



Emergence of divisional cooperation with negotiation and re-learning and evaluation of flexibility in continuous cooperative patrol problem

Ayumi Sugiyama¹ · Vourchteang Sea¹ · Toshiharu Sugawara¹

Received: 23 January 2018 / Revised: 21 October 2018 / Accepted: 24 November 2018 /

Published online: 11 December 2018

© Springer-Verlag London Ltd., part of Springer Nature 2018

Abstract

We propose a learning and negotiation method to enhance divisional cooperation and demonstrate its flexibility for adapting to environmental changes in the context of the multi-agent cooperative problem. We now have access to a vast array of information, and everything has become more closely connected. However, this makes tasks/problems in these environments complicated. In particular, we often require fast decision-making and flexible responses to follow environmental changes. For these requirements, multi-agent systems have been attracting interest, but the manner in which multiple agents cooperate is a challenging issue because of the computational cost, environmental complexity, and sophisticated interaction between agents. In this work, we address the *continuous cooperative patrol problem*, which requires cooperation based on high autonomy, and propose an autonomous learning method with simple negotiation to enhance divisional cooperation for efficient work. We also investigate how this method can have high flexibility to adapt to change. We experimentally show that agents with our method generate several types of role sharing in a bottom-up manner for effective and flexible divisional cooperation. The results also show that agents using our method appropriately change their roles in different environmental change scenarios and enhance the overall efficiency and flexibility.

Keywords Divisional cooperation · Multi-agent system · Continuous patrolling · Autonomous learning

1 Introduction

Autonomous decision-making and collaboration with multiple agents have been required for various applications in computer science. Ongoing developments in information and communication technology now enable us to easily obtain almost any information we desire, and everything is now more closely connected due to innovations such as the Internet of

✉ Ayumi Sugiyama
sugi.ayumi@ruri.waseda.jp

¹ Department Computer Science and Communications Engineering, Waseda University, Tokyo 1698555, Japan

things. On the other hand, these developments have dramatically increased the amount of information to be processed and cause frequent changes of environments. It is difficult to follow these changes with only centralized control systems in a top-down manner because the systems and environments are constantly growing without any centralized manager and the problems occurring there are becoming complicated. Thus, distributed systems in which multiple agents work through mutual coordination and/or cooperation to cover the entire environment and follow the changes occurring there by making autonomous decisions are more suitable in these environments.

Portugal and Rocha [13] stated that the patrol problem with multiple agents is a good case study for multi-agent systems in general. They first summarized the recent methods for cooperative patrolling, and from the discussion about recent progress, they stated that flexibility, resource utilization, interference, and a balanced workload would be required the future patrol problems. They also insisted that the cooperative patrol problem has the common characteristics required in multi-agent systems such as autonomy, distribution, communication constraints between agents, and scalability in terms of the number of agents and dimensions of the environment. Thus, this problem is a good benchmark to evaluate intelligent cooperative behaviors in multi-agent systems.

In this paper, we address one of the more sophisticated issues that require high agent autonomy: the *continuous cooperative patrol problem* (CCPP). In the CCPP, multiple autonomous agents continuously monitor and/or move around a given area without any pre-defined cooperation methods nor optimal visiting routes. In conventional patrol problems, all nodes have the same priority for visitation, and the agents' purpose is to visit all nodes with the same frequency (like the traveling salesman problem) and reduce the intervals of visits to monitor. On the other hand, nodes in the CCPP have different visitation requirements, which reflect that events in nodes occur with different probabilities, where the locations (nodes) with high probabilities indicate important locations such as easy-to-dirty regions in the vacuum cleaning domain, and location that require a high-security level where no events must be missed in the security patrolling domain. Thus, we have to reduce unawareness of events, i.e., minimize the periods from the time when an event occurs to the time when the occurrence of the event is identified by one of the agents. Another important requirement of the CCPP in our applications is that agents have to suspend operating periodically; for example, if the agents are autonomous robots, they need to charge their batteries at regular intervals. Therefore, they have to coordinate their behaviors by taking into account such a suspension as well as meeting the visitation requirement. We expect that good solutions to the CCPP would lead to enormous benefits in a variety of applications such as security surveillance and cleaning tasks.

Thus, problems similar to the CCPP are being actively studied by many researchers in the multi-agent system context. Particularly, we think that it is crucial that appropriate divisional cooperation, such as division of labor between agents, can be achieved by using only lightweight negotiation protocols based on locally observable characteristics because the cost of communication and negotiation with others always affects the performance of distributed systems [5,19]. Adam Smith [15] mentioned that the advantages of division of labor during the Industrial Revolution are (1) the improvement of skills by specializing on a specific type of work, (2) the cost reduction due to avoiding work change, and (3) the promotion of tool developments for efficiency. More importantly, he insisted that division of labor was structured as a result of individual selfish behavior in society. If a central agent could obtain all the updated complete information and could perfectly control all other agents, appropriate divisional cooperation would easily be realized. Unfortunately, this is infeasible in real-world applications because it requires a high computational cost due to the complexity

of environment and the heterogeneity of agents' behavior and unlimited communication bandwidth; thereby, it is expected that agents realize only divisional cooperation based on the autonomous decisions. Likewise, in the CCPP, any single agent cannot cover the entire area for visiting all nodes with the required frequency. Thus, dividing the area into a number of subareas and allocating them to individual agents as the responsible areas seem to be a better solution. However, studies on how agents with only local information and limited communication capability mutually generate satisfactory divisional cooperation in a bottom-up manner are not sufficiently clear for real-world applications.

We can consider two approaches to divisional cooperation in the CCPP in the context of multi-agent systems. The first is partitioning an area into separate subareas explicitly, each of which is allocated to one or a few agents as responsible areas [1,3,7,19]. Although this approach is a basic style of divisional cooperation and can easily prevent redundancy and conflicts between agents, it is not flexible enough to follow the environmental changes because the convergence speed decreases as the number of agents and the size of area increase. The second approach is not dividing the area but having the agents learn their own targets and behavioral strategies to increase their own performance [2,14]. This approach aims to establish a bottom-up cooperative manner in an entire system. We expected that this approach would have high flexibility for environmental changes. However, the mutual interference between agents caused by autonomous decisions is likely to be complicated, and the effect of the agents' local decision on the whole performance is unforeseeable; thus, the optimal methods for this approach have still not been studied well.

Therefore, we have also investigated the second approach to the CCPP, and we address the problem of how autonomous agents with limited communication capability cooperate only on the basis of local information to reduce the duration of unawareness of events in the context of the CCPP, where events occur in accordance with the probabilities specified for each location. Agents are also required to periodically suspend patrolling to charge their batteries for subsequent continuous operation. We previously proposed an autonomous learning meta-strategy to find the appropriate strategy in accordance with the environmental characteristics [20]. We also revised it so that agents indirectly learn which locations are important for their local perspectives by taking into account the behavior of other agents [17]. Their studies can improve the efficiency of coordinated patrolling but did not consider flexibility for the environmental changes. For this purpose, we focus on how the agents realize divisional cooperation in a bottom-up manner without complicated negotiation that may greatly increase computational cost.

The contribution of our research is to clarify the mechanism by which the autonomous agents implicitly realize divisional cooperation by individual learning and simple negotiation without needing to reach a complicated consensus as mentioned in Adam Smith [15]. We also investigate why such divisional cooperation facilitates the adaptation to environmental changes. For this purpose, we introduce the concept of "responsible nodes" to agents and demonstrate that simple negotiation with this concept enhances effective and flexible divisional cooperation in a bottom-up manner. Although we preliminarily reported these methods elsewhere [16,18], their formulation, analysis, and evaluation were not sufficiently described. Actually, we fully conducted new experiments with additional experimental conditions for this paper. We then analyzed these results and the structures of divisions of labor, and then discussed that agents can balance their workload and why the proposed mechanisms promoted divisional cooperation with improved flexibility and efficiency. In essence, we found that individual agents using our method gradually identified their own role to play, *specialist* or *generalist*, with the progress of learning, and the generated role-sharing structure as well

as regional divisional structure in the multi-agent system improved efficiency and flexibility of the adaptation to environmental changes.

This paper is organized as follows. First, we first discuss a number of related studies in Sect. 2. In Sect. 3, we describe our models of the environment and agents and introduce the CCPP. Section 4 presents our method and explains how agents decide which locations should be included in their responsible areas. Section 5 evaluates the effectiveness of our method and investigates its flexibility in the two basic scenarios where some agents suddenly halt and the probabilities of event occurrence in the environment suddenly change, and we explain what factors could improve the flexibility. We conclude with a brief summary in Sect. 6.

2 Related work

There have been a number of studies on the multi-agent patrolling problem. Portugal and Rocha [13] summarized the development of cooperative patrolling methods. In this research, they stated that the optimal solution for the patrolling problem depends on the environmental structure and the number of agents. In addition, they stated that a method based on the traveling salesman problem (TSP) often outperforms other strategies for most cases except in dynamic environments and environments expressed by a large graph or graphs containing long edges. They introduced not only optimal solutions but also heuristic approaches. They concluded from their summary that methods based on reinforcement learning have good distribution characteristics and usually derive the adaptive behavior of agents, which are required in many real-world domains.

There are two approaches for cooperation between agents in the patrolling problem. The first is the area division approach which divides the whole area into some subareas. This approach easily enables agents to avoid conflict with each other. Ahmadi and Stone [1] proposed a negotiation method to determine responsible subareas for individual agents by exchanging their boundary information. Elor and Bruckstein [3] achieved segmentation of the whole area in a continuous cleaning application by using indirect communications. Agents using this approach autonomously extend their responsible area on the basis of ant-pheromone and balloon models to balance the sizes of their responsible areas. Kato and Sugawara [7] proposed an autonomous and cooperative negotiation method for segmentation in a situation where dirt was accumulated differently in an environment. Suseki et al. [19] proposed a proportion regulation method inspired by task allocation in social insects. Two common issues of these area division approaches are the convergence speed and the flexibility. If the numbers of agents and subareas are large or an environment dynamically changes, the convergence speed of arranging the responsible areas decreases, and re-allocating their responsible areas to agents through negotiation becomes difficult. Clustering methods are also considered in area division approaches if we regard target data to be separated in clusters as the events to detect or the nodes to visit for patrolling. Li et al. [8] proposed a density- and distance-based hybrid clustering method in the context of multi-target detections with multi-sensor/scanning, where they assumed that the target data varied and the varying speed was slower than the scanning speed, so they could neglect the varying of the target data between different scans. We would expect to find the appropriate division by using a clustering method in a patrolling problem only when the environmental change is stable compared to the agents' activity. Popescu et al. [12] proposed a patrolling method on a wireless sensor network, in which agents collect the saved data from sensors with limited storage from their local viewpoint independently.

The second approach is not dividing the area, but agents with this approach identify their planning/behavior strategies. Sampaio et al. [14] proposed a gravity-based model in which the node to which no agents visit recently has strong gravity and the gravitation depends on distance between each agent and the node; thus, each agent visits nearby node with higher gravitation and avoids conflict. Cheng and Dasgupta [2] studied a decentralized management system considering real-world limitations such as communication range, noise in communications, and memory size. Jordan et al. [6] proposed a planning algorithm based on game theory in which self-interested agents reflected on interactions such as conflicts and congestions in the agents' utility to avoid conflict. This algorithm used the so-called taxation schemes, and a third party imposed a tax on the agent system involved in conflicts. The taxation schemes enable self-interested and non-cooperative agents to avoid conflicts. However, detecting the conflict by agents with a local view becomes difficult. Yoneda et al. [20] proposed an autonomous learning method to identify an appropriate planning strategy to determine the next targets according to knowledge on where dirt would accumulate easily. Then, they showed that the strategies selected by agents were different according to the environmental characteristics observed locally by agents. We extended their method and proposed a meta-strategy, called the *adaptive meta-target decision strategy with learning of dirt accumulation probabilities* (AMTDS/LD), that enables agents to autonomously decide planning strategies by learning where they should visit more frequently in a given area for a continuous cleaning application [17]. Agents with AMTDS/LD indirectly learn other agents behavior by learning about the environment and avoiding conflicts with others; for example, they could avoid redundant visits to nodes where so many agents frequently visit the same nodes with the previous method. These methods are more flexible, but how agents with local information in a dynamic environment, where the learning of agents is often not stable, avoid the conflict is still a challenging issue.

In recent years, the number of studies on patrolling problems based on a dynamic environment has increased. In general, the purpose of patrolling problems is to minimize the average or longest visitation interval between two visits of nodes in a given graph. This means that the environment was static and all nodes have the same visit priority. Ahmadi and Stone [1] also assumed that the event to be found was generated stochastically. Agents had to learn which location was important and changed the visit frequency to a node according to the environmental change. Othmani-Guibourg et al. [10] proposed a model assuming an environment in which edges in the environment dynamically changed, although the priority of nodes was the same. Pasqualetti et al. [11] studied a patrol problem that assumes that each node has a different priority in the context of multi-robot patrolling, but this model assumed a simple cycle graph, and the number of nodes and agents was small.

Generating divisional cooperation such as division of labor is a good solution to avoid conflicts. Ghavami et al. [4] proposed a learning method in which agents identify the human actor's social preference and also proposed a negotiation method in which agents reached the agreement instead of human actors for spatial urban land use planning. They introduced facilitators and decision makers for negotiation, and the decision makers had two roles: speaker and listener. The speaker revises the plan and proposes alternative plans, and the listeners express opinions on the plan. This negotiation between them enables individual agents to avoid conflicts, but reaching consensus is difficult and facilitators that decide the duration of negotiation phases often cause bottlenecks when the number of agents increases. However, how multiple autonomous agents should generate divisional cooperation in a bottom-up manner in a dynamic environment has been unclear. For example, there have been studies on how divisional cooperation based on the swarm system like that of social insects is achieved without a central manager [5,19].

Jones and Matarić [5] proposed an adaptive division of labor in large-scale systems. They conducted an experiment using a concurrent foraging problem where robots forage for two different types of foods in the environments to divide agents into two groups so that the ratio between the groups is identical to that between the numbers of the two types of foods, but they assumed that the numbers of foods were given in advance. Menezes et al. [9] proposed a negotiation algorithm based on an auction mechanism for patrolling. In their method, the appropriate ratio of agents to be allocated to individual tasks is not given but is decided by auction. However, the computational cost for negotiation becomes higher when the number of agents and the complexity of the environment increase.

We also proposed a new model of the dynamical patrolling problem, called the CCPP, and the methods that enhanced the emergence of divisional cooperation by interaction between individual agents with learning and using a simple negotiation. In the CCPP, agents move around in the environment to detect events that occurred stochastically. We assumed that agents have limited batteries and they have to stop to charge for subsequent patrolling. In [16], we introduced the notion of sets of responsible nodes to agents and proposed a simple negotiation that tried to balance the workload by adjusting the sets of responsible nodes in a bottom-up manner. As a result, the agents could generate role sharing in which some agents visit specific nodes and others move around in other large areas. Moreover, the agents could build some teams based on the structure of the environment although they did not negotiate with others to generate teams. We also analyzed the mechanisms to realize the divisional cooperation and found that their teams have high flexibility in divisional cooperation by changing their roles dynamically [18].

However, our previous studies were only examined using limited experimental scenarios, and the mechanism for autonomous agents to adapt to the changes has not been fully analyzed. Therefore, in this paper, we combine the insight of our previous researches with additional experiments. In particular, we evaluated our method in different change scenarios to evaluate the flexibility of our method.

3 Model

In the CCPP, events occur at each node with a different frequency, and agents move around to detect the events in the environment. Here, we explain the model based on the CCPP used in this study. Our model is an extension of the one by Sugiyama and Sugawara [17].

3.1 Environment

We introduce discrete time with units called *ticks* in which events occur and agents move and decide their strategy. The environment for agents to patrol is described by $G = (V, E)$ that can be embedded into \mathbb{R}^2 . $V = \{v_1, \dots, v_m\}$ is the set of nodes to visit, and v has coordinates such that $v = (x_v, y_v)$. E is the set of edges, and edge e has the length l_e . Agents can move to an adjacent node connected by an edge, but if obstacles $R_o (\subset V)$ exist, agents cannot move to that node. All nodes have an *event occurrence probability value* $p(v)$ ($0 \leq p(v) \leq 1$). A high value of $p(v)$ means the event will frequently occur. The number of neglected events without visiting (or monitoring) v at time t is expressed by $L_t(v)$. $L_t(v)$ is updated based on $p(v)$ at every tick by

$$L_t(v) \leftarrow \begin{cases} L_{t-1}(v) + 1 & \text{(if an event occurs),} \\ L_{t-1}(v) & \text{(otherwise).} \end{cases} \quad (1)$$

When an agent visits node v at time t , the neglected events at v are cleared and $L_t(v)$ is set to 0.

The requirement of the CCP is to minimize the values of $L_t(v)$ by visiting important nodes. Therefore, we define the performance measure D_{t_s, t_e} during the interval from t_s to t_e to evaluate our method as

$$D_{t_s, t_e}(s) = \sum_{v \in V} \sum_{t=t_s+1}^{t_e} L_t(v), \quad (2)$$

where $t_s < t_e$ and s is the strategy selected by agents. We explain this strategy in Sect. 3.3. $D_{t_s, t_e}(s)$ expresses the cumulative neglected duration for interval $(t_s, t_e]$ when agents use strategy s , so a smaller D_{t_s, t_e} indicates a better system performance.

3.2 Agent

We make two assumptions that simplify the problem so as to focus on cooperation among agents and the effect of divisional cooperation on robustness. First, multiple agents can be at the same node. This may be impossible in two-dimensional space, but many notable collision avoidance algorithms have been proposed, so we believe we can use this assumption. Second, agents know their own and others' locations. We believe this is a reasonable assumption because recent positioning technology, such as that of global positioning systems, is high precision, and external observable information such as location is easier to understand than internal information.

Let $A = \{1, \dots, n\}$ be a set of agents. The position of agent i at time t is represented as $v_t^i \in V$. Agent i has a battery with a limited capacity, so it must periodically return to its charging base v_{base}^i to charge its battery for continuous patrolling (the control algorithm for this is outside the scope of this paper [17]). Agent i learns and estimates the degree of importance $p^i(v)$ of node v , and i has a set of these importance as $P^i = \{(v, p^i(v)) | v \in V\}$. The importance can be expressed as a numerical value and differs from $p(v)$ in that each agent has a different belief for $p^i(v)$: for example, if some agents frequently visit a node, the importance of the node is low for other agents. Agent i can get the time at which any agent visits node v most recently because of the assumption described above, and i can be used to calculate the elapsed time $I_t^i(v)$ from t_{visit}^v to current time t as

$$I_t^i(v) = t - t_{\text{visit}}^v. \quad (3)$$

Agent i estimates the priority to visit a node at time t as $EL_t^i(v)$ using $p^i(v)$ and $I_t^i(v)$ as

$$EL_t^i(v) = p^i(v) \cdot I_t^i(v). \quad (4)$$

We explain how agents learn $p^i(v)$ in Sect. 4.

Communication between agents is often limited, and frequent communication is costly, so we consider these factors to model communications between agents. We denote the Euclidean distance between agents i and j as $m(v^i, v^j)$. This ignores the partitions and walls. Agents have a *communication range*, $d_{\text{com}} (> 0)$, and i can communicate with agent j at time t only when $m(v_t^i, v_t^j) < d_{\text{com}}$. To avoid a cost increase due to excessive communication, we also

define the minimum interval $T_{\text{limit}} (> 0)$. Agent i stores the last communication time with j as $T_{\text{last}}^{i,j}$, and if $T_{\text{limit}} \geq t - T_{\text{last}}^{i,j}$ at current time t , i does not communicate with j .

3.3 Planning in agents

Agent i patrols by repeating the following sequence. First, i decides the target node, v_{tar}^i , according to *target decision strategy* s . Second, i generates the path to v_{tar}^i according to a *path planning strategy*. Finally, i goes to v_{tar}^i using the generated path.

We introduce some target decision strategies below. We briefly explain these strategies; the details are discussed in [16,20].

Random selection (R): i randomly selects v_{tar}^i from V .

Probabilistic greedy selection (PGS): i randomly selects v_{tar}^i from N_g highest nodes according to the values of estimated priority $EL_t^i(v)$, where N_g is a positive integer.

Prioritizing unvisited interval (PI): i selects v_{tar}^i that was not recently visited; it selects it from the N_i highest nodes according to the interval $I_t^i(v)$ in V , where N_i is a positive integer and $I_t^i(v)$ is defined in Sect. 4.1.

Balanced neighbor-preferential selection (BNPS): i first selects a nearby node whose $EL_t^i(v)$ is large. After i has moved around nearby nodes, it selects v_{tar}^i by PGS.

Adaptive meta-target decision strategy (AMTDS): i learns the appropriate strategy from a given set of strategies $S = \{s_1, \dots, s_n\}$ [20] and selects a strategy to decide a target with ε -Greedy policy. Agents with AMTDS change their strategy according to the situation of the environment. They can obtain $p(v)$ before patrolling and set the value as $p^i(v)$ all the time. In this paper, we set S as $S = \{\text{R}, \text{PGS}, \text{PI}, \text{BNPS}\}$.

AMTDS with learning of dirt accumulation probability (AMTDS/LD): AMTDS/LD [17] is an extension of AMTDS. Agents with AMTDS/LD also change their strategy, but they cannot obtain $p(v)$ before patrolling. They learn $p^i(v)$ during their patrolling.

We introduce two path planning strategies. The first is the shortest path strategy using Dijkstra's algorithm. The second is the *gradual path generation* (GPG) method. An agent using the GPG method visits some nodes with higher $EL_t^i(v)$ if the nodes are near the generated shortest path. We found that the GPG method usually outperformed the simple shortest path strategy, so we only used GPG in our experiment. The details of these algorithms have been described elsewhere [20].

4 Proposed method

Our objective in proposing this method is to enhance effective flexible divisional cooperation from micro-behaviors such as autonomous individual learning and one-on-one negotiation. We call our method *AMTDS with learning of event probabilities and enhancing divisional cooperation* (AMTDS/EDC). The basic idea of our method is that each agent independently decides nodes which the agent is responsible for by learning the importance of each node, which is expressed by a nonnegative real value. For this purpose, we introduce a set of *responsible nodes* and a simple negotiation algorithm that uses the size and center of the responsible nodes to enhance divisional cooperation. The proposed model and method are based on those of a previous study [16], but we extend the model for flexibility to adapt to environmental changes.

Agents using our method do not strictly decide which agent should be responsible for certain nodes and the visit frequency. Instead, agents using our method suggest to other agents

which nodes seem to be important by delegating $p^i(v)$ to others during negotiation (described later), but these agents continue to visit there to some degree. On the other hand, an agent to whom the important nodes are suggested by another agent reflects the received values onto the local model. Then, they attempt to check whether the received nodes are really important, which are performed by temporarily increasing the visit frequency. Our proposed negotiation method is quite simple, and no additional negotiation such as agreement/disagreement and reallocation is performed.

4.1 Learning importance and responsible node

When agent i visits node v , $p^i(v)$ is updated from $I_t^i(v)$ as

$$p^i(v) \leftarrow \begin{cases} (1 - \beta)p^i(v) + \beta \frac{1}{I_t^i(v)} & \text{(if events on } v \text{ are cleared),} \\ (1 - \beta)p^i(v) & \text{(otherwise),} \end{cases} \tag{5}$$

where β ($0 < \beta \leq 1$) is the learning ratio

Here, we introduce the set of *responsible nodes* $V_{\text{self}}^i \subset V$. Agent i basically decides its next target v_{tar}^i from V_{self}^i (not V), but when i selects R or PI as the target decision strategy, it decides v_{tar}^i from V , since the purpose of these strategies is exploration. Agent i updates V_{self}^i when i returns to the charging base. i sorts the elements of P^i in descending order of $p^i(v)$ and defines V_{self}^i as the set of the first N_{self}^i nodes in P^i , where N_{self}^i expresses the size of V_{self}^i . If the values of $p^i(v)$ are identical for different nodes, one of them is selected randomly. We set the initial value of V_{self}^i as $V_{\text{self}}^i = V$, so N_{self}^i initially equals $|V|$ and is adjusted through the negotiation.

In addition, we introduce just two parameters calculated from V_{self}^i for negotiation. The first parameter is the *total amount of importance* of its responsible nodes $p_{\text{sum}}^i (\geq 0)$ and is calculated as

$$p_{\text{sum}}^i = \sum_{v \in V_{\text{self}}^i} p^i(v). \tag{6}$$

p_{sum}^i expresses the total burden of tasks for which i is responsible because a node with high $p^i(v)$ requires frequent visits. The second parameter is the *barycenter*, $C^i = (x_c^i, y_c^i)$ of V_{self}^i , that is the node in V closest to (x_c^i, y_c^i) , where x_c^i and y_c^i are calculated as

$$x_c^i = \sum_{v \in V_{\text{self}}^i} \frac{p^i(v)}{p_{\text{sum}}^i} x_v, \text{ and } y_c^i = \sum_{v \in V_{\text{self}}^i} \frac{p^i(v)}{p_{\text{sum}}^i} y_v. \tag{7}$$

When we define the shortest path length from node v_p to v_q as $d(v_p, v_q)$, if $d(C^i, v) < d(C^j, v)$, we assume that the cost of agent i to visit node v is smaller than that of j to visit this node. $d(v_p, v_q)$ is the shortest path length which is sum of l_e from node v_p to v_q . Note that because C^i may not be the element of V , we approximate the value of $d(C^i, v)$ by using $d(v_c, v)$, where $v_c \in V$ is the node closest to C_i based on $m(C_i, v)$.

4.2 Negotiation between agents

Agents using our method individually try to improve the elements of V_{self}^i by simple negotiation for more effective patrolling. In this negotiation, agents do not fully decide who is

responsible for the node; rather, agent i entrusts a number of important nodes to j if it is more appropriate for j to handle them.

We introduce two types of negotiations. The first is negotiation for balancing total amount of $p^i(v)$, p_{sum} , in which agents try to balance the learned importance $p^i(v)$. Nodes with higher value of $p^i(v)$ require frequent visits, so agents with higher p_{sum} have much responsibility. When there is big difference between the value of p_{sum}^i and one of p_{sum}^j : Agent i with higher p_{sum} delegates the $p^i(v)$ of some nodes that are not important for agent i to agent j with lower p_{sum} ; thus, i concentrates more on the important tasks, and j will be able to widely explore locations considered important. The second negotiation is for improving the performance, and agents carefully trade-off the responsible nodes between agents when their p_{sum} is almost identical. Hence, agent i delegates the $p^i(v)$ of some nodes that are important for agent i and another agent j can visit the nodes at a lower cost; thus, both agents i and j can decrease the cost to patrol in their responsible nodes.

4.2.1 Negotiation for balancing tasks

If condition

$$1 + T_c < p_{\text{sum}}^i / p_{\text{sum}}^j \tag{8}$$

is satisfied, agents i and j negotiate to balance their p_{sum} . T_c ($0 < T_c \ll 1$) is the threshold value to judge whether there is a difference of responsibility between i and j . Then, i calculates the ordered set

$$V_{\text{self}}^{i,j} = \{v \in V_{\text{self}}^i \mid d(C^i, v) > d(C^j, v)\},$$

where the elements are sorted by $p^i(v)$ in descending order. Then, i selects the smallest e_g (positive integer) nodes that are not so important to i in $V_{\text{self}}^{i,j}$ (i.e., from the tail), and i delegates its $p^i(v)$ to j as

$$\begin{aligned} p^j(v) &\leftarrow p^j(v) + p^i(v) \times (1 - \delta), \\ p^i(v) &\leftarrow p^i(v) \times \delta, \end{aligned} \tag{9}$$

where δ ($0 < \delta < 1$) is the ratio to delegate. e_g is determined on the basis of the ratio of p_{sum}^i to p_{sum}^j :

$$e_g = \min \left(N_{\text{self}}^i - 1, N_{g\text{max}}^i, \left\lceil \frac{p_{\text{sum}}^i}{p_{\text{sum}}^j} \times \gamma \right\rceil \right), \tag{10}$$

where $N_{g\text{max}}^i$ ($0 < N_{g\text{max}}^i < N_{\text{self}}^i$) is the upper limit to prevent big fluctuations, and γ ($0 < \gamma$) is adjustment ratio to resolve imbalance. After agents delegate or receive $p^i(v)$, agents i and j update their sizes of responsible nodes by

$$\begin{aligned} N_{\text{self}}^i &\leftarrow N_{\text{self}}^i - e_g \\ N_{\text{self}}^j &\leftarrow \min(|V|, N_{\text{self}}^j + e_g). \end{aligned} \tag{11}$$

4.2.2 Negotiation for trade-off of responsibility

If condition

$$1 - T_c < p_{\text{sum}}^i / p_{\text{sum}}^j < 1 + T_c \tag{12}$$

is satisfied, they negotiate to improve their V_{self}^i by swapping $p^i(v)$ of several important nodes. i selects the first N_{cmax}^i nodes from the head of $V_{self}^{i,j}$, and then i delegates those $p^i(v)$ to j according to Eq. 9. Then, e_g is determined as

$$e_g = \min \left(N_{self}^i - 1, N_{cmax}^i \right), \tag{13}$$

where $N_{cmax}^i (> 0)$ is the upper limit. Note that nodes with high $p^i(v)$ incur relatively high burdens, so N_{cmax}^i must be a small constant and be much less than N_{gmax}^i . Then, they update their sizes of responsible nodes by using Eq. 11. When Eq. 12 is satisfied, agent j is also likely to send a part of the learned $p^j(v)$ to i , so these processes occur in the opposite direction.

5 Experiments and discussion

5.1 Experimental setting

To evaluate our method, we prepared a large environment for agents to patrol (Fig. 1) that consists of six rooms (Room 0-5), a corridor, and a number of nodes where events occur frequently. It was represented by a 101×101 2-dimensional grid space with several obstacles, and we set a length of all edges to one. We made the environment using C#. We set $p(v)$ for $v \in V$ as

$$p(v) = \begin{cases} 10^{-3} & \text{if } v \text{ was in a red region,} \\ 10^{-4} & \text{if } v \text{ was in an orange region, and} \\ 10^{-6} & \text{otherwise,} \end{cases} \tag{14}$$

where the colored regions are as shown in Fig. 1. We provide two types of environments with different event occurrence probabilities, Office and Office2, to evaluate the flexibility to adapt to environmental change. The event occurrence probabilities of each room of Office2 are obtained by shifting the probabilities of Office clockwise by one room, which are shown

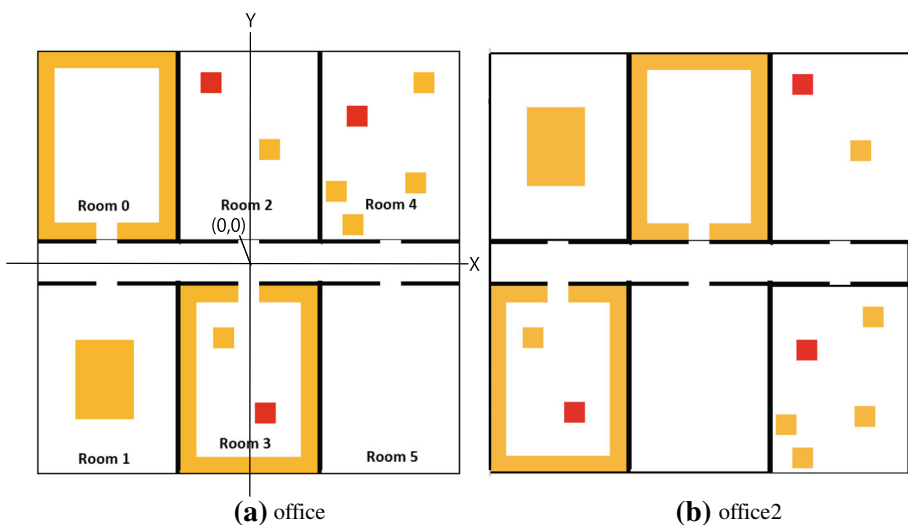


Fig. 1 Experimental environment

Table 1 Values of parameters used in Sec. 3

Model	Parameter	Value
PGS	N_g	5
PI	N_i	5
AMTDS/LD	β	0.05
	ε	0.05
Communication	d_{com}	5
	T_{limit}	10,800

Table 2 Value of parameters used in Sec. 4

Strategy	Parameter	Value
AMTDS/EDC	$N_{g\text{max}}$	100
	$N_{c\text{max}}$	10
	T_c	0.05
	γ	10
	δ	0.5

in Fig. 1. We set the number of agents, $|A|$, to 20 and set their charging base, v_{base}^i , on $v_{\text{base}}^i = (0, 0)$ for $\forall i \in A$. Agents start their patrol from their v_{base}^i and must periodically return to v_{base}^i before their battery runs out. The capacity of the battery in each agent enables them to move at most 900 ticks, and it requires 2700 ticks for a full charge when the battery is completely empty. The maximum cycle of movements and charges is 3600 ticks, so we measure $D_{t_s, t_e}(s)$ every 3600 ticks.

In addition to AMTDS/LD (LD) and AMTDS/EDC (EDC), we introduce *AMTDS/EDC without responsible nodes* (EDCRN) as the target decision strategy of agents. Agents with EDCRN decide the target node and negotiate with others in almost the same way as for EDC. The difference between EDCRN and EDC is that agents with EDCRN are always responsible for all nodes in an environment, which means they do not change their N_{self} with negotiation. The parameter values used in the model and our method are listed in Tables 1 and 2.

5.2 Efficiency evaluation

Before we evaluate the flexibility to adapt to changes in environment, we evaluate the efficiency of EDC. In this experiment, 20 agents move around to detect events in the Office environment shown in Fig. 1a. We have already discussed the difference in efficiency between EDC and LD [16]. In this paper, we additionally introduce EDCRN for more detailed analysis of factors to realize efficient and flexible divisional cooperation. The reason for improvement in efficiency is also related to flexibility to change, so we explain it here. Note that agents with EDC have limited responsible nodes, and agents with LD and EDCRN are responsible for the whole environment.

Figure 2 plots the improvement of $D(s)$ over time. EDCRN finally decreased $D(s)$ by 13.6%, and EDC finally decreased $D(s)$ by 27.2% compared with that of LD. The difference between EDCRN and LD is that agents with EDCRN negotiate and transfer their responsibility with others. Therefore, the results indicate both the negotiation and having a responsibility for a limited area are important for effective divisional cooperation.

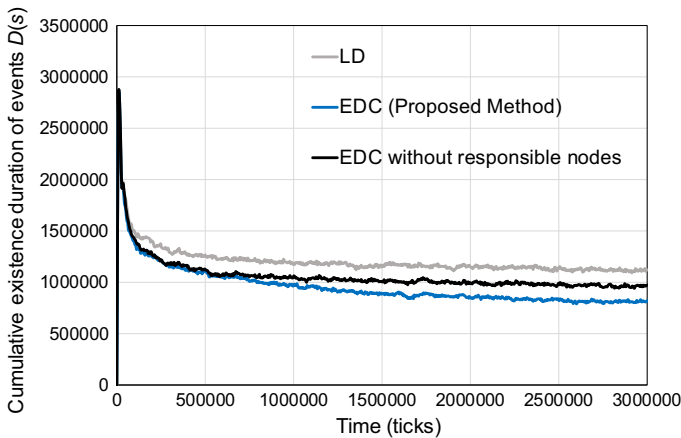


Fig. 2 Improvement in $D(s)$ over time

We measured the working time of each agent for each node to investigate the structure of divisional cooperation. Figure 3a–c plot the working time of agents with LD, EDCRN, and EDC in individual rooms during the last 1,000,000 ticks, and the 20 agents in these figures are sorted in descending order of working time in Room 3. Figure 3c shows that agents with EDC mainly worked in one or two rooms with much more bias than LD (Fig. 3a) and EDCRN (Fig. 3b). Agents with LD indirectly learned the behavior of others through learning $p^i(v)$ which became lower when other agents frequently visited the node. Therefore, agents targeted different nodes from other agents. However, their divisional cooperation was not enough. Agents with EDCRN delegated their $p^i(v)$ through negotiation. However, compared to EDC, the bias of the working time of agents with EDCRN was smaller. In particular, Room0 and Room1 were visited by more agents because there were regions with a much higher value of $p(v)$, the red regions in Fig. 1, in Room2 and Room4, so agents that were responsible for the whole environment could not ignore these regions and had to regularly visit these regions. Many agents that could not ignore these regions reduced the opportunities to visit well-learned nodes that were primarily visited by them. Also, as the opportunities for detecting events were dispersed to many agents, learning the appropriate visit frequency of a node became more difficult because the appropriate visit frequency depended on the behavior of other agents that often target the node.

Divisional cooperation was promoted by agents with EDC since they divided the agents into four groups like teams without any negotiation to make a team, and each team mainly visited a different room. An agent that detected a node required frequent visits and had a higher value of p_{sum}^i and decreased its size of responsibility N_{self}^i through the negotiation. As a result, the barycenter C^i of the agent approached the detected node. Since an agent whose C^i approached a detected node was delegated responsibility for nodes around the detected node, the agent specialized on the node and moved around the node. Moreover, switching the costs of a specialized agent to move around also reduced the costs because the agent focused on a specific node. Agents try to maximize the number of detected events. Therefore, agents do not give higher priority to a node that has a higher value of $p(v)$ and was visited by many agents when the agent can find a node where it detects more events. As a result, the appropriate visit frequency to an area was automatically adjusted according to the number of agents that visited the node and the visit frequency throughout the system. Our method

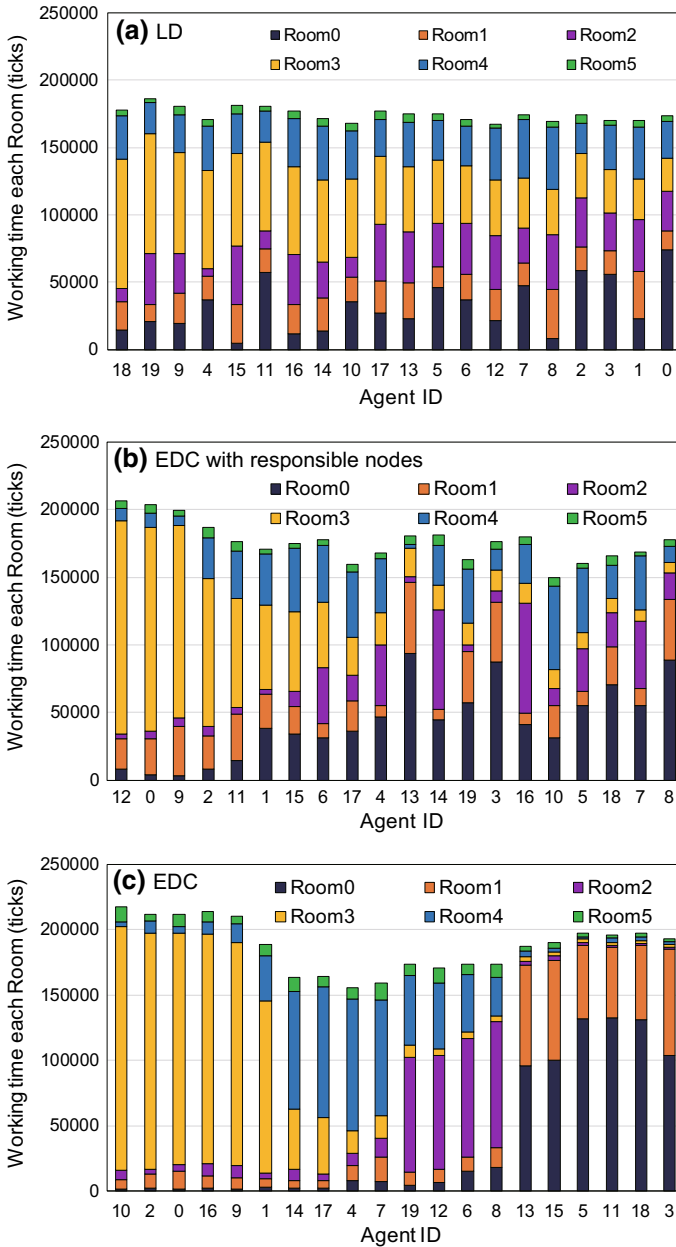


Fig. 3 a Distribution of working time during last 1,000,000 ticks in agents with LD. b Distribution of working time during last 1,000,000 ticks in agents with EDCRN. c Distribution of working time during last 1,000,000 ticks in agents with EDC

realized selection and integration by micro-behavior such as negotiation between agents, and its micro-behavior has created macro-divisional cooperation as a team.

Our method also created role sharing based on the size of responsibility. Figure 4 plots the size of responsible node N_{self}^i of agents with EDC at 3,000,000 ticks. This figure is

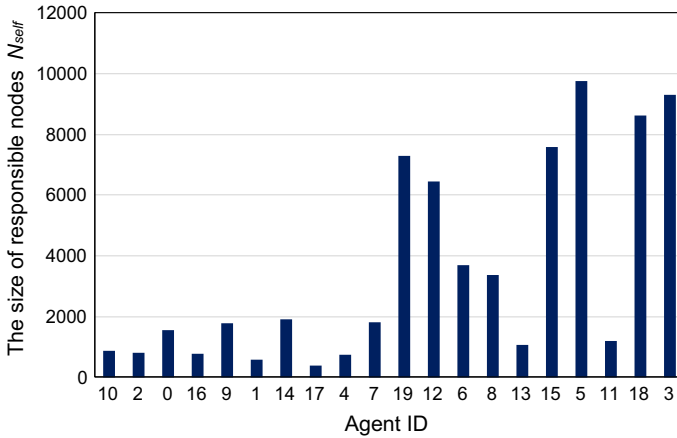


Fig. 4 Size of responsible nodes N_{self}^i at 3,000,000 ticks when agents use AMTDS/EDC as a target decision strategy

sorted in same order to Fig. 3c. This figure indicates that some agents focus on specific nodes (*specialists*), and some agents move around a larger area (*generalists*). Specialists occurred due to selection and integration. Agents that were specialists could move around with high accuracy in their responsible nodes because these agents focused on a specific area and have many opportunities to learn the importance of the nodes. Generalists occurred due to the negotiation for balancing tasks. Agents that were generalists were delegated the responsibility of some nodes that were not focused on by specialists. They explore a larger area and did not need to visit high $p(v)$ nodes because the specialists frequently visited them instead. We consider that the role sharing which is a divisional cooperation also raised the efficiency of the patrols.

In conventional region segmentation methods, agents have a specific role and limited responsible nodes, but the responsible area is usually connected, and the ratio of agents to be deployed in each area has been given. In contrast, agents using our method divided their responsible area into discontinuous nodes in a bottom-up manner and autonomously determined how to visit each area.

5.3 Flexibility evaluation for stop of agents

Next, we evaluate the flexibility to adapt to change. We experimented and analyzed our method in two scenarios of environmental change. In the first scenario, several agents suddenly halted and did not communicate with other agents. In the second scenario, the event occurrence probabilities of each node suddenly changed. In this section, we discuss the results of the first scenario.

We stopped ten agents selected randomly at 1,000,000 ticks; after that they restarted at 2,000,000 ticks. The stopped agents could not move and could not communicate with the other agents when they were stopped, and the other agents did not know they were stopped. Figure 5 plots the improvement in $D(s)$ over time. Compared to LD, EDCRN decreased $D(s)$ by 23.1% after the stop, and EDC greatly prevented the deterioration of efficiency after the stop, where $D(s)$ decreased by 36.6% at the peak. This result shows that agents with EDC flexibly responded to the change. Moreover, EDC outperformed LD and EDCRN before/after agents stopped at all times.

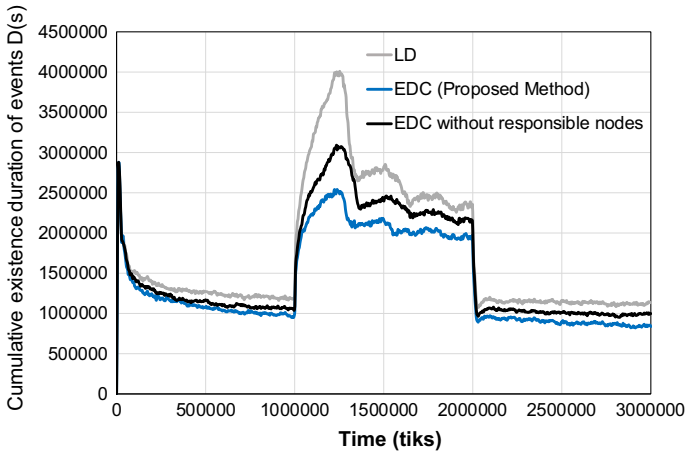


Fig. 5 Improvement in $D(s)$ over time in a scenario where some agents halt

We analyzed how agents with EDC flexibly reacted to the stop of agents. Figure 6a shows the change in p_{sum}^i of agents with EDC without a stop, and Fig. 6b shows the change in N_{self}^i of these agents in an experimental trial. Note that this experimental trial was randomly selected from thirty trials, and similar characteristics were observed in other trials. Figure 6a shows that the value of p_{sum}^i greatly changed after some agents stopped because the agents discovered and learned a remaining task that a stopped agent was primarily responsible for. Similarly, Fig. 6b shows that agents changed their value of N_{self}^i by negotiation according to the change in the value of p_{sum}^i after some agents stopped. Interestingly, some generalists (agents 1, 3, 12) were drastically decreasing the value of N_{self}^i , and some specialists (agents 2, 4) were drastically increasing it. We conclude that generalists who move around widely could quickly find remaining uncovered tasks, then the value of p_{sum}^i of the generalists became higher than the value of other agents so some generalists change their role to specialists. In contrast, a specialist with lower value of p_{sum}^i was delegated a part of $p^i(v)$ of some nodes and moved around widely as generalists after environmental change.

Figure 6b also shows that several agents (agents 11, 13, 19) do not change their role before and after the change in environment. There are two reasons. One is that other agents discovered and responded to changes, so they were able to continue working as usual. p_{sum}^i of these agents (agents 13, 19) changed little. Everyone responding to the change causes great confusion, but in our method, agents that do not respond to change in accordance with the situation autonomously appear. The other is that although the amount of work had changed, there was no other agents to delegate responsibility. The value of p_{sum}^i of agent 11 had changed significantly; so in the usual case, agent 11 delegated its responsibility to others by negotiating for fairness. However, agent 11 was a specialist and was responsible for only a few nodes. Agent 11 had few chances to delegate its responsibility when there were few agents that could execute tasks that agent 11 was responsible for with a lower cost compared with barycenter C^i and workload p_{sum}^i . Therefore, after another agent restarted at 2,000,000 ticks, agent 11 was able to decrease its p_{sum}^i and change its role. In our negotiation, agents compared just the workload and the barycenter, but agents automatically improved the information about the importance of each node and dynamically changed their role to adapt to change.

Specialists greatly contribute to efficiency as mentioned in Sect. 5.2, and generalists greatly contribute to adapting to change. We conducted an additional experiment in which the top

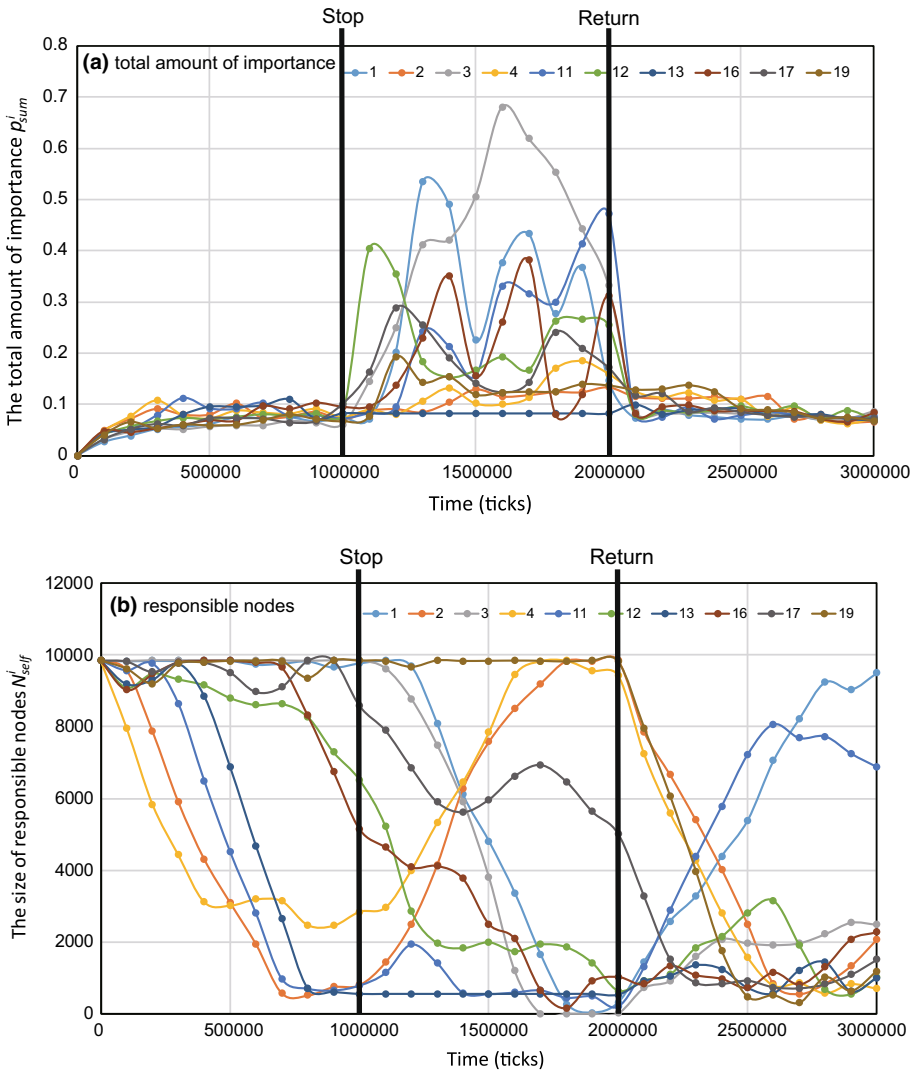


Fig. 6 **a** Change in p_{sum}^i . **b** Change in N_{self}^i

ten (generalist) or bottom ten (specialist) N_{self}^i agents with EDC were stopped. Figure 7 plots the improvement in $D(s)$ of the average of 30 trials. It shows that it is inefficient when the generalists stopped. As generalists moved around widely in environments, they discovered the change at an early stage. Also, in contrast to specialists, there were many other agents that generalists could delegate their responsibility to. However, as specialists focused on specific nodes, generalists greatly reduced switching costs to visit there and moved around in a wider area and improved overall efficiency. Therefore, agents with EDCRN that are responsible for the whole environment without roles such as specialist and generalist did not sufficiently respond to change. As a result, we conclude that the roles of both between specialists and generalists are important for flexibility, and agents with our method decide and adjust their roles automatically.

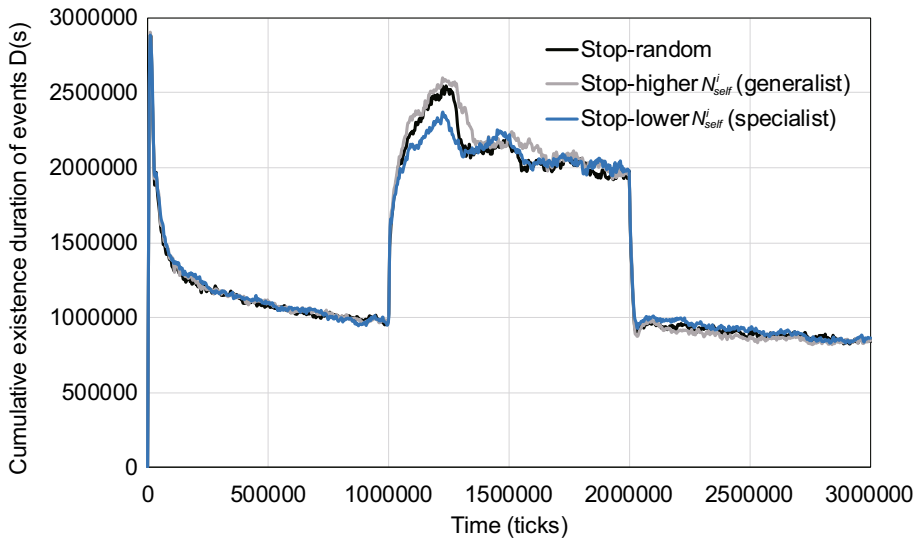


Fig. 7 Improvement in $D(s)$ over time when specialist or generalist halted

5.4 Flexibility evaluation for change in environment

In this section, we evaluate the flexibility of LD, EDCRN, and EDC in the second scenario where the event occurrence probabilities of the environment suddenly change. The number of agents was 20, and the event occurrence probabilities were changed from those of Office shown in Fig. 1a to those of Office2 shown in Fig. 1b at 1,000,000 ticks. After that, the probabilities were changed from those of Office2 to those of Office at 2,000,000 ticks. Figure 8 plots the improvement in $D(s)$ over time. Compared to LD, EDCRN decreased $D(s)$ by 4.3% after the stop, and EDC decreased $D(s)$ by 6.7% at the peak. At 2,000,000 ticks, EDCRN decreased $D(s)$ by 13.2% after the stop, and EDC decreased $D(s)$ by 25.1% compared with LD. As with the results of the first scenario, the efficiency of EDC was best, EDCRN was in the middle, and LD was the worst. However, the difference of $D(s)$ between LD, EDCRN, and EDC at the peak was much smaller than the difference of the first scenario.

The changes in the second scenario affect agents much more than changes in the first scenario. In the first scenario, as ten agents stopped, the number of agents to consider the agents' behavior decreased, and the remaining agents just additionally learned tasks that the stopped agent did. However, in the second scenario, the information learned by agents greatly diverged from the actual event occurrence probability after the change. Moreover, almost all agents changed their behaviors and must respond to changes in other agents' behaviors at the same time.

For such a complex change, agents with LD and EDC responded to the changes in a different way from the first scenario. Figure 9 plots the number of selected target decision strategies of first scenario in LD and EDC over time, and Fig. 10 plots one of the second scenarios in LD and EDC over time. Comparing Figs. 9 and 10, there is a big difference in the selected strategies at a certain time immediately after the environmental change. In Fig. 10, at first many agents selected the R or PI strategy, next, the PGS strategy is selected, and after that, the BNPS strategy is selected by many agents. In contrast to the case of the first scenario, the changes in the number of selected target decision strategies in the second scenario are

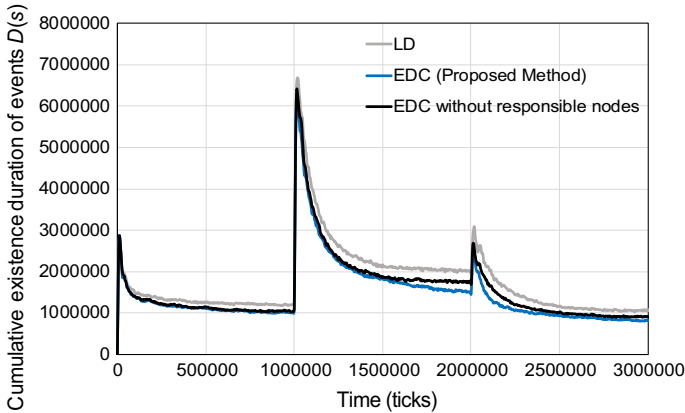


Fig. 8 Improvement in $D(s)$ over time

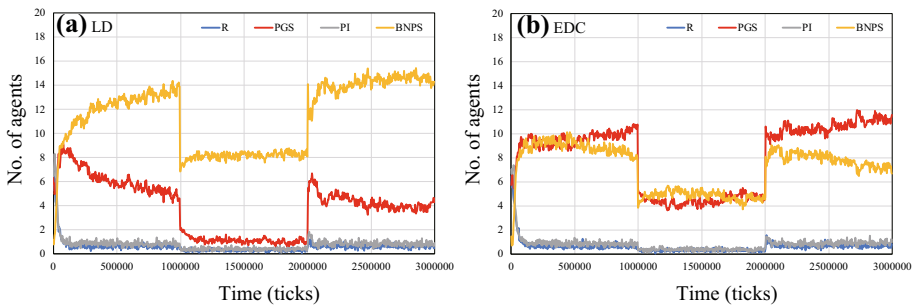


Fig. 9 Number of selected target decision strategies over time in scenario where some agents halt

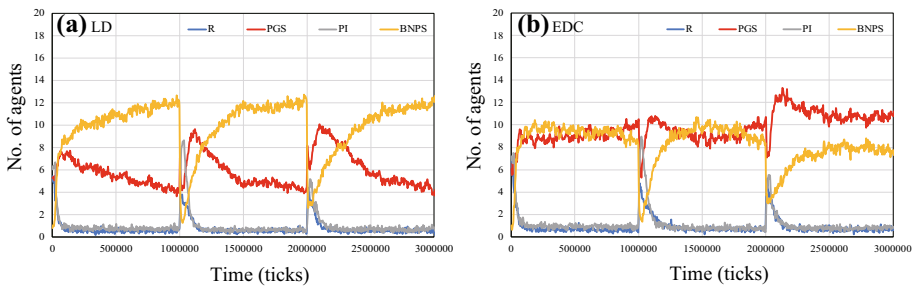


Fig. 10 Number of selected target decision strategies over time in scenario where events occurrence probabilities change

similar between LD and EDC because the difference in peak efficiency was much smaller between LD, EDCRN, and EDC than in the first scenario. After the environment changed, $p^i(v)$ and $EL_i^i(v)$ of each agent became inaccurate. Therefore, many agents selected a PI strategy that did not use a value of $p^i(v)$ unlike GPS and BNPS. After agents discovered a new location with many events, the GPS strategy outperformed the other strategies. At that time, the learning and behavior of other agents were not stable yet, so competition did not occur that much. However, as the behaviors became stable, agents selected the BNPS strategy

because it was effective in a situation where the learning of an agent was not enough and uncertainty in targeting a far node was big. At the second change, since the past knowledge remained in the agents, the agents with EDC that could efficiently redistribute their knowledge effectively adapted to change, so Fig. 9b shows that agents with EDC continued to select the PGS strategy after the change. When only the learning of probabilities of each node and negotiation were not able to adapt to change, the agents using our method changed the target decision strategy which strongly affects agents' behavior and adapted to change in a different way.

6 Conclusion

We focused on the CCPP, which requires higher autonomy and cooperation between agents, and investigated the mechanisms with which agents autonomously realize divisional cooperation and the group of agents that could follow environmental changes by revising their divisional cooperation structure. First, we proposed the autonomous learning and negotiation method in which agents generate the cooperation regime only with local interaction. Our method did not partition responsible areas nor did it explicitly decide which agents are responsible for each node. Instead, agent i using our method learns the importance of nodes, P^i , by its local viewpoint and identifies the set of the responsible nodes, V_{self}^i . Then, i partially transfers the values of importance of the responsible nodes to other agents and *vice versa*, and i decreases/increases the size of the responsible nodes, N_{self}^i . Our experimental evaluation revealed that our method enhanced divisional cooperation compared to our previous method. We found that our method developed two distinct roles, specialist and generalist, and that agents were able to move around with different patterns. This is one reason our method outperformed the previous one. With the negotiation and individual learning, the specialist agents have more responsibility on the specific nodes that usually require high visitation, and the generalist agents moved around the wider area to cover other nodes.

Then, we conducted two additional experiments to evaluate the flexibility of our method to adapt to environmental changes. In the first experimental scenario, some agents suddenly halted; thus, other agents have to find some areas that were neglected. The results suggest that our method could flexibly adapt to this situation. In this scenario, the generalists played an important role; they could quickly find the neglected nodes and identified them as nodes to visit by re-calculating the importance of nodes. This might force a few generalists to change to specialists temporarily, but other agents were not affected by this environmental change. After the halted agents returned to the environments, they mutually start to cover some important nodes, and all agents converged to another stable structure of divisional cooperation. This also indicates that our method is advantageous compared to the area-partitioning approaches that required more time to follow the change because the interference of agent using these approaches will be limited between the neighboring agents. This may result in slow adaptation to the changes in large environments.

In the second scenario, the event occurrence probabilities in the environment largely changed suddenly. This change led to the invalidity of learned results so far and forced agents to re-learn the importance of many nodes; thereby, the behaviors of other agents were also altered. Actually, our experimental results showed that only re-learning and negotiation were not enough to quickly adapt to the changes, unlike in the previous scenario. However, agents using our method could respond to the changes in a different way. First, they temporarily changed their target decision strategies to "R" or "PI", which does not use learned P^i , and then they start re-learning the importance and perform negotiation with others. After

a sufficient amount of processing, they gradually adopted the appropriate target decision strategies. This result suggests that the diversity of strategies facilitates the flexibility to follow the environmental changes.

In this research, we considered the efficiency of event detection. In terms of sustainability, we think that the efficiency of energy saving is also required, so we will try to propose a method in which agents save their energy when the quality of work is sufficient for the requirements. We also plan to compare our method with a deterministic method such as the TSP-based approach and will clarify the environmental characteristics in which our method effectively leads to divisional cooperation structure in our next research.

Acknowledgements This work was partly supported by JSPS KAKENHI Grant No. 17KT0044 and Grant-in-Aid for JSPS Research Fellow (JP16J11980).

References

- Ahmadi M, Stone P (2006) A multi-robot system for continuous area sweeping tasks. In: Proceedings of the 2006 IEEE international conference on robotics and automation. pp 1724–1729
- Cheng K, Dasgupta P (2007) Dynamic area coverage using faulty multi-agent swarms. In: IEEE/WIC/ACM international conference on intelligent agent technology, 2007. IAT 07, pp 17–23
- Elor Y, Bruckstein AM (2009) Multi-(a)gent graph patrolling and partitioning. In: Proceedings of the 2009 IEEE/WIC/ACM international joint conference on web intelligence and intelligent agent technologies. pp 52–57
- Ghavami SM, Taleai M, Arentze T (2017) An intelligent spatial land use planning support system using socially rational agents. *Int J Geogr Inf Sci* 31(5):1022–1041. <https://doi.org/10.1080/13658816.2016.1263306>
- Jones C, Mataric MJ (2003) Adaptive division of labor in large-scale minimalist multi-robot systems. In: Proceedings of the 2003 IEEE/RSJ international conference on intelligent robots and systems (IROS 2003) vol 2. pp 969–1974
- Jordán J, Torreño A, de Weerd M, Onaindia E (2017) A better-response strategy for self-interested planning agents. *Appl Intell* 48:1020–1040
- Kato C, Sugawara (2013) Decentralized area partitioning for a cooperative cleaning task. In: Boella G, Elkind E, Savarimuthu B, Dignum F and Purvis M (eds), PRIMA 2013: principles and practice of multi-agent systems. Lecture notes in computer science, vol 8291. Springer Berlin Heidelberg, pp 470–477
- Li T, la Prieta Pintado FD, Corchado JM, Bajo J (2017) Multi-source homogeneous data clustering for multi-target detection from cluttered background with misdetection. *Appl Soft Comput* 60:436–446
- Menezes T, Tedesco P, Ramalho G (2006) Negotiator agents for the patrolling task. In: Sichman JS, Coelho H, Rezende SO (eds) Proceedings of advances in artificial intelligence-IBERAMIA-SBIA 2006: 2nd International Joint Conference, 10th Ibero-American Conference on AI, 18th Brazilian AI Symposium, Ribeirão Preto, Brazil, October 23–27, 2006. Springer, Berlin, Heidelberg, pp 48–57
- Othmani-Guibourg M, Fallah-Seghrouchni AE, Farges JL, Potop-Butucaru M (2017) Multi-agent patrolling in dynamic environments. In: 2017 IEEE international conference on agents (ICA). pp 72–77
- Pasqualetti F, Durham JW, Bullo F (2012) Cooperative patrolling via weighted tours: performance analysis and distributed algorithms. *IEEE Trans Robot* 28(5):1181–1188
- Popescu MI, Rivano H, Simonin O (2016) Multi-robot patrolling in wireless sensor networks using bounded cycle coverage. In: 2016 IEEE 28th international conference on tools with artificial intelligence (ICTAI). pp 169–176
- Portugal D, Rocha R (2011) A survey on multi-robot patrolling algorithms. In: Camarinha-Matos LM (ed) Proceedings of technological innovation for sustainability: second IFIP WG 5.5/SOCOLNET doctoral conference on computing, electrical and industrial systems, DoCEIS 2011, Costa de Caparica, Portugal, February 21–23, 2011. Springer, Berlin, Heidelberg, pp 139–146
- Sampaio P, Ramalho G, Tedesco P (2010) The gravitational strategy for the timed patrolling. In: 2010 22nd IEEE international conference on tools with artificial intelligence (ICTAI). pp 113–120
- Smith A (1776) An inquiry into the nature and causes of the wealth of nations, number smith1776 In: History of economic thought books. McMaster University archive for the history of economic thought. <https://ideas.repec.org/b/hay/hetboo/smith1776.html>. Accessed 22 Dec 2017

16. Sugiyama A, Sea V, Sugawara T (2016) Effective task allocation by enhancing divisional cooperation in multi-agent continuous patrolling tasks, In: 2016 IEEE 28th international conference on tools with artificial intelligence (ICTAI). pp 33–40
17. Sugiyama A, Sugawara T (2015) Meta-strategy for cooperative tasks with learning of environments in multi-agent continuous tasks In: Proceedings of the 30th annual ACM symposium on applied computing. pp 494–500
18. Sugiyama A, Sugawara T (2017) Improvement of robustness to environmental changes by autonomous divisional cooperation in multi-agent cooperative patrol problem. In: Demazeau Y, Davidsson P, Bajo J, Vale Z (eds) Advances in practical applications of cyber-physical multi-agent systems: the paams collection. Springer, Cham, pp 259–271
19. Suseki K, Mizuguchi T, Sugawara K, Kosuge K (2005) Proportion regulation for division of labor in multi-robot system, In: Proceedings of the 2005 IEEE/RSJ international conference on intelligent robots and systems. pp 2339–2344
20. Yoneda K, Sugiyama A, Kato C, Sugawara T (2015) Learning and relearning of target decision strategies in continuous coordinated cleaning tasks with shallow coordination. *Web Intell* 13(4):279–294

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Ayumi Sugiyama received his B.S. and M.S. degrees in computer engineering from Waseda University, Japan, in 2014 and 2016. He is currently a doctoral student in Computer Science and Communications Engineering at Waseda University. His research interests include multi-agent systems, artificial intelligence, behavioral science, and ethology. He is a Research Fellow (DC1) of Japan Society for the Promotion of Science. He received the Distributed Systems Best Paper Award at Proceedings of the 30th ACM/SIGAPP Symposium On Applied Computing in 2015.



Vourchteang Sea is a Ph.D. graduate from the Department of Computer Science and Communications Engineering at Waseda University, Japan, receiving a Doctor of Engineering in Computer Science on September 2018. Her research interests lie in the areas of distributed computing, machine learning, computer vision, and multi-agent systems. She is the recipient of the best student paper award of IIAI-AAI 2015. She received her Bachelors and Masters degrees in Computer Science from Royal University of Phnom Penh in 2012 and Waseda University in 2015, respectively.



Toshiharu Sugawara is a professor of Department of Computer Science and Engineering, Waseda University, Japan, since April, 2007. He received his B.S. and M.S. degrees in Mathematics, 1980 and 1982, respectively, and a Ph.D., 1992, from Waseda University. In 1982, he joined Basic Research Laboratories, Nippon Telegraph and Telephone Corporation. From 1992 to 1993, he was a visiting researcher in Department of Computer Science, University of Massachusetts at Amherst, USA. His research interests include multi-agent systems, distributed artificial intelligence, machine learning, internet working, and network management. He is a member of IEEE, ACM, AAAI, Internet Society, The Institute of Electronics, Information and Communication Engineers (IEICE), Information Processing Society of Japan (IPSJ), Japan Society for Software Science and Technology (JSSST), and Japanese Society of Artificial Intelligence (JSAI).