Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 465–474

Procedia
Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Energy-Efficient Strategies for Multi-Agent Continuous Cooperative Patrolling Problems

Lingying Wu[a], Ayumi Sugiyama[a], Toshiharu Sugawara[a,*]

[a]*Department of Computer Science and Communications Engineering, Waseda University, Tokyo 1698555, Japan*

Abstract

Whereas research of the multi-agent patrolling problem has been widely conducted from different aspects, the issue of energy minimization has not been sufficiently studied. When considering real-world applications with a trade-off between energy efficiency and level of perfection, it is usually more desirable to minimize the energy cost and carry out the tasks to the required level of quality instead of fulfilling tasks perfectly by ignoring energy efficiency. This paper proposes a series of coordinated behavioral strategies and an autonomous learning method of target decision strategies to reduce of energy consumption on the premise of satisfying quality requirements in continuous patrolling problems by multiple cooperative agents. We extended our previous method of target decision strategy learning by incorporating a number of behavioral strategies, with which agents individually estimate whether the requirement is reached and then modify their action plans to reduce energy consumption. It is experimentally shown that agents with the proposed methods learn to decide the appropriate strategies based on energy cost and performance efficiency and are able to reduce energy consumption while cooperatively meeting the given requirements of quality.

Keywords: Multi-agent systems; Continuous patrolling; Learning; Cooperation; Energy efficiency

1. Introduction

Along with recent advances in robotics and computational technologies, robot applications have gained popularity in real-world environments. However, restrictions of speed, movement, and battery capacity restrict the performance of single-robot systems. Therefore, collaboration between multiple robots is required to cover a busy or large environment. From the 1990s, the application mode of robots has been changing from cell to system, making cooperative robotics an important field in artificial intelligence [1]. In spite of partial failures in the system or changes in the work environment, multi-robot systems which are fault-tolerant can complete tasks by compensation and cooperation.

[*] Corresponding author. Tel.: +81-3-5286-2596 ; fax: +81-3-5286-2596.
E-mail address: lingying.wu@isl.cs.waseda.ac.jp, a.sugiyama@isl.cs.waseda.ac.jp, sugawara@waseda.jp

10.1016/j.procs.2019.09.201

In this study, we tackle the *continuous cooperative patrolling problem* (CCPP) [18] by multiple autonomous agents, which are intelligent programs that control robots to move around the given area. Multi-agent CCPP is the abstract problem for many real-world applications, including cleaning, security, and surveillance patrolling tasks with non-uniform visit frequencies for given purposes. One important assumption for the multi-agent CCPP is that shallow communication is used instead of sophisticated communication, because agents may only have CPUs with limited performance, a constrained amount of memory, and limited battery capabilities. As might be expected, it is reasonable to suppose that agents must decide their action plans and routes on the basis of limited information including knowledge from the local viewpoints and using narrow communication bandwidth. Moreover, owing to battery and CPU limitations, periodical return to the charging base for recharging is required to ensure robots' continuous performance.

To deploy actual systems in the real world, realistic scenarios in the multi-agent patrolling problem field must be considered [2]. Portugal and Rocha [3] pointed out that research in this field should be oriented toward solutions with applicability in the real-world. Practical applications draw attention to function effectiveness, performance requirements, and energy efficiency, but there is a trade-off between energy efficiency and level of perfection. Despite the fact that multi-agent patrolling has been investigated from many different perspectives over a long period of time, the issue of energy minimization has not been sufficiently studied. When considering real-world applications, rather than accomplishing the tasks perfectly by ignoring energy consumption, we usually place a higher value on reduction of energy cost. For instance, in area cleaning tasks, it is not necessary to keep the cleanliness of the environment extremely low all the time. Instead, we would prefer the agents to cooperatively satisfy the given requirement of cleanliness with the lowest possible energy cost.

The contribution of this paper is to propose a series of agents' behavioral strategies by aiming to improve energy efficiency. First, we extend our previous method *adaptive meta-target decision strategy* (AMTDS) [16] so that agents with the extended method autonomously select appropriate target decision strategies during planning by monitoring the local energy consumption and number of handled events. Then, we propose algorithms for estimating the status of the work environment and evaluating self-importance, which enable agents to individually judge whether the given requirement of quality is satisfied and to understand their contribution degree regarding the system's purpose based on their recent and expected performance. On top of that, we introduce two types of energy-saving behaviors, *homing* and *pausing*, for agents to take instead of moving on to the next target. Agents would only perform these behaviors when they consider that the requirement level has been reached. In addition, the more important they consider they are, the higher the probability is that the agents perform these behaviors. Agents with the homing behavior stop patrolling and head toward the charging base to charge, and those performing the pausing behavior rest at the charging base and do not move around for a certain interval of time. We experimentally show that our methods enabled agents to individually select appropriate target decision strategies and, more importantly, to reduce the energy cost while cooperatively maintaining the required levels of perfection.

2. Related Work

A number of studies have been devoted to continuous patrolling problems. Ahmadi and Stone [4] defined the formulation of a continuous area sweeping task and introduced an initial approach that non-uniformly visits the environment to minimize the estimated cost. They then extended the approach to a multi-robot scenario, where area partitioning by negotiation between agents was conducted [5]. Moreira et al. [6] argued that multi-agent patrolling can be a good benchmark for multi-agent systems and proposed a software simulator constructed strictly for the patrolling tasks. Santana et al. [7] solved the multi-agent patrolling problem using reinforcement learning by automatically adapting the strategies of agents to the environment. Portugal and Rocha [3] proposed a distributed approach based on Bayesian interpretation, which effectively solved the multi-robot patrolling problem with scalability and fault-tolerance. Acevedo et al. [8] described an approach for patrolling missions in irregular-shaped areas with heterogeneous aerial vehicles.

Due to the complexity of actual situations, various studies on reasoning coordination and cooperation between multiple robots were conducted. Hennes et al. [10] provided a collision avoidance system for multiple robots based on a velocity obstacle paradigm. Dinnissen et al. [12] developed an algorithm capable of deciding when and how the maps should be merged for solving the multi-robot simultaneous localization and mapping problem using reinforcement learning. Korsah et al. [13] proposed a taxonomy that handles the issues of interrelated utilities and constraints for task allocation problems. Liu and Shell [14] also introduced a dynamic approach for realizing large-scale partitioning.

In previous work, Yoneda et al. [16] proposed AMTDS, which is the autonomous reinforcement learning of the meta-strategy to decide the target decision strategies for coordination. With this method, agents investigate different strategies and individually identify the most effective ones to achieve perfect quality. Then, they improved the method by introducing self-monitoring to avoid performance degradation due to over-selection and make the method more practical. Sugiyama et al. [17] revised the method by incorporating environmental learning so that agents can perform cleaning tasks without knowledge of the dirty regions in advance. Sugiyama et al. further integrated simple negotiation for task allocations to prompt division of labor [18] and enabled agents to learn the appropriate activity cycle length [19]. However, energy consumption was not taken into consideration in the studies mentioned above, so agents made an all-out effort and focused on carrying out the tasks perfectly by disregarding energy efficiency.

Regarding the issue of energy conservation, only a few studies about multi-agent systems have handled it, while none of them dealt with the CCPP. Mei et al. [15] presented an energy-efficient motion planning approach for robot exploration, which selects the next target node based on orientation information and reduces repeated coverage. In contrast, this paper discusses on energy efficiency strategies from the viewpoint of algorithms for continuous patrolling problems. Consider practical requirements and the trade-off between level of perfection and energy efficiency, we proposed a series of behavioral strategies with the purpose of saving energy by avoiding unnecessary movement.

3. Model Description

We focus on the CCPP model, in which multiple autonomous agents move around the work environment and visit locations with required and different frequencies for given purposes. As the main purpose of this research, agents are expected to minimize the energy cost under the premise of satisfying given requirements. In addition, it is necessary for them to periodically return to charging bases and recharge to ensure continuous patrolling.

3.1. Environment

The environment in which agents move and work is described by graph $G = (V, E)$, where $V = \{v_1, ...v_m\}$ is the set of nodes with coordinates $v = (x_v, y_v)$, and E is the set of edges. The length of each edge in E can be assumed to be one by adding dummy nodes if necessary. We introduce a discrete time unit called *tick*. In one tick, events occur on nodes, agents individually decide their action plan, and they can move to one of the neighboring nodes along the edges and then work on the nodes they visit.

Each node has a value of *event occurrence probability* denoted as P_v for node $v \in V$. In the case of area cleaning tasks, an event corresponds to the accumulation of dirt, so P_v represents the probability that one piece of dirt has accumulated at v per tick. In the case of security patrolling tasks, an event corresponds to the appearance of enemies or suspicious events, so P_v indicates the probability with which something dangerous has happened and the security level has increased at v per tick. The number of unhandled events on v at time t can be expressed as $L_t(v)$, which is updated based on P_v every tick by

$$L_t(v) \leftarrow \begin{cases} 0 & \text{if an agent has visited } v \text{ at } t, \\ L_{t-1}(v) + 1 & \text{if an event occurs with probability } P_v \text{ at } t, \\ L_{t-1}(v) & \text{otherwise.} \end{cases} \quad (1)$$

3.2. Agent

Let $A = \{1, ..., n\}$ be a set of agents, and $v^i(t) \in V$ be the position of agent $i \in A$ at time t. For simplicity, we assume that the agents know the structure of the environment, their own position and others' positions, and that multiple agents staying at the same node is allowed. An environment with these assumptions can be realized by applying algorithms for map creation [11], map merging [12], and collision avoidance [9, 10] and by equipping agents with indicators, such as infrared emission and reflecting devices. While the assumption that agents are given the knowledge of the environment in advance might restrict the range of applications, we will improve the method to handle the learning of environment in our future work. Nevertheless, sophisticated coordination should be avoided because agents may have restricted resources including limited CPU power and battery capacity. Each agent decides its action plans based on local view and shallow coordination, by which the agent can only exchange superficial data but does not acquire deep knowledge such as other agents' plans, long-term targets, and learned knowledge.

In this paper, agents are given an event occurrence probability $\{P_v | v \in V\}$, but do not know the actual value of $L_t(v)$. However, they can estimate it by calculating the expected value, $EL_t(v)$, from P_v and t^v_{visit}, which is the most recent time an agent (it may not be i) visited and dealt with the event on v. Agents can know t^v_{visit} since they have exchanged their positions with others following the above assumption. $EL_t(v)$ at any future time t is defined by

$$EL_t(v) = P_v \cdot (t - t^v_{visit}). \tag{2}$$

Note that even if agents do not know P_v for $v \in V$, they can learn through experience [17].

The battery in agent i is denoted by $B^i = (B^i_{max}, B^i_{cons}, k^i_{charge})$, where $B^i_{max} > 0$ is the maximal capacity of the battery, $B^i_{cons} > 0$ is the amount of battery consumption per tick, and parameter $k^i_{charge} > 0$ indicates the speed of charge. Let $b^i(t)$ represent the remaining battery capacity in i at time t. When i moves, $b^i(t)$ is updated by

$$b^i(t+1) \leftarrow b^i(t) - B^i_{cons} \tag{3}$$

every tick. When i charges its battery at the charging base, v^i_{base}, the required time for a full charge starting from t is proportional to the amount of battery consumption:

$$T^i_{charge}(t) = (B^i_{max} - b^i(t))/k^i_{charge}. \tag{4}$$

We assume that agents consume B^i_{cons} every time they move, regardless of the number of handled events. Accordingly, the amount of energy consumption by agent i from time $t-1$ to time t is defined by

$$E_t(i) = \begin{cases} 0 & \text{if } i \text{ is charging or stays at the same place at } t, \\ B^i_{cons} & \text{otherwise.} \end{cases} \tag{5}$$

The parameters B^i_{max}, B^i_{cons}, and k^i_{charge} can be independent of i, but they are assumed to be the same in this paper for simplicity. According to the above assumption, periodical return to charging bases is required for agents to ensure continuous patrolling, meaning that they must return to v^i_{base} before $b^i(t)$ becomes zero.

3.3. Planning in Agents

Planning in agents can be divided into two stages: *target decision* and *path generation*. The agent decides the target node, $v^i_{tar} \in V$, in the former stage and generates the appropriate path from the current node to v^i_{tar}. There are lots of algorithms to determine targets and paths. We use several simple strategies as proposing planning algorithms was not part of our main purpose.

3.3.1. Target Decision Strategies

Agent i decides v^i_{tar} based on (1) on which node the largest number of events is expected to occur and (2) which node is unlikely to be visited by other agents in a short amount of time. We extend the AMTDS [16], with which each agent learns to identify the appropriate strategy based on Q-learning from the following four strategies.

Random Selection (R) Agent i randomly selects v^i_{tar} from V.

Probabilistic Greedy Selection (PGS) Agent i selects one of the nodes with the highest expected number of unhandled events. Let $V^t_g \subset V$ be the set of $N_g > 0$ nodes with the highest values of $EL_t(v)$ for time t. i randomly selects v^i_{tar} from V^t_g, where randomness is introduced to avoid a high concentration of the targets selected by multiple agents.

Repulsive Selection (RS) Agent i selects the node with the longest summative distance from all agents. Let V^i_{rep} be the set of $N_{rep} > 0$ nodes that i randomly selected from V and $d(v_i, v_j)$ be the length of the minimum path between v_i and $v_j \in V$. Then, v^i_{tar} is decided as

$$v^i_{tar} = \arg\min_{v \in V^i_{rep}} \sum_{i \in A} d(v^i(t), v). \tag{6}$$

Balanced Neighbor-Preferential Selection (BNPS) BNPS is an advanced version of PGS. The basic idea is that if agent i estimates that there are nodes with higher values of $EL_t(v)$ in the neighborhood using the learned threshold, i selects v^i_{tar} from those nodes. Otherwise, i selects v^i_{tar} using PGS. Please refer to Yoneda et al.'s study [16] for a detailed explanation.

3.3.2. Path Generation Strategy

Before agent i generates the path to v^i_{tar}, it checks $b^i(t)$ in advance to confirm whether v^i_{tar} is reachable. Otherwise, i changes v^i_{tar} to v^i_{base} and then generates a path to return and charge its battery. Agent i uses the *gradual path generation* (GPG) method as path generation strategy. In general, agents move along the shortest path, but if there are nodes with larger number of unprocessed events near the path, agents with GPG go to these nodes and deal with these unprocessed events. Since an explanation of GPG is beyond the scope of this paper, please refer to [16] for more details. We chose this method as previous research [16] has shown that GPG always outperforms the simple shortest path strategy.

3.4. Performance Measures

Our purpose is to minimize the overall energy consumption on the premise of maintaining the requirements for quality, which is equal to the sum of unhandled events in the work environment. Therefore, we evaluate the proposed methods in two aspects: *cumulative existence duration of unhandled events*, D_{t_s,t_e}, and *total energy consumption*, C_{t_s,t_e}, for the interval from t_s to t_e. The former can have distinct definitions depending on the characteristics of application so that

$$D_{t_s,t_e} = \begin{cases} \sum_{v\in V} \sum_{t=t_s+1}^{t_e} L_t(v) & \text{in area cleaning tasks,} \\ \min_{v\in V, t_s<t\le t_e} L_t(v) & \text{in security patrol applications.} \end{cases} \quad (7)$$

Besides, C_{t_s,t_e} is defined in the same way for all types of applications by

$$C_{t_s,t_e} = \sum_{i\in A} \sum_{t=t_s+1}^{t_e} E_t(i). \quad (8)$$

Although smaller values of these measures are better, there is a trade-off between them. In our energy-efficient CCPP model, agents are expected to cooperatively carry out the tasks to the required extent with less energy. Given a value $D^{e,|A|}_{req}$, which is the requirement level of D_{t_s,t_e} for environment type e, instead of minimizing D_{t_s,t_e}, agents aim to minimize C_{t_s,t_e} and make D_{t_s,t_e} small enough to satisfy the condition $D_{t_s,t_e} \le D^{e,|A|}_{req}$.

4. Proposed Methods

Our proposed methods is called while agents execute their actions according to the generated plans. First, we describe the methods for estimating requirements and evaluating self-importance, with which agents decide the next action by taking into account the status of the environment and themselves. Next, we propose two behavioral strategies taken by agents as a substitute for moving toward the next target with the intention of decreasing the energy cost. Finally, we present a variation of the previous method AMTDS [16]. Fig. 1 shows an overview of the process of action selection in agents with the proposed methods.

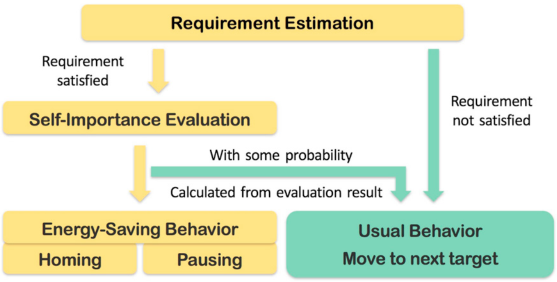

Fig. 1. Behavior selection in agents.

4.1. Requirement Estimation

To decide the next behavior, it is necessary for agents to know the current status of the environment. Each agent independently estimates the total number of unhandled events and then judges whether the given requirement is satisfied. For agent i at time t and in environment e, i randomly generates $V_{rand}(v^i(t))$, which is the set of N_{range} nodes it has visited, where N_{range} is a positive integer indicating the number of reference nodes. A larger value of N_range gives a more accurate estimation result, but also requires more expensive computational resources.

The estimated value, EV_t^i, is obtained the average value of $EL_t(v)$ in $V_{rand}(v^i(t))$:

$$EV_t^i = \frac{\sum_{v \in V_{rand}(v^i(t))} EL_t(v)}{N_{range}}. \tag{9}$$

When EV_t^i is smaller than the requirement level $D_{req}^{e,|A|}$, i assumes that the requirement has been achieved. It then decides the next action based on the result of *self-importance evaluation*, which is explained in the following section. Otherwise, i selects the next target node with one of the target decision strategies and generates a path to the destination.

4.2. Self-Importance Evaluation

Before executing next action, each agent evaluates its self-importance, which expresses its contribution degree, to understand how important it will be for the system. An agent considers its importance by taking into account (1) its recent performance and (2) whether it finds the important regions and is possible to process a large number of events in the subsequent behavior. Agent i evaluates its self-importance, $Imp^i(t)$, by comparing U_p^i (including U_s^i and U_l^i), its performance in the past from the short- and long-term viewpoints, and U_f^i, its expected performance in the near future. U_p^i and U_f^i are defined as u_{t_s,t_e}^i, the actual or expected number of handled events per tick from time t_s to time t_e:

$$U_p^i = u_{t_0,t_c}^i = \frac{\sum_{t_0 < t \le t_c} L_t(v^i(t))}{t_c - t_0}, \; t_0 = \begin{cases} t_c - T_s & \text{short term} \\ t_c - T_l & \text{long term} \end{cases} \tag{10}$$

$$U_f^i = u_{t_f,t_c}^i = \frac{\sum_{t_c < t \le t_f} EL_t(v^i(t))}{t_f - t_c}, \tag{11}$$

where t_c is the current time, T_s and T_l ($T_s < T_l$) are fixed integers, t_f is the future time when i arrives at the next target, $L_t(v^i(t))$ is the number of events dealt with by i at time t, and $EL_t(v^i(t))$ is the expected number of dealt with events at future time t. Note that to calculate U_f^i, v_{tar}^i is decided in advance. Then, $Imp^i(t)$ is obtained by

$$Imp^i(t) = \begin{cases} \frac{U_s^i + U_f^i}{U_l^i} & \text{if } U_s^i + U_f^i \le U_l^i, \\ 0 & \text{if } U_l^i = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{12}$$

4.3. Energy-Saving Behaviors

We propose two behavioral strategies aim to reduce energy, named *homing* and *pausing*, that agents adopt instead of moving to the next node. An agent will have a chance to perform these behaviors only when it supposes that the given requirement is met based on the local estimation. Note that only one of the two energy-saving strategy was applied during one experimental simulation. The probability of performing the energy-saving behavior is calculated from the result of self-importance evaluation by

$$P^i(t) = 1 - Imp^i(t). \tag{13}$$

4.3.1. Homing Behavior

Every time after agent i continuously moves for $T_{check} > 0$ ticks, it conducts requirement estimation and self-importance evaluation to decide whether to perform the homing behavior. In addition, i checks the remaining capacity of its battery $b^i(t)$ so that the action will only be taken under the constraint $b^i(t) < k_{homing} \cdot B_{max}^i$. The constraint is added to prevent agents from frequently returning to the charging base before they travel far away. By performing the homing behavior, i immediately sets v_{tar}^i to v_{base}^i to go back and charge its battery. On its way back, i keeps processing events but does not conduct requirement estimation or self-importance evaluation.

4.3.2. Pausing Behavior

Unlike the homing behavior, requirement estimation for the pausing behavior is conducted every time after agent i has its battery fully charged. According to the results, i decides whether to perform the pausing behavior. If it decides to do so, i simply stays at v^i_{base} for $T_{pausing} > 0$ ticks without processing any events. There is no limitation on the number of pausing behavior performed consecutively, which means that it is possible for an agent to adopt pausing behavior again and again based on local estimation results.

4.4. Autonomous Strategy Selection

As our main purpose is reducing the overall energy cost, we extend AMTDS and call the new method *AMTDS for energy saving and cleanliness* (AMTDS/ESC). With the extended method, agents choose appropriate target decision strategies from the number of handled events per unit of energy using reinforcement learning. A larger number of handled events and a smaller energy cost are preferred.

Suppose that agent i selects v^i_{tar} with strategy s, where s is one of the target decision strategies described in the previous section. After i moves to v^i_{tar} along the path generated by GPG, it calculates the reward of s by

$$r^i_{t_0,t_0+d_{travel}} = \frac{\sum_{t_0<t\leq d_{travel}} L_t(i) / \sum_{t_0<t\leq d_{travel}} E_t(i)}{d_{travel}}, \tag{14}$$

where d_{travel} is the length of travel from time t_0 when i started until the time it arrived at its target. Subsequently, the Q-value of s is updated as

$$Q(s) \leftarrow (1-\alpha)\cdot Q(s) + \alpha \cdot r^i_{t_0,t_0+d_{travel}}. \tag{15}$$

As noted above, to gain higher rewards, agents choose the strategy with which they can handle more events with less energy. Therefore, agents with AMTDS/ESC learn to select the strategy that minimizes the energy cost and maximizes the number of handled events per tick at the same time. In addition, the ε-greedy method is used during learning.

5. Experimental Evaluation

The proposed methods are evaluated in a simulation environment similar to that in our previous work [16]. We introduce a series of behavioral strategies to the process of plan creation in agents and experimentally show that the methods enable agents to cooperatively reduce energy consumption while satisfying the given requirements. We only show the results of area cleaning application by multiple autonomous agents due to page limitation.

5.1. Experimental Setting

Table 1. Parameters for target decision strategies.

Methods	Parameters	Values
PGS	N_g	5
RS	N^i_{rep}	100
BNPS	α	0.1
	d_{th}	15
AMTDS/ESC	α	0.1
	ϵ	0.05

Four environments with different characteristics are prepared, as shown in Fig. 2, to investigate the behavior of agents. Each environment is represented by a two-dimensional grid graph, where G is defined as a 101×101 grid. A node v can be expressed by (x, y), where $-50 \leq x, y \leq 50$. The charging bases for all agents are located in the middle of each environment so that $\forall i \in A : v_{base} = v^i_{base} = (0, 0)$, and multiple agents can charge simultaneously. In some environments, there exist regions where dirt accumulates easily and thus are considered important, so agents would like to focus on visiting them. The coordinates, shapes, and P_v for $v \in V$ of these regions are outlined in Fig. 2. Dirt accumulates uniformly in Env. (a). On the contrary, the areas near the walls in Env. (b) and in the independent blocks scatter in Env. (d) are dirtier, meaning that dirt easily accumulates. Finally, Env. (c) is the most complicated environment that has both of the characteristics.

We deployed 20 agents in each environment, and all of them are assumed to be homogeneous: they use the same path generation strategy (GPG) and utilize one of the five target decision strategies (R, PGS, RS, BNPS, and

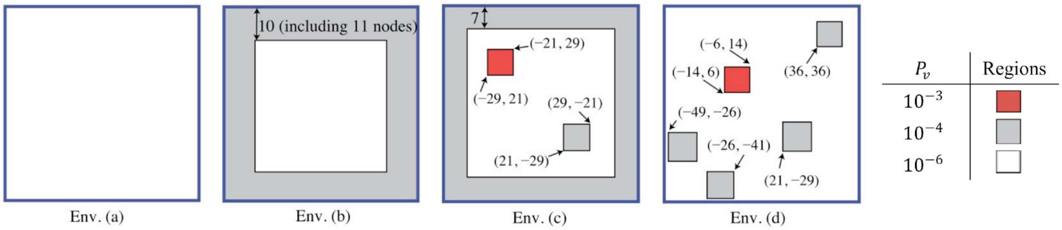

Fig. 2. Experimental environments [16]

AMTDS/ESC). With strategies other than AMTDS/ESC, agents always adopt a single target decision strategy. In contrast, agents with AMTDS/ESC independently select one from R, PGS, RS, and BNPS based on local learning. Note that in all experiments, agents are given $\{P_v | v \in V\}$ in advance. The parameter values used in target decision strategies are listed in Table 1. These values were determined by taking into account the size of the experimental environments and the number of agents but are not optimal.

The battery specifications of all agents are set as $(B_{max}, B_{cons}, k_{charge}) = (2700, 3, 1)$ in all experiments. As a result, an agent could continuously operate for up to 900 ticks and requires 2700 ticks for a full charge when the battery is running out of power, which makes the maximum cycle of operation and charge 3600 ticks. Therefore, when all the agents constantly work to make full use of their batteries without performing any energy-saving behavior, the theoretical maximal value of C_{t_s,t_e} per tick will be around 15. In the following experiments, we compare the resulting C_{t_s,t_e} to the theoretical maximal value for the purpose of evaluating the proposed methods.

To reduce the overall energy cost, agents estimate the status of environment and importance of themselves. According to the results, agents perform either the homing or pausing behavior instead of heading toward the next target under certain circumstances. The decision of action selection is individually made by each agent with their local viewpoints. We compare the values of D_{t_s,t_e} and C_{t_s,t_e} every 100 ticks. The experimental results below are the averages of five independent trials with different random seeds, and the length of each trial is 500,000 ticks.

5.2. Energy-Efficient Strategies

We compared the performance of three agent behavioral regimes. The first experiment is the control experiment, in which agents only take usual actions and ignore energy efficiency. In the second experiment, there are chances for agents to perform the homing behavior instead of exploring the environment so that they stop patrolling and return to the charging base and charge. In the third experiment, agents perform the pausing behavior with some probability after their batteries are fully charged so that they take a rest instead of leaving the charging base for patrolling.

The values of parameters for requirement estimation, self-importance evaluation, and energy-saving action are listed in Table 2. The quality requirements are determined under the principle of picking a value lower than that when agents make an all-out effort to work so that they are expected to take a rest appropriately for saving energy. Note that the values of all these parameters were determined by considering the size of experimental environments, the number of agents, and computational cost but are not optimal.

Table 2. Parameters for energy-efficient strategies.

Methods	Parameters	Values
Requirement Estimation	N_{range}	100
	$D^{a,20}_{req}$	45
	$D^{b,20}_{req}$	600
	$D^{c,20}_{req}$	400
	$D^{d,20}_{req}$	110
Self-Importance Evaluation	T_s	20
	T_l	50
Homing Behavior	T_{check}	100
	k_{homing}	$\frac{1}{3}$
Pausing Behavior	$T_{pausing}$	20

The performance measures of the total dirt amount for each environment with different agent behavioral regimes are shown in Fig. 3, where the ratio is the one of actual value D_{t_s,t_e} to the given requirement $D^{e,20}_{req}$. Note again that the smaller performance values are better in this figure. The red dotted lines represent the given requirements of cleanliness, which are set as benchmarks. The experimental results of the overall energy consumption are shown in Fig. 4, where energy consumption ratio is the one of actual consumption C_{t_s,t_e} to the theoretical maximal value.

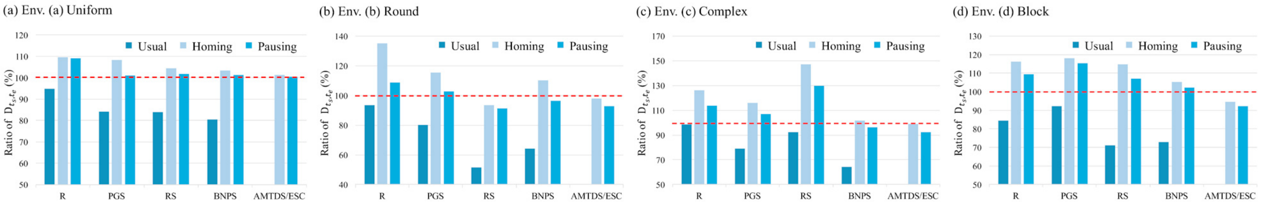

Fig. 3. Values of cumulative existence duration of dirt relative to given requirements.

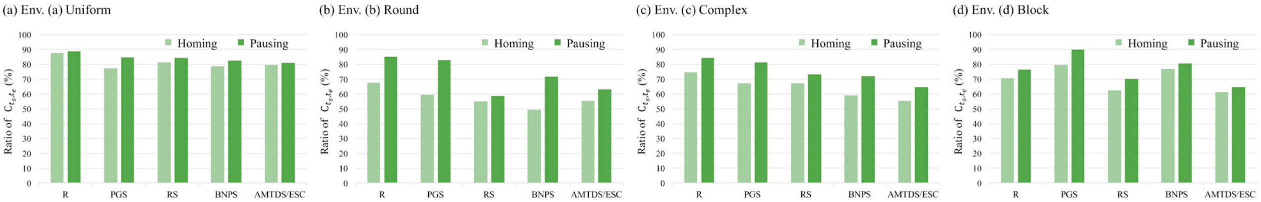

Fig. 4. Values of total energy consumption relative to theoretical values.

The results indicate that the proposal energy-efficient strategies successfully saves overall energy consumption while agents could meet the given requirement of total amount of remaining dirt in all circumstances. As shown in Fig. 4, the total energy consumption is reduced about 20%-50% with the homing behavior and 10%-35% with the pausing behavior. In respect of energy efficiency, agents in non-uniform environments give better performance compared to those in the uniform environment. A conceivable reason might be that the dirty nodes cause a significant difference between the dirt amount in different regions. As a result that in these environments, some agents find the dirty regions and move around to clean them, while others could not and choose to perform the energy-saving behavior. Furthermore, the performance of agents with single strategy varies with characteristics of the environment. Within single strategy regimes, RS performs the best in Env. (b), and BNPS gives better performance in Envs. (c) and (d). By contrast, agents with AMTDS/ESC give stable performance in all environments.

A key observation is that within the two energy-saving behaviors, the homing behavior always outperforms the pausing behavior in terms of lower values of energy consumption, and the values of the cumulative existence duration of dirt are closer to the given requirement levels. Since we set $T_{check} = 100$, requirement estimation is conducted every 100 ticks. During one operation cycle of an agent, requirement estimation could be conducted up to 9 times when adopting the homing behavior, while estimation could only be conducted after charging when adopting the pausing behavior. For this reason, agents with homing behavior have a higher chance of executing energy-efficient plan.

On the other hand, when comparing the two performance measures, we found that the proposed method of meta-strategy AMTDS/ESC always outperforms other single target decision strategy regimes, and more importantly, agents using the method are able to cooperatively satisfy the given requirements of cleanliness in all cases. Agents with the meta-strategy select the appropriate target decision strategies locally and respectively, which allows them to cooperatively clean the environment by adopting strategies with different characteristics. In prediction, the agents with the proposal AMTDS/ESC considerably reduce the amount of energy consumption as well as the remaining dirt. This also means that there is still room for improvement since we expect the requirements to be strictly met. In particular, for the pausing behavior in Envs. (b), (c), and (d), the values of dirt cumulative existence duration were 8% lower than the requirement, and thereby, agents could rest more to save more energy.

6. Conclusion and Future Work

By aiming to reduce of energy consumption in multi-agent CCPPs, we extended our previous learning method of target decision strategies. On top of that, we proposed a series of behavioral strategies for agents to independently estimate the current status of the work environment and understand their contribution degree. Based on the local estimation and self-evaluation results, agents choose to remain performing their usual behavior or to adopt energy-saving plans, including returning to the charging base (homing) or taking a break (pausing). The experimental results demonstrated that the proposed methods enable agents to reduce the energy cost while cooperatively maintaining the given requirements of quality perfection. Within the five target decision strategies, AMTDS/ESC was able to give the best performance in respects of cleanliness and energy consumption.

Our future plan covers several different aspects including learning of the environment, combination of energy-saving behaviors, and improvement of agent importance evaluation. In this paper, we assume that agents know the structure and event occurrence probability distribution of the environment. Nevertheless, this assumption restricts the range of applicable domain of the proposed method from large scale networks. Hence, we plan to improve the method by introducing learning of environment structure and event occurrence probability to make the method more practical. On the other hand, we believe that the combination of homing and pausing behaviors can achieve better performance, therefore we plan to integrate the two strategies as well as learning of parameters such as the length of pausing time and check interval. Moreover, the charging bases for all agents are located in the same place in this paper, whereas letting the charging bases be scattered over the environment or setting up multiple charging bases might make it easier for agents to perform energy-saving behavior. In our research plan, we will also focus on enabling agents to autonomously and individually evaluate their importance with regard to the system from their recent contribution and performance. With this functionality, a continuous system can eliminate old robots and introduce new ones without affecting the overall performance.

References

[1] Chen, Z., Lin, L., and Yan, G. (2001) “An Approach to Scientific Cooperative Robotics through MAS (multi-agent system)” *Robot* **23** (4): 368-373.

[2] Iocchi, L., Marchetti, L., and Nardi, D. (2001) “Multi-robot patrolling with coordinated behaviours in realistic environments” *IEEE IROS*: 2796-2801.

[3] Portugal, D., and Rocha, Rui P. (2013) “Distributed multi-robot patrol: A scalable and fault-tolerant framework” *Robotics and Autonomous Systems* **61** (12): 1572-1587.

[4] Ahmadi, M., and Stone, P. (2005) “Continuous area sweeping: A task definition and initial approach” *Proc. of the 12th International Conference on Advanced Robotics*: 316–323.

[5] Ahmadi, M., and Stone, P. (2006) “A multi-robot system for continuous area sweeping tasks” *Proc. of the 2006 IEEE International Conference on Robotics and Automation*: 1724– 1729.

[6] Moreira, D.M., Ramalho, G., and Tedesco, P.C. (2009) “SimPatrol - Towards the Establishment of Multi-agent Patrolling as a Benchmark for Multi-agent Systems” *Proceedings of the International Conference on Agents and Artificial Intelligence*: 570-575.

[7] Santana, H., Ramalho, G., Corruble, V., and Ratitch, B. (2004) “Multi-Agent Patrolling with Reinforcement Learning” *Proc. of the Third International Joint Conference on Autonomous Agents and Multiagent Systems* **3**: 1122–1129.

[8] Acevedo, J. J., C., B., Maza, I., and Ollero, A. (2013) “Distributed Approach for Coverage and Patrolling Missions with a Team of Heterogeneous Aerial Robots under Communication Constraints” *International Journal of Advanced Robotic Systems* **10** (1), pp. 28(1)–28(13).

[9] Cai, C., Yang, C., Zhu, Q., and Liang, Y. (2007) “Collision Avoidance in Multi-Robot Systems” *Proc. of the 2007 IEEE International Conference on Mechatronics and Automation*: 2795-2800.

[10] Hennes, D., Claes, D., Meeussen, W., and Tuyls, K. (2012) “Multi-robot collision avoidance with localization uncertainty” *Proc. of the 11th International Conference on Autonomous Agents and Multiagent Systems* **1**: 147–154.

[11] Hahnel, D., Burgard, W., Fox, D., and Thrun, S. (2003) “An efficient Fast SLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements” *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* **1**: 206–211.

[12] Dinnissen, P., Givigi, S., and Schwartz H. (2012) “Map merging of multi-robot slam using reinforcement learning” *IEEE International Conference on SMC*: 53–60.

[13] Ayorkor Korsah, G., Stentz, A. and Bernardine Dias, M. (2013) “A comprehensive taxonomy for multi-robot task allocation” *The International Journal of Robotics Research* **32** (12): 1495–1512.

[14] Liu, L., and Shell, D.A. Auton Robot (2012) “Large-scale multi-robot task allocation via dynamic partitioning and distribution” *Autonomous Robots* **33** (3): 291-307.

[15] Mei, Y., Lu, Y.-H., Lee, C., and Hu, Y. (2006) “Energy-efficient mobile robot exploration” *Proc. of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*: 505-511.

[16] Yoneda, K., Sugiyama, A., Kato, C., and Sugawara, T. (2015) “Learning and relearning of target decision strategies in continuous coordinated cleaning tasks with shallow coordination” *Web Intelligence* **13** (4): 279–294.

[17] Sugiyama, A., and Sugawara, T. (2015) “Meta-strategy for cooperative tasks with learning of environments in multi-agent continuous tasks” *Proc. of the 30th Annual ACM Symp. on Applied Computing*: 494–500.

[18] Sugiyama, A., Sea, V., and Sugawara, T. (2015) “Effective Task Allocation by Enhancing Divisional Cooperation in Multi-Agent Continuous Patrolling Tasks” *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*: 33–40.

[19] Sugiyama, A., Wu, L., and Sugawara, T. (2019) “Learning of Activity Cycle Length based on Battery Limitation in Multi-agent Continuous Cooperative Patrol Problems” *Proc. of the 11th International Conference on Agents and Artificial Intelligence* **1**: 62–71.