ISA Transactions 102 (2020) 208–220

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Research article

Potential game for dynamic task allocation in multi-agent system

Han Wu[a], Huiliang Shang[a,b,*]

[a] Research Center of Smart Networks and Systems School of Information Science and Engineering, Fudan University, Shanghai 200433, PR China
[b] Academy for Engineering and Technology, Fudan University, Shanghai 200433, PR China

ARTICLE INFO

Article history:
Received 13 June 2019
Received in revised form 3 March 2020
Accepted 5 March 2020
Available online 7 March 2020

Keywords:
Dynamic task allocation
Multi-agent system
Game theory
Log-linear learning

ABSTRACT

This paper proposes a novel distributed multi-agent dynamic task allocation method based on the potential game. Consider that the workload of each task may vary in a dynamic environment, and the communication range of each agent constrains the selectable action set. Each agent makes the decision independently based on the local information. Firstly, a potential game-theoretic framework is designed. Any Nash equilibrium is guaranteed at least 50% of suboptimality, and the best Nash equilibrium is the optimal solution. Furthermore, a time variant constrained binary log-linear learning algorithm is provided and the global convergence is proved under certain conditions. Finally, numerical results show that the proposed algorithm performs well in terms of global searching ability, and verify the effectiveness of the distributed dynamic task allocation approach.

1. Introduction

Cooperative task allocation plays a vital role in cooperative control in the field of multi-agent systems. It raises fundamental and novel problems in control theory and robotics and brings more significant challenges when it comes to a real-time assignment in a dynamic environment [1]. Generally, the cooperation of a swarm of agents could perform more complex tasks that a single agent can hardly achieve, for instance, cooperative radar jamming [2], surveillance [3], weapon-target assignment [4], and disaster management [5,6], etc. Driven by these demands, multiple agents must coordinate intelligently for successful task allocation.

How to cooperate intelligently to achieve an overall goal is crucial. The complexity of multi-agent dynamic task allocation mainly comes from the large cardinality of the systems and environmental uncertainty. Therefore, it is necessary to entrust individuals certain levels of decision-making such that they can adaptively adjust the assignment to finish the tasks efficiently. For this reason, this paper focuses on the general dynamic task allocation problem, where agents are allowed to change their choices in response to environmental change or the actions of other agents to obtain the desired state [1]. In such a single-task robot and multi-robot task (ST-MR) problem [7], an agent can only participate in at most one task while each task may require the combined effort of multiple agents, and the objective is to minimize the task completion time with limited computation resources. The motivation of this research is to design a distributed method to solve the dynamic ST-MR problem, and provide some theoretic supports for addressing applications such as surveillance, disaster management, etc.

As it is known that multi-agent task allocation is an NP-hard problem [7], and the computational requirement increases exponentially with the number of agents and tasks. Moreover, the problem becomes more complicated when the workload of each task may vary in a dynamic environment. An efficient distributed decision-making framework is expected as it has better robustness and scalability than centralized systems. However, the convergence performance of distributed decision-making systems may not as well as the centralized one; it is hard to guarantee global convergence, such as the market-based approaches [8,9]. Thus, the core challenge for distributed decision-making systems is to design the proper individual control policies which contribute to the global objective.

In this paper, a novel distributed multi-agent task allocation framework for the dynamic environment is designed based on the potential game. The potential game plays a critical role in engineering multi-agent systems and is formally defined by Monderer & Shapley [10]. The advantages of the potential game-theoretic framework include: (1) There are efficient approaches of designing the local decision rules [11,12] ; (2) Several learning algorithms, such as log-linear learning [13], fictitious play [14], and regret matching [15], etc., ensure to converge to Nash equilibrium (NE) in the potential game. The potential game model is designed at first. The single task reward function is defined as the submodular set function [16] with considering the properties of the task, and the global objective function is the discounted sum of all tasks' reward. Also, the agent individual utility function is

* Correspondence to: Room 301, Science building, Fudan University, 220 Handan Rd., Yangpu District, Shanghai 200433, PR China.
E-mail addresses: 17210720044@fudan.edu.cn (H. Wu), shanghl@fudan.edu.cn (H. Shang).

https://doi.org/10.1016/j.isatra.2020.03.004
0019-0578/© 2020 ISA. Published by Elsevier Ltd. All rights reserved.

designed as the normalized marginal contribution of global function according to the Wonderful Life Utility designing rule [17]. In the proposed interaction framework, each agent communicates with local agents and receives little information to formulate its decision. Furthermore, this paper also proves that any NE is guaranteed at least 50% of suboptimality, which is the state of being suboptimal, and the best NE is the global function maximizer.

Next, a time variant constrained binary log-linear learning (TVCBLL) algorithm is provided, where β is a time variant parameter, and the communication range constrains the action set of each agent. This paper proves that the TVCBLL converges to the global optimal solutions with probability one under certain conditions. Finally, numerical experiments validate the proposed method.

This paper is organized as follows. In the next section, some approaches for distributed multi-agent task allocation problems are reviewed. Section 3 introduces the game-theoretic model and proves the efficiency of the Nash equilibrium. Section 4 provides the time variant constrained binary log-linear learning algorithm and analyzes its convergence performance. Then, numerical experiments show that the TVCBLL performs well in terms of global searching ability and validate the proposed method in Section 5. Finally, Section 6 presents the summary and expectations.

2. Related works

Generally, the task allocation methods can be divided into three types: optimization-based approaches [18], market-based approaches [19], and game-theoretic approaches [20]. For the optimization-based task allocations approaches, a distributed version of the Hungarian method was proposed to solve the multi-robot routing and multi-robot orchestral problems [21]. Besides, Attiya et al. [22] proposed a simulated annealing based task allocation approach in heterogeneous distributed systems. In [23], a genetic algorithm, combined with some common heuristics, was designed for the dynamic task allocation. Although the optimization-based methods have an excellent performance of exploration and are widely used in many areas, it is hard to design appropriate local decision rules [19]; therefore, most optimization-based approaches are used for centralized systems.

Auction algorithm, a typical market-based approach, is an iterative procedure, where multiple bids of agents are compared to determine the best offer, with the final deals going to the highest bidders [9]. A consensus-based auction algorithm and a bundle auction algorithm are proposed for solving the task allocation problem of a fleet of autonomous robots [24]. Then, Lee et al. [8] designed an improved auction algorithm in the dynamic environment where considering the limited agent communication range. Although market-based approaches have good robustness and scalability, there exist some drawbacks, such as (i) the lack of efficient designing methods of individual control policies, and (ii) the deficient performance when introducing the necessary negotiations and penalty schemes.

Recently, plenty of research works on distributed game-theoretic decision-making have sprung up. From the perspective of whether an external authority exists to enforce rules, games can be divided into cooperative games and non-cooperative games. For the cooperative game, the Hedonic coalition formation game is first applied to solve the task allocation problem among many autonomous agents [25]. Jang et al. [26] proposed a novel framework in asynchronous environments based on hedonic games, which has good scalability and at least 50% of suboptimality, while this method cannot guarantee global convergence. For the non-cooperative game, Chapman et al. [5] proposed a potential game framework and applied a distributed search algorithm to solve multi-agent dynamic task allocation problems; however, the efficiency of Nash equilibrium and convergence of this algorithm cannot be guaranteed. Li et al. [27] designed the coordinated motion as a multi-player potential game with constraint action sets to solve the multiple unmanned aerial vehicles' cooperative search and surveillance problem. Our research is similar to these above works and aims at addressing multi-agent dynamic task allocation by using game theory, while there are two critical problems need to be solved: (i) the inefficiency of the game model, and (ii) the lack of real-time and global optimization ability of learning algorithm.

Also, consensus control is feasible for dynamic task allocation. Liu et al. [28] proposed a distributed controller to achieve admissible consensus performances of the agents and restrain the external disturbances. In [29], a distributed coordination controller is developed for task allocation among multiple robots with limited communications. However, this paper will give an easier way to design the local decision protocol from the perspective of game theory instead of consensus dynamics.

Toward the global optimality of multi-agent task allocation in a dynamic environment, the main contributions of this paper are as follows.

1. A new multi-agent dynamic task allocation game-theoretic model is proposed based on the potential game. The single task reward function is defined as the submodular set function, and the agent utility function is designed as the normalized marginal contribution of global utility. Furthermore, this paper proves that the game model is efficient as it guarantees at least 50% of suboptimality, and the best NE is the optimal solution.
2. A time variant constrained binary log-linear learning (TVCBLL) algorithm is provided for the game model. The parameter β is time variant, and the communication range constrains the action set of each agent. This paper proves that the TVCBLL converges to the global optimal solutions with probability one under particular conditions.
3. Numerical experiments validate the TVCBLL algorithm and confirm that the proposed multi-agent dynamic task allocation method is efficient.

3. Game model

3.1. Task model

Suppose that there exist a set of n agents $\mathcal{N} = \{1, 2, \dots, n\}$ and a set of m tasks $\mathcal{S} = \{1, 2, \dots, m\}$, and an agent cannot take part in multiple tasks at the same time. Let $\{\mathcal{S}\}_i$ represent the task that agent $i \in \mathcal{N}$ chooses, and $\{\mathcal{I}\}_j$ denote the set of agents for task $j \in \mathcal{S}$. A task j is located in a fixed position pos_j, where $pos_{(\cdot)}$ represents the two-dimensional coordinate. Besides, the position of agent i, pos_i, is the same as the chosen task's position. Let $d_{ij} = \left\| pos_i - pos_j \right\|$ denote the Euclidean distance. Consider an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, and each node $v \in \mathcal{V}$ represents a task, $e \in \mathcal{E}$ is the edge between the connected nodes and the weight $w \in \mathcal{W}$ corresponding to the Euclidean distance. Any two tasks, which Euclidean distance is less than the communication radius Rad, are connected by an edge. Besides, the relevant symbols through out this paper are summarized in Table 1 for convenient.

The workload of task j, $h_j \geq 0$, is the amount of work needed to finish it by the combined effort of agents. And the work capacity of agent i is denoted by ω_i, which represents the amount of work that agent i can execute per time unit. The work capacity of a group for task j is defined as the sum of every single agent's work

Table 1
Nomenclature.

Symbol	Description
$\mathcal{N}$	A set of n agents
$\mathcal{S}$	A set of m tasks
$\{\mathcal{S}\}_i$	The task that agent i chooses
$\{\mathcal{I}\}_j$	The set of agents or the group for task j
ω_i	The work capacity of agent i
Rad_i	The communication radius of agent i
v	The speed of the agent
h_j	The workload of task j
Δt	The time unit
T_j	The completion time of task j
T	The completion time of all tasks, $T = \max\{T_1, T_2, \ldots, T_m\}$
$\mathcal{A}_i$	The action set of agent i
$\mathcal{A}$	The joint action set, $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$
τ	The discount factor that reflects the rate of change of task reward
θ	The discount factor of workload
R_j	The reward of task j
U	The global utility function
U_i	The agent utility function
Φ	The potential function
C_i^t	The constrained action set of agent i at time t
π	The stationary distribution

capacity in this group, $\sum_{i\in\{\mathcal{I}\}_j}\omega_i$. To complete a task j, the work capacity for the task must satisfy

$$\sum_{i\in\{\mathcal{I}\}_j}\omega_i - h_j \geq 0. \tag{1}$$

For a task $j \in \mathcal{S}$, it would be better to possess a proper number of agents, such that (1) is satisfied, and the resource waste caused by unreasonable assignment can be reduced at the same time. Then the reward of task j is designed as

$$R_j = -\tau^{\sum_{i\in\{\mathcal{I}\}_j}\omega_i - h_j}, \tag{2}$$

where $\tau \in (0,1)$ is a discount factor that reflects the rate of change of reward with the work capacity of the group $\{\mathcal{I}\}_j$, or in other words, it represents the timeliness of the task. As τ closes to 1, the reward keeps a certain amount of growth even if (1) holds, such that more agents are willing to participate in the task; thus, this task can be completed faster. However, reward R_j is insensitive when $\sum_{i\in\{\mathcal{I}\}_j}\omega_i - h_j \geq 0$ if $\tau \to 0$, which means no more agents would benefit from this task after (1) was satisfied.

It is straightforward that set function $R_j(\{\mathcal{I}\}_j)$ monotonically increases while its slope decreases, and the upper bound is zero. Notably, the growth rate of R_j drops sharply when $\sum_{i\in\{\mathcal{I}\}_j}\omega_i - h_j \geq 0$ satisfies, and this prevents too many agents from participating in one task and encourages agents to choose the tasks which do not satisfy (1). Therefore, the designed reward function has a natural diminishing returns property, called submodularity, which means the marginal contribution of a single agent to task utility decreases as the size of the set of agents increases. This property would contribute to achieving the desired collective behavior.

The completion time of single task $j \in \mathcal{S}$ is defined as $T_j = h_j/\sum_{i\in\{\mathcal{I}\}_j}\omega_i$. According to (2), it is evident that T_j can be reduced by increasing the group work capacity $\sum_{i\in\{\mathcal{I}\}_j}\omega_i$ since $h_j \geq 0$ is a constant, and then R_j increases. Therefore, minimizing the task completion time T_j is equivalent to maximizing the reward of task j.

For a general task allocation problem, which involves multiple tasks and agents, the time to complete all tasks, T, is determined by the time of the last completed task, that is

$$T = \max\{T_1, T_2, \ldots, T_m\}. \tag{3}$$

3.2. Global utility and agent utility

The objective of the task allocation problem is to find an optimal assignment that minimizes the completion time T. From Section 3.1, the single task completion time T_j can be minimized by maximizing the task reward R_j. However, the work capacity of all the agents is finite, and it is necessary to make full use of limited work capacity to accomplish all the tasks. From the above analysis, the problem can be defined as follow:

$$\min_{\{e_{ij}\}} \max\{T_1, T_2, \ldots, T_m\} \tag{4}$$

subject to

$$\sum_{j\in\mathcal{S}} e_{ij} \leq 1, \quad \forall i \in \mathcal{N}, \tag{5}$$

$$e_{ij} \in \{0,1\}, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{S}, \tag{6}$$

where e_{ij} is a binary decision variable that indicates whether or not agent i is assigned to task j, and $T_j = h_j/\sum_{i\in\{\mathcal{I}\}_j}\omega_i$ is the completion time of task j.

Since the time of the last completed task determines T, one always expects all tasks are finished as soon as possible rather than some part of them. Then, the total reward of all tasks is denoted by

$$\widetilde{U} = \sum_{j=1}^{m} R_j = -\sum_{j=1}^{m} \tau^{\sum_{i\in\{\mathcal{I}\}_j}\omega_i - h_j}. \tag{7}$$

Notice that R_j has the property of diminishing marginal returns for every $j \in \mathcal{S}$, and maximizing $\widetilde{U}$ requires each task possesses a proper number of participants. Let $x_j = \sum_{i\in\{\mathcal{I}\}_j}\omega_i - h_j$ and $\mathbf{x} = (x_1, \ldots, x_m)$, and then maximizing $\widetilde{U}$ defined in (7) is equivalent to minimizing $F(\mathbf{x}) = \sum_{j=1}^{m}\tau^{x_j}$. According to the inequality of arithmetic and geometric means, since $\tau \in (0,1)$,

$$\sum_{j=1}^{m}\tau^{x_j} \geq m\sqrt[m]{\tau^{x_1}\tau^{x_2}\cdots\tau^{x_m}} \tag{8}$$

with equality if and only if $x_1 = x_2 = \cdots = x_m$. Therefore, $F(\mathbf{x})$ is minimized if and only if $x_1 = x_2 = \cdots = x_m$. Assume that $x_j = \sum_{i\in\{\mathcal{I}\}_j}\omega_i - h_j = C$ is a constant for all $j \in \mathcal{S}$, then the completion time is

$$T_j = \frac{h_j}{\sum_{i\in\mathcal{I}_j}\omega_i} = \frac{h_j}{h_j + C}. \tag{9}$$

If $C = 0$, then $\sum_{i\in\{\mathcal{I}\}_j}\omega_i - h_j = 0$ for all $j \in \mathcal{S}$, and $T_1 = \cdots = T_m = 1 = T$; thus, all tasks are completed simultaneously per unit time. In this ideal case, an optimal assignment to minimize the completion time T can be obtained by maximizing $\widetilde{U}$ defined in (7).

For the general scenario that $\sum_{i\in\{\mathcal{I}\}_j}\omega_i - h_j = 0$ for all $j \in \mathcal{S}$ does not satisfy, one can define the discount factor of workload as $\theta = \sum_{i\in\mathcal{N}}\omega_i/\sum_{j=1}^{m}h_j > 0$, and the discounted workload is

$$(h_1', h_2', \ldots, h_m') = \theta(h_1, h_2, \ldots, h_m), \tag{10}$$

and then $\sum_{i\in\mathcal{N}}\omega_i = \sum_{j=1}^{m}h_j'$. Now the global utility U can be defined as the discounted sum of rewards,

$$U = \sum_{j=1}^{m} R_j' = -\sum_{j=1}^{m}\tau^{\sum_{i\in\{\mathcal{I}\}_j}\omega_i - h_j'}, \tag{11}$$

and the following claim holds.

Claim 1. *An optimal assignment to minimize the completion time defined in (3) can be approximated by maximizing the global utility U.*

Proof. Let $x'_j = \sum_{i\in\{\mathcal{I}\}_j} \omega_i - h'_j$, it is obvious that

$$x'_1 + x'_2 + \cdots + x'_m = 0. \tag{12}$$

From (8), minimizing $F(\mathbf{x}') = \sum_{j=1}^{m} \tau^{x'_j}$ requires $x'_j = 0$ for all $j \in \mathcal{S}$, which means there exists an assignment such that all the tasks can be completed simultaneously in the task allocation problem. Therefore, the minimum task completion time is

$$T = T_j = \frac{h_j}{\sum_{i\in\mathcal{I}_j} \omega_i} = \frac{h_j}{h'_j} = \frac{1}{\theta}, \tag{13}$$

where the discounted workload h'_j is only used to calculate the assignment, and the actual value is still h_j for all $j \in \mathcal{S}$.

However, the condition $x'_1 = \cdots = x'_m = 0$ might not be satisfied due to the mismatch of the work capacity and the workload in some cases. Recall that (12) always satisfies, then the global utility U increases as x'_j approaches zero for all $j \in \mathcal{S}$, such that $T = \max\{T_1, \ldots, T_m\}$ approaches $1/\theta$.

In conclusion, the optimal assignment to minimize the completion time T can be approximated by maximizing the global utility U. Especially, minimizing T is equivalent to maximizing U when there exists an allocation scheme such that $\sum_{i\in\{\mathcal{I}\}_j} \omega_i - h'_j = 0$ satisfies for all $j \in \mathcal{S}$ (i.e., all the tasks can be completed at the same time.) $\square$

According to the global utility, the agent utility can be designed based on the potential game in the rest of this subsection. Let $G = \{\mathcal{N}, \{\mathcal{A}_i\}, \{U_i\}\}$ denote a game, where $\mathcal{N} = \{1, 2, \ldots, n\}$ is a set of n agents, $\mathcal{A}_i$ denotes the action set of agent i and $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$ is the joint action set, $U_i : \mathcal{A} \to \mathbb{R}$ is the utility function of agent i. For an action profile $a = (a_1, a_2, \ldots, a_n) \in \mathcal{A}$, $a_i \in \mathcal{A}_i$ is the action of agent i, a_i^0 is the null action of agent i (i.e., agent i is turned off), let $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ present the profile of agent actions other than agent i. In the task allocation problem, an agent chooses an action is equivalent to participating in a task; it means that the action set $\mathcal{A}_i$ is its selectable task set and a_i equal to the corresponding task.

Nash equilibrium, one of the central solution concepts for the non-cooperative game, represents a scenario for which no agent has an incentive to deviate unilaterally. The formal definition of Nash equilibrium (NE) is as follow:

Definition 1. For a game $G = \{\mathcal{N}, \{\mathcal{A}_i\}, \{U_i\}\}$, an action $a^{ne} \in \mathcal{A}$ is called a (pure) Nash equilibrium if for each agent $i \in \mathcal{N}$,

$$U_i(a_i^{ne}, a_{-i}^{ne}) = \max_{a_i\in\mathcal{A}_i} U_i(a_i, a_{-i}^{ne}). \tag{14}$$

In this paper, a game-theoretic framework is designed based on the potential game. A finite potential game has the Finite Improvement Property (FIP) and the Fictitious Play Property (FPP) [10], and the change of an agent's utility can be mapped to a global potential function. Next, the definition of the potential game and the essential corollary are presented.

Definition 2 (*Potential Game, [10]*)**.** A game $G = \{\mathcal{N}, \{\mathcal{A}_i\}, \{U_i\}\}$ is an exact potential game if there is a function $\Phi : \mathcal{A} \to \mathbb{R}$ such that $\forall i \in N, \forall a_{-i} \in \mathcal{A}_{-i}, \forall a'_i, a''_i \in \mathcal{A}_i$,

$$U_i(a'_i, a_{-i}) - U_i(a''_i, a_{-i}) = \Phi(a'_i, a_{-i}) - \Phi(a''_i, a_{-i}). \tag{15}$$

The function Φ is called the potential function. In a potential game, there exists at least one pure Nash equilibrium.

For simplicity, assume that $j = \{\mathcal{S}\}_i$ is the action or task that agent i chooses. The agent utility U_i is defined as the normalized marginal contribution of agent i to the global utility. Firstly,

$$\begin{aligned} &U(a_i, a_{-i}) - U(a_i^0, a_{-i}) \\ &= R'_j(a_i, a_{-i}) - R'_j(a_i^0, a_{-i}) \\ &= \tau^{\sum_{l\in\{\mathcal{I}\}_j\backslash i} \omega_l - h'_j} - \tau^{\sum_{l\in\{\mathcal{I}\}_j} \omega_l - h'_j} \\ &= (1-\tau^{\omega_i}) \cdot \tau^{\sum_{l\in\{\mathcal{I}\}_j\backslash i} \omega_l - h'_j}, \end{aligned} \tag{16}$$

where $\tau \in (0, 1)$, it is straightforward that

$$(1-\tau^{\omega_i}) \cdot \tau^{\sum_{l\in\{\mathcal{I}\}_j\backslash i} \omega_l - h'_j} \le (1-\tau^{\max\omega_i}) \cdot \tau^{-\max h'_j}, \tag{17}$$

let $M = (1-\tau^{\max\omega_i}) \cdot \tau^{-\max h'_j}$, then according to Wonderful Life Utility rule [17], the agent utility $U_i(a_i, a_{-i})$ can be defined as

$$U_i(a_i, a_{-i}) = \frac{1}{M}(1-\tau^{\omega_i}) \cdot \tau^{\sum_{l\in\{\mathcal{I}\}_j\backslash i} \omega_l - h'_j}. \tag{18}$$

In (17), $\max\omega_i$ and $\max h_j$ both exist; thus, M is a finite constant to normalize the agent utility such that $U_i(a_i, a_{-i}) \in [0, 1]$. According to (18), each agent requires information about the discounted workload of the task, h'_j, the group or the set of participants $\{\mathcal{I}\}_j$, and the work capacity of each agent in this group. Therefore, each agent needs to communicate with other agents to receive information about their chosen tasks and work capacity.

Corollary 1. *A game $G = \{\mathcal{N}, \{\mathcal{A}_i\}, \{U_i\}\}$ with U_i defined in (18), is a potential game while the potential function satisfies*

$$\Phi = \frac{1}{M} U = \frac{1}{M} \sum_{j=1}^{m} R'_j. \tag{19}$$

Proof. According to (18),

$$U_i(a_i, a_{-i}) = \frac{1}{M} U(a_i, a_{-i}) - \frac{1}{M} U(a_i^0, a_{-i}) \tag{20}$$

$$U_i(a'_i, a_{-i}) = \frac{1}{M} U(a'_i, a_{-i}) - \frac{1}{M} U(a_i^0, a_{-i}), \tag{21}$$

hence,

$$\begin{aligned} &U_i(a'_i, a_{-i}) - U_i(a_i, a_{-i}) \\ &= \frac{1}{M}[U(a'_i, a_{-i}) - U(a_i^0, a_{-i}) - U(a_i, a_{-i}) + U(a_i^0, a_{-i})] \\ &= \frac{1}{M} U(a'_i, a_{-i}) - \frac{1}{M} U(a_i, a_{-i}). \end{aligned} \tag{22}$$

The above equation satisfies (15) for every $i \in \mathcal{N}, \forall a_{-i} \in \mathcal{A}_{-i}, \forall a'_i, a''_i \in \mathcal{A}_i$. Thus, a game $G = \{\mathcal{N}, \{\mathcal{A}_i\}, \{U_i\}\}$ with U_i defined in (18) is a potential game with potential function $\Phi = U/M$. $\square$

3.3. The efficiency of Nash equilibriums

In general, a Nash equilibrium solution is not necessarily an optimal or even a suboptimal solution; likewise, the optimal solution may not be a Nash equilibrium. In this section, it will be proved that the optimal solution must be a Nash equilibrium in the proposed method, and any Nash equilibrium is guaranteed at least 50% of suboptimality. It means that $U(a^{ne}) \in [U(a^{opt})/2, U(a^{opt})]$ for any Nash equilibrium solution a^{ne}, where a^{opt} is the optimal solution.

This subsection would introduce some concepts of the submodular set function at first, and then prove that the designed global utility function satisfies the property of the submodular set function. Finally, the validity proof of Nash equilibrium is presented.

Set function $f : \mathcal{A} \to \mathbb{R}$ is submodular if one of the following two conditions is satisfied.

1. $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y), \forall X, Y \subseteq V$;
2. $f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y), \forall X, Y \subseteq V, X \subseteq Y, \forall x \in V \backslash Y$.

Besides, a set function f is non-decreasing if $f(X) \leq f(Y)$, $\forall X \subseteq Y \subseteq V$. Then the following corollary holds:

Corollary 2. *The global utility $U = \sum_{j=1}^{m} R_j'$ and potential function $\Phi = U/M$ are submodular set functions.*

Proof. Let $g(X) = -\sigma \cdot \tau^{\sum_{i \in X} \alpha_i}$, where $\sigma > 0$ is a constant, and $\alpha_i \geq 0$ belong to the set X for every $i \in N^*$. Assume that $\forall X, Y \subseteq V, X \subseteq Y, x \in V \backslash Y$, then

$$\sum_{i \in Y} \alpha_i - \sum_{i \in X} \alpha_i = \sum_{i \in Y \cup x} \alpha_i - \sum_{i \in X \cup x} \alpha_i \geq 0, \tag{23}$$

for $\tau \in (0, 1)$, it is obvious that $\tau^{\sum_{i \in Y} \alpha_i - \sum_{i \in X} \alpha_i} - 1 < 0$ and $\tau^{\sum_{i \in X \cup x} \alpha_i} \leq \tau^{\sum_{i \in X} \alpha_i}$, therefore,

$$\begin{aligned} &\tau^{\sum_{i \in X \cup x} \alpha_i} \cdot (\tau^{\sum_{i \in Y \cup x} \alpha_i - \sum_{i \in X \cup x} \alpha_i} - 1) \\ &\quad \geq \tau^{\sum_{i \in X} \alpha_i} \cdot (\tau^{\sum_{i \in Y} \alpha_i - \sum_{i \in X} \alpha_i} - 1), \end{aligned} \tag{24}$$

hence,

$$\tau^{\sum_{i \in Y \cup x} \alpha_i} - \tau^{\sum_{i \in X \cup x} \alpha_i} \geq \tau^{\sum_{i \in Y} \alpha_i} - \tau^{\sum_{i \in X} \alpha_i}, \tag{25}$$

then,

$$-\tau^{\sum_{i \in X \cup x} \alpha_i} + \tau^{\sum_{i \in X} \alpha_i} \geq -\tau^{\sum_{i \in Y \cup x} \alpha_i} + \tau^{\sum_{i \in Y} \alpha_i}. \tag{26}$$

According to the second condition of submodular set function, $\forall X, Y \subseteq V, X \subseteq Y, \forall x \in V \backslash Y$,

$$g(X \cup \{x\}) - g(X) \geq g(Y \cup \{x\}) - g(Y), \tag{27}$$

so $g(X) = -\sigma \cdot \tau^{\sum_{i \in X} \alpha_i}$ is a submodular set function. Let $\sigma = \tau^{-h_j'} > 0$ is a constant; then, $R_j = -\tau^{\sum_{i \in \{\mathcal{I}\}_j^k} \omega_i - h_j'}$ is a submodular set function. Moreover, the class of submodular functions is closed under a non-negative linear combination, and $M > 0$ is a finite constant; hence, $U = \sum_{j=1}^{m} R_j'$ and $\Phi = U/M$ are submodular set functions. □

This paper refers to the fraction of $U(a^{ne})$ with respect to $U(a^{opt})$ as the suboptimality of Nash equilibrium solutions, denoted by $U(a^{ne})/U(a^{opt})$. The following corollary determines the lower and upper bounds.

Corollary 3. *A game $G = \{\mathcal{N}, \{\mathcal{A}_i\}, \{U_i\}\}$ with $U_i(a_i, a_{-i})$ defined in (18), then any Nash equilibrium a^{ne} satisfies*

$$\frac{1}{2} U(a^{opt}) \leq U(a^{ne}) \leq U(a^{opt}), \tag{28}$$

where, a^{opt} is the optimal strategy.

Proof. First, the lower bound of the Nash equilibrium solution can be determined. By Corollary 2, the global utility $U = \sum_{j=1}^{m} R_j'$ and potential function $\Phi = U/M$ are non-decreasing submodular set functions, and $U_i(a_i, a_{-i}) = [U(a_i, a_{-i}) - U(a_i^0, a_{-i})]/M$ for every $i \in \mathcal{N}$. Based on Theorem 5 in [30], for any Nash equilibrium a^{ne},

$$U(a^{ne}) \geq \frac{1}{2} U(a^{opt}). \tag{29}$$

Second, it can be proved that the upper bound of the Nash equilibrium solution is $U(a^{opt})$. Supposing the strategy $a^{opt} = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ is not a Nash equilibrium, and then there exists at least one agent, say agent i, could increase its utility by changing its strategy σ_i to σ_i^*, thus $U_i(\sigma_i^*, \sigma_{-i}) - U_i(\sigma_i, \sigma_{-i}) > 0$. In a potential game, this assumption means that the global utility can be improved by changing the action of agent i, which is

$$U(\sigma_i^*, \sigma_{-i}) > U(\sigma_i, \sigma_{-i}) = U(\Omega). \tag{30}$$

Obviously, this result contradicts the assumption; thus, a^{opt} must be a Nash equilibrium.

In conclusion, $U(a^{opt})/2 \leq U(a^{ne}) \leq U(a^{opt})$ holds for any Nash equilibrium a^{ne} in game $G = \{\mathcal{N}, \{\mathcal{A}_i\}, \{U_i\}\}$ with $U_i(a_i, a_{-i})$ defined in (18). □

3.4. Dynamic task allocation

Consider the workload of each task may vary in a dynamic environment, and the allocation scheme requires fine-tuning at the right moments. In order to improve the performance, some agents need to stop their current work and engage in other tasks, while this change will take a toll on the system, such as the time cost when agents change their positions. Thus, each agent must make a rational choice after the environment changed.

In the above sections, a generic task allocation game-theoretic model is defined without considering the time, while this section will consider the dynamic task allocation in discrete time. Denote $[t_k, t_k + \Delta t)$ as the kth stage, where Δt is a time unit. In such a multi-stage task allocation problem, the allocation scheme is adjusted at each stage. Now, let $\{\mathcal{S}\}_i^k$ and $\{\mathcal{I}\}_j^k$ represent the task that agent $i \in \mathcal{N}$ chooses and the set of participants of task $j \in \mathcal{S}$ at stage k, respectively. An agent may move to another position in the next stage, assume that the speed of each agent is v, and the distance between any two tasks is less than that of an agent moves in a unit time Δt, then $d_{jj'} < v\Delta t$ holds for every $j, j' \in \mathcal{S}$.

Different from Section 3.1, in dynamic task allocation, an agent i may have different work capacity for different tasks at each stage. The reason is that an agent i may move to a new location to execute another task, while it cannot perform the task in the transfer process. Therefore, denote the work capacity of agent i to task j by

$$\omega_{ij} = \omega_i(\Delta t - \frac{d_{ij}}{v}), \tag{31}$$

where d_{ij}/v is the time cost of the transfer process, $(\Delta t - d_{ij}/v)$ is the actual working time of agent i for task j in a stage. It is obvious that ω_{ij} is small if agent i is far away from task j, so each agent tends to select a closer task.

The workload of task j at stage k is defined as h_j^k, and then in the next stage,

$$h_j^{k+1} = h_j^k - \Delta t \sum_{i \in \{\mathcal{I}\}_j^k} \omega_{ij} + add^{k+1}, \tag{32}$$

where add^{k+1} is the additional workload at stage $k+1$.

Fig. 1 shows the flow chart of multi-agent dynamic task allocation. Firstly, all the agents are assigned to tasks in the initialization step. Then, each agent moves to the position of the chosen task and executes it in stage k. In the meanwhile, the agent communicates with neighbors in the communication range to receive information about their chosen tasks and work capacity, and detects the change of tasks. Finally, based on the new information, each agent decides to participate in a new task or remain unchanged in the next stage.

In dynamic task allocation, the model introduced in Section 3.2 are applied in the Initialization and Adjustment steps. The only difference is that agent i needs to calculate ω_{ij} according to the current state at each stage and replace ω_i with ω_{ij}. Then the game approaches one of the best Nash equilibriums, i.e., the optimal assignment, in each stage until all tasks are completed.

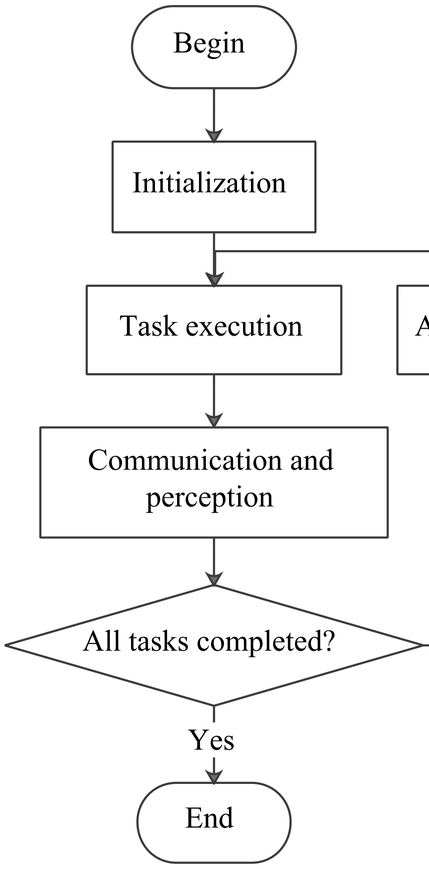


Fig. 1. The flow chart of multi-agent dynamic task allocation.

4. Learning algorithm

Many learning algorithms have good convergence, such as fictitious play [31], gradient play [14], log-linear learning [13], and some others [32,33]. Among them, log-linear learning, originally introduced by Blume, ensures that the action profile will be at a potential maximizer in potential games [34]; however, the fixed parameter β constraints the performance since maximizing the potential function requires $\beta \rightarrow \infty$ as a game proceeds. To improve the performance of log-linear learning, a time variant constrained binary log-linear learning (TVCBLL) algorithm is designed for multi-agent dynamic task allocation problem, and then the TVCBLL is proved to converge to the optimal solution as time goes to infinity.

4.1. Time variant constrained binary log-linear learning

For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, each node $v \in \mathcal{V}$ represents a task (i.e., an action for the agent), $e \in \mathcal{E}$ is the edge between the connected nodes and the weight $w \in \mathcal{W}$ corresponding to the Euclidean distance. Any two nodes (tasks) connected by an edge are called neighbor nodes (adjacent tasks). The communication radius of agent i is denoted by Rad_i, and the agent can only choose the tasks within its communication range and receives information about the participants of these tasks. The constrained action/task set at time t is

$$C_i^t = \{v_j \in \mathcal{V} \mid \left\| pos_j - pos_i^{t-1} \right\| \le Rad_i, h_j > 0\}, \tag{33}$$

where v_j is the node that corresponding to task j in graph $\mathcal{G}$, pos_i^{t-1} is the position of agent i at time $t-1$, and h_j represents the workload of task j, $C_i^1 = \mathcal{V}$. Next, the following assumption is necessary to prove the convergence of proposed algorithm.

Assumption 1. Suppose the multi-agent system satisfies the following conditions.

1. The undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ is connected;
2. Agent $i \in \mathcal{N}$ could only select the action/task in the constrained action/task set C_i^t at time t;
3. The communication radius of any agent is no less than the distance between any two adjacent tasks, i.e., $\forall i \in \mathcal{N}$, $\forall w \in \mathcal{W}$, $Rad_i \ge \max w = Rad$.

Follow Assumption 1, there exists a sequence of actions $a_i^0 \rightarrow a_i^1 \rightarrow \cdots \rightarrow a_i^m$ that agent $i \in \mathcal{N}$ could select any action/task $a_i^m \in \mathcal{S}$ through finite numbers of state transitions.

The procedure of the TVCBLL algorithm is given as follow:

(1) Set $t := 1$, randomly initialize the actions/tasks of all agents, and set the maximum iterations, $Max_{iterations}$;
(2) Randomly choose an agent $i \in \mathcal{N}$ to be allowed to update its action/task, while the others remain unchanged;
(3) Update the constrained action/task set of agent i at time, C_i^t, by (33);
(4) Select a trial action/task $\hat{a}_i$ from C_i^t with probability:

$$\begin{aligned} p[\hat{a}_i = a_i] &= \frac{1}{N_i}, \quad \forall a_i \in C_i^t \backslash a_i(t-1) \\ p[\hat{a}_i = a_i(t-1)] &= 1 - \frac{|C_i^t| - 1}{N_i} \end{aligned} \tag{34}$$

where $N_i = \max_{a_i \in \mathcal{A}_i} |C_i|$ is the maximum cardinal number of all the possible constrained action/task set of agent i;
(5) Calculate $U_i(\hat{a}_i, a_{-i})$ and $U_i(a_i(t-1), a_{-i})$ according to (18). Chooses $\hat{a}_i$ or remains $a_i(t-1)$ with probability:

$$\begin{aligned} p_{\hat{a}_i(t)} &= \frac{e^{\beta U_i(\hat{a}_i, a_{-i}(t-1))}}{e^{\beta U_i(\hat{a}_i, a_{-i}(t-1))} + e^{\beta U_i(a_i(t-1), a_{-i}(t-1))}} \\ p_{a_i(t)} &= 1 - p_{\hat{a}_i(t)}, \end{aligned} \tag{35}$$

where $\beta(t) = \beta_0 + \ln(\lambda t + 1)/c$ is a time variant parameter, $t \ge 0, \beta_0 \ge 0, \lambda \ge 1, c \in N^*$;
(6) Set $t := t + 1$. If $t < Max_{iterations}$, return to step (2);
(7) End. Return the action profile, a_t.

From the implementation and computational point of view, the chosen agent in each iteration only calculates $U_i(\hat{a}_i, a_{-i}(t-1))$ and $U_i(a_i(t-1), a_{-i}(t-1))$ according to (18), and selects the trial action/task or remains unchanged based on probability distribution (35). In the communication process, an agent receives information about the discounted workload of the trial task and communicates with the participants to obtain the agents' work capacity in this group. Therefore, the process requires little information exchange, and the TVCBLL algorithm provides an implemented way to solve the task allocation problem.

4.2. Analysis

The traditional log-linear learning with the fixed parameter β defines an ergodic homogeneous Markov chain [13], which guarantees the existence of unique stationary distribution and the convergence property. As $\beta \rightarrow \infty$, the agent will choose the action that provides a higher reward, while the agent will select $a_i(t-1)$ and $\hat{a}_i$ with equal probability as $\beta \rightarrow 0$. Consequently, β determines how likely an agent is to select a suboptimal action, and it is crucial that β approximates infinity as $t \rightarrow \infty$.

In this paper, β is designed to be a time variant parameter, and the action set is constrained. Since $\beta(t)$ is time variant, the Markov chain is no longer time independent; the inhomogeneous Markov chain theory is considered to analyze the convergence of the TVCBLL algorithm.

Firstly, some concepts used for the convergence analysis are introduced.

Definition 3 (*Scrambling Matrix, [35]*). A transition matrix P is called scrambling matrix if for any two rows, say α and β, there exists at least one column, γ, such that both $p_{\alpha\gamma} > 0$ and $p_{\beta\gamma} > 0$.

Notice that a transition matrix (or stochastic matrix) is a nonnegative real square matrix with each row summing to 1. Then, a measure of the scrambling matrix is called "scrambling power" defined as

$$sp(P) = \min_{\alpha,\alpha'} \sum_{\beta} \min(p_{\alpha\beta}, p_{\alpha'\beta}). \tag{36}$$

According to (36), transition matrix P is not a scrambling matrix if and only if $sp(P) = 0$. Besides, P is called *regular* if some power of P has all positive entries. With the above notions, the following lemma from [35] is given.

Lemma 1. *For each $t \in N^*$, if the transition matrix P_t is regular and has the same pattern, which means the arrangement of non-zero elements in each P_t is the same, then there exists an integer $c > 0$, such that $H_{t,c} = \prod_{i=t}^{t+c-1} P_i$ is a scrambling matrix, therefore, $sp(H_{t,c}) > 0$.*

For an inhomogeneous Markov chain, there exists a unique stochastic row vector $\boldsymbol{\pi}^*$, called stationary distribution. If the inhomogeneous Markov chain is strongly ergodic [36], then $\lim_{n\to\infty} \boldsymbol{\pi} H_{t,n} = \boldsymbol{\pi}^*$ holds for any initial distribution $\boldsymbol{\pi}$, and all the weight of the stationary distribution $\boldsymbol{\pi}^*$ is on the joint actions that maximize the potential function as $\beta \to \infty$. Therefore, it is necessary to prove the strong ergodicity of the inhomogeneous Markov chain $M_{\beta(t)}$ induced by the TVCBLL algorithm. Next, the criteria for weak ergodicity and the sufficient conditions of strong ergodicity are given as follows:

Lemma 2 (*Weak Ergodicity, [35]*). *A Markov chain is weakly ergodic if and only if there exists a subdivision of the sequence of trials into blocks at trials numbered $t_1, t_2, t_3, \ldots$, such that $\sum_{i=1}^{\infty} sp(P_{t_i,m_i}) > \infty$, where $m_i = t_{i+1} - t_i$.*

Lemma 3 (*Strong Ergodicity, [37]*). *An inhomogeneous Markov chain $M_{\beta(t)}$ is strongly ergodic if the following conditions holds:*

1. *$M_{\beta(t)}$ is weakly ergodic.*
2. *For each t, P_t is a regular matrix with a stationary distribution $\boldsymbol{\pi}_t$.*
3. *The stationary distribution $\boldsymbol{\pi}_t$ satisfies*

$$\sum_{t=1}^{\infty} \sum_{a\in\mathcal{A}} |\pi_{t+1}^a - \pi_t^a| < \infty. \tag{37}$$

Before proving the weak and strong ergodicity of $M_{\beta(t)}$ for the proposed TVCBLL algorithm, the conclusions about constrained binary log-linear learning (CBLL) algorithm [38] and the TVCBLL algorithm are presented as follows:

Lemma 4. *For a finite potential game with potential function $\varphi(\cdot)$, if Assumption 1 holds, then the constrained binary log-linear learning (CBLL) algorithm induced a homogeneous Markov process over the state space $\mathcal{A}$. Also, there exists the unique stationary distribution π^a for any strategy $a \in \mathcal{A}$ as*

$$\pi^a = \frac{e^{\beta\varphi(a)}}{\sum_{\tilde{a}\in\mathcal{A}} e^{\beta\varphi(\tilde{a})}}. \tag{38}$$

Proof. This proof is similar to Theorem 3.1 of [38]. Based on Assumption 1, the Markov process that the CBLL induced is irreducible and aperiodic; therefore, there exists a unique stationary distribution. Next, it will be shown that $\pi^a P(a,b) = \pi^b P(b,a)$ holds for any state $a, b \in \mathcal{A}$, where $P(a,b)$ is the transition probability from state a to b since π^a is unique when this condition satisfies.

Based on (35),

$$P(a,b) = \frac{1}{n}\begin{cases} p_{b_i}, & \text{if} \quad a_i \neq b_i, a_{-i} = b_{-i} \\ \sum_{i=1}^{n} p_{b_i}, & \text{if} \quad a = b \\ 0, & otherwise. \end{cases} \tag{39}$$

It is easy to verify that $\pi^a P(a,b) = \pi^b P(b,a)$ when the latter two conditions of (39) meet. Consequently, this proof focus on the first situation where $a^i \neq b^i, a^{-i} = b^{-i}$. For the CBLL algorithm,

$$\begin{aligned} \pi^a P(a,b) &= \frac{e^{\beta\varphi(a)}}{\sum_{\tilde{a}\in\mathcal{A}} e^{\beta\varphi(\tilde{a})}} \frac{1}{n} \frac{1}{N_i} \frac{e^{\beta U_i(b)}}{e^{\beta U_i(a)} + e^{\beta U_i(b)}} \\ \pi^b P(b,a) &= \frac{e^{\beta\varphi(b)}}{\sum_{\tilde{a}\in\mathcal{A}} e^{\beta\varphi(\tilde{a})}} \frac{1}{n} \frac{1}{N_i} \frac{e^{\beta U_i(a)}}{e^{\beta U_i(a)} + e^{\beta U_i(b)}}, \end{aligned} \tag{40}$$

where $N_i = \max_{a_i \in \mathcal{A}_i} |C_i|$ is the maximum cardinal number of all the possible constrained action set of agent i.

Moreover, recall that $\varphi(a) + U_i(b) = \varphi(b) + U_i(a)$ always holds in a potential game, thus

$$\begin{aligned} \pi^a P(a,b) &= \frac{e^{\beta\varphi(a) + \beta U_i(b)}}{nN_i(e^{\beta U_i(a)} + e^{\beta U_i(b)}) \sum_{\tilde{a}\in\mathcal{A}} e^{\beta\varphi(\tilde{a})}} \\ &= \frac{e^{\beta\varphi(b) + \beta U_i(a)}}{nN_i(e^{\beta U_i(a)} + e^{\beta U_i(b)}) \sum_{\tilde{a}\in\mathcal{A}} e^{\beta\varphi(\tilde{a})}} \\ &= \pi^b P(b,a), \end{aligned} \tag{41}$$

which shows that the CBLL algorithm would induce a homogeneous Markov process with unique stationary distribution π^a. □

Corollary 4. *For any given time $t > 0$, the transition matrix P_t of inhomogeneous Markov chain $M_{\beta(t)}$ that the TVCBLL algorithm induced is a regular matrix.*

Proof. For any given time $t > 0$, $\beta(t)$ is a constant, the Markov chain induced by the CBLL has stationary distribution $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_{|A|}\}$ and $\boldsymbol{\pi} = \boldsymbol{\pi} P_t$. It shows that $\pi_j = \lim_{k\to\infty} p_{ij}^{(k)} \geq 0$ holds for any row i and column j of transition matrix at a given time, then all the elements of column j of P_t are π_j under the stationary distribution. Therefore, the transition matrix P_t of $M_{\beta(t)}$ that the TVCBLL algorithm induced is a regular matrix at any given time $t > 0$. □

Proposition 1. *For a finite potential game $G = \{\mathcal{N}, \{\mathcal{A}_i\}, \{U_i\}\}$, where U_i is defined as (18), the proposed TVCBLL algorithm induces an inhomogeneous Markov chain $M_{\beta(t)}$ that is weakly ergodic.*

Proof. Consider the time variant parameter $\beta(t) = \beta_0 + \ln(\lambda t + 1)/c, t \geq 0, \beta_0 \geq 0, \lambda \geq 1$, according to (39), for any $a_1, a_2 \in \mathcal{A}$,

$$\begin{aligned} P_t(a_1,a_2) &= \frac{1}{n}\frac{1}{N_i}\frac{e^{\beta U_i(a_2)}}{e^{\beta U_i(a_1)} + e^{\beta U_i(a_2)}} \\ &= \frac{1}{nN_i}\frac{e^{\beta_0}(\lambda t + 1)^{\frac{U_i(a_2)}{c}}}{e^{\beta_0}(\lambda t + 1)^{\frac{U_i(a_1)}{c}} + e^{\beta_0}(\lambda t + 1)^{\frac{U_i(a_2)}{c}}} \\ &= \frac{1}{nN_i}\frac{1}{1 + (\lambda t + 1)^{\frac{U_i(a_1) - U_i(a_2)}{c}}}. \end{aligned} \tag{42}$$

Since U_i is defined in (18), $U_i \in [0, 1]$, it is obvious that $U_i(a_1) - U_i(a_2) \leq 1$. Therefore, any positive element $P_t(a_1, a_2)$ of transition matrix P_t satisfies

$$P_t(a_1, a_2) \geq \frac{1}{nN_i} \frac{1}{2(\lambda t + 1)^{\frac{1}{c}}}. \tag{43}$$

According to Corollary 4, the transition matrix P_t is a regular matrix for any given time $t > 0$. Consider (39), it is straightforward that the arrangement of non-zero elements in each P_t at different stage remains unchanged, or in another word, the pattern of P_t remains the same. Let $H_{t,c} = \prod_{i=t}^{t+c-1} P_i$, by Lemma 1, an proper integer c can be chosen such that $H_{t,c}$ is a scrambling matrix,

$$sp(H_{t,c}) \geq \min_{a_1,a_2} h^{*(t,c)}_{a_1,a_2}, \tag{44}$$

where, $h^{*(t,c)}_{a_1,a_2}$ is the positive element of $H_{t,c}$. Next, based on Chapman–Kolmogorov equation and Assumption 1,

$$\begin{aligned} h^{*(t,c)}_{a_1,a_2} &= \sum_{b_1\in\mathcal{A}} \cdots \sum_{b_c\in\mathcal{A}} P_t(a_1, b_1)\cdots P_{t+c-1}(b_c, a_2) \\ &\geq \frac{1}{2^c n^c N_i^c} \frac{1}{(\lambda t + 1)^{\frac{1}{c}}} \frac{1}{(\lambda t + 2)^{\frac{1}{c}}} \cdots \frac{1}{(\lambda t + c)^{\frac{1}{c}}} \\ &\geq \frac{1}{2^c n^c N_i^c} \frac{1}{((\lambda t + c)^{\frac{1}{c}})^c}, \end{aligned} \tag{45}$$

therefore,

$$sp(H_{t,c}) \geq \frac{1}{2^c n^c N_i^c} \frac{1}{\lambda t + c}. \tag{46}$$

Then, the time sequence can be divided, $t_j = (j-1)c+1, j \in N^*$, and the length of each unit is c, according to (46),

$$\sum_{j=1}^{\infty} sp(H_{t_j,c}) \geq \frac{1}{2^c n^c N_i^c} \sum_{j=1}^{\infty} \frac{1}{\lambda j + c}, \tag{47}$$

it is straightforward that infinite series $\sum_{j=1}^{\infty} \frac{1}{\lambda j+c}$ diverges when $\lambda \geq 1, c \in N^*$, thus $\sum_{j=1}^{\infty} sp(H_{t_j,c}) > \infty$.

Based on Lemma 2, the inhomogeneous Markov chain $M_{\beta(t)}$ initiated by the TVCBLL algorithm is weakly ergodic. □

Proposition 2. *For a game $G = \{\mathcal{N}, \{\mathcal{A}_i\}, \{U_i\}\}$, with U_i defined in (18), the inhomogeneous Markov chain $M_{\beta(t)}$ induced by the TVCBLL algorithm is strongly ergodic.*

Proof. According to Lemma 3, three conditions need to be verified. Proposition 1 proved that the inhomogeneous Markov chain $M_{\beta(t)}$ initiated by the TVCBLL algorithm is weakly ergodic.

Then, the second condition can also be proved. For any given stage, $\beta(t) = \beta_0 + \ln(\lambda t + 1)/c$ is a constant, and P_t is a regular matrix according to Corollary 4. By Lemma 4, the stationary distribution in a potential game with potential function $\varphi(\cdot)$ is

$$\pi_t^a = \frac{e^{\beta(t)\varphi(a)}}{\sum_{\tilde{a}\in\mathcal{A}} e^{\beta(t)\varphi(\tilde{a})}} = \frac{(\lambda t + 1)^{\frac{\varphi(a)}{c}}}{\sum_{\tilde{a}\in\mathcal{A}}(\lambda t + 1)^{\frac{\varphi(\tilde{a})}{c}}}. \tag{48}$$

Finally, it is essential to prove the stationary distribution π_t^a in (48) satisfies

$$\sum_{t=1}^{\infty} \sum_{a\in\mathcal{A}} |\pi_{t+1}^a - \pi_t^a| < \infty. \tag{49}$$

This part of proof is leaved out as it is similar to Theorem 2 in [39], the only difference is the stationary distribution $\boldsymbol{\pi}_t$. □

From the above analysis, the inhomogeneous Markov chain $M_{\beta(t)}$ induced by the TVCBLL algorithm is strongly ergodic; thus, there must exist a unique stationary distribution. In a potential game, the strategy obtained by the TVCBLL algorithm is the potential function maximizer as the time approaches infinity (such that $\beta(t) \to \infty$).

Besides, it is vital to guarantee that $H_{t,c} = \prod_{i=t}^{t+c-1} P_i$ is a scrambling matrix in Proposition 1. By Lemma 1, whether matrix $H_{t,c}$ is scrambling is in the control of parameter c. According to [40], c should be chosen as the least positive integer, such that the following inequation holds.

$$(1 - n^{-n})^{\lfloor c/n \rfloor} \leq \frac{1}{2 + \varepsilon}, \tag{50}$$

where $\varepsilon > 0$ is a constant, n is the number of agents and $\lfloor c/n \rfloor$ is the greatest integer that no more than c/n.

As a consequence, the following main result can be formulated based on the above analysis.

Theorem 1. *For a finite potential game $G = \{\mathcal{N}, \{\mathcal{A}_i\}, \{U_i\}\}$ with U_i defined in (18), if the selectable action set of agents is constrained and Assumption 1 holds, the TVCBLL algorithm with $\beta(t) = \beta_0 + \ln(\lambda t+1)/c$ and c satisfies (50) could converge to the global optimal solutions with probability 1.*

Proof. According to Propositions 1 and 2, the inhomogeneous Markov chain $M_{\beta(t)}$ induced by the TVCBLL algorithm is strongly ergodic and has a unique stationary distribution π^a defined in (38). As the time approaches infinity, $\beta(t) \to \infty$, all the weight of the stationary distribution $\boldsymbol{\pi}$ is on the joint actions of the potential function maximizer. By (15), the global utility $U = M\varphi$, where M is a finite constant, thus, the TVCBLL algorithm could converge to one of the optimal solutions with probability 1 in game G. □

5. Experimental results

In this section, the proposed method is tested in a static environment, where the workload is fixed over time; therefore, only one allocation plan needs to be calculated. Then the dynamic task allocation is considered, which is a multi-stage allocation procedure introduced in Section 3.4, and the assignment is adjusted in every stage.

5.1. Static task allocation

This subsection is devoted to evaluating the effectiveness of the proposed game-theoretic task allocation method in the static environment. The following algorithms are compared with the TVCBLL algorithm:

- Better reply process (BRP): In the BRP, each agent tends to select an action, which can improve its reward, with a probability distribution. Young et al. [33] proved that the BRP converges to an equilibrium in the potential game. Firstly, the strict better reply set of agent i for any action $a_i \in \mathcal{A}_i$ is defined as

$$B_i(a) = \{a_i' \in \mathcal{A}_i | U_i(a_i', a_{-i}) > U_i(a)\}, \tag{51}$$

then, at each time $t > 0$, agent $i \in \mathcal{N}$ selects an action from $B_i(a)$ according to the following procedures:

  - If $B_i(a(t-1)) = \emptyset$, which means there is no better choice for agent i, then $p_{a_i(t-1)} = 1$.

– Otherwise, if $B_i(a(t-1)) \neq \emptyset$,

$$p_{a_i} = \epsilon, for \quad a_i = a_i(t-1)$$
$$p_{a_i'} = \frac{1-\epsilon}{|B_i(a(t-1))|}, \forall a_i' \in B_i(a(t-1)) \tag{52}$$
$$p_{a_i''} = 0, \forall a_i'' \notin B_i(a(t-1)) \cup \{a_i(t-1)\},$$

where, p_{a_i} is the probability that agent i selects a_i, and $\epsilon \in (0,1)$ is the inertia which reflects the reluctance to change actions.

- Best response algorithm (BRA): Different from the BRP, the BRA is a multi-round greedy algorithm. At each iteration, one agent is randomly chosen and allowed to select the best action from the constrained action set C_i^t.
- Log-linear learning algorithm (LLA): The log-linear learning algorithm with fixed parameter $\beta = \beta_0$ is considered. The LLA has the equilibrium selection feature, which could converge to the potential function maximizer. Moreover, the probability distribution is calculated by

$$p_{a_i} = \frac{e^{\beta U_i(a_i, a_{-i}(t-1))}}{\sum_{\tilde{a} \in \mathcal{A}} e^{\beta U_i(\tilde{a}_i, a_{-i}(t-1))}} \tag{53}$$

- Joint strategy fictitious play (JSFP): The JSFP is the variant of the fictitious play algorithm and provides similar asymptotic guarantees while alleviating the computational and observational challenges [31]. The critical difference between the JSFP and fictitious play is that each agent presumes the joint strategy of all other agents rather than the individual strategy of each agent. The average hypothetical utility of agent $i \in \mathcal{N}$ for each action $a_i \in \mathcal{A}_i$ is defined as

$$\begin{aligned}\bar{U}_i^{a_i}(t) &= \frac{1}{t}\sum_{\tau=0}^{t-1} U_i(a_i, a_{-i}(\tau)) \\ &= \frac{t-1}{t}\bar{U}_i^{a_i}(t-1) + \frac{1}{t}U_i(a_i, a_{-i}(t-1))\end{aligned} \tag{54}$$

At time t, agent i calculates its average hypothetical utility and selects the action according to the rule

$$a_i(t) = \begin{cases} arg \max_{a_i \in \mathcal{A}_i} \bar{U}_i^{a_i}(t) \quad with \quad Pr = (1-\epsilon), \\ a_i(t-1) \quad with \quad Pr = \epsilon, \end{cases} \tag{55}$$

where, $\epsilon \in (0,1)$ is inertia.

Two scenarios are considered: (i) 20 agents are assigned to 4 tasks, and (ii) 100 agents are assigned to 20 tasks. The Monte Carlo simulations are utilized for each scenario, and over 100 runs generate the results of each algorithm. At each run, all the agents are randomly assigned and the maximal iteration is set as 1000. In all the scenarios, the time unit Δt is one hour (h).

Assume that if the number of agents is 5, parameter c can be chosen as 1.08×10^4 with $\varepsilon = 10^{-10}$, according to (50). Parameter c is chosen as 9.303×10^7 when the number of agents is 8, while $\beta(t) = \beta_0 + \ln(\lambda t + 1)/c$ almost remains unchanged as time goes on. Besides, parameter c that satisfies (50) is huge when $n = 20$. To make the TVCBLL algorithm more efficient and practical when there are more agents in the environment, the constraint of c is relaxed and c is set as $c = 100 = \lambda$, $\beta_0 = 2000$. Notice that there is no theoretical support of global convergence if $c = 100$ and $n \geq 20$, since the inequality (50), which is one of the sufficient conditions of Theorem 1, no longer holds. This is one of the reasons why the TVCBLL convergence to the global optimal solution under certain conditions. However, the following tests will show that the TVCBLL algorithm performs well compared with many other algorithms in terms of global searching ability.

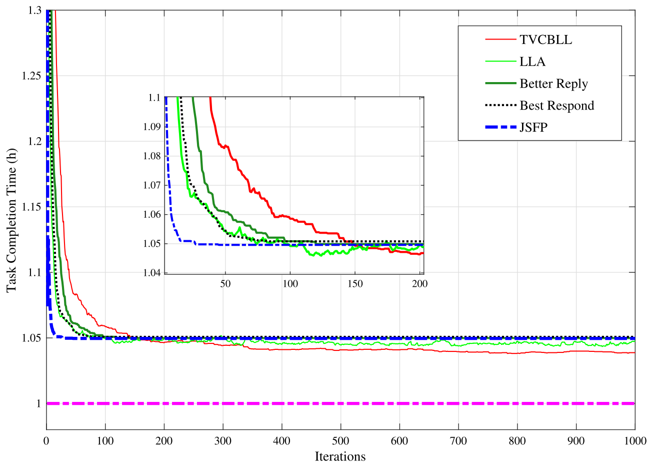


Fig. 2. Convergence performance of 5 algorithms in the static environment with 20 agents and 4 tasks. The pink dotted lines indicate the value of the optimal solution, which is 1 h.

Table 2
Assignment results of different algorithms for the first scenario ($n = 20, m = 4$). Simulations are repeated over 100 runs for each method.

Algorithm	Indicators				
	P_0	*Var*	δ	T	*Cost*
Better reply	7.5%	0.00076	4.98%	1.050 h	0.063 s
Best response	5.8%	0.00063	5.08%	1.051 h	0.054 s
JSFP	6.7%	0.00059	4.92%	1.049 h	0.368 s
LLA	8.1%	0.00102	4.73%	1.047 h	0.177 s
TVCBLL	**13.3%**	0.00087	**3.88%**	**1.039 h**	0.161 s
Optimal	100%	–	–	1 h	–

In the first scenario, the work capacity of each agent is randomly set as an integer between 1 and 10 while making sure the sum of all agents' work capacity is 100, and the workload of 4 tasks are initialized as different positive values while making sure their sum is 100. Furthermore, the optimal solution calculated by the exhaustive search is 1 h. Fig. 2 shows the convergence curves of five algorithms, and the pink dotted line indicates the value of the optimal solution. Table 2 denotes the correlated indicators: T is the task completion time defined in (3); P_0 indicates the proportion of optimal solutions T^{opt} over the 100 runs; *Var* is the variance; $\delta = (T - T^{opt})/T^{opt}$ is the difference ratio of T and T^{opt}; *Cost* record the computation time that each algorithm converges.

From the perspective of result efficiency, the performance of the BRP, BRA, and JSFP are about 1.05 h, while the LLA and the TVCBLL are better than them. Especially, the proposed TVCBLL approximates the optimal solution as time goes to infinity. Although the LLA has better assignments than the BRP, BRA, JSFP and is similar to the TVCBLL, the LLA is not robust as other algorithms, and it often makes an inferior decision due to the strong randomness. Therefore, it is shown that the variance of the LLA in Table 2 is the biggest. The TVCBLL algorithm gets over this drawback as $\beta \to \infty$ when time goes to infinity, such that each agent would choose the most appropriate task with probability 1. Consider the indicator P_0, which reflects the global search ability, the P_0 of the BRP is slightly better than the BRA as better reply process has a certain randomness when agents choosing the actions, and this property would help to escape local optima. Moreover, the TVCBLL has good exploration ability at the beginning, since it has a certain randomness, while it has good exploitation ability when $\beta \to \infty$; therefore, the P_0 of the TVCBLL is the largest of the five. The indicator δ is the difference ratio of T and T^{opt}, and the lowest δ of the TVCBLL indicates that the proposed learning algorithm is more likely to find the global optimal solutions. The results imply that the perfect balance between exploration and exploitation ability is of significant importance to convergence to better solutions, and the TVCBLL is a better trade-off algorithm than the LLA by adopting the time variant parameter β.

Next, the convergence dynamics are considered. Fig. 2 shows that the TVCBLL requires more iterations than any other methods while the JSFP converges in a few iterations. However, it does not mean that the TVCBLL requires much computing resources. From Table 2, the *Cost* of the TVCBLL is better than the JSFP. The reason is that each agent, in the JSFP, needs to observe the past joint strategy of others and presumes their next joint strategy according to empirical frequency. As a consequence, the JSFP is more computational than the TVCBLL, and it would be impractical when the size of agents is large. Also, recall that the chosen agent only computes the utilities of trial action and current action in each iteration for the TVCBLL according to (35), while, for the LLA, it calculates utilities of all the selectable actions, that is why the *Cost* of the LLA is higher than the TVCBLL. As it is mentioned in the previous section, the TVCBLL requires little information exchange, i.e., the discounted workload of the trial task/action and the work capacity of the participants in the same task group, so it is a simple and implemented learning algorithm. As for the BRP and the BRA, these two greedy algorithms have similar performance with the LLA in the short run according to Fig. 2. Actually, the TVCBLL algorithm is also greedy if β_0 is large enough since each agent would always choose the best action.

For the second scenario, the work capacity of each agent and workload of each task are set in the same way as the first scenario, but the work capacity of 100 agents is 400, and the workload of 20 tasks is 800. The optimal solution calculated by the exhaustive search is 2 h. The task completion time obtained by the BRP, BRA, and JSFP is about 2.2 shown in Fig. 3. Table 3 denotes the correlated indicators for the second scenario. In line with the first scenario, the TVCBLL has the best convergence performance where the optimal solution is 2.142 h despite requiring many iterations, and other indicators also demonstrate the effectiveness of the TVCBLL, especially the proportion of optimal solution P_0 of the TVCBLL is much large than the others.

Compare these two scenarios, the solutions obtained by the TVCBLL algorithm are 0.161 s and 0.212 s respectively, and approximate to the optimal solutions as time goes to infinity in both situations. Moreover, the TVCBLL algorithm is tested in more complex environments, where the number of agents is increased to 200 and 300, respectively, while the number of tasks remains unchanged at 20. We found that the TVCBLL algorithm still performs well in these complex problems. However, the performance of the TVCBLL algorithm will drop if the number of tasks increases, and this is because the chosen agent selects only one trial task in each iteration, and it is hard to select the optimal trial task when the task set is large. Therefore, the algorithm is suitable for the task allocation problem that the number of agents is higher than that of tasks.

Although there exist some limitations for the TVCBLL algorithm, such as it requires many iterations and the parameter $\beta(t)$ grows very slow as time goes on, it is still a practical algorithm in engineering systems and suitable for dynamic task allocation problems.

5.2. Dynamic task allocation

Section 5.1 has shown the efficiency of the proposed method. For many applications in the dynamic environment, such as surveillance, disaster management, and weapon-target assignment, et al. it is vital to converge to an optimal or suboptimal solution in a short time. The TVCBLL algorithm satisfies such a requirement.

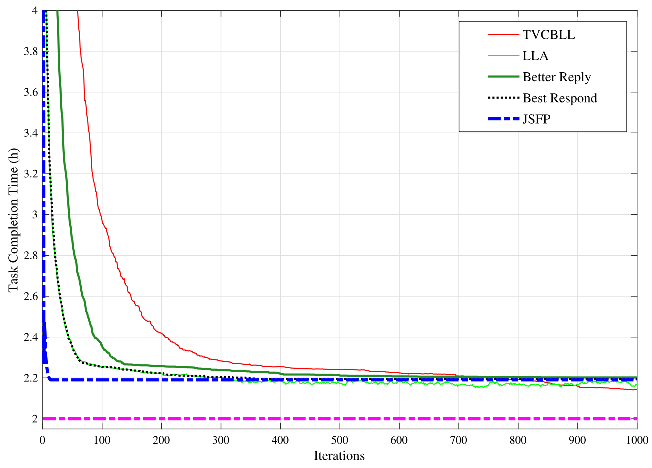


Fig. 3. Convergence performance of 5 algorithms in the static environment with 100 agents and 20 tasks. The pink dotted lines indicate the value of the optimal solution, which is 2 h.

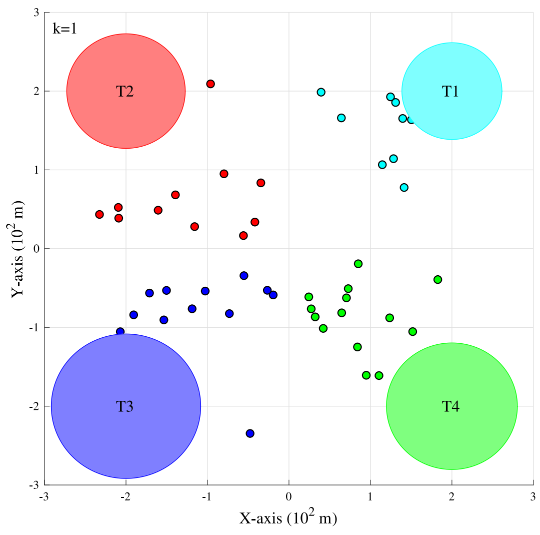
Table 3
Assignment results of different algorithms for the first scenario ($n = 100, m = 20$). Simulations are repeated over 100 runs for each method.

Algorithm	Indicators				
	P_0	*Var*	δ	T	*Cost*
Better reply	15.6%	0.00854	10.12%	2.202 h	0.147 s
Best response	15.0%	0.00724	9.72%	2.194 h	0.140 s
JSFP	18.8%	0.00970	9.52%	2.190 h	1.091 s
LLA	27.1%	0.01405	8.44%	2.169 h	0.346 s
TVCBLL	**34.4%**	0.01095	**7.11%**	**2.142 h**	0.212 s
Optimal	100%	–	–	2 h	–

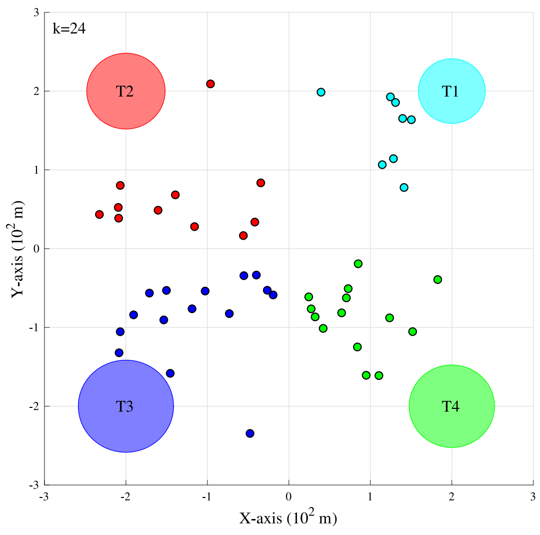
In the dynamic task allocation, only the situation with 48 agents and 4 tasks is considered, where the work capacity of agents and workload of tasks are both initialized as 300. The simulation is implemented in a 600 m × 600 m square area, which is discretized into 6 × 6 grids.

Fig. 4 shows the dynamic assignment results of four phases. Four rounds with different colors, located in the four quadrants of coordinates, represent tasks, and the size of each round represents its workload. Each circle implies an agent, and the color corresponding to the task it has selected. Notice that the agents are randomly distributed at corresponding quadrant, while the coordinates of the agent in Fig. 4 do not represent its actual position, as mentioned before, the location of an agent is the same as its chosen task. Assume that an agent $i \in \mathcal{N}$ executes the task $j \in \mathcal{S}$ for Δt seconds, then the workload of task j decreases $\omega_i \Delta t$ accordingly. It can be seen that the workload of all tasks has been reduced at $k = 24$. Next, to verify the adaptability of the proposed method in response to the unexpected increase of the workload, let the workload of task 3 increase at $k = 25$; thus the area of task 3 in (c) is bigger than that of in (b). As a consequence, some agents prefer to participate in task 3 after they found that the workload of this task is higher than the others, and (c) shows the adjusted assignment at $k = 26$. Then in the following phases, there are only a few agents change their decisions to adjust the assignment slightly, and the final assignment is shown in (d). In the next phases, i.e., $k = 57$, all the tasks are completed.

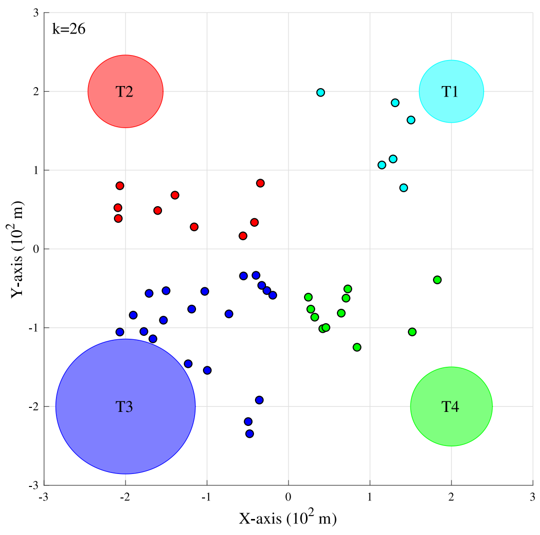
Fig. 5 indicates the remaining workload of each task, the update frequency is 0.02 h, and the initial workloads of tasks are 80, 70, 55, 95, respectively. The slope of each curve reflects its group work capacity, and the smaller the slope, the larger the work capacity for a task. As is shown, the workload of task 3 increased at $k = 25$; thus, the slope of task 3 decreased after the



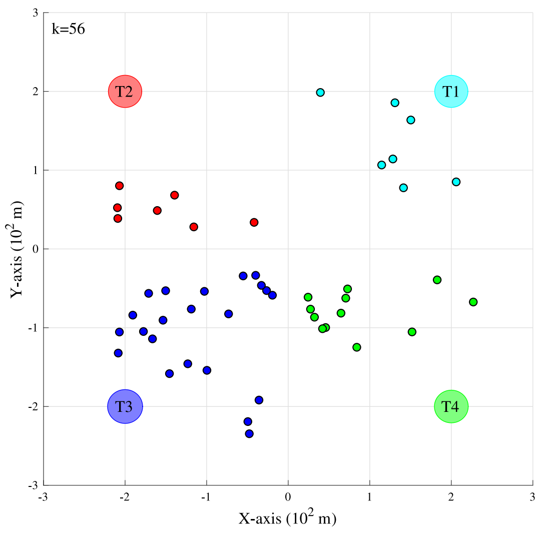
(a) The assignment of stage $k = 1$



(b) The assignment of stage $k = 24$



(c) The assignment of stage $k = 26$



(d) The assignment of stage $k = 56$

Fig. 4. The initial assignment ($k = 1$) and dynamic adjustment ($k = 24, 26, 56$) are shown in the above figures. Four rounds with different colors (i.e., cyan, red, blue, and green respectively) represent the four tasks, and the size of each round represents its remaining workload. Each circle implies an agent, and the color corresponding to the task it has selected. The workload of task 3 has increased at $k = 25$ such that the size of round T3 has grown.

additional workload emerged while the slopes of others increase. It means that some agents decided to participate in tasks 3 at $k = 26$ after they discovered the environmental change. Finally, all the tasks were completed at $k = 57$ almost simultaneously. It is essential that all the tasks were completed at the same time, as presented in Section 3.2. In the static environment where the problem needs only one allocation plan, it must be the optimal assignment that all the tasks completed simultaneously. Since the time of the last completed task determines T, slightly adjusting the assignment to complete tasks simultaneously is expected.

The simulation results in the dynamic environment imply that the proposed framework help each agent make a reasonable choice in response to the unexpected environmental change. With the variation of environment, the chosen agent would quickly evaluate the utility of one trial action/task to decide whether or not to execute it. Consequently, the distributed decision-making game-theoretic model leads to an efficient co-operative behavior, and the proposed dynamic allocation method could adjust the assignment properly to accelerate convergence and could be applied to a more complex environment. However, the proposed method, as discussed previously, is unsuitable for the situations where the number of the task is very large, since the learning algorithm is asynchronous where only one agent is selected, and only one trial action is evaluated for this chosen agent. In summary, the proposed framework provides good adaptability to the dynamic environment, and the TVCBLL algorithm contributes to the convergence of the optimization process.

6. Conclusion

To the cooperation of a swarm robotic system, this paper proposed an efficient game-theoretic framework for addressing the multi-agent dynamic task allocation problem. A large number of agents, which have limited computing and communication capacities, make their own decisions based on local information. The effectiveness of the designed game-theoretic model is proved according to the submodularity of global utility function. Then, the convergence of provided TVCBLL algorithm is also analyzed based on inhomogeneous Markov chain theory. Numerical experiments validate the proposed game-theoretic framework and

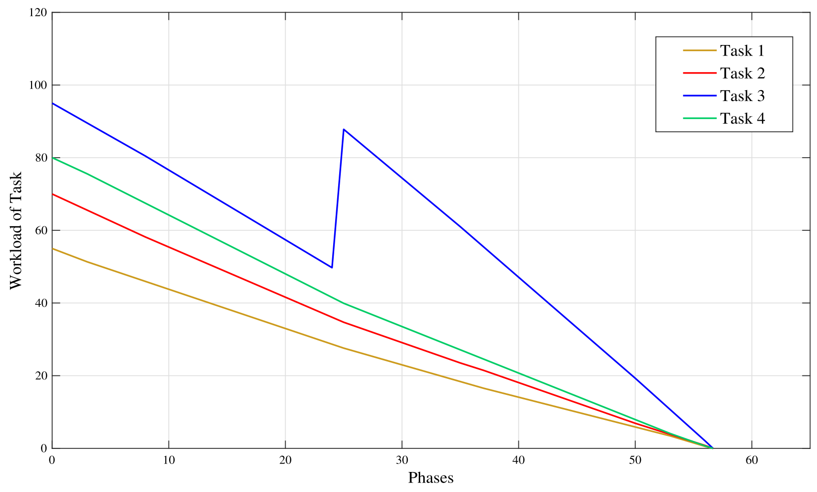


Fig. 5. The curves indicate the workload of 4 tasks with update frequency 0.02 h for dynamic task allocation. The workload of task 3 has increased at $k = 25$. All the tasks are completed at stage $k = 57$, the completion time is 1.14 h.

learning algorithm. In conclusion, the multi-agent dynamic task allocation approach has great potential for many applications, such as forest fire fighting, logistics operation, and surveillance, et al.

The future works will be (1) compare our work with the existing excellent works in applications, (2) improve the effectiveness of the game model, such that the lower bound of Nash equilibrium can be improved, and (3) consider the feasibility of our method in asynchronous environments since it is difficult for every agent to update the state and receives information synchronously.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Lerman K, Jones C, Galstyan A, Matarić MJ. Analysis of dynamic task allocation in multi-robot systems. Int J Robot Res 2006;25(3):225–41.
[2] Jang I, Jeong J, Shin HS, Kim S, Tsourdos A, Suk J. Cooperative control for a flight array of uavs and an application in radar jamming. IFAC-PapersOnLine 2017;50(1):8011–8.
[3] Saska M, Vonásek V, Chudoba J, Thomas J, Loianno G, Kumar V. Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. J Intell Robot Syst 2016;84(1–4):469–92.
[4] Xin B, Chen J, Peng Z, Dou L, Zhang J. An efficient rule-based constructive heuristic to solve dynamic weapon-target assignment problem. IEEE Trans Syst Man Cybern A 2011;41(3):598–606.
[5] Chapman AC, Micillo RA, Kota R, Jennings NR. Decentralised dynamic task allocation: a practical game: theoretic approach. In: Proceedings of the 8th international conference on autonomous agents and multiagent systems-volume 2. International Foundation for Autonomous Agents and Multiagent Systems; 2009, p. 915–22.
[6] Erdelj M, Natalizio E, Chowdhury KR, Akyildiz IF. Help from the sky: Leveraging uavs for disaster management. IEEE Pervasive Computing 2017;16(1):24–32.
[7] Gerkey BP, Matarić MJ. A formal analysis and taxonomy of task allocation in multi-robot systems. Int J Robot Res 2004;23(9):939–54.
[8] Lee DH, Zaheer SA, Kim JH. A resource-oriented, decentralized auction algorithm for multirobot task allocation. IEEE Trans Autom Sci Eng 2015;12(4):1469–81.
[9] Liu L, Shell DA. Optimal market-based multi-robot task allocation via strategic pricing.. In: Robotics: Science and systems, Vol. 9. 2013, p. 33–40.
[10] Monderer D, Shapley LS. Potential games. Games Econom Behav 1996;14(1):124–43.
[11] Li N, Marden JR. Designing games for distributed optimization. IEEE J Sel Top Sign Proces 2013;7(2):230–42.
[12] Marden JR, Shamma JS. Game theory and distributed control. In: Handbook of game theory with economic applications, Vol. 4. Elsevier; 2015, p. 861–99.
[13] Blume LE. The statistical mechanics of strategic interaction. Games Econom Behav 1993;5(3):387–424.
[14] Shamma JS, Arslan G. Dynamic fictitious play, dynamic gradient play, and distributed convergence to nash equilibria. IEEE Trans Automat Control 2005;50(3):312–27.
[15] Hart S, Mas-Colell A. A simple adaptive procedure leading to correlated equilibrium. Econometrica 2000;68(5):1127–50.
[16] Fujishige S. Submodular functions and optimization, Vol. 58. Elsevier; 2005.
[17] Wolpert DH, Tumer K. An introduction to collective intelligence. 1999, arXiv preprint cs/9908014.
[18] Badreldin M, Hussein A, Khamis A. A comparative study between optimization and market-based approaches to multi-robot task allocation. Adv Artif Intell 2013;2013:12.
[19] Khamis A, Hussein A, Elmogy A. Multi-robot task allocation: A review of the state-of-the-art. In: Cooperative robots and sensor networks 2015. Springer; 2015, p. 31–51.
[20] Shi HY, Wang WL, Kwok NM, Chen SY. Game theory for wireless sensor networks: a survey. Sensors 2012;12(7):9055–97.
[21] Chopra S, Notarstefano G, Rice M, Egerstedt M. A distributed version of the hungarian method for multirobot assignment. IEEE Trans Robot 2017;33(4):932–47.
[22] Attiya G, Hamam Y. Task allocation for maximizing reliability of distributed systems: A simulated annealing approach. J Parallel Distrib Comput 2006;66(10):1259–66.
[23] Page AJ, Keane TM, Naughton TJ. Multi-heuristic dynamic task allocation using genetic algorithms in a heterogeneous distributed system. J Parallel Distrib Comput 2010;70(7):758–66.
[24] Choi HL, Brunet L, How JP. Consensus-based decentralized auctions for robust task allocation. IEEE Trans Robot 2009;25(4):912–26.
[25] Saad W, Han Z, Basar T, Debbah M, Hjorungnes A. Hedonic coalition formation for distributed task allocation among wireless agents. IEEE Trans Mob Comput 2011;10(9):1327–44.
[26] Jang I, Shin HS, Tsourdos A. Anonymous hedonic game for task allocation in a large-scale multiple agent system. IEEE Trans Robot 2018;(99):1–15.
[27] Li P, Duan H. A potential game approach to multiple uav cooperative search and surveillance. Aerosp Sci Technol 2017;68:403–15.
[28] Liu X, Xie Y, Li F, Gui W. Admissible consensus for homogenous descriptor multiagent systems. IEEE Trans Syst Man Cybern: Syst 2019.
[29] Jin L, Li S. Distributed task allocation of multiple robots: A control perspective. IEEE Trans Syst Man Cybern: Syst 2016;48(5):693–701.
[30] Vetta A. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In: The 43rd annual IEEE symposium on foundations of computer science, 2002. proceedings. IEEE; 2002, p. 416–25.
[31] Marden JR, Arslan G, Shamma JS. Joint strategy fictitious play with inertia for potential games. IEEE Trans Automat Control 2009;54(2):208–20.
[32] Young HP. Individual strategy and social structure: An evolutionary theory of institutions. Princeton University Press; 2001.
[33] Young HP. Strategic learning and its limits. OUP Oxford; 2004.
[34] Marden JR, Shamma JS. Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation. Games Econom Behav 2012;75(2):788–808.

[35] Hajnal J, Bartlett M. Weak ergodicity in non-homogeneous Markov chains. In: Mathematical proceedings of the cambridge philosophical society, Vol. 54. Cambridge University Press; 1958, p. 233–46.
[36] Isaacson DL, Madsen RW. Markov chains theory and applications. 1976.
[37] Tatarenko T. Proving convergence of log-linear learning in potential games. In: 2014 American control conference. IEEE; 2014, p. 972–7.
[38] Marden JR, Arslan G, Shamma JS. Connections between cooperative control and potential games illustrated on the consensus problem. In: 2007 European control conference (ECC). IEEE; 2007, p. 4604–11.
[39] Sun C. A time variant log-linear learning approach to the set k-cover problem in wireless sensor networks. IEEE Trans Cybern 2018;48(4):1316–25.
[40] Tatarenko T. Log-linear learning: Convergence in discrete and continuous strategy potential games. In: 53rd IEEE conference on decision and control. IEEE; 2014, p. 426–32.