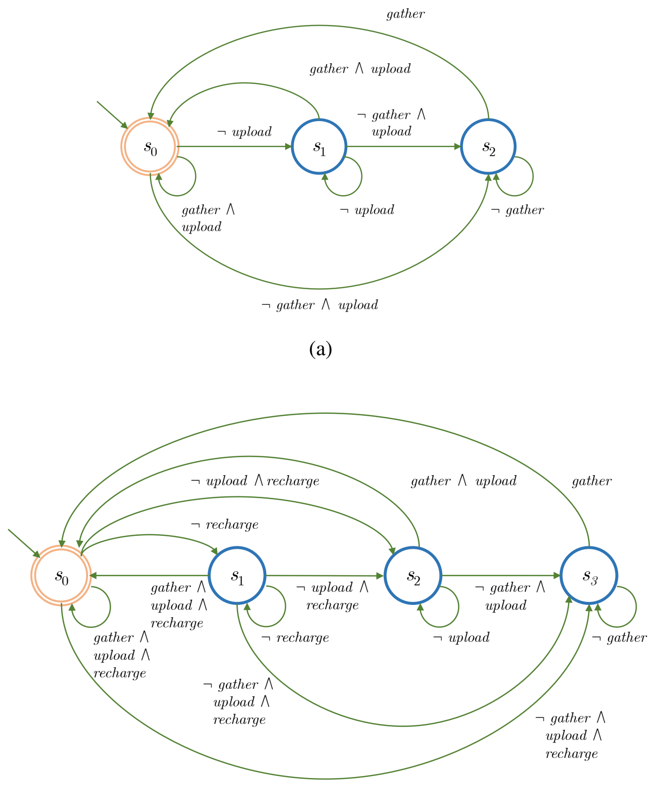
$$\exists \xi \in R_{\mathbf{T}},\ \xi \models \varphi \wedge (\Box \diamond \pi \wedge \Box \diamond \gamma), \mathcal{E}(\rho) = \mathcal{E}(\xi),$$

(ii)

$$J_{\xi}(\mathbb{T}^{\pi}) = k_{\pi,\gamma}J_{\xi}(\mathbb{T}^{\gamma}).,and$$

(iii) the cost function (2) is minimized after conditions *(i)* and *(ii)* are satisfied.

IV. Synthesizer of LTL-based Opacity Runs



(a)

(b)

Fig. 2. (a) is the NBA $\mathbf{B}$ constructed from LTL specification $\phi := \Box \diamond \mathit{gather} \wedge \Box \diamond \mathit{upload}$, and (b) is the NBA $\mathbf{B}'$ constructed from modified LTL specification $\phi' := (\Box \diamond \mathit{gather} \wedge \Box \diamond \mathit{upload}) \wedge \Box \diamond \mathit{recharge}$ for LTL-based opacity.

In this section, we construct the synthesizer for LTL-based opaque runs and propose the algorithm that finds the optimal opaque run. According to [6], to synthesize optimal run for TS $\mathbf{T}$ under the LTL specification $\phi := \varphi \wedge (\Box \diamond \pi)$, an NBA $\mathbf{B}$ is constructed based on the LTL formula ϕ. Then, the optimal run can be obtained in the product automaton $\mathbf{P} = \mathbf{T} \times \mathbf{B}$. Although the optimal run can be synthesized via the aforementioned algorithm correct-by-construction, yet the synthesized run may not be *LTL-based opaque* such that the mission specification will reveal to the intruder.

Our goal is to synthesize the LTL-opaque run for TS $\mathbf{T}$, while guaranteeing its optimality. We use the mission specification $\phi' := \varphi \wedge (\Box \diamond \pi \wedge \Box \diamond \gamma)$ to construct NBA $\mathbf{B}$ such that the corresponding product automaton $\mathbf{P}$ contains the states in which APs π and γ are satisfied, respectively. We use $S_{\mathbf{P},\pi} \in S_{\mathbf{P}}$ and $S_{\mathbf{P},\gamma} \in S_{\mathbf{P}}$ to denote the accepting states for APs π and γ in the product automaton $\mathbf{P}$, respectively. Besides, with a slight abuse of notation, we use $a \in (a, b, c, \dots)$ to denote a is an element of tuple $(a, b, c, \dots)$, and we use $\emptyset$ for none of the APs are satisfied for the current state. Since the product automation $\mathbf{P}$ constructed from TS $\mathbf{T}$ and LTL specification $\phi' := \varphi \wedge (\Box \diamond \pi \wedge \Box \diamond \gamma)$ can be seen as an NBA, the corresponding accepting language will satisfy proposition φ infinitely many times, i.e., if the accepting language of product automaton $\mathbf{P}$, contains the states of $s_{\pi,\mathbf{P}} \in S_{\mathbf{P},\pi}$, and $s_{\gamma,\mathbf{P}} \in S_{\mathbf{P},\gamma}$, then the correctness of the LTL specification and repeated satisfactions of APs π and γ will be guaranteed. Thus, we have the following lemma.

Lemma 1: Given a product automaton $\mathbf{P}$ constructed from TS $\mathbf{T}$ and NBA of LTL formula $\phi' := \varphi \wedge (\Box \diamond \pi \wedge \Box \diamond \gamma)$, if a accepted sequence of automaton $\mathbf{P}$ contains the states $s_{\pi,\mathbf{P}} \in S_{\mathbf{P},\pi}$ and $s_{\gamma,\mathbf{P}} \in S_{\mathbf{P},\gamma}$, then the corresponding sequence in TS $\mathbf{T}$ will satisfy the LTL specification φ and the APs π and γ infinitely often.

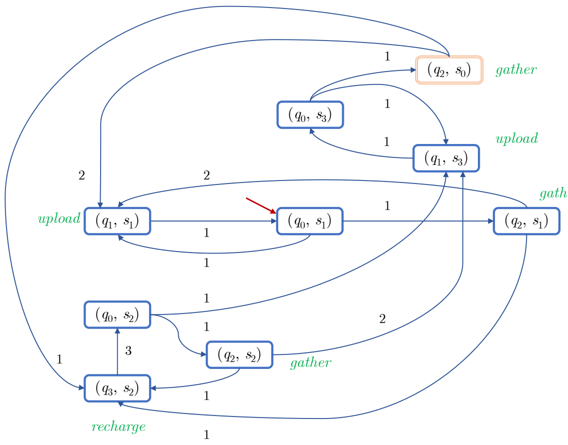


Fig. 3. Product automaton $\mathbf{P} = \mathbf{T} \times \mathbf{B}'$ with respect to TS $\mathbf{T}$ and LTL specification $\phi' := (\Box \diamond gather \wedge \Box \diamond upload) \wedge \Box \diamond recharge$.

To find the loops which guarantee correctness and opacity, we construct the structure of the synthesizer from the product automaton, which can be formally given as follows.

Definition 2: The synthesizer is a bipartite transition system $\mathbf{A} := (Y, Z, y_0, \delta_{YZ}, \delta_{ZY})$ where

(i) $Y = \{(\{(q,s)\}, \alpha) \in 2^{S_{\mathbf{P},\pi}} \times \Pi | \exists s \in (q,s), s \in \Pi_o\}$ is a finite set of Y-states,

(ii) $Z = \{(\{(q,s)\}, \alpha) \in 2^{S_{\mathbf{P},\pi}} \times \Pi | \exists s \in (q,s), s \in \Pi_{uo}\}$ is a finite set of Z-states,

(iii) $y_0 = (S_{\mathbf{P},\pi}, \pi \in Y)$ is the set of initial states,

(iv) $\delta_{YZ} \subseteq Y \times Z$ is the transition relation from Y-states to Z-states,

(v) $\delta_{ZY} \subseteq Z \times Y$ is the transition relation from Z-states to Y-states.

Inductively, in synthesizer $\mathbf{A}$, the Y-states are for the set of states with observable APs, and The Z-states are for the possible reachable states from the state of Y-states, in which the APs of a Z-states are unobservable or unchanged.

In synthesizer $\mathbf{A}$, the set of information states Y and Z are updated according to the following rules. Given a Y-state $y \in Y$ and an unobservable AP $\gamma' \in \Pi_{uo}$, the following Z-states z' can be obtained by its *unobservable reach*

$$UR_{\gamma'}(y) = (\{(q',s') \in S_{\mathbf{P}} | \exists (q,s) \in y, \exists (q',s') \in S_{\mathbf{P}}, \\ (s,\gamma',s') \in \delta_{\mathbf{B}}, \delta_{\mathbf{P}}((q,s),(q',s'))!\}, \gamma'),$$

where notation $(\cdot)!$ denotes "is defined". Besides, we define

$$UR_{\emptyset}(y) = (\{(q,s) \in S_{\mathbf{P}} | \exists (q,s) \in y\} \cup \\ \{(q',s') \in S_{\mathbf{P}} | \exists (q,s) \in y, \exists (q',s') \in S_{\mathbf{P}}, \\ s' = \emptyset, \delta_{\mathbf{P}}((q,s),(q',s'))!\}, \emptyset).$$

Similarly, given a Z-state $z \in Z$ and an observable AP $\pi' \in \Pi_o$, the following Y-states y' can be obtained by its *observable reach*

$$NX_{\pi'}(z) = (\{(q',s') \in S_{\mathbf{P}} | \exists (q,s) \in z, \exists (q',s') \in S_{\mathbf{P}}, \\ (s,\pi',s') \in \delta_{\mathbf{B}}, \delta_{\mathbf{P}}((q,s),(q',s'))!\}, \pi').$$

Inductively, the *unobservable reach* is for the possible reachable states from the states within a Y-state to the states with unobservable APs via trace-empty paths, while the *unobservable reach* for a null AP yields the reachable states with no traces and the states within the input Y-states. The *observable reach*is for the possible reachable states from the states within a Z-state to the states with observable APs via the path of no traces. According to the aforementioned information states update rules, the following remark is given for the upper bound of the maximal number of states of the synthesizer $\mathbf{A}$.

Remark 1: Consider a TS with the number of $\mathcal{Q}$ states, if the length of the LTL formula is denoted as $\|\phi\|$, then the upper bound of the maximal number of states of the synthesizer is $2^{\mathcal{Q}+\|\phi\|+2}$.

Considering the run of the synthesizer $\mathbf{A}$ can be represented as $r_{\mathbf{A}} = p_0 p_1 \dots p_i \dots$, $p \in Y \cup Z$,$i \in \mathbb{N}$. For states q_i q_{i+1}, there exists a set of states $S_i \in p_i$, $S_{i+1} \in p_{i+1}$ such that $\exists s_i \in S_{\mathbf{P}} \cap S_i$ and $\exists s_{i+1} \in S_{\mathbf{P}} \cap S_{i+1}$ of the corresponding product automaton $\mathbf{P}$. If the transition relation $(s_i, s_{i+1}) \in \delta_{\mathbf{P}}$, then the subsequence $s_i s_{i+1}$ can be directly obtained from the synthesizer $\mathbf{A}$. Otherwise, there exists an optimal path $s_i \dots s_{i+1}$ in the corresponding product automaton $\mathbf{P}$ in which only APs $\pi_i \in p_i$ and $\pi_{i+1} \in p_{i+1}$ are satisfied. Now, with a slight abuse of notations, we extend the *event* operator for the product automaton $\mathbf{P}$ and the synthesizer $\mathbf{A}$ such that

$$\mathcal{E}(r_{\mathbf{P}}) = \cup_{i=0}^{n} \{s_i\} \cap \Pi_o, s_i \in (q_i, s_i),$$

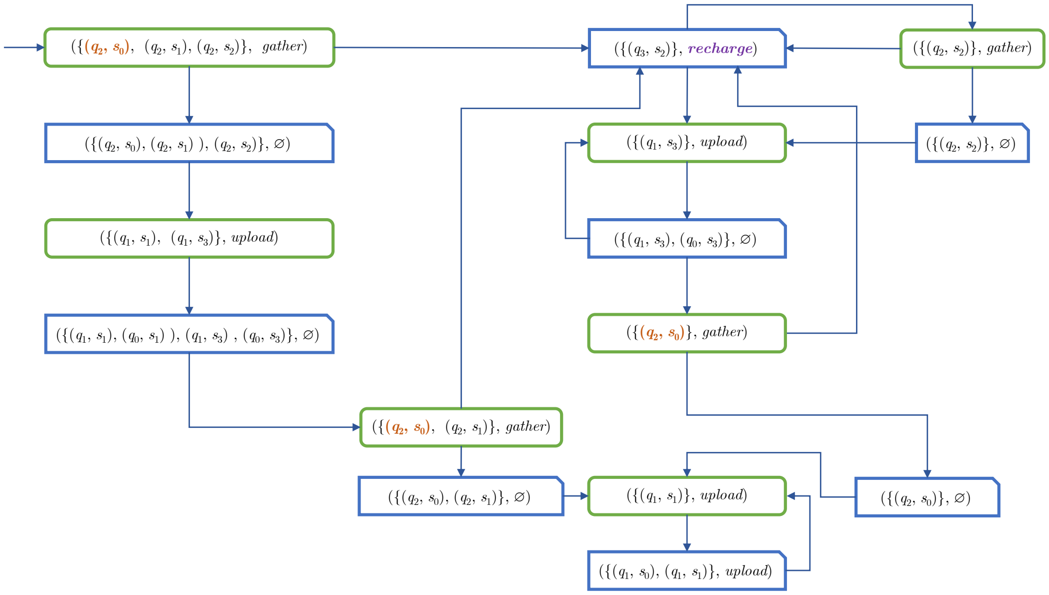

$(\{(q_2, s_0), (q_2, s_1), (q_2, s_2)\}, gather)$
$(\{(q_2, s_0), (q_2, s_1)), (q_2, s_2)\}, \varnothing)$
$(\{(q_1, s_1), (q_1, s_3)\}, upload)$
$(\{(q_1, s_1), (q_0, s_1)), (q_1, s_3) , (q_0, s_3)\}, \varnothing)$
$(\{(q_3, s_2)\}, recharge)$
$(\{(q_2, s_2)\}, gather)$
$(\{(q_1, s_3)\}, upload)$
$(\{(q_2, s_2)\}, \varnothing)$
$(\{(q_1, s_3), (q_0, s_3)\}, \varnothing)$
$(\{(q_2, s_0)\}, gather)$
$(\{(q_2, s_0), (q_2, s_1)\}, gather)$
$(\{(q_2, s_0), (q_2, s_1)\}, \varnothing)$
$(\{(q_1, s_1)\}, upload)$
$(\{(q_2, s_0)\}, \varnothing)$
$(\{(q_1, s_0), (q_1, s_1)\}, upload)$

Fig. 4. Synthesizer $\mathbf{A}$ constructed for product automaton $\mathbf{P} = \mathbf{T} \times \mathbf{B}'$.

and

$$\mathcal{E}(r_{\mathbf{A}}) = \cup_{i=0}^{n}\{\alpha_i\} \cap \Pi_o, \alpha_i \in (\{(q_i, s_i)\}, \alpha_i),$$

where $i, n \in \mathbb{N}$, and the sequences

$$\begin{aligned} r_{\mathbf{P}} =& (q_0, s_0)(q_1, s_1)\ldots(q_n, s_n)\ldots \\ r_{\mathbf{A}} =& (\{(q_0, s_0)\}, \alpha_0)(\{(q_1, s_1)\}, \alpha_1)\ldots \\ & (\{(q_n, s_n)\}, \alpha_n)\ldots. \end{aligned}$$

Then, we give the following lemma for the existence of the correspondence between the sequences of synthesizer $\mathbf{A}$ and product automaton $\mathbf{P}$.

Lemma 2: Given a sequence of states in $r_{\mathbf{A}}$ a synthesizer $\mathbf{A}$, there will be a set of corresponding sequences $r_{\mathbf{P}}$ in the product automaton $\mathbf{P}$ such that the satisfaction of observable APs are identical, i.e., $\mathcal{E}(r_{\mathbf{P}}) = \mathcal{E}(r_{\mathbf{A}})$.

Once the synthesizer is constructed, we use Algorithm 1 to find the optimal opaque run for TS $\mathbf{T}$. Besides, the correctness of Algorithm 1 is illustrated in Theorem 1.

Theorem 1: Given a TS $\mathbf{T}$, a high-level mission specification in the form of LTL formula $\phi := \varphi \wedge (\Box \diamond \pi)$, and a set of observable APs $\Pi_o \in \Pi$, if for each state $q \in \mathcal{Q}$, the number of APs $\|L(q)\| \leq 1$, then the optimal LTL-opaque run can be synthesized by Algorithm 1.

Algorithm 1: OPTIMAL RUN WITH OPACITY-PRESERVATION

```
Input: TS $\mathbf{T}$;
       LTL formula $\phi := \varphi \wedge (\Box \diamond \pi)$;
Output: Optimal opaque runs $\rho^*$;
$suffix\_set \leftarrow \emptyset$;
$\mathcal{R} \leftarrow \emptyset$;
foreach $\gamma \in \Pi_{uo} \backslash \{\pi\}$ do
    Construct NBA from LTL formula
      $\phi' := \varphi \wedge (\Box \diamond \pi \wedge \Box \diamond \gamma)$;
    Construct product automaton $\mathbf{P} = \mathbf{T} \times \mathbf{B}'$;
    Construct synthesizer $\mathbf{A}$ from product automaton $\mathbf{P}$;
    Find all cycles $\xi_{\mathbf{P}suffix} \in R_{suffix}$ containing $y_0$, at
      least one Y-state $(p, \pi)$, and at least one Z-state
      $(p', \gamma)$, $\exists p, p' \subseteq 2^{S_{\mathbf{P}}}$ within synthesizer $\mathbf{A}$;
    foreach $\xi_{\mathbf{P},suffix} \in R_{suffix}$ do
        if $\exists k_{\pi,\gamma}, J_{\xi}(\mathbb{T}^{\pi}) = k_{\pi,\gamma} J_{\xi}(\mathbb{T}^{\gamma})$ then
            Add $\xi_{\mathbf{P}suffix}$ to suffix_set;
foreach $\xi_{\mathbf{P},suffix} \in R_{suffix}$ do
    Project $\xi_{\mathbf{P},suffix}$ back to TS $\mathbf{T}$;
    Find the corresponding optimal prefix for the projected
      suffix cycle;
    Find the corresponding run $\rho$ satisfying the condition
      of transition cost and LTL specification;
    Add run $\rho$ to $\mathcal{R}$;
Find the optimal opaque $\rho^*$ from $\mathcal{R}$ with minimal
  transition cost $J_{\rho}(\mathbb{T}^{\pi})$;
Find the shortest path $\rho_0$ from $q_0$ to the optimal run $\rho^*$;
return Optimal opaque run $\rho_0\rho^*$
```

V. Case Studies

In this section, we demonstrate a case study to illustrate the effectiveness of our proposed algorithm. Consider a surveillance robotic system in a road network with intersections, the abstracted system is denoted $\mathbf{T}$ illustrated in Figure 1. The high-level mission specification is given as

$$\phi := \Box \diamond gather \wedge \Box \diamond upload \tag{5}$$

where the AP $\pi = upload$. In (5), the LTL specification ϕ means the robot must gather and upload data infinitely often, and the maximal time between successive uploads of the data must be minimized. Assume the behavior of the robot is partially observable, i.e., the observable set of APs $\Pi_o = \{gather, upload\}$, while the unobservable set of APs $\Pi_{uo} = \{recharge\}$. Besides, there exists an intruder that monitors the behavior of the robot, and infers the intention via the observable trace and the corresponding overall transition cost.

To avoid the optimal AP $\pi = upload$ being revealed to the intruder, we first let AP $\gamma = recharge$. Then, we convert the modified LTL specification $\phi' := (\Box\Diamond gather \wedge \Box\Diamond upload) \wedge \Box\Diamond recharge$ to the NBA $\mathbf{B}'$ illustrated in Figure 2 (b). and construct the product automaton $\mathbf{P} = \mathbf{T} \times \mathbf{B}'$ illustrated in Figure 3 and the corresponding synthesizer $\mathbf{A}$ illustrated Figure 4. As illustrated in Figure 4, the cycles

$$\begin{aligned}&\xi_{\mathbf{P},suffix,1}\\ =&((\{(q_3,s_2)\},recharge)\ ((\{(q_1,s_3)\},upload)\\ &(\{(q_1,s_3),(q_0,s3)\},\emptyset))^k\ (\{(q_2,s_0)\},gather))^\omega\\ &\xi_{\mathbf{P},suffix,2}\\ =&((\{(q_3,s_2)\},recharge)\ (\{(q_2,s_2)\},gather)\ (\{(q_2,s_2)\},\emptyset)\\ &((\{(q_1,s_3)\},upload)\ (\{(q_1,s_3),(q_0,s_3)\},\emptyset))^k\\ &(\{(q2,s0)\},gather))^\omega\end{aligned}$$

can be found, where $k \in \mathbb{N}_{>0}$ is an arbitrary positive integer, $(r')^k$ is for the subsequence r' will be repeated for k times, and $(r')^\omega$ is for the subsequence r' will be repeated infinitely many times. Then, one can easily check that the corresponding subsequence

$$\xi_{suffix,1} = (q_3\ q_0\ (q_1\ q_0)^k\ q_2)^\omega$$

is projected from $\xi_{\mathbf{P},suffix,1}$ and

$$\xi_{suffix,2} = (q_3\ q_0\ q_2\ (q_1\ q_0)^k\ q_2)^\omega$$

is projected from $\xi_{\mathbf{P},suffix,2}$. However, for suffix cycle $\xi_{suffix,1}$, the cost functions $J_\xi(\mathbb{T}^\pi) = J_\xi(\mathbb{T}^\gamma) = 6 + 2k$, while for suffix cycle $\xi_{suffix,2}$, the cost functions $J_\xi(\mathbb{T}^\pi) = 8 + 2k$ and $J_\xi(\mathbb{T}^\gamma) = 4 + 2k$. For both of the suffix cycles, the condition *(ii)* of LTL-based opacity is satisfied. Besides, for suffix cycles $\xi_{suffix,1}$ and $\xi_{suffix,2}$, the trace-equivalent run is $\rho_{suffix} = ((q_1 q_0)^{l+1} q_2)^\omega$, and the corresponding cost function $J_\rho(\mathbb{T}^\pi) = 2+2l$. Considering the optimal transition cost of $\xi_{suffix,1}$ and $\xi_{suffix,2}$ is 4, we let $l = 3$ such that the optimal opaque run is

$$\rho = (q_0\ q_2\ q_1\ q_0\ q_1\ q_0\ q_1)^\omega,$$

and the corresponding trace-equivalent run is

$$\xi = (q_0\ q_2\ q_3\ q_0\ q_1)^\omega.$$

One can easily check that for optimal run ρ there exists the run ξ such that the sets of observable APs $\mathcal{E}(\rho) = \mathcal{E}(\xi) = \{gather, upload\}$, and the cost function $J_\rho(\mathbb{T}^\pi) = J_\xi(\mathbb{T}^\pi) = J_\xi(\mathbb{T}^\gamma)$, which meet the definition of LTL-based opacity. Thus, the run ρ obtained from Algorithm 1 is optimal opaque, i.e., the intruder cannot determine for sure the AP π is $upload$ or $recharge$ via the synthesized optimal run.

VI. Conclusion

In this paper, we have proposed an opacity-preserving path planning method with opacity preservation. To this end, we first extend the notion of opacity and put forward the definition of LTL-based opacity. Then, a new structure named *synthesizer* is developed for finding all LTL-based opaque runs under the LTL specifications. An algorithm is proposed to find the optimal opaque run via the synthesizer. Finally, a case study is demonstrated to illustrate the effectiveness of our method. In our further research, the path planning problem with opacity preservation for multi-agent systems and stochastic systems will be investigated.

References

[1] C. Zu, C. Yang, J. Wang, W. Gao, D. Cao, and F.-Y. Wang, "Simulation and field testing of multiple vehicles collision avoidance algorithms," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 4, pp. 1045–1063, 2020.

[2] Z. Zuo, C. Liu, Q.-L. Han, and J. Song, "Unmanned aerial vehicles: Control methods and future challenges," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 4, pp. 601–614.

[3] Z. Peng, J. Wang, D. Wang, and Q.-L. Han, "An overview of recent advances in coordinated control of multiple autonomous surface vehicles," *IEEE Trans. Ind. Inf.*, vol. 17, no. 2, pp. 732–745, 2020.

[4] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT Press, Cambridge, MA., 2008.

[5] A. M. Ayala, S. B. Andersson, and C. Belta, "Formal synthesis of control policies for continuous time markov processes from time-bounded temporal logic specifications," *IEEE Trans. Autom. Control*, vol. 59, no. 9, pp. 2568–2573, 2014.

[6] S. L. Smith, J. Tumova, C. Belta, and D. Rus, "Optimal path planning for surveillance with temporal-logic constraints," *Int. J. Rob. Res.*, vol. 30, no. 14, pp. 1695–1708, 2011.

[7] A. Ulusoy, S. L. Smith, X. C. Ding, C. Belta, and D. Rus, "Optimality and robustness in multi-robot path planning with temporal logic constraints," *Int. J. Rob. Res.*, vol. 32, no. 8, pp. 889–911, 2013.

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

[9] G. Wagner and H. Choset, "Subdimensional expansion for multirobot path planning," *Artif. Intell.*, vol. 219, pp. 1–24, 2015.

[10] Y. Xie, X. Yin, S. Li, and M. Zamani, "Secure-by-construction controller synthesis for stochastic systems under linear temporal logic specifications," in *2021 60th IEEE Conf. Decis. Control*, pp. 7015–7021, 2021.

[11] Y.-C. Wu and S. Lafortune, "Comparative analysis of related notions of opacity in centralized and coordinated architectures," *Discrete Event Dyn. Syst.*, vol. 23, no. 3, pp. 307–339, 2013.

[12] Y. Guo, X. Jiang, C. Guo, S. Wang, and O. Karoui, "Overview of opacity in discrete event systems," *IEEE Access*, vol. 8, pp. 48731–48741, 2020.

[13] Z. Ma, X. Yin, and Z. Li, "Verification and enforcement of strong infinite-and K–step opacity using state recognizers," *Autom.*, vol. 133, p. 109838, 2021.

[14] Y. Ji, Y.-C. Wu, and S. Lafortune, "Enforcement of opacity by public and private insertion functions," *Autom.*, vol. 93, pp. 369–378, 2018.

[15] Y.-C. Wu and S. Lafortune, "Synthesis of insertion functions for enforcement of opacity security properties," *Autom.*, vol. 50, no. 5, pp. 1336–1348, 2014.

[16] B. Brandin, R. Su, and L. Lin, "Supervisory control of time-interval discrete event systems," *IFAC-PapersOnLine*, vol. 53, no. 4, pp. 217–222, 2020.

[17] X. Yin and S. Lafortune, "A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems," *IEEE Trans. Autom. Control*, vol. 61, no. 8, pp. 2140–2154, 2015.

[18] Y. Ji, X. Yin, and S. Lafortune, "Opacity enforcement using nondeterministic publicly known edit functions," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4369–4376, 2019.

[19] X. Yin and S. Li, "Synthesis of dynamic masks for infinite-step opacity," *IEEE Trans. Autom. Control*, vol. 65, no. 4, pp. 1429–1441, 2019.

[20] B. Ramasubramanian, L. Niu, A. Clark, L. Bushnell, and R. Poovendran, "Privacy-preserving resilience of cyber-physical systems to adversaries," in *2020 59th IEEE Conf. Decis. Control*, pp. 3785–3792, 2020.

[21] X. Yu, X. Yin, S. Li, and Z. Li, "Security-preserving multi-agent coordination for complex temporal logic tasks," *Control Eng. Pract.*, vol. 123, p. 105130, 2022.

[22] S. Yang, X. Yin, S. Li, and M. Zamani, "Secure-by-construction optimal path planning for linear temporal logic tasks," in *2020 59th IEEE Conf. Decis. Control*, pp. 4460–4466, 2020.