PAPER *Special Section on Circuits and Systems*

Multi-Agent Surveillance Based on Travel Cost Minimization

Kyohei MURAKATA†, ***Nonmember***, **Koichi KOBAYASHI**†a), ***and*** **Yuh YAMASHITA**†, ***Members***

SUMMARY The multi-agent surveillance problem is to find optimal trajectories of multiple agents that patrol a given area as evenly as possible. In this paper, we consider the multi-agent surveillance problem based on travel cost minimization. The surveillance area is given by an undirected graph. The penalty for each agent is introduced to evaluate the surveillance performance. Through a mixed logical dynamical system model, the multi-agent surveillance problem is reduced to a mixed integer linear programming (MILP) problem. In model predictive control, trajectories of agents are generated by solving the MILP problem at each discrete time. Furthermore, a condition that the MILP problem is always feasible is derived based on the Chinese postman problem. Finally, the proposed method is demonstrated by a numerical example.

key words: *Chinese postman problem, mixed integer programming, surveillance problem, travel cost*

1. Introduction

A cyber physical system (CPS) is a system composed of physical and information components. There are many applications such as energy management [5] and healthcare [17]. In this paper, we focus on the persistent surveillance problem, which is closely related to a smart city. The persistent surveillance problem that find trajectories of multiple agents to patrol a given area has many applications such as city safety management and disaster rescue.

The persistent surveillance problem has been studied in [1], [4], [6]–[8], [10]–[13], [15]. In surveillance of city areas and buildings, it is useful to model a surveillance area as a graph (see, e.g., [1], [8], [10]–[13]). Furthermore, it is important to apply a model predictive control (MPC) method to the surveillance problem. Using the policy of MPC, trajectories of agents are generated by solving an optimization problem at each discrete time [3], [14]. For example, when it is detected that some agents stop in failures, trajectories of remained agents are automatically changed. The authors has developed MPC-based surveillance methods for a area given by a graph in [8], [10], [12], [13].

In this paper, we focus on travel costs (i.e., total travel distances) of agents. We consider the problem of finding trajectories of agents that minimize travel costs under constraints on the surveillance performance. The surveillance performance is based on the penalty of each node to evaluate the unattended time. If constraints are satisfied, then agents may stop. Such behavior is desirable from the viewpoint of power saving.

In MPC, it is important to guarantee the feasibility of an optimization problem solved at each discrete time. For the above surveillance problem, we derive a feasibility condition under the assumption that the initial locations of agents can be arbitrarily set. Then, the Chinese postman problem (CPP) plays an important role. CPP is the problem of finding a shortest closed walk of an undirected graph in which each edge is traversed at least once, rather than exactly once [16]. Using a solution of CPP, we can derive a feasibility condition such that the number of agents and the surveillance performance are characterized. Initial locations of agents can also be determined from a solution of CPP. In also [11], CPP has been used in the surveillance problem. In [11], a solution of CPP is used as trajectories of agents. In this paper, a solution of CPP is used in placement of initial locations and a feasibility condition.

This paper is organized as follows. In Sect. 2, the optimal surveillance problem is formulated. In Sect. 3, this problem is reduced to a mixed integer linear programming (MILP) problem. In Sect. 4, we derive a feasibility condition. In Sect. 5, we present a numerical example to demonstrate the proposed method. In Sect. 6, we conclude this paper.

Notation: For the finite set A, let $|A|$ denote the number of elements of A. Let $\mathcal{R}$ denote the set of real numbers. Let $1_{m\times n}$ ($0_{m\times n}$) denote the $m \times n$ matrix whose elements are all one (zero). Let I_n denote the $n \times n$ identity matrix. For simplicity, we sometimes use the symbol 0 instead of $0_{m\times n}$, and the symbol I instead of I_n. For the matrix M, let $M^\top$ denote the transpose matrix of M. For the two matrices X and Y, let $X \otimes Y$ denote the Kronecker product of X and Y.

2. Problem Formulation

A surveillance area is given by an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $V = \{1, 2, \ldots, n\}$ is the set of nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of undirected edges. We assume that an agent can move according to a given graph, and behavior of an agent is expressed by a discrete-time system. The number of agents is given by m. Let $q_j(k) \geq 0$ denote the travel cost of the agent $j \in \{1, 2, \ldots, m\}$ until time k. The travel cost from a certain node to other node is defined by the length of the path (i.e., the number of edges of the path).

We present a simple example.

Manuscript received February 3, 2023.
Manuscript revised May 21, 2023.
Manuscript publicized July 19, 2023.

†The authors are with the Graduate School of Information Science and Technology, Hokkaido University, Sapporo-shi, 060-0814 Japan.
a) E-mail: k-kobaya@ssi.ist.hokudai.ac.jp

DOI: 10.1587/transfun.2023KEP0015

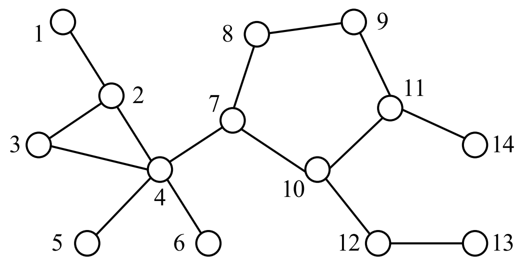

Fig. 1 Example of undirected connected graphs. Nodes except for 1 and 13 have self-loops, but these are omitted.

Example 1: Consider the case of a single agent ($m = 1$). Suppose that the surveillance area is given by the graph in Fig. 1. Suppose also that the initial location and the initial travel cost are given by the node 4 and $q_1(0) = 0$, respectively. Then, the candidates of the locations at the next time are constrained to the set $\{2, 3, 4, 5, 6, 7\}$. If the agent moves from the node 4 to 2, the travel cost is updated as $q_1(1) = 0+1 = 1$. Next, if the agent moves from the node 2 to 3, the travel cost is updated as $q_1(2) = 1 + 1 = 2$. Finally, if the agent stays at the node 3 (i.e., the self-loop $(3, 3)$ is chosen), the travel cost is not changed (i.e., $q_1(3) = 2 + 0 = 2$).

Next, we introduce the notion of the penalty for each node. The penalty $p_i(k) \geq 0$, $i \in \{1, 2, \ldots, n\}$ is defined as follows:

$$p_i(k+1) = \begin{cases} 0 & \text{if some agent is located} \\ & \text{on the node } i \text{ at time } k, \\ p_i(k) + 1 & \text{otherwise.} \end{cases} \quad (1)$$

When a node is not monitored by any agent, the penalty for this node increases. Hence, the penalty for each node can be used for evaluating the performance of surveillance.

The optimal surveillance problem is formulated as follows.

Problem 1: For the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the update rule (1) of the penalty, suppose that the current location of agents, the current penalty $p_i(t)$, the current total travel cost $q_j(t)$, and the prediction horizon $N \geq 1$ are given (t is the current time). Then, find trajectories of m agents minimizing the following cost function:

$$J = \sum_{j=1}^{m} q_j(t+N) \quad (2)$$

subject to the following constraint for each node $i \in \{1, 2 \ldots, n\}$:

$$p_i(k) \leq c, \quad k \in \{t, t+1 \ldots, t+N\}, \quad (3)$$

where $c > 0$ is a given scalar.

In this problem, the cost function (2) represents a sum of travel costs of all agents in the time interval $[t, t+N]$. By the constraint (3), the performance of surveillance is guaranteed.

3. Reduction of Problem 1 to an MILP Problem

In this section, based on [13], we consider reducing Problem 1 to an MILP problem.

To model the travel cost of each agent, each undirected edge except for self-loops is represented by two directed edges (arcs). That is, the undirected edge (i, j) is represented by two arcs (i, j) and (j, i). Let $\bar{\mathcal{E}}$ denote the set of arcs enlarged by the above method. The set $\bar{\mathcal{E}}$ is denoted by $\bar{\mathcal{E}} = \{1, 2, \ldots, |\bar{\mathcal{E}}|\}$, where each element may be represented by the pair of nodes. We define binary variables as follows:

- $\delta_{i,j}(k)$: $\delta_{i,j}(k) = 1$ if the agent j is located on the node i at time k. Otherwise $\delta_{i,j}(k) = 0$.
- $\delta_i(k)$: $\delta_i(k) = 1$ if at least one agent is located on the node i at time k. Otherwise $\delta_i(k) = 0$.
- $\xi_{l,j}(k)$: $\xi_{l,j}(k) = 1$ if the agent j is located at the node a at time $k-1$, and is located at the node b at time k, where $l = (a, b)$. Otherwise $\xi_{l,j}(k) = 0$.

From the definition

$$\xi_{l,j}(k) = \delta_{a,j}(k-1)\delta_{b,j}(k) \quad (4)$$

holds, where $l \in \mathcal{I}_{\text{out}}(a)$ and $l \in \mathcal{I}_{\text{in}}(b)$. In addition, we impose the following equality constraint:

$$\sum_{l=1}^{|\bar{\mathcal{E}}|} \xi_{l,j}(k) = 1, \quad j \in \{1, 2, \ldots, m\}.$$

If this constraint is satisfied, then $\sum_{i=1}^{n} \delta_{i,j}(k) = 1$ is also satisfied. The relation between $\delta_{i,j}(k)$ and $\delta_i(k)$ is given by the following linear inequalities for each $i \in \{1, 2, \ldots, n\}$ [8]:

$$\delta_{i,j}(k) \leq \delta_i(k) \leq \sum_{j'=1}^{m} \delta_{i,j'}(k), \quad j \in \{1, 2, \ldots, m\} \quad (5)$$

We also define binary variable vectors as follows:

$$\bar{\delta}(k) := \begin{bmatrix} \delta_1(k) & \delta_2(k) & \cdots & \delta_n(k) \end{bmatrix}^\top,$$
$$\xi_j(k) := \begin{bmatrix} \xi_{1,j}(k) & \xi_{2,j}(k) & \cdots & \xi_{|\mathcal{E}|,j}(k) \end{bmatrix}^\top.$$

First, using $\delta_i(k)$, the penalty $p_i(k)$ is modeled by

$$p_i(k+1) = (1 - \delta_i(k))(p_i(k) + 1). \quad (6)$$

The lower bound of $p_i(k)$ is 0, and the upper bound of $p_i(k)$ is given by $\bar{p} < \infty$ ($\bar{p}$ can be determined based on a given graph). Then, $z_i(k) := \delta_i(k)p_i(k) - 1$ is equivalent to the following linear inequalities [2]:

$$\begin{cases} -1 \leq z_i(k) \leq \bar{p}\delta_i(k) - 1, \\ p_i(k) - \bar{p}(1 - \delta_i(k)) - 1 \leq z_i(k) \leq p_i(k) - 1. \end{cases} \quad (7)$$

Hence, (6) can be represented by

$$p_i(k+1) = p_i(k) - z_i(k) - \delta_i(k)$$

and (7).

Next, using $\xi_{l,j}(k)$, the input-output relation at the node i can be represented by

$$\sum_{l\in\mathcal{I}_{\rm out}(i)} \xi_{l,j}(k+1) = \sum_{l\in\mathcal{I}_{\rm in}(i)} \xi_{l,j}(k),\ \ i\in\{1,2,\ldots,n\}.$$

From this expression, we can obtain

$$0 \le Eu_j(k) - F\xi_j(k) \le 0,\ \ j\in\{1,2,\ldots,m\},$$

where $u_j(k) := \xi_j(k+1)$, $j\in\{1,2,\ldots,m\}$, and $E,F\in\{0,1\}^{n\times|\bar{\mathcal{E}}|}$ is derived from the graph $\mathcal{G}$ (see also [9]). Furthermore, from (4), $\delta_{i,j}(k) = \sum_{l\in\mathcal{I}_{\rm out}(i)}\xi_{l,j}(k+1)$ holds. Then, the inequalities (5) can be rewritten as

$$Eu_j(k) \le \bar{\delta}(k) \le \sum_{j=1}^{m} Eu_j(k),\ \ j\in\{1,2,\ldots,m\}.$$

Hence, $\delta_{i,j}(k)$ may not be used.

Finally, using $u_j(k)(=\xi_j(k+1))$, the time evolution of the traveling cost for the agent j can be modeled by

$$q_j(k+1) = q_j(k) + Wu_j(k),\ \ j\in\{1,2,\ldots,m\},$$

where $W := [w_1\ w_2\ \cdots\ w_{|\bar{\mathcal{E}}|}]$,

$$w_i = \begin{cases} 0 & \text{if the edge } i \text{ is a self-loop,} \\ 1 & \text{otherwise.} \end{cases}$$

From the above preparations, the move of agents, penalties (the constraint (3) is included), and travel costs are modeled by the following mixed logical dynamical system model [2]:

$$\begin{cases} x(k+1) = Ax(k) + Bv(k), \\ Cx(k) + Dv(k) \le G, \end{cases} \tag{8}$$

where

$$\begin{aligned} x(k) &= [p_1(k),\ldots,p_n(k),q_1(k),\ldots,q_m(k), \\ &\qquad \xi_1^\top(k),\ldots,\xi_m^\top(k)]^\top \in \mathcal{R}^{n+m}\times\{0,1\}^{|\bar{\mathcal{E}}|m}, \\ v(k) &= [z_1(k),z_2(k),\ldots,z_n(k),\bar{\delta}^\top(k), \\ &\qquad u_1^\top(k),\ldots,u_m^\top(k)]^\top \\ &\qquad \in \mathcal{R}^n\times\{0,1\}^{n+|\bar{\mathcal{E}}|m}. \end{aligned}$$

The matrices A, B, C, D, and G are derived as follows:

$$A = \begin{bmatrix} I_n & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I_m & 0 \end{bmatrix},\ B = \begin{bmatrix} -I_n & -I_n & 0 \\ 0 & 0 & I_{|\bar{\mathcal{E}}|m} \\ 0 & 0 & I_m\otimes W \end{bmatrix},$$

$$C = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline I_n & 0 & 0 \\ -I_n & 0 & 0 \\ \hline 0 & 0 & I_n\otimes F \\ 0 & 0 & -I_n\otimes F \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & -I_m & 0 \end{array}\right],$$

$$D = \left[\begin{array}{ccc} -I_n & 0 & 0 \\ I_n & -\bar{p}I_n & 0 \\ \hline -I_n & \bar{p}I_n & 0 \\ I_n & 0 & 0 \\ \hline 0 & 0 & -I_n\otimes E \\ 0 & 0 & I_n\otimes E \\ \hline 0 & -I_{mn} & I_m\otimes E \\ 0 & I_{mn} & -1_{m\times m}\otimes E \\ \hline 0 & 0 & 0 \end{array}\right],$$

$$G = \left[\begin{array}{c} 1_{n\times 1} \\ -1_{n\times 1} \\ \hline (\bar{p}+1)1_{n\times 1} \\ -1_{n\times 1} \\ \hline 0 \\ 0 \\ \hline 0 \\ 0 \\ \hline -c1_{m\times 1} \end{array}\right].$$

From the state equation in (8), we can obtain

$$\bar{x} = \bar{A}x(t) + \bar{B}\bar{v}, \tag{9}$$

where

$$\begin{aligned} \bar{x} &= [x^\top(t),\ldots,x^\top(t+N)]^\top, \\ \bar{v} &= [v^\top(t),\ldots,v^\top(t+N-1)]^\top, \end{aligned}$$

$$\bar{A} = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix},\ \bar{B} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A^{N-1}B & \cdots & AB & B \end{bmatrix}.$$

From the linear inequality in (8), we can obtain

$$\bar{C}\bar{x} + \bar{D}\bar{v} \le \bar{G}, \tag{10}$$

where $\bar{C} = [I_N\otimes C\ \ 0]$, $\bar{D} = I_N\otimes D$, and $\bar{G} = 1_{N\times 1}\otimes G$. Moreover, (2) can be rewritten as

$$\begin{aligned} J &= L\bar{x}, \\ L &= \left[0_{1\times(n+m+m|\bar{\mathcal{E}}|)N}\ \ 0_{1\times n}\ \ 1_{1\times m}\ \ 0_{1\times m|\bar{\mathcal{E}}|}\right]. \end{aligned} \tag{11}$$

Thus, using (9), (10), and (11), Problem 1 is equivalently rewritten as the following MILP problem:

Problem 2:

$$\begin{aligned} \text{given}\quad & x(t) \\ \text{find}\quad & \bar{v}\in(\mathcal{R}^n\times\{0,1\}^{n+|\bar{\mathcal{E}}|m})^N \\ \text{minimize}\quad & J = L\bar{B}\bar{v} + L\bar{A}x(t) \\ \text{subject to}\quad & (\bar{C}\bar{B}+\bar{D})\bar{v} \le \bar{G} - \bar{C}\bar{A}x(t). \end{aligned}$$

The MILP problem obtained can be solved by using a suitable free/commercial solver. According to the policy of model predictive control [3], [14], the optimal trajectory of agents can be generated by solving the MILP problem at each time.

Remark 1: In this paper, we consider an undirected graph as a mathematical model of surveillance areas. The above method can be extended to a weighted directed graph.

Remark 2: The computational complexity of MILP problems depends on the number of decision variables and so on (especially, the number of binary decision variables). The important point is that an MILP problem is NP-hard. Hence, the computation time for solving an MILP problem becomes extremely longer with an increase in the number of binary decision variables. One of the methods to overcome this issue is to decompose a given graph. For each decomposed graph, each agent is assigned in advance, and the surveillance problem for a single agent is solved. See [8] for further details.

4. Derivation of Feasibility Condition

In this section, based on CPP, we derive a condition on the number of agents, the surveillance performance, and the prediction horizon such that Problem 2 (the MILP problem) is always feasible. The surveillance performance is evaluated by c in the inequality (3). Here, we make the following assumption.

Assumption 1: Initial locations of all agents can be set arbitrarily. In addition, the initial location of each agent is different.

Using a solution of CPP, we can determine initial locations of agents.

CPP is the problem of finding a shortest closed walk of an undirected graph in which each edge is traversed at least once, rather than exactly once [16]. In the case of the undirected graph shown in Fig. 1, one of the solutions of CPP can be derived as the closed walk shown in Fig. 2, where self-loops are ignored. Let κ denote the number of nodes in the closed walk, which is a solution of CPP. In the closed walk shown in Fig. 2, κ is given by $\kappa = 22$.

First, we explain a motivating example.

Example 2: Consider the graph shown in Fig. 1. Assume that the initial penalties of nodes are given by $p_i(0) = 0$, $i \in \{1, 2, \ldots, 14\}$. Suppose that the number of agents is given by three ($m = 3$), and the initial locations of agents 1, 2, and 3 are given by nodes 1, 7, and 14, respectively (gray nodes in Fig. 2). Suppose also that $c = 7$ and $N = 8$. Then, one of the trajectories (closed walks) that Problem 1 is always feasible can be derived as follows:

Agent 1: $1 \to 2 \to 3 \to 4 \to \cdots \to 2 \to 1$,

Agent 2: $7 \to 10 \to 12 \to 13 \to \cdots \to 4 \to 7$,

Agent 3: $14 \to 11 \to 9 \to 8 \to \cdots \to 11 \to 14$.

In the case where the number of agents is two ($m = 2$), if $c = 10$ and $N = 11$, then there exist trajectories that Problem 2 is always feasible. Thus, initial locations, feasibility, and performance can be discussed using a solution of CPP. □

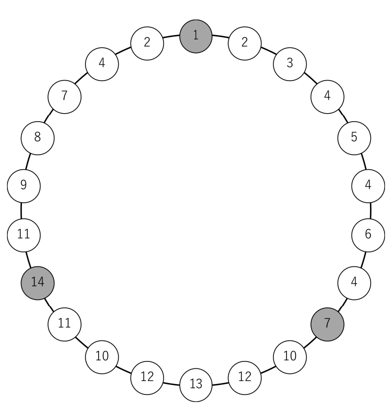

Fig. 2 Solution of CPP for the graph shown in Fig. 1.

Next, we consider a general case. We further impose the following constraint for Problem 2:

- The location of agent $i \in \{1, 2, \ldots, m\}$ at time $t + N$ is either one of the locations of agents at time t.

This constraint can be represented by

$$\bar{\delta}(t + N) = \bar{\delta}(t). \tag{12}$$

where $\bar{\delta}(t+N)$ is not a decision variable in Problem 2. Noting that $\delta_{i,j}(k) = \sum_{l \in \mathcal{I}_{\rm in}(i)} \xi_{l,j}(k)$ holds from (4), $\bar{\delta}(t + N)$ can be derived as a binary vector satisfying the following inequality:

$$Fu_j(t + N - 1) \leq \bar{\delta}(t + N) \leq \sum_{j=1}^{m} Fu_j(t + N - 1). \tag{13}$$

From (12) and (13), the additional constraint is derived as

$$Fu_j(t + N - 1) \leq \bar{\delta}(t) \leq \sum_{j=1}^{m} Fu_j(t + N - 1). \tag{14}$$

Problem 2 with (14) can also be rewritten as an MILP problem. Thus, we have the following theorem.

Theorem 1: Under Assumption 1 and $p_i(0) = 0$ (i.e., the initial penalty is given by zero), Problem 2 with the inequality constraint (14) is always feasible if the following condition holds:

$$\frac{\kappa}{m} \leq c + 1 \leq N. \tag{15}$$

Proof : First, from the definition (1) of the penalty, the update of the penalty by the location information is delayed one discrete time. Hence, the prediction horizon N should be equal to or greater than $c + 1$. That is, we can obtain $N \geq c + 1$.

Next, under $N \geq c + 1$, m agents monitor locations except for the current locations of agents, such that (3) is satisfied. From this fact, we can obtain $mc \geq \kappa - m$. We can obtain (15) from $N \geq c + 1$ and $mc \geq \kappa - m$.

Finally, under the condition (15), the feasible solution of Problem 2 with (14) can be derived from the solution of

CPP. □

Using the condition (15), we can characterize the relation between initial locations, feasibility, performance, and the number of agents. In the previous motivating example, from $\kappa = 22$ and $m = 3$, we can obtain $c \geq 19/3 = 6.3333$. Then, we set $c = 7$. Of course, we can determine the number of agents. When $c = 5$ is given, we have $m \geq 22/6 = 3.6667$. That is, at least four agents are needed. Initial locations can be determined from the closed walk obtained from CPP.

We remark that a closed walk obtained by solving CPP is not unique in general. However, the number (κ) of nodes in the optimal closed walk is unique. Hence, the fact that a closed walk obtained by solving CPP is not unique does not affect the performance (c) and the number of agents (m).

5. Numerical Example

We present a numerical example. We use the setting of Example 2 ($m = 3$, $c = 7$, and $N = 8$). Figure 3 shows trajectories of three agents. From this figure, we see that trajectories are not necessarily the same as a solution for CPP. Figure 4 shows time response of penalties of nodes 6, 8, 13, and 14. From this figure, the constraint on penalties ($p_i(k) \leq 7$) is satisfied in these nodes. We confirmed that $p_i(k) \leq 7$ holds in all nodes. In the case of $N = 7$, Problem 1 is infeasible at a certain time. Hence, in this example, the conditions (15) are tight.

Next, we compare the proposed method with our previously proposed methods [8], [10], [13]. In the proposed method, since the travel cost is minimized, agents sometimes stop (the performance is guaranteed by (3)). See Fig. 3. In [8], [10], since the penalty for each node is minimized, agents continue to move. In [13], agents may stop by introducing a fuel constraint for each agent. However, the problem formulation is more complicated than the proposed method. Thus, by the proposed method, energy saving can be easily considered.

Finally, we comment about the computation time to solve the MILP problem. The worst computation time was 0.3867 sec, and the mean computation time was 0.0881 sec. Here, we used the computer with CPU: Intel Core i9-11900K 3.50 GHz and Memory: 32 GB, and used IBM ILOG CPLEX Optimizer 12.7.1 as an MILP solver. Thus, the MILP problem in this case can be solved fast.

6. Conclusion

In this paper, we proposed a multi-agent surveillance method based on travel cost minimization. The optimal surveillance problem was reduced to an MILP problem through an MLD system model. Furthermore, we discussed the relation between initial locations, feasibility, and performance based on CPP. Furthermore, we derived a condition on the number of agents, the surveillance performance, and the prediction horizon such that the MILP problem is always feasible. The obtained condition is very simple, and is useful for determining the number of agents and the performance.

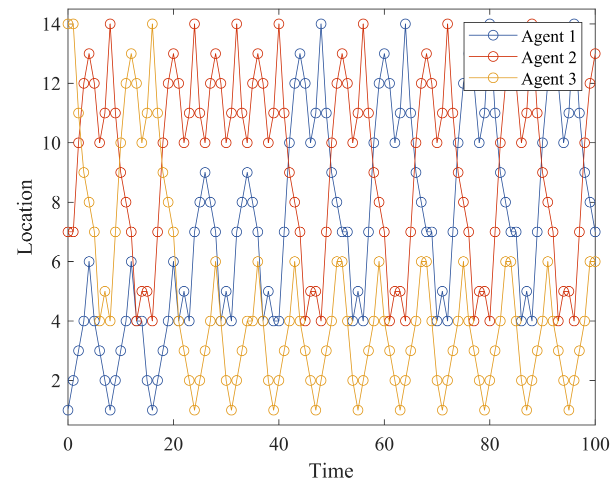

Fig. 3 Trajectories of three agents.

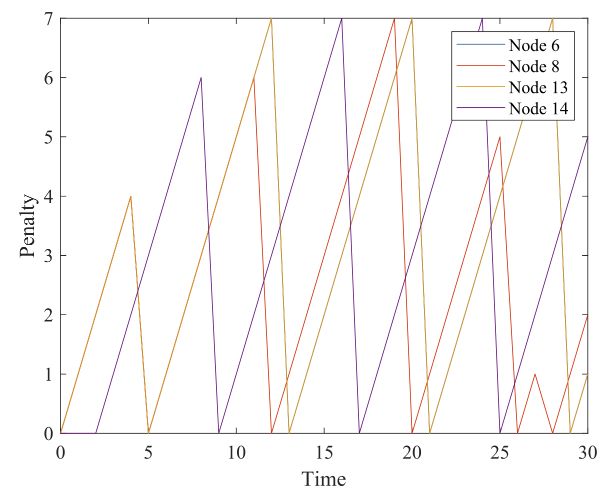

Fig. 4 Time response of penalties of nodes 6, 8, 13, and 14.

One of the future efforts is to develop a distributed optimization method for large-scale systems. It is also important to apply the proposed method to real applications. In implementations, it is significant to consider cyber security issues.

This work was partly supported by JSPS KAKENHI Grant Numbers JP21H04558, JP22K04163, JP23H01430.

References

[1] D. Aksaray, K. Leahy, and C. Belta, "Distributed multi-agent persistent surveillance under temporal logic constraints," Proc. 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems, pp.174–179, 2015.
[2] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," Automatica, vol.35, no.3, pp.407–427, 1999.
[3] E.F. Camacho and C.B. Alba, Model Predictive Control, Springer, 2008.
[4] C.G. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," IEEE Trans. Autom. Control, vol.58, no.4, pp.947–961, 2013.
[5] M.H. Cintuglu, O.A. Mohammed, K. Akkaya, and A.S. Uluagac, "A

survey on smart grid cyber-physical system testbeds," IEEE Commun. Surveys Tuts., vol.19, no.1, pp.446–464, 2016.
[6] Y. Elmaliach, N. Agmon, and G.A. Kaminka, "Multi-robot area patrol under frequency constraints," Ann. Math. Artif. Intell., vol.57, no.3–4, pp.293–320, 2009.
[7] J.G.M. Fu and M.H. Ang, Jr., "Probabilistic ants (PAnts) in multi-agent patrolling," Proc. 2009 IEEE/ASME Int'l Conf. on Advanced Intelligent Mechatronics, pp.1371–1376, 2009.
[8] M. Kido, K. Kobayashi, and Y. Yamashita, "MPC-based surveillance over graphs by multiple agents," SICE Journal of Control, Measurement, and System Integration, vol.10, no.3, pp.253–258, 2017.
[9] K. Kobayashi and J. Imura, "Deterministic finite automata representation for model predictive control of hybrid systems," Journal of Process Control, vol.22, no.9, pp.1670–1680, 2012.
[10] K. Kobayashi, M. Kido, and Y. Yamashita, "Computationally efficient model predictive control for multi-agent surveillance systems," IEICE Trans. Fundamentals, vol.E102-A, no.2, pp.372–378, Feb. 2019.
[11] K. Kajita and E. Konaka, "Hard-to-predict routing algorithm from intruders for autonomous surveillance robots," Proc. 44th Annual Conference of the IEEE Industrial Electronics Society, pp.2540–2545, 2018.
[12] R. Masuda, K. Kobayashi, and Y. Yamashita, "Optimal monitoring of multiple agents via time sequence-based modeling," SICE Journal of Control, Measurement, and System Integration, vol.11, no.3, pp.169–173, 2018.
[13] R. Masuda, K. Kobayashi, and Y. Yamashita, "Dynamic surveillance by multiple Aagents with fuel constraints," IEICE Trans. Fundamentals, vol.E103-A, no.2, pp.462–468, Feb. 2020.
[14] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert, "Constrained model predictive control: Stability and optimality," Automatica, vol.36, no.6, pp.789–814, 2000.
[15] F. Pasqualetti, A. Franchi, and F. Bullo, "On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms," IEEE Trans. Robot., vol.28, no.3, pp.592–606, 2012.
[16] O. Sokmen, S. Emec, M. Yilmaz, and G. Akkaya, "An overview of Chinese postman problem," Proc. 3rd Int'l Conf. on Advanced Engineering Technologies, 2019.
[17] Y. Zhang, M. Qiu, C.-W. Tsai, M.M. Hassan, and A. Alamri, "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data," IEEE Syst. J., vol.11, no.1, pp.88–95, 2015.

Kyohei Murakata received the B.E. degree in 2021 and the M.I.S degree in 2023 from Hokkaido University. His research interests include cyber-physical systems.

Koichi Kobayashi received the B.E. degree in 1998 and the M.E. degree in 2000 from Hosei University, and the D.E. degree in 2007 from Tokyo Institute of Technology. From 2000 to 2004, he worked at Nippon Steel Corporation. From 2007 to 2015, he was an Assistant Professor at Japan Advanced Institute of Science and Technology. From 2015 to 2022, he was an Associate Professor at Hokkaido University. Since 2023, he has been a Professor at the Faculty of Information Science and Technology, Hokkaido University. His research interests include discrete event and hybrid systems. He is a member of IEEE, IEEJ, IEICE, ISCIE, and SICE.

Yuh Yamashita received his B.E., M.E., and Ph.D. degrees from Hokkaido University, Japan, in 1984, 1986, and 1993, respectively. In 1988, he joined the faculty of Hokkaido University. From 1996 to 2004, he was an Associate Professor at the Nara Institute of Science and Technology, Japan. Since 2004, he has been a Professor of the Graduate School of Information Science and Technology, at Hokkaido University. His research interests include nonlinear control and nonlinear dynamical systems. He is a member of SICE, ISCIE, IEICE, RSJ, and IEEE.