Effective Area Partitioning in a Multi-Agent Patrolling Domain for Better Efficiency

Katsuya Hattori and Toshiharu Sugawara[a]
Department of Computer Science and Communications Engineering, Waseda University, Shinjuku, Tokyo 1698555, Japan

Keywords: Multi-Agent System, Patrolling Problem, Division of Labor, Negotiation, Cooperative Agent.

Abstract: This study proposes a cooperative method for a multi-agent continuous cooperative patrolling problem by partitioning the environment into a number of subareas so that the workload is balanced among multiple agents by allocating subareas to individual agents. Owing to the advancement in robotics and information technology over the years, robots are being utilized in many applications. As environments are usually vast and complicated, a single robot (agent) cannot supervise the entire work. Thus, cooperative work by multiple agents, even though complicated, is indispensable. This study focuses on cooperation in a bottom-up manner by fairly partitioning the environment into subareas, and employing each agent to work on them as its responsibility. However, as the agents do not monitor the entire environment, the decentralized control may generate unreasonable shapes of subareas; the area are often unnecessarily divided into fragmented enclaves, resulting in inefficiency. Our proposed method reduced the number of small and isolated enclaves by negotiation. Our experimental results indicated that our method eliminated the minute/unnecessary fragmented enclaves and improved performance when compared with the results obtained by conventional methods.

1 INTRODUCTION

In recent years, robotic applications have attracted attention in many fields due to the development of advanced hardware, such as high-functional sensors and actuators related to robot technology and information technology. Robots are particularly required to play an active role in fields that entail repetitive tasks or operations in inaccessible areas. However, if the workspace is vast and/or complicated and requires various abilities, it is not realistic to work only with a single robot owing to physical and performance limits, such as battery capacity, movement speed, and limited work capability. The advancement in mobile wireless communication technology enables efficient real-time communication among robots and coordination and cooperation among multiple robots.

To control the collaborative activities of multiple *agents*, which are the abstraction of robots in a general framework, we consider the *multi-agent continuous cooperative patrolling problem*. Possible applications of this problem are area cleaning and security/surveillance patrolling by multiple agents. In this problem, agents are required to divide the given task so that the burden on each agent is as fair as possible; this also improves overall efficiency and results in uniform quality of task outputs. To achieve fair and effective division of labor, we consider a method of explicitly partitioning a working *responsible area* (RA) to a number of smaller areas and assigning each agent to the partitioned area through communication/negotiation between agents. One of the difficulties in fair partition of areas is that simple partition into equal sizes may not be appropriate because, for example, (1) some partitioned area/room is distant from the charging/storage locations of agents, (2) some rooms are more important than others so agents have to visit them more frequently than others, (3) some areas contain obstacles/slopes that makes patrolling inefficient, and (4) the shape of a partitioned area is complicated so it takes longer time to cover the area. Nevertheless, agents have to consider these factors to fairly partition the working area in a decentralized manner.

Several studies have attempted to achieve collaborative work by dividing work areas (Kato and Sugawara, 2013; Ahmadi and Stone, 2006; Elor and Bruckstein, 2009; Nasir et al., 2016). One disadvantage of such distributed methods is that there is no agent that manages the entire area; hence, the shapes

[a] https://orcid.org/0000-0002-9271-4507

Hattori, K. and Sugawara, T.
Effective Area Partitioning in a Multi-Agent Patrolling Domain for Better Efficiency.
DOI: 10.5220/0010241102810288
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 1*, pages 281-288
ISBN: 978-989-758-484-8

of the divided areas are left unattended. For example, agents proposed in Kato and Sugawara (Kato and Sugawara, 2013) may generate split (and often chopped) RAs due to the local decision. This kind of inefficient division leads to a decrease in overall performance. Conversely, agents often have to generate disconnected RAs. For example, if rooms have narrow entrance doors or are led by narrow passages that must be allocated to only one agent (an actual example is shown later), the splitting of RAs is mandatory. Therefore, in such an environment, if we introduce a constraint that each RA must be connected, the divided working areas cannot be balanced.

Therefore, we propose a method for partitioning the entire area into a number of RAs for individual agents to obtain a balanced workload without *unnecessarily* splitting the areas. We experimentally demonstrated that our proposed method exhibited better performance than that by the conventional method (Kato and Sugawara, 2013). We also investigated the performance when the environment had many obstacles and when the environment characteristics are not uniform and show that in all such cases, our method outperformed the conventional method.

2 RELATED WORK

There are a number of studies on multi-agent continuous patrolling problems (Huang et al., 2019). We can classify them roughly into two based on the methods for solving them. In the first type of method, the agents share the entire area to move around without dividing the area into smaller subareas. Carrillo and Rapp (Carrillo and Rapp, 2020) proposed a method to identify patrolling policies for multiple agents with limited visibility regions and non-deterministic patrolling paths. Yoneda et al. (Yoneda et al., 2013) propose a method in which agents individually determine their exploration algorithms using reinforcement learning to contribute more toward the shared goal. They also assumed that agents' intermittent behaviors depend on the battery charge and agents decide to explore the environment further or return to their charging base depending on their battery capacity. Our study also assumes the charging base, and that agents have to return to the base before run-out. Sugiyama et al. (Sugiyama et al., 2019), who also used the model of cyclic charging activities, consider the cycle of agent patrolling while shifting the time phase to visit each location. However, because environments are not partitioned, there is a possibility that more than two agents patrol the same area, which may be redundant and hence, unnecessary actions increase.

In the second type of method for solving continuous patrolling problems, the environment is fairly divided into a number of subareas and each of them is allocated to an agent to move around in a balanced manner. In the method proposed by Nasir et al. (Nasir et al., 2016), the leader agent divides the environment and determines and allocates the exploring area to individual member agents. However, this requires centralized control by the leader agent, and hence, its failure affects the entire system. Ahmadi and Stone (Ahmadi and Stone, 2006) introduced area division based on boundary relationships between agents. If there were overlapping locations, they were transferred to the agents which frequently visited those areas. Elor and Bruckstein (Elor and Bruckstein, 2009) proposed a model based on balloon expansion so that individual agents can fairly divide the environment into subareas of the same size in a bottom-up manner. However, given the obstacles and non-uniform structures in the environment, divisions of the same size are not always fair from the viewpoint of agents' workload. Moreover, these methods did not consider the constraints due to battery capacity.

By contrast, Kato et al. (Kato and Sugawara, 2013) introduced the constraint on the battery capacity and proposed the partitioning method for fair workload among agents. Their method for area partitioning is based on the expansion power, like the current study, that reflects the degree to which the agent has completed the work in its RA. However, because their method generated many fragments of RAs (Kato and Sugawara, 2013), the performance often decreased or could not be applied in a complicated environment. In this study, agents divide their RAs according to the shape of the environment so that agents with the method can be applied to even complex environments. We also reduce the unnecessary fragments of the area of responsibility to improve the efficiency of event collection/observation in multi-agent patrolling problem.

3 BACKGROUND AND PROBLEM

3.1 Environment

We introduce discrete time $t \geq 0$, whose unit is a *step*. The *multi-agent continuous cooperative patrolling problem* can be expressed by $(G, A, \mathcal{P})$, where $A = \{1, \cdots, N\}$ is a set of agents, $G = (V, E)$ is a connected graph embeddable into two-dimensional Euclidean space, $V = \{v_1, \cdots, v_x\}$ is the set of nodes, E is the set of edges e_{v_i,v_j} connecting two nodes

$v_i, v_j \in V$, and $\mathcal{P} = \{p_v\}_{v \in V}$ $(0 \le p_v \le 1)$ is the *distribution of event probability*, i.e., a collection of probabilities of event occurrence at $v \in V$. An event at node v is required to be observed or monitored by one of the agents, and thus, agents are necessary to visit these nodes in proportion to the values of these probabilities. For example, nodes where high security levels are required has higher event probabilities in a security patrolling domain and nodes with higher probabilities are more likely to get dirty in a vacuum cleaning domain. Therefore, we assume that $\mathcal{P}$ is given to all agents.

We assume that events are accumulated if no agents have monitored/observed them. Therefore, for $\forall v \in V$, the amount of accumulated events $L_t(v)$, is incremented by one with p_v at t, that is,

$$L_{t+1}(v) = \begin{cases} L_t(v) + 1 & \text{with probability } p_v \\ L_t(v) & \text{otherwise} \end{cases} \quad (1)$$

If the agent visits node v at time t, it has observed or monitored v, and so the amount of accumulated event at v is cleared $(L_{t+1}(v) = 0)$. However, agents cannot know the actual values of $L_t(v)$; hence, they estimate $L_t(v)$ using the expected value $E(L_t(v))$ using $p_v \in \mathcal{P}$.

We can assume that the length of an edge is 1 by adding dummy nodes whose event probabilities are zero. Therefore, agents at any node can move to its adjacent node in one step. Agent i has a charging base v^i_{base} $(\in V)$ that is specified initially and that it has to return to before run-out. Agent i has its own RA V^i_t at time t, which is a subset of V including v^i_{base}, and where it must move around to maintain. For a fair workload, i has to adjust its own RA by negotiation with its collaborative agents.

3.2 Model of Agent

We introduced three assumptions in the model of agents. First, agents have the map where they patrol. We believe that in many applications, the map of the environment is often known; hence, this assumption is plausible. Furthermore, many map creation algorithms have been proposed so far. As we focus on the fair workload by appropriate area partitioning, we assume that maps can be made by using these algorithms if necessary. Second, multiple agents can exist in the same node and collision is not possible. In reality, a collision may occur when multiple agents try to move to the same node if it is narrow. However again, because we focus on fair workload, we assume that collision avoidance algorithms will be used for actual applications. Finally, each agent has a battery with a finite capacity. Therefore, it cannot continue patrolling forever, and alternately repeats the exploring state and the charging state.

Agents have two states: the *active state* and *charging state*. Agents in the active state move around the environment (or their own RA) to find unobserved events in accordance with their exploring strategies, whereas agents in the charging state charge at their charging base. The states of the battery in agent i is specified by parameters B^i_{max} B^i_{drain} B^i_t, where B^i_{max} is the maximal charge capacity, B^i_{drain} is the battery consumption per step when i is moving, and B^i_t is the remaining battery capacity. Therefore, the charging time t^i_{ch} to make the battery full is proportional to the consumed energy; hence, it is

$$t^i_{ch} = k^i_{ch}(B^i_{max} - B^i_t), \quad (2)$$

$k^i_{ch} > 0$ is the *charging constant*. For simplicity, agents start from their base after full charge.

For $\forall v \in V$, we define the potential $Pot^i(v)$ of v as the required amount of energy to reach the charging base. Let $Len(v,u)$ be the shortest distance from node v to node u, i.e., the number of edges in the shortest path between them. It can then be defined by

$$Pot^i(v) = Len(v, v^i_{base}) \times B^i_{drain} \quad (3)$$

When i attempts to move from node v_1 to node v_2, it compares B^i_t and $Pot^i(v_2)$ and if

$$B^i_t < Pot^i(v_2) + Len(v_1, v_2) \times B^i_{drain} \quad (4)$$

is satisfied, i gives up on v_2 and returns to its charging base v^i_{base} to prevent battery run-out.

The purpose of agents is to visit nodes in their RA V^i_t as frequently as possible by considering the event probability of nodes using a certain exploring algorithm. In this paper, because we focus on the method of area partitioning, we used a simple algorithm for exploring, the directed depth-first search (DDFE) (Kato and Sugawara, 2013), which is briefly described as follows.

When agent i leaves the charging base at t, it sets the node v_0 which has the largest expected value $E(L_t(v))$ in V^i_t and moves to v_0 along the shortest route. When i reaches v_0, v_0 is pushed into the internal stack and its unvisited adjacent nodes in V^i_t are added to the open list. Then, i selects one of them randomly and moves there. Agent i pushes the current node into the stack and the adjacent nodes that are unvisited and are not in the open list are added to the open list. This is repeated as long as i can select unvisited adjacent nodes. When i cannot select an unvisited node, it pops the top node from the stack and moves back to that node, and then i selects an unvisited node from the open list. After repeating this operation if i returns to v_0 and cannot select another unvisited node, it returns to the charging base along the shortest path. Note that when it moves along the shortest path, all nodes in this path may not be in V_t^i; therefore, i may pass through the areas that are the responsibility of other agents.

3.3 Deciding RA

We will briefly explain how agents coordinate with each other to decide their RAs in the environment using the conventional method. Please see (Kato and Sugawara, 2013) for details. Initially, agent $i \in I$ has a small area within the distance d_{init} from its charging base v_{base}^i as an initial RA $V_0^i = \{v \in V | Len(v, v_{base}^i) \leq d_{init}\}$. We assume that $v_{base}^i \in V_t^i$ for $\forall t$.

First, agent i calculates the sum of all expected values

$$E(L(V_t^i)) = \sum_{v \in V_t^i} E(L(v)).$$

This value also indirectly shows how completely an agent has patrolled its area of responsibility. The reciprocal of this value, $\varepsilon(i,t)$, is called the *expansion power* of i at time t. Agents calculate the expansion power whenever they return to the charging base, and retain the values until the next change in calculation.

When agents expand their RAs, they attempt to include some boundary nodes $\{u_1, \dots, u_K\}$ of V_t^i (where $u_i \notin V_t^i$) and nodes that are not too far from the base v_{base}^i. Then, if u_i is not included, in other RAs, u_i is added to V_t^i. If $u_i \in V_t^j$ for $j \in I$ ($i \neq j$), they begin to negotiate and the agent whose expansion power is larger includes the nodes into their RA.

3.4 Isolated Enclaves

One significant drawback of the conventional method is that agents will have their RAs unnecessarily split by others' expansion behaviors; there will be many isolated enclaves. We define an *enclave* as follows. For the current RA V_t^i, let us consider $G_t^i = (V_t^i, E_t^i)$ where $E_t^i = \{e_{uv} \in E | u, v \in V_t^i\}$. Then, for $v \in V_t^i$, the connected nodes $V_{conn}^i(v)$ of i is the set of nodes in V_t^i reachable from v only along the edges in E_t^i. Similarly, we can define the *connected component* $G_{conn}^i(v) = (V_{conn}^i(v), E_{conn}^i(v))$, where $E_{conn}^i(v) = \{e_{uv} \in E | u, v \in V_{conn}^i(v)\}$. Let $\mathcal{G}_{conn}^i$ be the set of all connected components of i. Then, $G_{conn}^i(v)$ is an *enclave* if it does not include i's base ($v_{base}^i \notin V_{conn}^i(v)$). Obviously, connected components are exclusive and their union is equal to V_t^i. As mentioned before, some enclaves are indispensable to cover the entire environment, but unnecessary enclaves, especially small and scattered enclaves, will significantly reduce the system's performance.

3.5 System Evaluation Criteria

The purpose of this research is to allow agents to patrol the environment and more frequently visit important nodes by appropriately partitioning the environment into RAs, so that their workload is fair and balanced. Because our target applications are, for example, patrolling for security surveillance, large area cleaning, and environmental/sensor data collection, agents should visit the nodes in their RA without leaving any nodes unattended for a long time. For this purpose, we evaluate the systems performance using the average value $D_{t_s,t_e}(V)$ of the number of remaining events of all nodes in period $[t_s, t_e]$, which is defined by

$$D_{t_s,t_e}(V) = \sum_{t=t_s}^{t_e} L_t(V)/(t_e - t_s + 1). \quad (5)$$

Therefore, the smaller the value of $D_{t_s,t_e}(V)$, the better is the performance of the method.

The target of the proposed method is efficient patrolling by reducing the number of enclaves, and so, we also investigate the number of enclaves of their RAs. Note that t_s and t_e in $D_{t_s,t_e}(V)$ is often omitted (so $D(V)$).

4 PROPOSED METHOD

We propose a method for multiple autonomous agents to individually decide their own RA by partitioning the environment without small isolated enclaves. This method consists of the *determination of expansion nodes*, which is almost identical to the conventional method (Kato and Sugawara, 2013), and two negotiation phases for the *arrangement of overlapped area assignments* to maintain a balanced workload and *delegate isolated enclaves* to reduce small and isolated areas.

4.1 Determination of Expansion Nodes

Agent $i \in A$ attempts to gradually expand the RA by including a number of nodes that are not in but at the boundaries of V_t^i, when it has almost finished the work in the RA and still has enough battery capacity to move more. First, when i leaves the charging base v_{base}^i at the time t_b, it calculates the estimated value $E(L_{t_b+\gamma}(V_{t_b}^i))$ of the remaining amount of the event after γ (> 0) steps from t_b. The reason for calculating the estimated number of unobserved events at time $t_b + \gamma$ instead of at t_b is that (1) i tries to expand the RA at a certain time after leaving the base and (2) it wants to compare the state of unobserved events if it does not move after it works actually.

After agent i starts from v^i_{base} at t_b, it records $N_{vis}(t)$, which is the number of nodes visited by t ($> t_b$), and $N_d(t)$, which is the sum of the observed events. Then, when conditions (6) and (7) are satisfied, the agent tries to expand the RA.

$$N_{vis}(t) \geq R_1 \cdot |V^i_t| \qquad (6)$$

$$N_d(t) \geq R_2 \cdot E(L_{t_b+\gamma}(G^i_{t_b})), \qquad (7)$$

where R_1 and R_2 are the parameters to adjust expansion activity ($0 \leq R_1, R_2 \leq 1$). To avoid excessive area expansion, agents attempt to expand the RA only once after leaving their bases.

When agent i tries to expand its own area, it determines the set of nodes to be included I^i.

(1) Agent i selects a set of nodes B that is adjacent to its own RA.

(2) i selects k_{inc} (> 0) nodes from B that are not included in I^i_{exp} (explained later) and have the shortest distance from its charging base v^i_{base}, and sets them as I^i_{inc}, where k_{inc} is an integer.

(3) Nodes in I^i_{inc} and nodes adjacent to an element in $I^i_{inc} \setminus I_{exp}$ are defined as the set I^i.

Thus, if $I^i = \varnothing$, i does not expand the area.

4.2 Arrangement of Overlapped Areas

After determining the expansion nodes, agents decide which agent should work for individual nodes in I^i. For every $v \in I^i$, when $v \in V^j_t$, i requests $\varepsilon(j,t)$, from j and i includes v to V^i_{t+1} only when $\varepsilon(j,t) < \varepsilon(i,t)$. If $\varepsilon(j,t) \geq \varepsilon(i,t)$, then v is added to I^i_{exp}.

I^i_{exp} is the set of nodes that should not be included in I^i for a while to prevent frequent challenges and failures. Hence, once a node is included in I^i_{exp}, it will be excluded from I^i_{exp} after the area expansion is performed k_{avoid} (> 0) times. Furthermore, in Step (2) of the expansion node determination, i adds all nodes that are not included in other's RA and are omitted from counting k_{avoid} nodes.

4.3 Reducing Isolated Small Enclaves

To reduce the occurrence of unnecessary enclaves, agent i negotiates with other agents to delegate isolated enclaves to more appropriate agents. To decide an enclave that should be delegated to other agents, i will select small enclaves distant from the base.

For $\forall G^i_{conn,k} \in \mathcal{G}^i_{conn}$, we calculate the mean distance from the base v^i_{base} by

$$dis(G^i_{conn,k}) = \frac{1}{|V^i_{conn,k}|} \sum_{v \in G^i_{conn,k}} Len(v, v^i_{base}) \qquad (8)$$

Then, the enclaves, which are the connected components that do not contain v^i_{base}, and satisfy the following condition become the candidates to be delegated.

$$dis(G^i_{conn,k}) > \frac{|V^i_{conn,k}|}{R_3 \cdot |V^i_t|}, \qquad (9)$$

where R_3 is a small positive number to decide the balance between the size and distance of the connected components. An enclave $G^i_{conn,k}$ will be delegated to agent j that has the largest number of nodes adjacent $V^i_{conn,k}$ to its RA. This calculation can be done during i charges.

Then, agent i communicates with agent j so that i delegates $G^i_{conn,k}$ to j, without considering their expansion powers. Therefore, j may temporally have a large RA. However, its expansion power decreases so that j's RA is gradually diminished by other agents; eventually, the overall load will become fair. Note that all enclaves are not deleted; only small enclaves far from v^i_{base} will be delegated. Finally, when agent i moves from enclave $G^i_{conn,k}$ to another $G^i_{conn,k'}$, it takes the shortest path between two enclaves. Therefore, i finds the pair $(v, v') \in G^i_{conn,k} \times G^i_{conn,k'}$ s.t. $Len(v, v')$ is the smallest (if there are more than two pairs, one of the pairs is selected randomly), and i follows the shortest path between v and v'. Note that the pair of length $Len(v, v')$ is defined as the distance between two areas and denoted by $dist(G^i_{conn,k}, G^i_{conn,k'})$, We can clearly define the distance between any subareas in the same way.

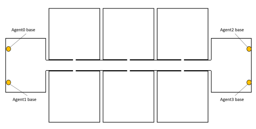
5 EXPERIMENTAL RESULTS

We conducted three experiments using three different environments (Fig. 1) to verify that our proposed method does not generate unnecessary fragments of RAs in various environments. We set $A = \{1,2,3,4\}$ (agent i is indicated as Agent+i in all figures below). Other initial parameters in this experiment are listed in Table 1. All data shown below are averages for every 3600 steps (because the maximum cycle of operation and charging is 3600 steps) taken from 100 independent experimental runs.

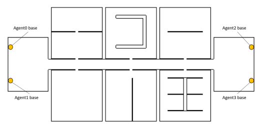
To compare the performance with those of the conventional method (Kato and Sugawara, 2013), we adopted the DDFE method to explore. Because the

Table 1: Agent's Parameters.

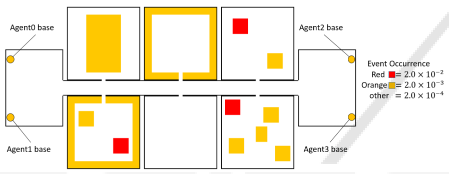
Description	Parameter	Value
Parameters for area expansion	R_1	0.4
	R_2	0.4
	γ	300
	k_{avoid}	90
Ratio for enclave negotiation	R_3	0.005
Battery consumption per step	B_{drain}	1
Battery capacity	B_{max}	900
Charging constant	$k_{ch} = k^i_{ch}$	3
Radius of initial RA	d_{init}	2



(a) Uniform environment



(b) Environment with obstacles


Agent0 base
Agent2 base
Agent1 base
Agent3 base
Event Occurrence
Red 🟥= 2.0×10^{-2}
Orange 🟧= 2.0×10^{-3}
other = 2.0×10^{-4}

(c) Biased environment

Figure 1: Three Environments.

DDFE assumes that the RA is connected (its performance considerably decreases if not), we connected the fragmented RAs with paths between the nearest area; then, the agent moves in its RA with the connected paths. Let us set $\mathcal{G}_{temp} = \mathcal{G}^i_{conn}$. Agent i selects two connected components G^i_1 and G^i_2 from $\mathcal{G}^i_{temp}$, whose distance is the smallest and connects them with one of the shortest path. Then, G^i_1 and G^i_2 are removed from $\mathcal{G}_{temp}$ and instead, union of $G^i_1 \cup G^i_2 \cup \{n \in V | n \text{ is on the connected path}\}$ is added to $\mathcal{G}_{temp}$. Agent i repeats this process until $\mathcal{G}_{temp}$ becomes a singleton. The generated area consisting of $\mathcal{G}_{temp}$ is used as the connected RA of i.

5.1 Uniform Environment

In the first experiment, we evaluate the proposed method by comparing its performance with that by the conventional method in the uniform environment where the event probability is identical in all nodes and defined as $p_v = 2 \times 10^{-4}$ for $\forall v \in V$. The structure of the environment is shown in Fig. 1a, in which there is only one connection node between the corridor and each of the six rooms. Therefore, assuming that the

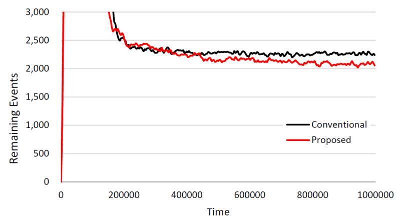


Figure 2: Remaining events $D(V)$ in uniform environment.

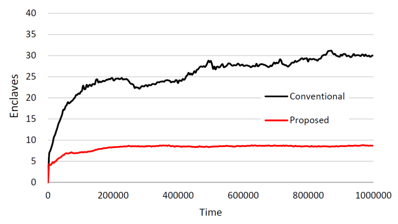
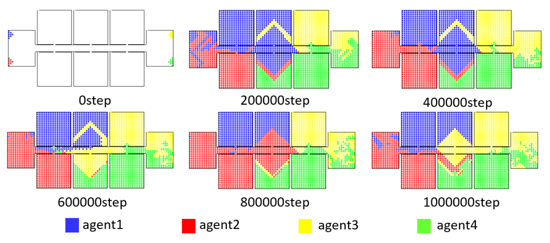


Figure 3: Number of enclaves (uniform environment).

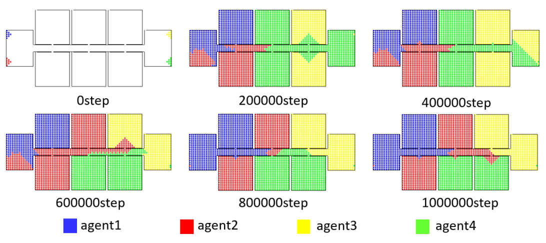
RA must be connected, each room must be patrolled by one agent, whereas the number of agents is four, making fair assignment of responsibilities impossible. Conversely, if we exclude this assumption, many enclaves are generated because no agent is looking at the whole state of the RAs, resulting in undesirable division in the conventional method. The locations of the battery charging bases for individual agents are in one of the wide spaces at the ends of the corridor as shown in Fig. 1a.

Figure 2 plots the number of remaining events in the environment every 3,600 steps over time. Note that the smaller the number of events, the better is the method. This figure indicates that the proposed method exhibited slightly better performance than the conventional method. In earlier steps, the performance of the proposed method seems lower, but this is caused by reallocating the enclaves to other agents, so agents' workloads are temporally not balanced. We counted the number of enclaves (including the connected component containing v^i_{base}) of the RA of each agent; this is shown in Fig. 3. This figure indicates that the number of enclaves of the conventional method gradually increased over time because in the area expansion, agents tried to include nodes that is decided only by the local viewpoint; thus, this expansion process may split the RA of other agents. Conversely, agents with the proposed method allocated the fragmented enclaved areas to other agents, even if such areas were generated, and thus, they can suppress the increase of the number of RAs. It should be noted that because $|A| = 4$, at least four connected components are necessary. Furthermore, in the experimental environment, the disconnected enclaves are mandatory to cover the entire environment.

Finally, we investigated how environment is di-

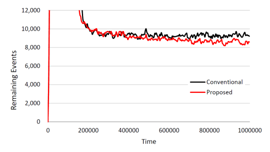


(a) Conventional method

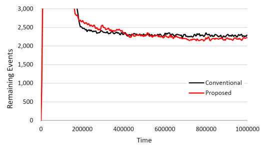


(b) Proposed method

Figure 4: Distribution map of RAs (uniform environment).

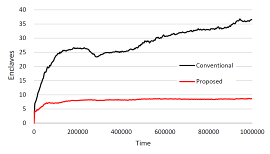


(a) Biased event probabilities

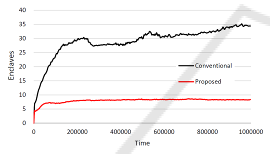


(b) With obstacles

Figure 5: Number of remaining events ($D(V)$).



(a) Biased event probabilities



(b) With obstacles

Figure 6: Number of enclaves.

vided into the RAs and how they are allocated to agents over time. The distribution maps of the enclaves of the RAs of all agents are shown in Fig. 4. Figure 4a indicates that RAs were fragmented in the whole environment, especially spaces near the battery bases, throughout the experiment. This was probably because the competition of RAs occurred frequently there, even if the size of the allocated areas were almost identical. By contrast, Fig. 4b shows the RAs are not unnecessarily divided, and thus, no fragmented areas seemed to exist. Note that the RAs were always varying. Of course, we can suspend the behavior of area expansion at a sufficient point if the environment is static. However, if the environment is dynamic and unexpected events, such as failure of an agent, introduction of new agents and deployment of obstacles, occur, agents should continue to perform the area expansion behavior to adapt to the changes.

5.2 Non-uniform Environment

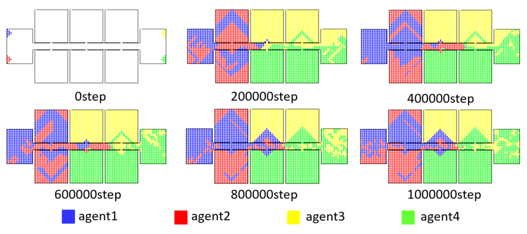
We conducted the same experiments in two different types of environments. First, is the biased environment (Fig. 1c), where there are a number of specific regions where more events are likely to occur or which are more important regions that agents must visit more frequently. The second environment (Fig. 1b) has a number of walls and obstacles in all rooms. The purpose of these experiments are to check if the proposed method can avoid the unnecessary, small, and disconnected RAs even in more complicated environments. Note that the event probabilities of white orange and red nodes are $p_v = 2.0 \times 10^{-4}$, $p_v = 2.0 \times 10^{-3}$ and $p_v = 2.0 \times 10^{-2}$ in Fig. 1.

Their performances, i.e., the values of $D(V)$, are plotted in Fig. 5. This figure shows that the proposed method outperformed the conventional method in both environments by suppressing unnecessary fragmented enclaves; we can also see this fact in both cases from Fig. 6. Actually, the number of enclaves was stable in the proposed method whereas it gradually increased over time in the conventional methods. Figure 5 also indicates that the convergence speed of agents with the proposed method was slightly slower; this is also the result of the negotiation to allocating enclaves to more appropriate agents because, in earlier stages, agents generated more enclaves aggressively by their area expansion processes.

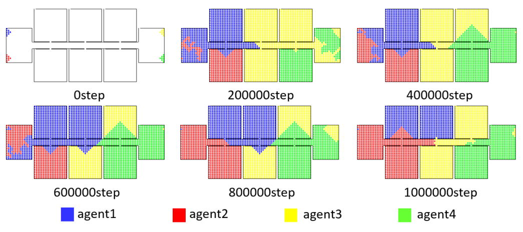
Finally, we generated maps to see how all RAs changed over time in these experiments; these maps are shown in Figs. 7 and 8. They show that the proposed method considerably reduces the number of enclaves by eliminating unnecessary fragments even if there are no global observers that monitor the entire environment. We have to note that because the proposed method suppressed the number of unnecessary enclaves, the degree of balance in the sizes of RAs of four agents slightly decreased; however, this is quite small and the resulting efficiency was improved, so we believe that it can be ignorable.

6 CONCLUSION

In this paper, we discuss a method to cover a large environment using multiple agents by partitioning it in a bottom-up manner to achieve fair and efficient cooperative executions of the continuous multi-agent patrolling problem. Although there are some studies to

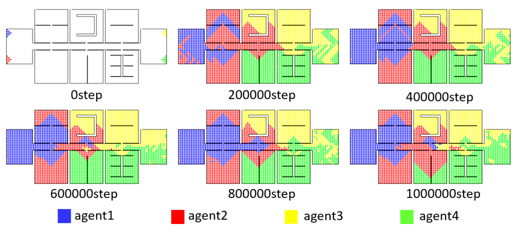


(a) Conventional method

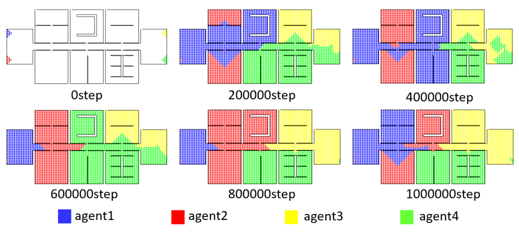


(b) Proposed method

Figure 7: Distribution map of RAs (biased environment).



(a) Conventional method



(b) Proposed method

Figure 8: Distribution map of RAs (biased environment).

achieve the balanced collaboration by area division, they often generated fragmented RAs due to the decentralized control wherein no one agent can see the entire situation, thus, resulting in inefficient cooperative work. We proposed a method in which agents do not generate unnecessary enclaves of RAs by allocating fragmented parts of the RA to more appropriate agents through communication. The results indicated that our proposed method could reduce the unnecessary enclaves of the RAs and thus, could achieve efficient cooperative work in the various environments.

We would like to examine more complicated environments to apply our method to more realistic domains. We also plan to further improve the shape of RAs, especially eliminating enclaves or fill recesses in concave areas, for more efficiency.

ACKNOWLEDGEMENT

This paper is partly supported by JSPS KAKENHI grant number 17KT0044.

REFERENCES

Ahmadi, M. and Stone, P. (2005). Continuous area sweeping: A task definition and initial approach. In *Proc. of 12th Int. Conf. on Advanced Robotics (ICAR 2005), IEEE*, pages 316–323.

Ahmadi, M. and Stone, P. (2006). A multi-robot system for continuous area sweeping tasks. In *Proc. of 2006 IEEE Int. Conf. on Robotics and Automation (ICRA 2006)*, pages 1724–1729.

Carrillo, P. and Rapp, B. (2020). Stochastic multi-robot patrolling with limited visibility. *Journal of Intelligent & Robotic Systems*, 97(2):411–429.

Elor, Y. and Bruckstein, A. (2009). Multi-a(ge)nt graph patrolling and partitioning. In *Proc. of the 2009 IEEE/WIC/ACM Int. Joint Conf. on Web Intelligence and Intelligent Agent Technology, Vol. 2, IEEE Computer Society*, pages 52–57.

Huang, L., Zhou, M., Hao, K., and Hou, E. (2019). A survey of multi-robot regular and adversarial patrolling. *IEEE/CAA Journal of Automatica Sinica*, 6(4):894–903.

Kato, C. and Sugawara, T. (2013). Decentralized area partitioning for a cooperative cleaning task. In *Proc. of the 16th Int. Conf. on Principles and Practice of Multi-Agent Systems (PRIMA-2013)*, pages 470–477.

Nasir, A., Salam, Y., and Saleem, Y. (2016). Multi-level decision making in hierarchical multi-agent robotic search teams. *The Journal of Engineering*, 2016(11):378–385.

Sugiyama, A., Wu, L., and Sugawara, T. (2019). Improvement of multi-agent continuous cooperative patrolling with learning of activity length. *Agents and Artificial Intelligence*, pages 270–292, Cham. Springer Int. Publishing.

Yoneda, K., Kato, C., and Sugawara, T. (2013). Autonomous learning of target decision strategies without communications for continuous coordinated cleaning tasks. In *IEEE/WIC/ACM Int. Confs. on Web Intelligence and Intelligent Agent Technology*, pages 216–223.