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

Security-preserving multi-robot path planning for Boolean specification tasks using labeled Petri nets

Weijie Shi[1], Zhou He[2,*], *Member, IEEE*, Ziyue Ma[3], *Member, IEEE*, Ning Ran[4], and Xiang Yin[5], *Member, IEEE*

***Abstract*—This letter investigates the path planning of multi-robot systems for high-level tasks described by Boolean specifications and security constraints. We assume that the behavior of each robot can be identified and partially monitored by a passive intruder. The problem aims to plan an optimal path for each robot such that the tasks expressed in conjunction, disjunction, and negation for trajectories and final states are coordinately completed. The security constraints require that the intruder should never infer the final locations of a set of robots called secret robots. In order to solve this problem, labeled Petri nets are adopted to model the mobile capability of the multi-robot systems. Then an integer linear programming problem is proposed to find an optimal solution (if it exists) such that the Boolean specification is fulfilled, while the securities of secret robots are preserved. Finally, the effectiveness of the proposed method is illustrated through several simulation studies.**

***Index Terms*—Labeled Petri net, Cyber-security, Multi-robot system, Boolean specification, Path planning**

I. Introduction

Multi-robot systems (MRSs) are extensively employed in industrial systems, intelligent logistics systems, and automated warehouses since they can coordinately finish complex tasks with high efficiency [1], [2]. The path planning of multiple robots plays an essential role in the analysis and management of MRSs. The conventional research on path planning mainly concentrates on low-level tasks that are characterized as determining the optimal paths for a group of robots to optimize some factors such as travel time and distance, while adhering to certain requirements, e.g., obstacle avoidance, task assignment, time window constraint, and collision-free execution [3]–[5].

Path planning for high-level tasks that are expressed by *linear temporal logics* (LTL) and *Boolean specifications* attracts considerable attention in recent decades [6]–[9]. It is shown that these high-level tasks can be used in many practical problems, such as task gathering, patrolling path planning, and intelligent navigation [10]. Due to the compact representation of the state space of MRSs, Petri nets (PNs) are widely used to plan the paths of multiple robots to achieve some given high-level tasks [11]–[13]. The trajectories of the MRSs and the Boolean specification tasks can be transformed into a set of linear constraints [13]. Therefore, the path planning problem of MRSs for Boolean specification tasks can be optimally resolved by solving an integer linear programming problem (ILPP). This method is further extended to handle general LTL problems in unknown maps [14]. However, it mainly concentrates on the viability of LTL tasks and does not ensure the optimality of paths. Instead of transforming the Boolean specification tasks into linear constraints, control places are designed to implement the logical constraints of tasks according to the structural characteristics of PNs, and an efficient ILPP is developed in [15].

In order to cooperatively finish sophisticated tasks, each robot needs to interact with the control center via networks. However, it may threaten the privacy and security of the system if the information is observed and some key behaviors are inferred by an external intruder. Therefore, security preserving problems in the robot task planning problems receive much attention in recent years [16]–[18], e.g., the information flow security requirement of the system can be characterized by the notion of opacity, and the task planning can be viewed as an accessibility problem [19]. In [16], the problem of planning an optimal infinite path for a single robot to fulfill LTL tasks with security constraints is introduced. It is assumed that the behavior of the robot can be acquired by a passive intruder. The security constraints require that the intruder should never infer the initial location of the robot. Based on the graph search techniques in the product of the twin-WTS and the Büchi automaton, an efficient algorithm is developed. However, few efforts deal with the security and privacy path planning problem in the multi-agent scenario, where the security and privacy of each individual robot in the system should be considered.

In this letter, we study the path planning of MRSs for Boolean specification tasks with security constraints. We assume that the behavior of each robot is partially monitored by an external intruder via a generic observation mapping. Our goal is to find a plan for each robot of the system such that the Boolean specification tasks are fulfilled with minimal cost while the final positions of some robots are preserved against the external intruder.

Recently, the path planning of MRSs for LTL tasks with security constraints is reported in [18]. Compared with [18], our work has the following main differences. First, the security constraint in [18] requires that the intruder can never identify

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61803246, 61903119, and 62061136004, the China Postdoctoral Science Foundation under Grant No. 2019M663608, Shaanxi Provincial Natural Science Foundation under Grant Nos. 2022JM-323 and 2023-JC-YB-564, the Foundation of Hebei Education Department under Grant BJ2021008, and the Fundamental Research Funds for the Central Universities under Grant No. JB210413. (*Corresponding author: Zhou He*)

[1] Weijie Shi is with the School of Electro-Mechanical Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China (e-mail: 201605011420@sust.edu.cn)

[2] Zhou He is with the School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China (e-mail: hezhou@sust.edu.cn)

[3] Ziyue Ma is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China (e-mail: maziyue@xidian.edu.cn)

[4] Ning Ran is with the College of Electronic and Information Engineering, Heibei University, Baoding 071002, China (e-mail: ranning87@hotmail.com)

[5] Xiang Yin is with the Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yinxiang@sjtu.edu.cn)

for sure that some specific individual agent is executing the secret tasks. In our work, however, the security constraint requires that the intruder should never infer the final (or initial) locations of the secret robots. Second, the approach for handling multi robots in [18] is to construct the entire product automaton by considering the N-product state-space. In this work, we use Petri nets to model MRSs, which is known to be more efficient without explicit state-space enumeration.

The main contribution of this work is summarized as follows. First, labeled Petri nets are adopted to model the mobile capability of the MRSs and the observation structure of the intruder, such that the state-space explosion of MRSs can be avoided. Second, some methods are proposed to transform the Boolean specification tasks and the security constraints into linear algebraic constraints. Third, an ILPP is developed to obtain an optimal path for each individual robot such that the final positions of the secret robots are preserved. Finally, the developed approach is further extended to the situation where the initial positions of the secret robots are preserved against the external intruder.

This letter is structured in six sections. Some necessary preliminaries are introduced in Section II. Boolean specification tasks and problem formulation are presented in Section III. A security-preserving planning method is developed in Section IV. In Section V, case studies are proposed to illustrate the effectiveness of the proposed approach. Finally, a conclusion is given in Section VI.

II. Preliminary

A Petri net (PN) is a 4-tuple $N = (P, T, Pre, Post)$, where P is a set of n places represented by circles; T is a set of m transitions represented by bars; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the *pre-* and *post-incidence functions* that specify the arcs, respectively, which are also denoted by matrices $\mathbb{N}^{n \times m}$, where $\mathbb{N} = \{0, 1, 2, \ldots\}$ represents the set of nonnegative integers. The *incidence matrix* is defined by $C = Post - Pre \in \mathbb{Z}^{n \times m}$, where $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$ denotes the set of integers.

A *marking* is described as an n-component vector $M \in \mathbb{N}^n$ and the number of tokens of place p at marking M is denoted by $M(p)$. A *PN system* $\langle N, M_0 \rangle$ is a net N with an *initial marking* M_0.

A transition t_i is *enabled* at M if $M \geq Pre(\cdot, t_i)$, denoted by $M[t_i\rangle$, where $Pre(\cdot, t_i)$ represents the column of the matrix Pre associated with transition t_i. An enabled transition t_i may fire yielding a new marking M' with

$$M' = M + C \cdot \vec{t_i}, \tag{1}$$

where $\vec{t_i}$ is an m-dimensional firing vector corresponding to the i-th canonical basis vector.

Let $\sigma = t_0 t_1 \ldots t_h$ be a sequence of transitions with length h, a transition $t \in T$ appears in sequence σ is represented by $t \in \sigma$. The notation $M[\sigma\rangle$ represents that σ is enabled at M and $M[\sigma\rangle M'$ represents that the firing of σ yields M'. We denote by $\vec{\sigma} \in \mathbb{N}^m$ the *firing count vector* whose i-th component represents the number of times that transition t_i appears in sequence σ.

A *labeled Petri net* (LPN) is a 4-tuple $G = (N, M_0, E, \lambda)$ where N is a PN, M_0 is an initial marking, E is the alphabet (a set of labels), and $\lambda : T \to E \cup \{\varepsilon\}$ is a labeling function which maps each transition $t \in T$ a label $e \in E$ or the empty word ε [20].

In this letter, we assume that the intruder has complete knowledge of the system but partial observation of the behavior of each individual robot [7]. Therefore, the transition set T can be further divided into the set of observable transitions T_o and the set of unobservable transitions T_u, where $T = T_o \cup T_u$ and $T_o \cap T_u = \varnothing$. Moreover, if $t \in T_o$ then $\lambda(t) = e \in E$, otherwise $\lambda(t) = \varepsilon$. Note that, a label e can be associated with more than one transition. We denote by $T(e) = \{t \in T_o | \lambda(t) = e\}$ the set of transitions associated with the same label $e \in E$.

The labeling function can be extended to a sequence $\sigma = t_1 t_2 \ldots t_h \in T^*$, such that $\lambda(\varepsilon) = \varepsilon$ if $\sigma = \varepsilon$; otherwise $\lambda(\sigma) = \lambda(t_1)\lambda(t_2)\ldots\lambda(t_h)$. We denote by $w \in E^*$ the word that is observed with the sequence $\sigma \in T^*$ fires. We use $M_1[w\rangle M_2$ to represent that there exists a sequence $\sigma \in T^*$ such that $w = \lambda(\sigma)$ and the firing of $\sigma \in T^*$ at M_1 yields M_2. We denote by $\sigma_o \in \sigma$ (resp., $\sigma_u \in \sigma$) the sequence of σ composed of the observable (resp., unobservable) transitions and $\vec{\sigma}_o$ (resp., $\vec{\sigma}_u$) the corresponding firing vector.

In this work, we consider a group of k identical robots $R = \{r_1, r_2, \ldots, r_k\}$ moving in the same workspace $S = \{s_1, s_2, \ldots, s_n\}$ that is partitioned into n *cells*. Let $\Re = (S, R)$ be an MRS, it can be easily modelled by an LPN [7], [12]. Particularly, we model a cell $s \in S$ by a place $p \in P$ and the movement of a robot from cell s to an adjacent cell by a transition t. In addition, each transition $t \in T$ is assigned a label $e \in E \cup \{\varepsilon\}$ to denote the observation of the corresponding movement from the intruder's point of view. For example, when a robot moves from cells s_i to s_j, the observed word w for the intruder will be $w = \lambda(t_q)$, where t_q is a transition of the LPN G that models the movement. Note that if cell s_j is unobservable for the intruder, then the observed word will be empty, i.e., $w = \varepsilon$. We mention that the LPN model for the MRS $\Re$ is a particular subclass of PNs called *state machine* where each of its transition has exactly one input place and one output place.

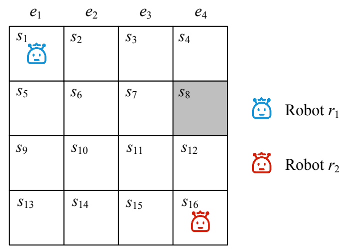


Fig. 1. A multi-robot system of Example 1.

Example 1: Let us consider an example that consists of two robots working in a 4×4 workspace as shown in Fig. 1, where $S = \{s_1, s_2, \ldots, s_{16}\}$ and $R = \{r_1, r_2\}$. Each movement of the robot will generate an observation according to the column information of the destination (when the robot moves to the first, second, third, and fourth columns, the labels e_1, e_2, e_3, and e_4 will be triggered, respectively). Note that cell s_8 (the gray cell) is an unobservable cell which means that every movement to s_8 from its adjacent cell will not be observed by the intruder, i.e., the observed word for the intruder will be ε.

The LPN model G for the MRS $\Re = (S, R)$ is shown in Fig. 2. It consists of 16 places $P = \{p_1, p_2, \ldots, p_{16}\}$ (each of which represents a cell), 48 transitions $T = \{t_1, t_2, \ldots, t_{48}\}$ (each of which represents a movement of a robot from one cell to another adjacent cell), and four labels $E = \{e_1, e_2, e_3, e_4\}$. Transitions in colors green, red, orange, blue, and black represent that their labels are e_1, e_2, e_3, e_4, and ε, respectively, i.e., $T(e_1) = \{t_2, t_3, t_4, t_{16}, t_{17}, t_{18}, t_{30}, t_{31}, t_{32}, t_{44}\}$, $T(e_2) = \{t_1, t_6, t_7, t_8, t_{15}, t_{20}, t_{21}, t_{22}, t_{29}, t_{34}, t_{35}, t_{36}, t_{43}, t_{46}\}$, $T(e_3) = \{t_5, t_{10}, t_{11}, t_{12}, t_{19}, t_{24}, t_{25}, t_{26}, t_{33}, t_{38}, t_{39}, t_{40}, t_{45}, t_{48}\}$, $T(e_4) = \{t_9, t_{14}, t_{27}, t_{37}, t_{41}, t_{42}, t_{47}\}$, and $T_u = \{t_{13}, t_{23}, t_{28}\}$. $\Diamond$

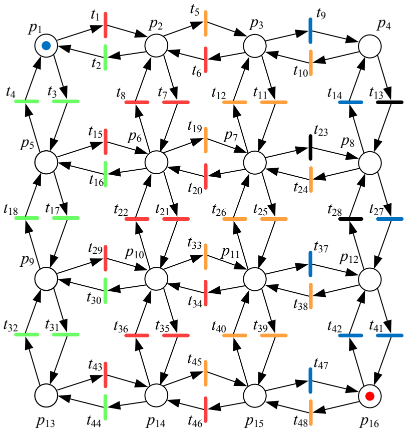


Fig. 2. The LPN model corresponding to the MRS in Fig. 1.

III. Boolean Specification Tasks and Problem Formulation

In this section, we give the definition of the task specification and the security constraint, and formulate the path planning problem.

A. Task specification and security constraint

Task specification: In an MRS, robots need to work cooperatively to finish a global task. In this letter, we consider a high-level task described by a Boolean specification formula. We denote by $\Omega = \{\mathbf{\Pi}_1, \mathbf{\Pi}_2, \ldots, \mathbf{\Pi}_{|\Omega|}\}$ the *set of regions of interest*, where region $\mathbf{\Pi}$ represents a set of cells of interest. The basic unit of the Boolean specification formula is an atomic proposition in $\Omega_t \cup \Omega_f$, where $\Omega_t = \{\Pi_1, \Pi_2, \ldots, \Pi_{|\Omega|}\}$ and $\Omega_f = \{\pi_1, \pi_2, \ldots, \pi_{|\Omega|}\}$. Each atomic proposition $\Pi \in \Omega_t$ (resp., $\pi \in \Omega_f$) represents that region $\mathbf{\Pi}$ should be visited or avoided along the trajectories (resp., at the final state). The Boolean specification formula is defined as the logical relationship (i.e., conjunction $\wedge$, disjunction $\vee$, and negation $\neg$) between atomic propositions.

Particularly, the global task for the MRS is given as a Boolean specification formula φ of the following form:

$$\varphi = Y \wedge A \wedge U. \tag{2}$$

- The sub-specification $Y = y_1 \wedge \cdots \wedge y_q$ represents the logical requirements on the *task regions*, where
$$y_i = \bigvee_{\Pi \in \Omega_{y_i}} \Pi, \ \Omega_{y_i} \subseteq \Omega_t. \tag{3}$$
It requires that at least one of the cells of region $\mathbf{\Pi}$ whose atomic proposition Π belongs to Ω_{y_i} should be visited along the trajectories.
- The subspecification A represents the logical requirements on the *forbidden regions* that should be avoided, where
$$A = \bigwedge_{\Pi \in \Omega_a} (\neg \Pi) = \neg (\bigvee_{\Pi \in \Omega_a} \Pi), \ \Omega_a \subseteq \Omega_t. \tag{4}$$
It requires that all cells of region $\mathbf{\Pi}$ whose atomic proposition Π belongs to Ω_a should always be avoided along the trajectories.
- The sub-specification $U = U_1 \wedge \cdots \wedge U_d$ represents the logical requirements on the *final regions*, where
$$U_i = \bigvee_{\pi \in \Omega_{u_i}} \pi, \ \Omega_{u_i} \subseteq \Omega_f. \tag{5}$$
It requires that at least one of the cells of region $\mathbf{\Pi}$ whose atomic proposition π belongs to Ω_{u_i} should eventually be occupied by a robot.

Security constraint: In this letter, we assume that the intruder has the full knowledge of the MRS, i.e., the map of the workspace S, the initial location/state of each robot (modelled by the initial marking M_0 of the LPN model). The internal state of the system is not directly available to the intruder during the execution of the task. However, the behavior of each robot can be identified and partially monitored by the intruder via the labeling function and the observed word [16].

In the rest of the letter, we denote by $M_{i,j}$ the i-th state of robot $r_j \in R$ and by $\rho_j = M_{0,j}[\sigma_{u_{1,j}}\rangle M_{u_{1,j}}[\sigma_{o_{1,j}}\rangle M_{o_{1,j}} \ldots M_{u_{h,j}}[\sigma_{o_{h,j}}\rangle M_{o_{h,j}}$ a *path trajectory* starting from an initial state $M_{0,j}$ for robot r_j, where $h \in \mathbb{N}$ is a designed parameter. We assume that a robot can advance maximum one cell at each state (i.e., $\|\vec{\sigma}_{o_{i,j}}\|_1 \leq 1$, $\|\vec{\sigma}_{u_{i,j}}\|_1 \leq 1$, $i = 1, \ldots, h$), hence the maximum intermediate states (markings) of each robot is $2h$. Recall that σ_o and σ_u denote the observable and unobservable firing sequence, respectively.

Let $R_s \subseteq R$ be a set of *secret robots* whose final states need to be kept confidential for safety, the security constraint requires that for each secret robot $r_j \in R_s$, there should exists another different path trajectory $\rho'_j = M_{0,j}[\sigma'_{u_{1,j}}\rangle M'_{u_{1,j}}[\sigma'_{o_{1,j}}\rangle M'_{o_{1,j}} \ldots M'_{u_{h,j}}[\sigma'_{o_{h,j}}\rangle M'_{o_{h,j}}$ starting from the same initial state $M_{0,j}$ such that:

$$\begin{aligned} &\lambda(\sigma_{o_{1,j}}) \ldots \lambda(\sigma_{o_{h,j}}) = \lambda(\sigma'_{o_{1,j}}) \ldots \lambda(\sigma'_{o_{h,j}}) \quad (6a) \\ &M_{o_h} \neq M'_{o_h} \quad (6b) \end{aligned} \tag{6}$$

The first condition implies that the observation of the paths ρ and ρ' of a secret robot are identical from the intruder's point of view, and the second condition indicates that the final states of the two paths ρ_j and ρ'_j should be different. As a consequence, the intruder cannot determine the final states of the secret robots.

B. Problem formulation

Combining the results discussed above, we formulate the security-preserving multi-robot path planning problem as follows.

Problem 1: Given an MRS $\Re = (S, R)$ that contains k mobile robots $R = \{r_1, \ldots, r_k\}$, a known workspace that contains n cells $S = \{s_1, \ldots, s_n\}$, a Boolean specification task φ in the form of (2), and a set of secret robots $R_s \subseteq R$, we aim to plan a *path trajectory* (if it exists) for each robot such that both the Boolean specification task and the security constraint are achieved at the final state while the total travel distance of the MRS $\Re$ is minimized.

IV. Planning algorithm

In this section, we present some methods to transform the logical Boolean specification and the security constraint into linear algebraic constraints. Then, we develop an ILPP to obtain an optimal solution for Problem 1.

A. Logical Boolean specification

For each sub-specification $y_i \in Y$, we define an n-component characteristic vector $v_{y_i} = [v_{y_i}(1), \ldots, v_{y_i}(n)] \in \{0,1\}^{1\times n}$, where

$$v_{y_i}(j) = \begin{cases} 1 & \text{if } s_j \in \mathbf{\Pi} \text{ \& } \Pi \in \Omega_{y_i}, \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to observe that, for an intermediate marking $M_{i,j}$, if $v_{y_i} \cdot M_{i,j} \geq 1$ holds , then the sub-specification y_i is satisfied. We denote by $V_Y = [v_{y_1}, \ldots, v_{y_q}]$ the characteristic matrix of sub-specification Y.

For each sub-specification $U_i \in U$, we define $v_{u_i} = [v_{u_i}(1), \ldots, v_{u_i}(n)] \in \{0,1\}^{1\times n}$ as the characteristic vector of U_i and $V_U = [v_{u_1}, \ldots, v_{u_d}]$ as the characteristic matrix of U. In particular,

$$v_{u_i}(j) = \begin{cases} 1 & \text{if } s_j \in \mathbf{\Pi} \text{ \& } \Pi \in \Omega_{u_i}, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, the characteristic vector of sub-specification A is defined as $V_A = [v_a(1), \ldots, v_a(n)] \in \{0,1\}^{1\times n}$ and $v_a(j) = 1$ if $s_j \in \mathbf{\Pi}$ and $\Pi \in \Omega_A$, otherwise $v_a(j) = 0$.

The logical Boolean specification (2) can be transformed into a set of linear constraints as follows:

$$\left\{\begin{array}{ll} V_Y \cdot \sum\limits_{j=1}^{k} \sum\limits_{i=1}^{h} (M_{u_{i,j}} + M_{o_{i,j}}) \geq \vec{1}, & (7a) \\ V_A \cdot \sum\limits_{j=1}^{k} \sum\limits_{i=1}^{h} (M_{u_{i,j}} + M_{o_{i,j}}) = 0, & (7b) \\ V_U \cdot \sum\limits_{j=1}^{k} M_{o_{h,j}} \geq \vec{1}. & (7c) \end{array}\right. \qquad (7)$$

Note that $\sum\limits_{j=1}^{k} \sum\limits_{i=1}^{h} (M_{u_{i,j}} + M_{o_{i,j}})$ and $\sum\limits_{j=1}^{k} M_{o_{h,j}}$ represent the number of visiting times of each cell along the trajectories and the number of robots located in each cell at the final state, respectively. Therefore, constraint $(7a)$ imposes that for each characteristic vector v_{y_i} there exists at least one marking $M_{i,j}$ that satisfies $v_{y_i} \cdot M_{i,j} = 1$, which means that at least one robot r_j will complete the task corresponding to sub-specification y_i. Constraint $(7b)$ guarantees that all robots will never enter the forbidden regions along their path trajectories. Finally, constraint $(7c)$ implies that each final region will be occupied by at least one robot at the final state.

B. Security constraint

Let $\mathcal{L} : T \times E \to \{0,1\}$ be the *labeling incidence function* that specifies the label corresponding to each transition, where $\mathcal{L}(i,j) = 1$ if $\lambda(t_i) = e_j \in E$, otherwise, $\mathcal{L}(i,j) = 0$. Note that for an unobservable transition t_i, the row of the matrix $\mathcal{L}$ associated with transition t_i will be all zeros, i.e., $\mathcal{L}(i,\cdot) = 0$.

As we discussed in subsection III-A, the planned path trajectory of a secret robot $r_j \in R_s$ starting from an initial state $M_{0,j}$ can be represented as:

$$\rho_j = M_{0,j}[\sigma_{u_{1,j}}\rangle M_{u_{1,j}}[\sigma_{o_{1,j}}\rangle M_{o_{1,j}} \ldots M_{u_{h,j}}[\sigma_{o_{h,j}}\rangle M_{o_{h,j}}.$$

In order to satisfy the security constraint for a secret robot r_j, it is required that there exists another different path trajectory starting from the same initial state $M_{0,j}$ such that

$$\rho'_j = M_{0,j}[\sigma'_{u_{1,j}}\rangle M'_{u_{1,j}}[\sigma'_{o_{1,j}}\rangle M'_{o_{1,j}} \ldots M'_{u_{h,j}}[\sigma'_{o_{h,j}}\rangle M'_{o_{h,j}}.$$

In the following proposition, we show how to represent the security constraints by linear algebraic constraints.

Proposition 1: The security constraint (6) can be transformed into a set of linear constraints as follows:

$$\left\{\begin{array}{ll} \left.\begin{array}{l} M_{u_{i,j}} = M_{o_{i-1,j}} + C \cdot \vec{\sigma}_{u_{i,j}}, \\ M_{o_{i-1,j}} - Pre \cdot \vec{\sigma}_{u_{i,j}} \geq \vec{0}, \\ M_{o_{i-1,j}} = M_{u_{i,j}} + C \cdot \vec{\sigma}_{o_{i,j}}, \\ M_{u_{i,j}} - Pre \cdot \vec{\sigma}_{o_{i,j}} \geq \vec{0}, \end{array}\right\} & (8a) \\ \sum\limits_{t\in T_o} \vec{\sigma}_{o_{i,j}}(t) \leq 1, \ \sum\limits_{t\in T_u} \vec{\sigma}_{u_{i,j}}(t) \leq 1, & (8b) \\ \left.\begin{array}{l} M'_{u_{i,j}} = M'_{o_{i-1,j}} + C \cdot \vec{\sigma}'_{u_{i,j}}, \\ M'_{o_{i-1,j}} - Pre \cdot \vec{\sigma}'_{u_{i,j}} \geq \vec{0}, \\ M'_{o_{i-1,j}} = M'_{u_{i,j}} + C \cdot \vec{\sigma}'_{o_{i,j}}, \\ M'_{u_{i,j}} - Pre \cdot \vec{\sigma}'_{o_{i,j}} \geq \vec{0}, \end{array}\right\} & (8c) \\ \sum\limits_{t\in T_o} \vec{\sigma}'_{o_{i,j}}(t) \leq 1, \ \sum\limits_{t\in T_u} \vec{\sigma}'_{u_{i,j}}(t) \leq 1, & (8d) \\ M_{o_{0,j}} = M_{0,j}, \ M'_{o_{0,j}} = M_{0,j}, & (8e) \\ \mathcal{L}^T \cdot \sigma_{o_{i,s}} = \mathcal{L}^T \cdot \sigma'_{o_{i,s}}, \forall r_s \in R_s, & (8f) \\ \left.\begin{array}{l} y \cdot M'_{o_{h,s}} - y \cdot M_{o_{h,s}} + z_s \cdot H \geq 1, \\ y \cdot M'_{o_{h,s}} - y \cdot M_{o_{h,s}} - \bar{z}_s \cdot H \leq -1, \\ y = [1, 2, \ldots, n], \\ z_s + \bar{z}_s = 1, \ z_s, \bar{z}_s \in \{0,1\}, \ \forall r_s \in R_s, \end{array}\right\} & (8g) \\ i = 1, 2, \ldots, h, \ j = 1, 2, \ldots, k, & (8h) \end{array}\right. \qquad (8)$$

where $H \in \mathbb{R}_{\geq 0}$ is a constant satisfying $H \geq n$.

Proof: Constraints $(8a)$, $(8c)$, and $(8h)$ guarantee the correctness of the path trajectory according to (1). Constraints $(8b)$ and $(8d)$ imposes that a robot can advance maximum one cell at each state. Constraint ($8e$) ensures that the starting positions of the two path trajectories ρ_j and ρ'_j are identical. Constraint ($8f$) indicates that the observations of ρ_j and ρ'_j for a secret robot $r_s \in R_s$ are equivalent at each state from the intruder's point of view, which consequently implements condition ($6a$).

Constraint ($8g$) ensures that the final positions of the two path trajectories will be different as follows. Suppose that the final positions of the two path trajectories are identical, i.e., $M'_{o_h,s} = M_{o_h,s}$. Equation ($8g$) can be simplified as:

$$\begin{cases} z_s \cdot H \geq 1, & (9a) \\ -\bar{z}_s \cdot H \leq -1, & (9b) \\ z_s + \bar{z}_s = 1,\ z_s, \bar{z}_s \in \{0, 1\}. & (9c) \end{cases} \tag{9}$$

To satisfy Eqs. ($9a$) and ($9b$), it must be $\bar{z}_s = 1$ and $z_s = 1$. However, in this situation condition ($9c$) is violated. Therefore, the final positions of the two path trajectories for secret robots are different, i.e., $M'_{o_h,s} \neq M_{o_h,s}$.

C. Path Planning for MRS with an ILPP

Combining the above results, we develop an ILPP to solve the security-preserving multi-robot path planning for Boolean specification tasks (i.e., Problem 1) as as follows:

$$\min w \cdot \sum_{j=1}^{k} \sum_{i=1}^{h} (\vec{\sigma}_{u_{i,j}} + \vec{\sigma}_{o_{i,j}})$$

$$\left\{\begin{array}{ll}
\left.\begin{array}{l} M_{u_{i,j}} = M_{o_{i-1,j}} + C \cdot \vec{\sigma}_{u_{i,j}}, \\ M_{o_{i-1,j}} - Pre \cdot \vec{\sigma}_{u_{i,j}} \geq \vec{0}, \\ M_{o_{i-1,j}} = M_{u_{i,j}} + C \cdot \vec{\sigma}_{o_{i,j}}, \\ M_{u_{i,j}} - Pre \cdot \vec{\sigma}_{o_{i,j}} \geq \vec{0}, \end{array}\right\} & (10a) \\
\sum_{t \in T_o} \vec{\sigma}_{o_{i,j}}(t) \leq 1, \quad \sum_{t \in T_u} \vec{\sigma}_{u_{i,j}}(t) \leq 1, & (10b) \\
\left.\begin{array}{l} M'_{u_{i,j}} = M'_{o_{i-1,j}} + C \cdot \vec{\sigma}'_{u_{i,j}}, \\ M'_{o_{i-1,j}} - Pre \cdot \vec{\sigma}'_{u_{i,j}} \geq \vec{0}, \\ M'_{o_{i-1,j}} = M'_{u_{i,j}} + C \cdot \vec{\sigma}'_{o_{i,j}}, \\ M'_{u_{i,j}} - Pre \cdot \vec{\sigma}'_{o_{i,j}} \geq \vec{0}, \end{array}\right\} & (10c) \\
\sum_{t \in T_o} \vec{\sigma}'_{o_{i,j}}(t) \leq 1, \quad \sum_{t \in T_u} \vec{\sigma}'_{u_{i,j}}(t) \leq 1, & (10d) \\
M_{o_{0,j}} = M_{0,j},\ M'_{o_{0,j}} = M_{0,j}, & (10e) \\
\mathcal{L}^T \cdot \sigma_{o_{i,s}} = \mathcal{L}^T \cdot \sigma'_{o_{i,s}}, \forall r_s \in R_s, & (10f) \\
\left.\begin{array}{l} y \cdot M'_{o_{h,s}} - y \cdot M_{o_{h,s}} + z_s \cdot H \geq 1, \\ y \cdot M'_{o_{h,s}} - y \cdot M_{o_{h,s}} - \bar{z}_s \cdot H \leq -1, \\ y = [1, 2, \ldots, n], \\ z_s + \bar{z}_s = 1,\ z_s, \bar{z}_s \in \{0, 1\},\ \forall r_s \in R_s, \end{array}\right\} & (10g) \\
i = 1, 2, \ldots, h,\ j = 1, 2, \ldots, k, & (10h) \\
\left.\begin{array}{l} V_Y \cdot \sum_{j=1}^{k} \sum_{i=1}^{h} (M_{u_{i,j}} + M_{o_{i,j}}) \geq \vec{1}, \\ V_A \cdot \sum_{j=1}^{k} \sum_{i=1}^{h} (M_{u_{i,j}} + M_{o_{i,j}}) = 0, \\ V_U \cdot \sum_{j=1}^{k} M_{o_{h,j}} \geq \vec{1}. \end{array}\right\} & (10i)
\end{array}\right. \tag{10}$$

Note that the goal of Problem 1 is to plan a path trajectory for each robot such that both the Boolean specification and the security constraint are achieved at the final state, while the total travel distance of the MRS is minimized. The objective function of (10) accounts for the total travel cost/distance of the MRS, where $w = [w(t_1), w(t_2), \ldots, w(t_m)]$ is an m-dimensional non-negative vector corresponding to the distance associated with each transition. Note that the parameter h is a predesigned number, and $2h$ represents the maximum intermediate states (markings) of each robot. Constraints ($10a$-$10h$) conjointly enforce the security constraints, and constraints ($10a$-$10e$) and ($10i$) conjointly enforce the Boolean specification tasks. The optimal solution $\sigma_j = \sigma_{u_{1,j}} \sigma_{o_{1,j}} \ldots \sigma_{u_{h,j}} \sigma_{o_{h,j}}$ ($j = 1, ..., k$) of ILPP (10) represents the sequence of the firing count vector of the MRS that corresponds to the trajectory of each robot.

Complexity discussion. The optimization problem (10) is a standard ILPP whose computational complexity is commonly characterized by the numbers of variables and constraints. Problem (10) has at most $4k \cdot h \cdot (n+m) + 2k$ variables and $k \cdot h \cdot (8n+m+4) + 2k \cdot n + 3k + 2n + 1$ constraints. In practice, it can usually be efficiently solved by linear programming tools such as LINGO and CPLEX.

Remark 1: The security-preserving multi-robot path planning algorithm can be easily extended to the situation where the intruder does not know the initial state (i.e., position) of each individual robot but has the information of the final state (i.e., marking $M_{h,j}$) due to some reasons. However, the intruder may infer the initial positions of the secret robots through the observation of the execution of the task. We aim to plan a path trajectory for each robot such that the initial positions of the secret robots will be hidden for the intruder at the final state. This problem can be solved by replacing constraints ($10e$) and ($10g$) with the following constraints:

$$\begin{array}{ll}
M_{o_{h,j}} = M_{h,j},\ M'_{o_{h,j}} = M_{h,j}, & (10'e) \\
\left.\begin{array}{l} y \cdot M'_{o_{0,s}} - y \cdot M_{o_{0,s}} + z_s \cdot H \geq 1, \\ y \cdot M'_{o_{0,s}} - y \cdot M_{o_{0,s}} - \bar{z}_s \cdot H \leq -1, \\ y = [1, 2, \ldots, n], \\ z_s + \bar{z}_s = 1,\ z_s, \bar{z}_s \in \{0, 1\},\ \forall r_s \in R_s, \end{array}\right\} & (10'g)
\end{array}$$

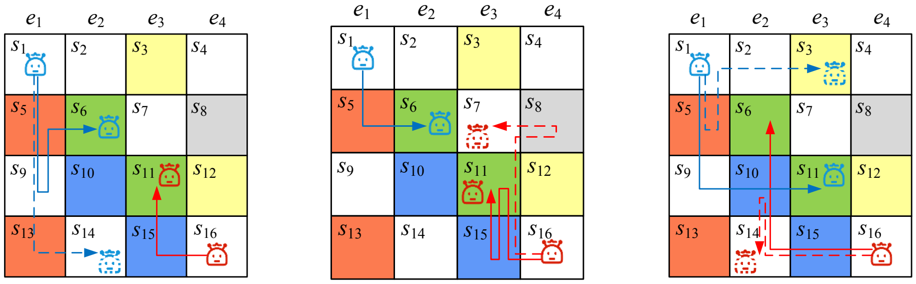
V. Simulation results

In this section, the effectiveness of the developed security-preserving multi-robot path planning algorithm is illustrated by the MRS $\Re$ discussed in Example 1. The developed algorithm is implemented by MATLAB with YALMIP subroutines.

There exist two robots r_1 (blue one) and r_2 (red one) completing tasks in the workspace. Initially, r_1 and r_2 are located in cells s_1 and s_{16}, respectively. We assume that the set of regions of interest is $\Omega = \{\mathbf{\Pi}_1, \mathbf{\Pi}_2, \mathbf{\Pi}_3, \mathbf{\Pi}_4, \mathbf{\Pi}_5, \mathbf{\Pi}_6\}$, where $\mathbf{\Pi}_1 = \{s_5\}$, $\mathbf{\Pi}_2 = \{s_{10}, s_{15}\}$, $\mathbf{\Pi}_3 = \{s_{13}\}$, $\mathbf{\Pi}_4 = \{s_6\}$, $\mathbf{\Pi}_5 = \{s_3, s_{12}\}$, and $\mathbf{\Pi}_6 = \{s_{11}\}$. The global task for the MRS $\Re$ is as follows:

$$\varphi = (\Pi_1 \vee \Pi_3) \wedge \Pi_2 \wedge (\neg \Pi_5) \wedge \pi_4 \wedge \pi_6,$$

where $\Omega_{y_1} = \{\mathbf{\Pi}_1, \mathbf{\Pi}_3\}$, $\Omega_{y_2} = \{\mathbf{\Pi}_2\}$, $\Omega_A = \{\mathbf{\Pi}_5\}$, $\Omega_{u_1} = \{\mathbf{\Pi}_4\}$, and $\Omega_{u_2} = \{\mathbf{\Pi}_6\}$. It requires that at least one of cells s_5 or s_{13} (orange cells in Fig. 3) and one of cells s_{10} or s_{15} (blue cells in Fig. 3) should be visited along the planned trajectories, respectively. Cells s_3 and s_{12} (yellow cells) belong to the forbidden region $\mathbf{\Pi}_5$ that should always be avoided. Cells s_6 and s_{11} (green cells) belong to the final region $\mathbf{\Pi}_4$ and $\mathbf{\Pi}_6$ that should be visited by a robot at the final state, respectively.



(a) case 1: $R_s = \{r_1\}$.

(b) case 2: $R_s = \{r_2\}$.

(c) case 3: $R_s = \{r_1, r_2\}$.

Fig. 3. Simulation results of MRS $\Re$ for the Boolean specification task φ and different security constraints R_s.

For the MRS $\Re$ with Boolean specification task φ, three different security constraints are investigated and the simulation results are shown in Fig. 3. For exemplification purposes, we consider the unitary distance vector $w = \vec{1}$ for ILPP (10). Let $R_s = \{r_1\}$, which means that the final state of the robot r_1 needs to be hidden. By assuming that the maximal number of intermediate states (markings) of each robot is 8 (i.e., $h = 4$) and solving ILPP (10), we can obtain one of the optimal transition firing sequences starting from the initial state as follows:

$$\text{robot } r_1 : \ t_3 t_{17} t_{18} t_{15},$$
$$\text{robot } r_2 : \ t_{48} t_{40},$$

with total travel distance equals six. These sequences correspond to two actual path trajectories (solid line) as depicted in Fig. (3a). In addition, we also present a possible path of r_1 (dash line) that has the same observations as the actual one from the intruder's point of view. It is easy to observe that r_1 takes extra movements before arriving region s_6 in order to meet the security requirement. The simulation results in Figs. (3b) and (3c) correspond to the situation where $R_s = \{r_2\}$ (i.e., r_2 is a secret robot) and $R_s = \{r_1, r_2\}$ (i.e., both r_1 and r_2 are secret robots), respectively.

VI. Conclusion

In this letter, we study the path planning of MRSs for Boolean specification tasks with security requirements. By modelling the mobile capability of the MRSs with labeled Petri net models, an optimal ILPP solution is developed to plan a path for each individual robot of the system such that the Boolean specification task is fulfilled with the minimal cost, while the final positions of a set of secret robots are preserved against the external intruder. In addition, we show that the developed approach can be easily extended to the situation where the initial positions of some robots need to be hidden. In future work, we would like to extend the developed approach to other types of security constraints, e.g., k-step opacity.

References

[1] C. He, Y. Wan, Y. Gu, and F. Lewis, "Integral reinforcement learning-based multi-robot minimum time-energy path planning subject to collision avoidance and unknown environmental disturbances," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 983–988, 2020.

[2] S. Zhang and F. Pecora, "Online sequential task assignment with execution uncertainties for multiple robot manipulators," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6993–7000, 2021.

[3] P. Boström-Rost, D. Axehill, and G. Hendeby, "On global optimization for informative path planning," *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 833–838, 2018.

[4] C. Miao, G. Chen, C. Yan, and Y. Wu, "Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm," *Computers and Industrial Engineering*, vol. 156, 107230, 2021.

[5] H. Ma, W. Hoenig, L. Cohen, T. Uras, H. Xu, S. Kumar, N. Ayanian, and S. Koenig, "Overview: A hierarchical framework for plan generation and execution in multirobot systems," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 6–12, 2017.

[6] W. Shi, Z. He, W. Tang, W. Liu, and Z. Ma, "Path planning of multi-robot systems with Boolean specifications based on simulated annealing," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6091–6098, 2022.

[7] P. Lv, G. Luo, X. Yin, Z. Ma, and S. Li, "Optimal multi-robot path planning for cyclic tasks using Petri nets," *In Proceedings of the International Workshop on Discrete Event Systems*, pp. 9–15, 2022.

[8] M. Kloetzer and C. Mahulea, "Path planning for robotic teams based on LTL specifications and Petri net models," *Discrete Event Dynamic Systems*, vol. 30, no. 1, pp. 55–79, 2020.

[9] Y. Yan, D. Cheng, J. Feng, H. Li, and J. Yue, "Survey on applications of algebraic state space theory of logical systems to finite state machines," *Science China Information Sciences*, vol. 66, no. 1, 111201, 2023.

[10] C. Mahulea, M. Kloetzer, and R. González, *Path planning of cooperative mobile robots using discrete event models*, John Wiley & Sons, 2020.

[11] J. Luo, Y. Wan, W. Wu, and Z. Li, "Optimal Petri-net controller for avoiding collisions in a class of automated guided vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4526–4537, 2020.

[12] Z. He, R. Zhang, N. Ran, and C. Gu, "Path planning of multi-type robot systems with time windows based on timed colored Petri nets," *Applied Sciences*, vol. 12, no. 1, 6878, 2022.

[13] C. Mahulea and M. Kloetzer, "Robot planning based on Boolean specifications using Petri net models," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2218–2225, 2017.

[14] C. Mahulea, E. Montijano, and M. Kloetzer, "Distributed multirobot path planning in unknown maps using Petri net models," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2063–2068, 2020.

[15] H. Zhang, J. Luo, and J. Long, "Multi-robot path planning using Petri nets," *In Proceedings of the International Conference on Verification and Evaluation of Computer and Communication Systems*, pp. 15–26, 2020.

[16] S. Yang, X. Yin, S. Li S, and M. Zamani, "Secure-by-construction optimal path planning for linear temporal logic tasks," *In Proceedings of the 59th IEEE Conference on Decision and Control*, pp. 4460–4466, 2020.

[17] A. Saboori and C.N. Hadjicostis, "Coverage analysis of mobile agent trajectory via state-based opacity formulations," *Control Engineering Practice*, vol. 19, no. 9, pp. 967–977, 2011.

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

[19] C. Hadjicostis, "Trajectory planning under current-state opacity constraints," *IFAC-PapersOnLine*, vol. 51, no. 7, pp. 337–342, 2018.

[20] X. Cong, M. Fanti, A. Mangini, and Z. Li, "On-line verification of current-state opacity by Petri nets and integer linear programming," *Automatica*, vol. 94, pp. 205–213, 2018.