Security-Based Path Planning of Multi-Robot Systems by Partially Observed Petri Nets and Integer Linear Programming

Zhou He, *Member, IEEE*, Jiaxing Yuan, Ning Ran, *Member, IEEE*, and Xiang Yin, *Member, IEEE*

***Abstract*—This letter addresses the security-based path planning of multi-robot systems with Boolean specification tasks. Specifically, we consider the existence of an external intruder that is capable of partially monitoring the behavior of each robot. The security constraint requests that the intruder cannot infer whether the system has accessed the secret region based on the monitoring (constraint I), or the intruder cannot uniquely identify whether a specific robot has accessed the secret region (constraint II). In order to allocate the most energy-efficient pathway for every robot to fulfill the Boolean specification and security constraint, we utilize partially observed Petri nets to model the mobility capabilities of multi-robot systems and the observation of the intruder. Then, we provide an integer linear programming based solution that can generate optimal paths for the system. The validity of the proposed methodology is demonstrated through simulation studies.**

***Index Terms*—Discrete event system, partially observed Petri net, multi-robot, information security, high-level task.**

I. Introduction

MULTI-ROBOT systems (MRSs) are increasingly utilized across a range of fields, including intelligent manufacturing, medical services, and autonomous driving systems [1], [2], [3]. Path planning holds significant importance in MRSs. Traditional path planning researches primarily address the low-level tasks [4], [5]. In recent years, high-level tasks described by Boolean specification and linear temporal logic (LTL) have found wide application in practical problems and received great attention, such as resource collection and monitoring patrols [6], [7], [8].

Owing to their compact representation of state space, Petri nets (PNs) are extensively used for modeling and analyzing multi-robot systems [9], [10], [11]. By utilizing PNs structural properties and state equations, paths of MRSs and Boolean specification tasks can be expressed by a set of linear equations. As a consequence, the optimal solution for the path planning problem of MRSs with Boolean specification tasks can be obtained by solving an integer linear programming problem (ILPP) [11]. For cyclic tasks described by LTLs, an efficient optimal path planning algorithm is developed based on the construction of the basis reachability graph of a product PN model [12]. In order to tackle large-scaled MRSs, an efficient heuristic method based on the simulated annealing algorithm is developed such that near optimal solutions can be found in a reasonable time [13].

To handle tasks cooperatively, each robot of MRSs may be required to share local information and communicate with the control center through networks. However, external intruders may use malicious network attacks to obtain some critical information of the robot, thus posing a severe threat to the security and privacy of the system. Therefore, path planning of MRSs with security and privacy concerns has also received widespread attention recently [14], [15], [16], [17].

This letter studies the security-based path planning of MRSs with Boolean specification tasks. Particularly, we consider two types of security constraints and propose an optimal solution based on ILPPs. Our goal is to allocate the minimal cost path for each robot such that the Boolean specification tasks are fulfilled while ensuring that the intruder cannot infer whether the system has accessed the secret regions based on the monitoring, or/and the intruder cannot uniquely identify whether a specific robot has accessed the secret regions.

Recently, multi-robot path planning with LTL tasks and security constraints is investigated in [16]. Based on the construction of a product automaton that synchronizes the entire system and the copy patterns for security constraints I and II, a graph search algorithm is developed to obtain an optimal plan. In this letter, Petri nets are used to model MRSs, which do not require the complete enumeration of the entire state space. In [17], the security constraints are imposed independently on each robot and the secret instant is fixed for the initial or final state. In this letter, the security constraints essentially extend that of [17] in two folds: (i) first,

Manuscript received 3 January 2024; revised 26 February 2024; accepted 13 March 2024. Date of publication 25 March 2024; date of current version 10 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62373234, Grant 62373132, and Grant 62061136004; in part by the Shaanxi Provincial Natural Science Foundation under Grant 2023-JC-YB-564; in part by the Foundation of Hebei Education Department under Grant BJ2021008; and in part by the Chunhui Project of Ministry of Education of China under Grant HZKY20220257. Recommended by Senior Editor A. P. Aguiar. *(Corresponding author: Xiang Yin.)*

Zhou He and Jiaxing Yuan are with the School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China (e-mail: hezhou@sust.edu.cn; 220611015@sust.edu.cn).

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

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

Digital Object Identifier 10.1109/LCSYS.2024.3381182

the security instant of our interest is arbitrary along the entire path. Specifically, we do not require when the secrete region is accessed by robots; (ii) second, the security constraint is not imposed independently on each robot. Instead, we have to consider the joint behavior of the entire team. Specifically, in security constraint II, if a secrete region has been visited by a specific robot, there exists another different robot with a possible path that also visited it. Therefore, we have to capture the correlation between each pair of paths. In this letter, some new methods are developed to convert security constraints I and II and their correlations into a set of linear algebraic equations. Then, an ILPP solution is proposed to obtain an optimal plan that both secure and specifications are achieved.

This letter is divided into following sections. The essential preparations are presented in Section II. Section III describes the task specification and problem formulation. The proposed path planning method is elaborated in Section IV. The effectiveness of the presented method is demonstrated through some case studies in Section V. Finally, conclusion and future work are presented in Section VI.

II. Preliminary

A. Labelled Petri Net

A Petri net (PN) is composed of 4-tuple $N = (P, T, Pre, Post)$, where P is a finite set of n places denoted by circles; T is a finite set of m transitions denoted by bars; $Pre{:}P \times T \longrightarrow \mathbb{N}^{n\times m}$ and $Post : P \times T \longrightarrow \mathbb{N}^{n\times m}$ respectively are *pre-* and *post-incidence matrices* which state the arcs connecting places and transitions, where $\mathbb{N} = \{0, 1, 2, \ldots\}$ is the set of non-negative integers. The incidence matrix is defined as $C = Post - Pre \in \mathbb{Z}^{n\times m}$, where $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$ denotes the set of integers.

In a PN, the state is a *marking* that is represented by an n-component vector $M \in \mathbb{N}^n$. The count of tokens at a place p within marking M is represented by $M(p)$. A PN system $\langle N, M_0\rangle$ is a net N initially marked by M_0.

A transition t_i is *enabled* at a marking M if $M \geq Pre(\cdot, t_i)$, indicated as $M[t_i\rangle$, where $Pre(\cdot, t_i)$ denotes the corresponding column in the pre-incidence matrix Pre for transition t_i. Firing an enabled transition t_i results in a new marking M', denoted by $M[t_i\rangle M'$. This process is captured by the following *state equation*:

$$M' = M + C \cdot \vec{t}_i \tag{1}$$

where $\vec{t}_i$ represents an m-dimensional firing vector that associates with the i-th canonical basis vector.

A *run* of a PN system $\langle N, M_0\rangle$ is defined as a transition sequence $\sigma = t_1 t_2 \ldots t_h \in T^*$ that satisfies the condition $M_0[t_1\rangle M_1[t_2\rangle M_2 \ldots [t_h\rangle M_h$ (denoted as $M_0[\sigma\rangle M_h$ for simplicity). Let $\vec{\sigma} \in \mathbb{N}^m$ represent the *firing count vector* such that its i-th element indicates the frequency of occurrence of transition t_i within the sequence σ.

A *labeled Petri net* (LPN) system $G = (N, M_0, E, \lambda)$ is a PN system with an alphabet E (a set of labels), and a labeling function $\lambda : T \to E \cup \{\varepsilon\}$ that assigns a label $e \in E$ or the empty label ε to every transition $t \in T$. A transition t with a label $e \in E$ is said to be observable (i.e., $\lambda(t) \in E$). Otherwise t is said to be unobservable (i.e., $\lambda(t) = \varepsilon$). Thus, the set of transitions T is divided into two distinct sets: the observable transition set $T_o = \{t \in T | \lambda(t) \in E\}$ with $m_o = |T_o|$, and the unobservable transition set $T_u = \{t \in T | \lambda(t) = \varepsilon\}$ with $m_u = |T_u|$, where $T = T_o \cup T_u$ and $T_o \cap T_u = \varnothing$.

For an observable transition sequence $\sigma_o \in T_o^*$ that is composed of observable transitions, we use an m_o-dimensional firing vector $\vec{\sigma}_o$ to represent the number of occurrences of observable transitions in σ_o. Analogously, we define an m_u-dimensional firing vector $\vec{\sigma}_u$ for an unobservable transition sequence $\sigma_u \in T_u^*$.

The labeling function λ can be further extended to a transition sequence $\sigma = t_1 \ldots t_h \in T^*$, such that $\lambda(\sigma) = \lambda(t_1) \cdots \lambda(t_h)$ if $\sigma \neq \varepsilon$, otherwise $\lambda(\sigma) = \varepsilon$. An observation of a run $\sigma \in T^*$ is a label sequence $w = \lambda(\sigma)$ that is composed of labels of observable transitions in σ.

Definition 1: Given a PN $N = (P, T, Pre, Post)$ and a subset $T_A \subseteq T$, the T_A-induced subnet of N is a PN $N_A = (P, T_A, Pre_A, Post_A)$, where Pre_A and $Post_A$ are the restrictions of Pre and $Post$ to $P \times T_A$, i.e, N_A is the net obtained from N by removing all transitions in $T \backslash T_A$, which is denoted by $N_A \angle_{T_A} N$.

According to Definition 1, the matrices $C_u = Post_u - Pre_u$ and $C_o = Post_o - Pre_o$ are used to represent the restrictions of the incidence matrix C to T_u and T_o, respectively.

B. Modelling of Multi-Robot System

In this letter, we consider an MRS $\mathfrak{R} = (S, R)$ that consists of k identical robots working in a static workspace that is divided into n *cells*, where $R = \{r_1, \ldots, r_k\}$ and $S = \{s_1, \ldots, s_n\}$. An external intruder has full knowledge of the system, but only partial observation of each robot's behavior. An MRS $\mathfrak{R} = (S, R)$ and the observation behavior of the intruder can be modelled by an LPN as follows. Each cell $s_i \in S$ is modelled by a place $p_i \in P$, the action of moving from any cell to another neighborhood cell is modelled by a transition $t_j \in T$, each robot is modelled by a token in the LPN, and the state of the MRS is represented by a marking M. We assign each transition a label $e \in E \cup \{\varepsilon\}$ to denote the observation of the movement from the view of the intruder. When robots move to an unobservable cell, the observed results of the external intruder will be empty, i.e., $w = \varepsilon$.

III. Task Specification and Problem Statement

A. Boolean Specification

In the rest of this letter, we study a global high-level task that is characterized by a Boolean specification formula. Let $\Omega = \{\Pi_1, \Pi_2, \ldots\}$ be a set of regions of interest. The fundamental unit that makes up a Boolean specification formula is an atomic proposition in $\Omega_t = \{\Pi_1, \Pi_2, \ldots\}$ or $\Omega_f = \{\pi_1, \pi_2, \ldots\}$. Atomic propositions $\Pi \in \Omega_t$ and $\pi \in \Omega_f$ mean that region Π must be visited at least one time along the trajectory, and at least one of the robots should end up in region Π, respectively. A Boolean specification formula can be formed from the logical relations (i.e., disjunction $\vee$, conjunction $\wedge$, and negation $\neg$) between atomic propositions.

Specifically, the Boolean specification φ consists of three sub-specifications

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

- The sub-specification $Y = y_1 \wedge \cdots \wedge y_q$ denotes the logical constraints on the *task regions* such that

$$y_i = \bigvee_{\Pi \in \Omega_{y_i}} \Pi, \ \ \Omega_{y_i} \subseteq \Omega_t. \tag{3}$$

It indicates that region $\mathbf{\Pi}$ that corresponding to atomic proposition $\Pi \in \Omega_{y_i}$ should be visited along the trajectory.

- The sub-specification A denotes the logical constraints on the *forbidden regions* such that

$$A = \bigwedge_{\Pi \in \Omega_a} (\neg \Pi) = \neg \left(\bigvee_{\Pi \in \Omega_a} \Pi \right), \ \Omega_a \subseteq \Omega_t. \tag{4}$$

It indicates that region $\mathbf{\Pi}$ that corresponding to atomic proposition $\Pi \in \Omega_a$ should not be accessed by any robot.

- The sub-specification $U = u_1 \wedge \cdots \wedge u_d$ denotes the logical constraints on the *final regions* such that

$$u_i = \bigvee_{\pi \in \Omega_{u_i}} \pi, \ \Omega_{u_i} \subseteq \Omega_f. \tag{5}$$

It indicates that region $\mathbf{\Pi}$ that corresponding to atomic proposition $\pi \in \Omega_{u_i}$ should eventually be occupied by a robot.

B. Security Constraint

We consider an intruder that owns complete knowledge of the MRS, including the map of workspace S and the system's initial state (modeled by the initial markings M_0 of the LPN system). The system's internal states cannot be directly accessed by the intruder during the execution of tasks. However, the intruder can identify and partially monitor the behavior of each robot.

At each state, a robot has the capability to move to one neighborhood cell at most, hence the path trajectory of each robot can be represented by a sequence of movement strategies that are modeled by unobservable and observable transition sequences of LPN. Let $M_{j,i}$ be the state of robot r_j at i-th step, a *path* ϱ_j of robot r_j is a sequence of states

$$\varrho_j = M_{j,0} M_{j,u_1} M_{j,o_1} \ldots M_{j,u_h} M_{j,o_h}, \tag{6}$$

with a sequence of runs (i.e., movement strategies) $\sigma_j = \sigma_{j,u_1} \sigma_{j,o_1} \ldots \sigma_{j,u_h} \sigma_{j,o_h}$ such that

$$M_{j,0}[\sigma_{j,u_1}\rangle M_{j,u_1}[\sigma_{j,o_1}\rangle M_{j,o_1} \ldots M_{j,u_h}[\sigma_{j,o_h}\rangle M_{j,o_h}, \tag{7}$$

where $M_{j,0}$ is the initial state of robot r_j, $h \in \mathbb{N}$ is a designed parameter that denotes the maximal number of observable movements, $\sigma_{j,u_i} \in T_u^*$ and $\sigma_{j,o_i} \in T_o^*$ $(i = 1, \ldots, h)$ are observable transition sequence and unobservable transition sequence respectively, and M_{j,u_i} and M_{j,o_i} are unobservable state and observable state respectively. The observation of a path ϱ_j for the intruder is represented by

$$\lambda(\sigma_j) = \lambda(\sigma_{j,o_1}) \lambda(\sigma_{j,o_2}) \cdots (\sigma_{j,o_h}).$$

We aim to protect the identity of specific robot performing some certain important tasks that are intended to remain concealed from the intruder. Such important behaviors can be represented by visiting certain secret regions $\mathbf{\Pi}_s \in \Omega$. We write $\varrho_j \rightsquigarrow \mathbf{\Pi}_s$ (resp., $\varrho_j \nrightsquigarrow \mathbf{\Pi}_s$) to represent that path ϱ_j has (resp., has not) visited secret regions. Two different type of security constraints are defined as follows.

Security constraint I: The intruder cannot determine whether the system has accessed secret regions or not. Specifically, if any robot $r_j \in R$ visits a secret region $\mathbf{\Pi}_s$ along the planned path, i.e., $\varrho_j \rightsquigarrow \mathbf{\Pi}_s$, there exists another possible path ϱ_j' for robot j with a sequence of runs σ_j' such that:

$$\varrho_j \rightsquigarrow \mathbf{\Pi}_s, \tag{8a}$$

$$\varrho_j' \nrightsquigarrow \mathbf{\Pi}_s, \tag{8b}$$

$$\lambda(\sigma_j) = \lambda(\sigma_j'). \tag{8c}$$

In another word, security constraint I imposes that if any robot r_j of the system visits a secret region along the planned path, there exists a possible path for the same robot r_j that avoids the secret region, while both paths must be indiscernible to the intruder. However, the intruder may determine that robot r_j is the only possible robot that may visit the secret region. Then, the intruder may focus its attention on tracking this specific robot r_j and take actions in the future. This results in the establishment of following security constraint.

Security constraint II: If any robot r_j visits a secret region $\mathbf{\Pi}_s$ along the planned path ϱ_j, there exists another robot $r_l \in R$ $(r_l \neq r_j)$ with a possible path ϱ_l such that:

$$\varrho_j \rightsquigarrow \mathbf{\Pi}_s, \tag{9a}$$

$$\varrho_l \rightsquigarrow \mathbf{\Pi}_s, \ l \neq j. \tag{9b}$$

In another word, security constraint II requires that if any robot of the system has visited a secret region along the planned path, there should exist another robot with a possible path that has also visited the secret region. When both security constraints I and II are considered, the intruder can neither determine whether the secret region has been accessed nor uniquely suspect whether a specific robot has accessed the secret region, which enhances the security of the system.

C. Problem Statement

In this letter, we aim to find an optimal path ϱ_j for each robot r_j such that both the Boolean specification and the security constraints are fulfilled. Let $\omega(\varrho_j)$ denote the cost of path ϱ_j. The security-based multi-robot path planning problem can be formulated as follows.

Problem 1: Consider an MRS $\mathfrak{R} = (S, R)$, an LPN system $G = (N, M_0, E, \lambda)$ that models the MRS $\mathfrak{R}$, a global task φ, a set of secrete regions $\mathbf{\Pi}_s$, we aim to solve the following problem:

$$\min \ \sum_{j=1}^{k} \omega(\varrho_j)$$

$$s.t. \begin{cases} \varphi \text{ is satisfied,} \\ \text{condition (8) is satisfied,} \\ \text{condition (9) is satisfied.} \end{cases}$$

IV. Planning Algorithm

A. Boolean Specification Transformation

Recall that the global task is described by $\varphi = Y \wedge A \wedge U$, where $Y = y_1 \wedge \cdots \wedge y_q$ denotes the logic requirements on the task regions. We define an eigenvector of n-component $v_{y_i} = [v_{y_i}(1), v_{y_i}(2), \ldots, v_{y_i}(n)] \in \{0, 1\}^{1 \times n}$ for each $y_i \in Y$, where $v_{y_i} = 1$ if $s_j \in \Pi \ \& \ \Pi \in \Omega_{y_i}$, otherwise $v_{y_i} = 0$. Let $V_Y = [v_{y_1}, \ldots, v_{y_q}]$ be the characteristic matrix of Y. When condition $v_{y_i} \cdot M_{j,i} \geq 1$ holds for marking $M_{j,i}$, it means that the task region corresponding to sub-specification y_i is visited by robot j at step i, i.e., sub-specification y_i is satisfied.

For each sub-specification $u_j \in U$, we denote by $v_{u_j} = [v_{u_j}(1), v_{u_j}(2), \ldots, v_{u_j}(n)] \in \{0,1\}^{1\times n}$ the eigenvector of u_j and by $V_U = [v_{u_1}, v_{u_2}, \ldots, v_{u_q}]^T$ the characteristic matrix of U, where $v_{u_j} = 1$ if $s_j \in \Pi$ & $\Pi \in \Omega_{u_j}$, otherwise $v_{u_j} = 0$. When condition $v_{u_j} \cdot M_{j,h} \geq 1$ holds for marking $M_{j,h}$, it means that the final region corresponding to sub-specification u_i is satisfied at the final state, since h denotes the maximal number of observable movements.

Similarly, for sub-specification A, we define a characteristic vector $V_A = [v_a(1), \ldots, v_a(n)] \in \{0,1\}^{1\times n}$, where $v_a(j) = 1$ if $s_j \in \Pi$ & $\Pi \in \Omega_a$, otherwise $v_a(j) = 0$. Therefore, task $\varphi = Y \wedge A \wedge U$ that imposes logical requirement on the trajectory and final state can be converted into the following linear restrictions:

$$\begin{cases} V_Y \cdot \sum_{j=1}^{k}\sum_{i=1}^{h} \left(M_{j,u_i} + M_{j,o_i}\right) \geq \vec{1}, & \text{(10a)} \\ V_A \cdot \sum_{j=1}^{k}\sum_{i=1}^{h} \left(M_{j,u_i} + M_{j,o_i}\right) = 0, & \text{(10b)} \\ V_U \cdot \sum_{j=1}^{k} M_{j,o_h} \geq \vec{1}, & \text{(10c)} \end{cases}$$

where h represents the maximal number of movements of robots, and M_{j,u_i} and M_{j,o_i} denote the unobservable state and observable state respectively.

B. Security Constraint Transformation

A path of robot $r_j \in R$ can be represented by a finite sequence of unobservable and observable states

$$\varrho_j = M_{j,0}M_{j,u_1}M_{j,o_1}, \ldots M_{j,u_h}M_{j,o_h}.$$

According to Eqs. (7) and (1), following equation should hold to ensure the correctness of path ϱ_j.

$$\begin{cases} \left.\begin{aligned} &M_{j,u_i} = M_{j,o_{i-1}} + C_u \cdot \vec{\sigma}_{j,u_i}, \\ &M_{j,o_{i-1}} - Pre_u \cdot \vec{\sigma}_{j,u_i} \geq \vec{0}, \\ &M_{j,o_i} = M_{j,u_i} + C_o \cdot \vec{\sigma}_{j,o_i}, \\ &M_{j,u_i} - Pre_o \cdot \vec{\sigma}_{j,o_i} \geq \vec{0}, \end{aligned}\right\} & \text{(11a)} \\ \sum_{t\in T_o} \vec{\sigma}_{j,o_i}(t) \leq 1, \sum_{t\in T_u} \vec{\sigma}_{j,u_i}(t) = 0, & \text{(11b)} \\ \sum_{t\in T_u} \vec{\sigma}_{j,u_i}(t) \leq 1, \sum_{t\in T_o} \vec{\sigma}_{j,o_i}(t) = 0, & \text{(11c)} \\ M_{j,o_0} = M_{j,0}, & \text{(11d)} \\ \vec{\sigma}_{j,u_i} \in \mathbb{N}^{m_u}, \ \vec{\sigma}_{j,o_i} \in \mathbb{N}^{m_o}, & \text{(11e)} \\ i = 1, 2, \ldots, h. & \text{(11f)} \end{cases}$$

In particular, conditions (11a), (11d), (11e), and (11f) jointly ensure the correctness of the state transitions, and conditions (11b) and (11c) impose that robot is restricted to moving at most one cell per step.

We denote by $\mathcal{L} \in \{0,1\}^{|E|\times m}$ the *labeling incidence matrix* such that $\mathcal{L}(j,i) = 1$ if $\lambda(t_i) = e_j \in E$, otherwise $\mathcal{L}(j,i) = 0$. When t_i is an unobservable transition, the associated column are all zeros, i.e., $\mathcal{L}(\cdot, i) = 0$. In the subsequent propositions, we show how to use linear algebraic constraints to represent security constraints I and II.

Proposition 1: The security constraint I in Eq. (8) can be converted into following linear algebraic constraint

$$\mathcal{S}_1 \begin{cases} 1 - \sum_{i=1}^{h}\sum_{s_r\in\mathbf{\Pi}_s} \left(M_{j,u_i}(p_r) + M_{j,o_i}(p_r)\right) \leq z(j)\cdot H, & \text{(12a)} \\ \sum_{i=1}^{h}\sum_{s_r\in\mathbf{\Pi}_s} \left(M'_{j,u_i}(p_r) + M'_{j,o_i}(p_r)\right) \leq z(j)\cdot H, & \text{(12b)} \\ \left.\begin{aligned} &\mathcal{L}_o\cdot\sigma_{j,o_i} - \mathcal{L}_o\cdot\sigma'_{j,o_i} \leq H\cdot z(j)\cdot\vec{1}, \\ &\mathcal{L}_o\cdot\sigma_{j,o_i} - \mathcal{L}_o\cdot\sigma'_{j,o_i} \geq -H\cdot z(j)\cdot\vec{1}, \end{aligned}\right\} & \text{(12c)} \\ z(1) + \cdots + z(k) \leq k-1, & \text{(12d)} \\ z(j) \in \{0,1\}, & \text{(12e)} \\ j = 1, \ldots, k, \ \ i = 1, \ldots, h & \text{(12f)} \end{cases}$$

where $h \in \mathbb{N}$ is a pre-defined parameter that represents the maximal number of observable movements, and $H \in \mathbb{R}_{\geq 0}$ is a sufficiently large constant.

Proof: Conditions (12d) and (12e) ensure that at least one variable z should be zero. Suppose that $z(j) = 0$, constraint (12a) could be reduced to $1 \leq \sum_{i=1}^{h}\sum_{s_r\in\mathbf{\Pi}_s} (M_{j,u_i}(p_r) + M_{j,o_i}(p_r))$, which imposes that robot r_j visits the secrete region $\mathbf{\Pi}_s$ along the planed path ϱ_j, i.e., $\varrho_j \rightsquigarrow \mathbf{\Pi}_s$.

Then, condition (12b) can be simplified as $\sum_{i=1}^{h}\sum_{s_r\in\mathbf{\Pi}_s} (M'_{j,u_i}(p_r) + M'_{j,o_i}(p_r)) \leq 0$, which implies that there exits another possible path ϱ'_j of robot j that dose not visit the secrete region $\mathbf{\Pi}_s$, i.e., $\varrho'_j \nrightarrow \mathbf{\Pi}_s$.

In addition, condition (12c) can be rewritten in the following form:

$$\left.\begin{aligned} &\mathcal{L}_o\cdot\sigma_{j,o_i} - \mathcal{L}_o\cdot\sigma'_{j,o_i} \leq \vec{0}, \\ &\mathcal{L}_o\cdot\sigma_{j,o_i} - \mathcal{L}_o\cdot\sigma'_{j,o_i} \geq \vec{0}, \end{aligned}\right\} \Rightarrow \mathcal{L}_o\cdot\sigma_{j,o_i} = \mathcal{L}_o\cdot\sigma'_{j,o_i}. \quad \text{(13)}$$

This implies that from the perspective of an intruder, ϱ'_j and ϱ_j appear identical observation at every state, i.e., $\lambda(\sigma_j) = \lambda(\sigma'_j)$.

Suppose that $z(j) = 1$ $j \in \{1, \ldots, k\}$, we can simplify condition (12) as following form:

$$\begin{cases} 1 - \sum_{i=1}^{h}\sum_{s_r\in\mathbf{\Pi}_s} \left(M_{j,u_i}(p_r) + M_{j,o_i}(p_r)\right) \leq H, & \text{(a)} \\ \sum_{i=1}^{h}\sum_{s_r\in\mathbf{\Pi}_s} \left(M'_{j,u_i}(p_r) + M'_{j,o_i}(p_r)\right) \leq H, & \text{(b)} \\ \left.\begin{aligned} &\mathcal{L}_o\cdot\sigma_{j,o_i} - \mathcal{L}_o\cdot\sigma'_{j,o_i} \leq H\cdot\vec{1}, \\ &\mathcal{L}_o\cdot\sigma_{j,o_i} - \mathcal{L}_o\cdot\sigma'_{j,o_i} \geq -H\cdot\vec{1}, \end{aligned}\right\} & \text{(c)} \end{cases}$$

Since H is a sufficiently large number, these constraints become redundant. Therefore, the security constraint I in Eq. (8) is implemented by constraint (12). ■

Proposition 2: The security constraint II in Eq. (9) can be converted into following linear algebraic constraint

$$\mathcal{S}_2 \begin{cases} 1 - \sum_{i=1}^{h}\sum_{s_r\in\mathbf{\Pi}_s} \left(M_{j,u_i}(p_r) + M_{j,o_i}(p_r)\right) \leq z(j)\cdot H, & \text{(14a)} \\ z(1) + \cdots + z(k) \leq k-2, & \text{(14b)} \\ z(j) \in \{0,1\}, & \text{(14c)} \\ j = 1, \ldots, k, \ \ i = 1, \ldots, h, & \text{(14d)} \end{cases}$$

Proof: Similar to the proof of Proposition 1, condition (14b) guarantees that at least two variables $z(j)$ and $z(l)$ are equal to zero ($j \neq l$), which consequently guarantees that at least two robots r_j and r_l have visited the secrete region $\mathbf{\Pi}_s$ along the planed path ϱ_j according to condition (14a), i.e., $\varrho_j \rightsquigarrow \mathbf{\Pi}_s$ and $\varrho_l \rightsquigarrow \mathbf{\Pi}_s$. ■

C. Security-Based Path Planning Method

We denote by $w = [w(t_1), w(t_2), \ldots, w(t_m)] \in \mathbb{R}^{1\times m}$ the cost associated with each transition, i.e., cost corresponding to each movement. Based on above results, an optimal solution for Problem 1 can be obtained by ILPP (15). Particularly, the objective of Eq. (15) minimizes the overall cost of the system. Condition (15a) ensures the correctness of each planed path ϱ_j, condition (15b) ensures the correctness of another possible path ϱ_j', condition (15c) implements the Boolean specification tasks, and conditions (15d) and (15e) enforce the security constraints I and II, respectively. Therefore, the optimal solution $\sigma_j = \sigma_{j,u_1}\sigma_{j,o_1}, \ldots \sigma_{j,u_h}\sigma_{j,o_h}$ $(j = 1, \ldots, k)$ of ILPP (15) denotes the sequence of runs (i.e., movement strategy) of each robot. Note that the collision avoidance problem can be further solved by using supervisory control theory of discrete event systems [18].

$$\min\ w \cdot \sum_{j=1}^{k}\sum_{i=1}^{h}\left(\vec{\sigma}_{j,u_i} + \vec{\sigma}_{j,o_i}\right)$$

$$s.t.\begin{cases}
\left.\begin{array}{l}
M_{j,u_i} = M_{j,o_{i-1}} + C_u \cdot \vec{\sigma}_{j,u_i},\\
M_{j,o_{i-1}} - Pre_u \cdot \vec{\sigma}_{j,u_i} \geq \vec{0},\\
M_{j,o_i} = M_{j,u_i} + C_o \cdot \vec{\sigma}_{j,o_i},\\
M_{j,u_i} - Pre_o \cdot \vec{\sigma}_{j,o_i} \geq \vec{0},\\
\sum\limits_{t\in T_o} \vec{\sigma}_{j,o_i}(t) \leq 1,\ \sum\limits_{t\in T_u} \vec{\sigma}_{j,u_i}(t) = 0,\\
\sum\limits_{t\in T_u} \vec{\sigma}_{j,u_i}(t) \leq 1,\ \sum\limits_{t\in T_o} \vec{\sigma}_{j,o_i}(t) = 0,\\
M_{j,o_0} = M_{j,0},\\
M_{j,u_i}, M_{j,o_i} \in \mathbb{N}^n,\ \vec{\sigma}_{j,u_i} \in \mathbb{N}^{m_u},\ \vec{\sigma}_{j,o_i\in\mathbb{N}^{m_o}},
\end{array}\right\} & (15a)\\
\left.\begin{array}{l}
M'_{j,u_i} = M'_{j,o_{i-1}} + C_u \cdot \vec{\sigma}'_{j,u_i},\\
M'_{j,o_{i-1}} - Pre_u \cdot \vec{\sigma}'_{j,u_i} \geq \vec{0},\\
M'_{j,o_i} = M'_{j,u_i} + C_o \cdot \vec{\sigma}'_{j,o_i},\\
M'_{j,u_i} - Pre_o \cdot \vec{\sigma}'_{j,o_i} \geq \vec{0},\\
\sum\limits_{t\in T_o} \vec{\sigma}'_{j,o_i}(t) \leq 1,\ \sum\limits_{t\in T_u} \vec{\sigma}'_{j,u_i}(t) = 0,\\
\sum\limits_{t\in T_u} \vec{\sigma}'_{j,u_i}(t) \leq 1,\ \sum\limits_{t\in T_o} \vec{\sigma}'_{j,o_i}(t) = 0,\\
M'_{j,o_0} = M_{j,0},\\
M'_{j,u_i}, M'_{j,o_i} \in \mathbb{N}^n,\ \vec{\sigma}'_{j,u_i} \in \mathbb{N}^{m_u},\ \vec{\sigma}'_{j,o_i\in\mathbb{N}^{m_o}},
\end{array}\right\} & (15b)\\
\left.\begin{array}{l}
V_Y \cdot \sum\limits_{j=1}^{k}\sum\limits_{i=1}^{h}\left(M_{j,u_i} + M_{j,o_i}\right) \geq \vec{1},\\
V_A \cdot \sum\limits_{j=1}^{k}\sum\limits_{i=1}^{h}\left(M_{j,u_i} + M_{j,o_i}\right) = 0,\\
V_U \cdot \sum\limits_{j=1}^{k} M_{j,o_h} \geq \vec{1},
\end{array}\right\} & (15c)\\
\mathcal{S}_1, & (15d)\\
\mathcal{S}_2. & (15e)
\end{cases}$$

Complexity: The developed approach in (15) is an ILPP whose computational complexity is usually measured in terms of the number of variables and constraints. It has $4k \cdot h \cdot (m+n) + 2k$ variables and $2k \cdot h \cdot (5n+4) + 3k + 3n + 2$ constraints at most, where h is a pre-determined constant denoting the maximal number of observable movements, k represents the total number of robots, m signifies the number of transitions, and n denotes the number of cells of the map. The additional number of variables and constraints induced by security constraint II are $k\cdot|\mathbf{\Pi}_s|$ and $(k+1)\cdot|\mathbf{\Pi}_s|$, respectively, where $|\mathbf{\Pi}_s|$ denotes the number of secrete regions.

V. Case Study

In this section, several cases are discussed to illustrate the effectiveness of the developed approach. The developed algorithm is implemented by MATLAB and YALMIP subroutine. Note that although the security constraints considered in [16] are same as this letter, the method in [16] and our method are incomparable since the task specifications studied are different with each other.

A. Case 1: Security Constraint I

We use this example to demonstrate the validity of the proposed method for security constraint I, i.e., the intruder cannot determine that the system has visited the secret region. Suppose there are two robots r_1 and r_2 collecting resource in a 5×5 workspace as shown in Fig. 1(a). At each time, every robot can move to its neighborhood cell (if it exists) or stay at its current cell. Initially, robots r_1 and r_2 are located in cells s_{25} and s_5 respectively. We assume that the external intruder can only observe the column information of the workspace. Cell s_9 is an unobservable cell that any entrance to it cannot be observed. The regions of interest with special properties are as follows: s_6 and s_{16} are warehouses for storing resources; s_{13} and s_{19} contain foods; s_{17} contains water; s_7 contains critical medical supplies that should be kept confidential; and s_1 and s_{21} are obstacles.

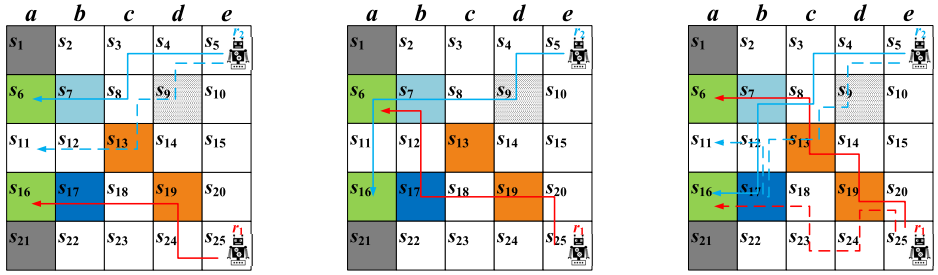
Let $\Omega = \{\mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_7\}$, where $\mathbf{\Pi}_1 = \{s_{13}\}$, $\mathbf{\Pi}_2 = \{s_{19}\}$, $\mathbf{\Pi}_3 = \{s_{17}\}$, $\mathbf{\Pi}_4 = \{s_6\}$, $\mathbf{\Pi}_5 = \{s_{16}\}$, $\mathbf{\Pi}_6 = \{s_1, s_{21}\}$, $\mathbf{\Pi}_7 = \{s_7\}$, and $\mathbf{\Pi}_s = \mathbf{\Pi}_7$. The task for the MRS is to collect foods, water, medicines and deliver them to the warehouse eventually, which can be represented by

$$\varphi = \Pi_7 \wedge (\Pi_1 \vee \Pi_2) \wedge \Pi_3 \wedge (\neg\Pi_6) \wedge \pi_4 \wedge \pi_5. \quad (16)$$

The security constraint I requests that the system should prevent the intruder from knowing that the system is collecting the medicines. For the sake of simplicity, the cost of each movement is set as one unit, i.e., $w = \vec{1}$. By assuming the maximal number of observable movements equals to seven (i.e., $h = 7$), the optimal paths of robot r_1 and r_2 can be obtained by solving ILPP (15) as shown in Fig. 1(a). Specifically, the paths in solid lines depict the actual routes taken by each robot, whereas the dashed lines illustrate possible routes which have identical observation of the actual routes from the view of the intruder, but avoid entering the secret region $\mathbf{\Pi}_7$.

B. Case 2: Security Constraint II

We use this example to demonstrate the effectiveness of the developed algorithm for security constraint II which requires the intruder cannot recognize that the specific robot r_j is the only possible robot that has accessed the secret region. The optimal paths of robot r_1 and r_2 can be obtained by solving ILPP (15) as shown in Fig. 1(b). It should be noticed that in



(a) (b) (c)

Fig. 1. Simulation results for cases 1, 2, and 3, where solid lines are the actual executed paths of robots and dashed lines are possible paths that appear identical observations of actual ones from the view of the intruder.

TABLE I
PERFORMANCE ANALYSIS OF CASES 1, 2 AND 3

	constraints	variables	solving time (s)
Case 1	3132	2942	0.798
Case 2	1534	1472	0.337
Case 3	3135	2944	0.838

order to ensure security, robot r_1 also visited the secret region $\mathbf{\Pi}_7$ so that the intruder may think that two robots have both passed through the secret region. Thus, it becomes impossible for the intruder to suspect r_2 as the only carrier of medicines.

C. Case 3: Security Constraints I and II

The aforementioned two cases illustrate security constraints I and II respectively. In this case, we consider security constraints I and II simultaneously. Particularly, it requests that the intruder cannot infer whether the system has accessed the secret region to collect medicines, and the intruder cannot uniquely identify whether a specific robot has accessed the secret region. The optimal paths of robot r_1 and r_2 can be obtained by solving ILPP (15) as shown in Fig. 1(c). Note that from the finite observation of each robot, the intruder is unable to deduce if the robot r_2 has visited the secret region, nor can it solely suspect robot r_2.

VI. CONCLUSION

In this letter, we explore security-based path planning of MRSs with high-level specification tasks. Suppose there exists an external intruder that is able to partially monitor the behavior of each robot, we aim to find an optimal path for each robot such that the global task is cooperatively completed, while the intruder cannot infer whether the system has accessed the secret region, or/and the intruder cannot uniquely identify whether a specific robot has accessed the secret region. We adopt partially observed Petri net to model the mobility of the MRSs and propose an integer linear programming method. For the future work, we aim to introduce dynamic uncertainties for the consider problem, such as motion uncertainties and environment uncertainties.

REFERENCES

[1] W. He, C. Xue, X. Yu, Z. Li, and C. Yang, "Admittance-based controller design for physical human–robot interaction in the constrained task space," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1937–1949, Oct. 2020.

[2] J. Huang, J. Zeng, X. Chi, K. Sreenath, Z. Liu, and H. Su, "Velocity obstacle for polytopic collision avoidance for distributed multi-robot systems," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3502–3509, Jun. 2023.

[3] C. Miao, G. Chen, C. Yan, and Y. Wu, "Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm," *Comput. Ind. Eng.*, vol. 156, Jun. 2021, Art. no. 107230.

[4] A. Madridano, A. Al-Kaff, D. Martin, and A. Escalera, "Trajectory planning for multi-robot systems: Methods and applications," *Expert Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114660.

[5] P. Wang, S. Gao, L. Li, B. Sun, and S. Cheng, "Obstacle avoidance path planning design for autonomous driving vehicles based on an improved artificial potential field algorithm," *Energies*, vol. 12, no. 12, p. 2342, 2019.

[6] M. Kloetzer and C. Mahulea, "Path planning for robotic teams based on LTL specifications and Petri net models," *Discrete Event Dyn. Syst.*, vol. 30, pp. 55–79, Mar. 2020.

[7] C. Mahulea, M. Kloetzer, and J. Lesage, "Multi-robot path planning with Boolean specifications and collision avoidance," *IFAC-PapersOnLine*, vol. 53, no. 4, pp. 101–108, 2020.

[8] M. Guo and M. Zavlanos, "Probabilistic motion planning under temporal tasks and soft constraints," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4051–4066, Dec. 2018.

[9] Z. He, Y. Dong, G. Ren, C. Gu, and Z. Li, "Path planning for automated guided vehicle systems with time constraints using timed Petri nets," *Meas. Control*, vol. 53, no. 9, pp. 2030–2040, 2020.

[10] 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," *Appl. Sci.*, vol. 12, no. 14, p. 6878, 2022.

[11] C. Mahulea and M. Kloetzer, "Robot planning based on Boolean specifications using Petri net models," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 2218–2225, Jul. 2018.

[12] P. Lv, G. Luo, Z. Ma, S. Li, and X. Yin, "Optimal multi-robot path planning for cyclic tasks using Petri nets," *Control Eng. Pract.*, vol. 138, Sep. 2023, Art. no. 105600.

[13] 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 Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6091–6098, Jul. 2022.

[14] G. Zhu, Z. Li, and N. Wu, "Online verification of K-step opacity by Petri nets in centralized and decentralized structures," *Automatica*, vol. 145, Nov. 2022, Art. no. 110528.

[15] S. Yang and X. Yin, "Secure your intention: On notions of pre-opacity in discrete-event systems," *IEEE Trans. Autom. Control*, vol. 68, no. 8, pp. 4754–4766, Aug. 2023.

[16] X. Yu, X. Yin, S. Li, and Z. Li, "Security-preserving multi-agent coordination for complex temporal logic tasks," *Control Eng. Pract.*, vol. 123, Jun. 2022, Art. no. 105130.

[17] W. Shi, Z. He, Z. Ma, N. Ran, and X. Yin, "Security-preserving multi-robot path planning for Boolean specification tasks using labeled Petri nets," *IEEE Control Syst. Lett.*, vol. 7, pp. 2017–2022, Jun. 2023.

[18] E. Dallal, A. Colombo, D. Del-Vecchio, and S. Lafortune, "Supervisory control for collision avoidance in vehicular networks using discrete event abstractions," *Discr. Event Dyn. Syst.*, vol. 27, pp. 1–44, Mar. 2017.