

A mutual-selecting market-based mechanism for dynamic coalition formation

*International Journal of Advanced
Robotic Systems*
January-February 2018: 1–10
© The Author(s) 2018
DOI: 10.1177/1729881418755840
journals.sagepub.com/home/arx



Bing Xie, Shaofei Chen, Jing Chen and LinCheng Shen

Abstract

This article presents a novel market-based mechanism for a dynamic coalition formation problem backgrounded under real-time task allocation. Specifically, we first analyze the main factors of the real-time task allocation problem, and formulate the problem based on the coalition game theory. Then, we employ a social network for communication among distributed agents in this problem, and propose a negotiation mechanism for agents forming coalitions on timely emerging tasks. In this mechanism, we utilize an auction algorithm for real-time agent assignment on coalitions, and then design a mutual-selecting method to acquire better performance on agent utilization rate and task completion rate. And finally, our experimental results demonstrate that our market-based mechanism has a comparable performance in task completion rate to a decentralized approach (within 25% better on average) and a centralized dynamic coalition formation method (within 10% less on average performance).

Keywords

Dynamic coalition formation, distributed method, market-based mechanism, mutual-selecting, multi-agent

Date received: 22 June 2017; accepted: 21 August 2017

Topic: Mobile Robots and Multi-Robot Systems

Topic Editor: Lino Marques

Associate Editor: M Bernadine Dias

Introduction

Since distributed multi-agent systems (DMASs) have been increasingly employed in real-world problems, such as disaster rescue,¹ sensor surveillance,² military operations,^{1,3,4} and wireless sensor networks,^{5,6,7} tasks with heterogeneous requirements usually emerge dynamically in those applications and need agents to cooperate to meet those requirements for successful execution. However, due to complex constraints both of task requirements and agent-employed environment, how to allocate agents to form real-time coalitions for complex task completion becomes a new challenge in research. For example, in a disaster rescue scenario, a rescue task requires first aid and transportation, where first aid helps to swiftly handle the wounded and ambulances help to transport the wounded to target areas. Some wounded need aid first and then being transported to

hospitals, which means such tasks demand a compounded coalition with heterogeneous resources for execution. Obviously, centralized approaches are not feasible or reliable in those systems because in such a DMAS each agent needs to move to the task location when assigned a task, which means the network between the central distributor and other agents is not guaranteed to always work well.^{8,9} To deal with this issue, we focus on the use of a distributed

Artificial Intelligence Lab, National University of Defense Technology, Changsha, Hunan, People's Republic of China

Corresponding author:

Bing Xie, Artificial Intelligence Lab, National University of Defense Technology, Changsha, Hunan, People's Republic of China.

Email: snn_xie12@163.com



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License

(<http://www.creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

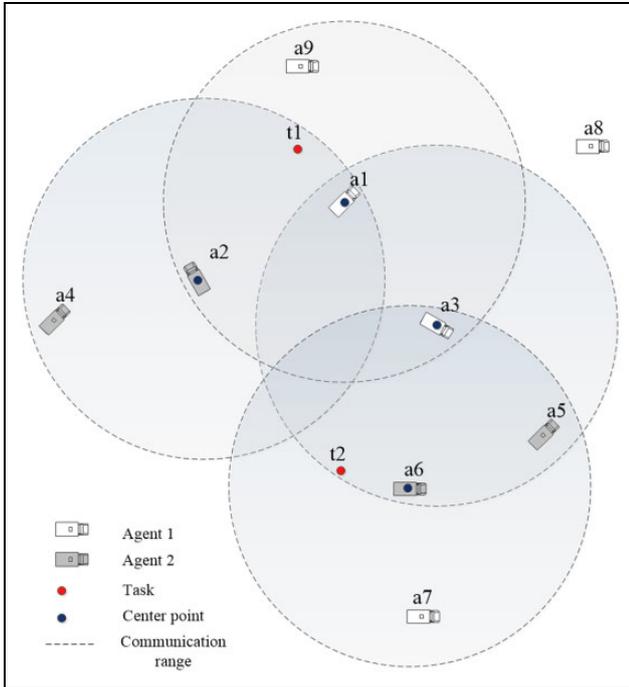


Figure 1. Neighborhood networks.

approach, in which individual agents negotiate directly with each other to form dynamic coalitions.

Although lots of researchers have been working on coalition formation problem, many of them focus on finding an optimal coalition structure under an ideal communication,^{10,11,12} and agents are usually assumed to be capable of transporting information to any other agents freely. Moreover, structures of networks employed in those applications are fixed though agents need to change their locations for executing tasks, which is apparently infeasible in reality. Generally, agents are with limited communication range that restraints their communication ability, and they usually communicate with each other via neighborhood networks, such as social networks^{3,4} which allow agents communicate with their neighbors directly, as in Figure 1. However, forming coalitions for timely emerging tasks under such networks faces many new challenges. In this article, we intend to design an approach on dynamic coalition formation (DCF), in which autonomous agents can cooperate with each other on their own decisions, with a goal of forming a global optimal coalition structure or an approximate global optimal coalition structure to execute timely emerging tasks.

Against this background, the DCF problem that we intend to address is about a set of agents assigning themselves to coalitions and executing dynamic tasks, without a central allocator. Each task has a deadline and a specific processing requirement over resources (e.g. injurers need first aid before 3 pm and then to be transported to a hospital in 30 min). Besides, tasks are dynamic, which means tasks reveal gradually over time in the system. This, in turn, means that we need a practical mechanism in response to

a real-time task flows in a timely manner. Moreover, as agents change their locations dynamically, we need to adopt a suitable network for information transportation. And, as the full set of tasks is not known at the outset, an agent needs to negotiate with other agents over a sequence of tasks to execute, with the goal of completing as many real-time tasks as possible, and of improving agent utilization.

Hereafter, based on all the objectives, we tackle this problem as follows. First, we define a global utility function which is constructed as if we were defining it for a centralized coalition formation problem. And then, we propose an approximating global utility function for a feasible calculation of DCF with a series of static coalition formation utility function. Then, we cast the approximating global utility as a sum of agent individual utilities, which enables agents to make their own decisions based on their local information. Finally, we propose a distributed mechanism to form dynamic coalitions for dynamic real-time tasks.

Given this context, our main contributions are presented in the following ways:

- We employ a practical social network model in a distributed multi-agent system for communication. As agents change their locations dynamically, we design a novel mechanism that provides reliable communication for agents forming task coalitions.
- We formulate the DCF problem based on coalition game, and then define a global utility function of the DCF. In addition, we utilize a series of static coalition formations to approximate the DCF process, based on which we propose an approximating global utility function for the feasible calculation of DCF.
- We propose a market-based mutual-selecting algorithm for agent DCF, which supports individual agents to make decisions based on local information and provides resource optimization in forming coalitions.

The remainder of the article is organized as follows: The “Related work” section introduces some related work on DCF, and the “Dynamic coalition formation” section describes our model of DCF and give out relative utility functions. In the “Distributed mechanism for DCF” section, we present our communication mechanism and a distributed algorithm for DCF. Experimental results and analysis are presented in the “Experiment analysis” section and the article is concluded in the “Conclusion” section.

Related work

Coalition formation, widely studied in the game theory and economics, has attracted much attention in the artificial intelligence area as a method of forming sophisticated agent teams to perform certain complex tasks.¹³ It is an

instance of a partition problem which is known as an NP-complete.¹⁴ Being an application of the coalition game theory, a coalition formation problem describes the possible occurring outcomes when the players decide to participate in a group of cooperative peers, referred to as coalitions. Static coalition formation has also been addressed in multi-agent systems where its goal is to find a coalition structure that maximizes the global utility of all coalitions.^{10,15} Yet, a DCF problem, as a kind of coalition formation problems, demands agents to spontaneously form coalitions to complete emerging tasks timeously and economically.

To deal with DCF problems, Shehory and Kraus in 1998 studied a computational task allocation problem via coalition formation¹³ and proposed algorithms, which are greedy distributed set partitioning and covering algorithms with low ratio bounds, for task allocation among computational agents in RETSINA (REusable Task-based System of Intelligence Networked Agents). In 2002, Klusch and Gerber, from the German Research Centre for Artificial Intelligence, studied DCF among rational software agents in open, heterogeneous, and distributed environments,¹⁶ such as the Internet and Web, and defined a set of cooperation methods, schemes, and technologies to beneficially cope with the DCF problem among agents in those environments. However, the DCF-S scheme does not guarantee an optimal solution in general. Bayram and Isil Bozma, from Bogazici University, emphasized on finding a feasible method of forming flexible coalitions for dynamic tasks.² They proposed a centralized algorithm to solve the issue.

As DCF has been increasingly employed in decentralized multi-agent systems, more attention is paid to distributed methods. Choi et al. concerned the dynamic decentralized task allocation problem in military operations.⁸ They presented a consensus-based decentralized auction algorithm utilized by a market-based decision strategy for noncooperative task allocation, and extended the algorithm for solving heterogeneous DCF.^{1,3,4} But the structures of coalitions in those studies are fixed, which constraints the flexibility of their algorithms. Chapman and Kota, from the University of Southampton, propose a distributed algorithm for a DCF problem in disaster rescue.⁹ They formulate the problem as a Markov game, analyze challenges in solving the Markov game, and then use a series of potential games to approximate the Markov game they built for figuring out a possible solution. The distributed algorithm they propose is capable of finding a solution to DCF. However, the algorithm assumes that all agents could communicate with any other agents, which is infeasible.

Thus, considering a feasible communicating model and a flexible coalitional structure for application in real DMASs, Gaston and desJardins, from the University of Maryland, Baltimore County, propose an agent-organized network model based on neighbor networks for communication in DMAS to form coalitions and execute dynamic

task sequences based on its applicability.¹⁷ They design adaptation policies for the social network to optimize the process of coalition formation, and propose a stochastic algorithm of coalition formation. In every time step, each agent could decide to either attempt to join a coalition or adapt its local network structure. Then, Glington and Scerri, from Carnegie Mellon University, further this research about self-adaption social networks on DCF under a distributed circumstance of sensor coalition formation problem.¹⁸ Their research mainly focuses on a heterogeneous agent coalition formation problem with time constraints. They propose two main kinds of policies: the performance-based policy and the structure-based policy. As in such a system, agents could make decisions to change their neighbors as they want based on their own profit assessment without considering reality limits. Through experiment analysis, they find that the structure-based policy has its roof of improvement while the performance-based policy works more effectively on the entire performance of the algorithm.

Based on the social network in DMASs, Ye and Zhang probe into a DCF problem of sensor surveillance.¹⁹ They present a novel coalition formation mechanism that is constructed by a market-based algorithm under a distributed social network. In their mechanism, agents are cataloged into two types: the initiator who initializes a task allocation and the participant who accepts the announced task. The initiator releases tasks it initialized through its social networks and forms coalitions. The participant chooses its task among the received tasks by a market-based algorithm for optimization. Once the tasks are refused by one participant it will be passed to the neighbors of the participant for coalition formation. And they also introduce a penalty mechanism for breaking coalitions formed to adapt the dynamic task environment, which helps reduce the cost of the coalition formation process and enhance the effectiveness of the global system. However, those distributed methods and mechanisms are not suitable for a DCF under an environment with agent varying locations.

Therefore, our method of solving DCF problems is to design a mechanism for approximating the optimal solution. And our approach is motivated by these ideas where the process of finding optimal agent coalitions is modeled as a coalition formation game. Our approach to approximate the DCF is motivated by a somewhat similar technique for producing approximating solutions to a Markov game using a series of potential games.⁵

Dynamic coalition formation

In this section, we utilize a coalition formation model with dynamic tasks,² beginning with a DCF problem formulation, and then defining a coalition character function and approximating global utility function and an agent individual utility function based on Shapley value definition.²⁰ At the end of this section, we introduce a social network

model^{17,18} applied in our study, and propose a negotiation mechanism and corresponding algorithms for DCF.

Problem formulation

A set of agents with heterogeneous resources are distributed in a target area for responding to dynamic tasks. Considering practical limits on agent communication range, those agents communicate with each other through a social network built on neighbor agents within the communication range. Tasks, with deadlines, heterogeneous requirements, and unpredicted locations, emerge dynamically over time. Agents are assumed to stay motionless until they are assigned to execute missions. And our objective is to find coalitions to respond to those dynamically emerging tasks.

Based on coalition game theory, the problem we study can be defined as $G = \langle A, T, u, (\succ_i)_{i \in A} \rangle$

- Agents, $A = \{a_1, \dots, a_{|A|}\}$ is a set of agents.
- Tasks, $T = \{t_1, \dots, t_{|T|}\}$ is a set of tasks.
- Utility: u is a utility function portraying agent a_i 's payoff and cost. Here cost refers to consuming time on moving.
- Preference: \succ_{a_i} describes agent a_i 's preference over coalitions. For instance, $C_k \succ_{a_i} C_j$ means that, for agent a_i , it prefers to join coalition C_k than coalition C_j .

And more specifically, notations of an agent and a task are given as follows.

For an agent $a_i \in A$, it can be defined as a tuple $a_i = \langle l, s, \mathbf{r}_{a_i} \rangle$, where l and s denote the location and status of agent a_i , and $\mathbf{r}_{a_i} = \langle r_1, \dots, r_n \rangle$ represents its resources. For task $t_i \in T$, it can be defined as a tuple $\langle l, s, \mathbf{r}_{t_i}, \tau_{dl}, w \rangle$, where l and s denote the location and status of task t_j , $\mathbf{r}_{t_j} = \langle r_1, \dots, r_n \rangle$ represents its requirement over resources, τ_{dl} and w denote the deadline and reward of task t_j , respectively. A set of agents that can perform task t_j is defined as C_j , which is a coalition on the agent set A . And detailed notations on the status of agents and tasks will be given in section coalition character function.

Here, all agents are assumed to be self-interested but honest. Thus, each agent shares its information with its neighbors, and has a neighbor information list, which is also shared with agents in its neighborhood. Note that, "self-interested" does not mean agents have intentions to violate the utilities of other agents, but is only used for optimization here. Since agents are honest, the information they share with each other is reliable. And we assume that the utility of each task is transferable, which means utilities can be transferred among coalition members freely. Also, all task rewards are assumed to be given when tasks emerge, and mutually independent.

Thus, how the executing coalition constructs and when the task will begin execution both will impact the utility of

corresponding coalitions. Obviously, resource excessiveness or resource insufficiency cause low agent utilizations. Also, for a certain task the earlier it gets started the more efficient the system will be. Therefore, based on those considerations, we can define a coalition character function,²¹ which is the fundament of a global utility function, in the "Coalition character function" section.

Coalition character function

Coalition character function is a function $v : 2^A \rightarrow \mathbb{R}$ that assigns a value to every possible coalition, revealing a given coalition C_j 's utility with a given task t_j via encoding resource sufficiency, resource excessiveness, and members' proximity to the task location. It can be defined as

$$v(C_j, t_j) = \begin{cases} \beta(C_j, t_j)^{-\tau_{t_j}^C}, & \text{if } C_j \text{ satisfy the requirement} \\ 0, & \text{else} \end{cases} \quad (1)$$

where the term $\beta(C_j, t_j)$ is defined as

$$\beta(C_j, t_j) = r(t_j) \cdot \left(1 - \left| \frac{h(C_j) - h(t_j)}{h(t_j)} \right| \right) \quad (2)$$

where $r(t_j)$ is task t_j 's reward, which is a given constant when t_j emerges in the system. And function $h(C_j)$ can be defined as $h(C_j) = \boldsymbol{\omega} \cdot \mathbf{r}^{C_j}$, where $\boldsymbol{\omega}$ is a weight vector and \mathbf{r}^{C_j} is a heterogeneous resource vector presenting resources possessed by coalition C_j . So, $h(C_j)$ can also be expressed as $h(C_j) = \omega_1 r_1 + \dots + \omega_i r_i + \dots + \omega_n r_n$ with one item representing one resource, where r_i is a quantized resource and ω_i is its weight.

And based on the notion of coalition character function, we can have the preference relation \succ defined in detail in theorem 1.

Theorem 1 [preference]. For agent $a_i \in A$, $C_k \succ_{a_i} C_j$ if and only if $v(C_k) > v(C_j)$, where C_k and C_j are both coalitions obtaining a_i .

Apparently, a coalition utility will be reduced when the coalition ability is beyond or below the task requirement. Meanwhile, $\tau_{t_j}^C$ is the estimated time when the coalition could arrive at the task location, and generally we use the time of the latest agent arriving. Obviously, the earlier a coalition arrives, the greater a coalition utility is.

Approximating global utility

Based on coalition character function, we can define a global utility of this coalition game as follows. The goal of a coalition game is to find an optimal coalition structure to maximize global utility

$$u_g(CS^*) = \max \sum_{t_j \in T} \sum_{C_j \in CS^*} v(C_j, t_j) \quad (3)$$

As a DCF problem, the coalition formation process goes with time, and un-emerging tasks remain unknown. Thus,

finding a solution to a dynamic coalition game seems to be impossible as now, and we try to approximate it with a series of static coalition formation game to find an approximating solution.

At each time step, a DCF problem can be viewed as a static coalition formation problem with known tasks emerged. That is, for any time period $[\tau, \tau + \xi]$, where τ is the starting time of this period, and ξ is a time amount, if ξ is a small time amount, we can view this coalition formation as static coalition formation. Then we have

$$u_g^{\tau, \xi}(CS_{[\tau, \tau + \xi]}^*) = \max \sum_{t_j \in T_{[\tau, \tau + \xi]}} \sum_{C_j \in CS^*} v(C_j, t_j) \quad (4)$$

where $CS_{[\tau, \tau + \xi]}^*$ is a current optimal coalition structure for period $[\tau, \tau + \xi]$, $u_g^{\tau, \xi}(CS_{[\tau, \tau + \xi]}^*)$ is a current global utility, and $T_{[\tau, \tau + \xi]}$ is a set of tasks emerging in this time slice.

In this way, at each time step, the DCF problem is approximated by a static coalition formation game with complete task information over the next ξ time steps. Then, the global utility functions in each approximating coalition formation games are defined as equation (4). Based on the definition of Shapley value, we can design a function to distribute the total utility of a coalition to its members. Each assignment of the utility for an agent can be viewed as an agent individual utility.

Individual utility definition

As assumed above, coalition utility could be transferable among agents and agents can only participate in one task at a time. Based on the definition of Shapley value for each agent, we can define agent individual utility for one task. Our main principle of the distribution is that, what degree is the contribution of each coalition member on the task. Thus, for agent $a_i \in C_j$, its individual utility on task t_j that coalition C_j executes can be defined as

$$u_{t_j}(a_i) = v(C_j, t_j) \cdot g(a_i, t_j) \quad (5)$$

where $g(a_i, t_j)$ is the contribution function of agent a_i in coalition C_j on task t_j . And the entire utility of agent a_i depends on tasks it participates

$$u(a_i) = \sum_{t_j \in T_{a_i}} v(C_j, t_j) \cdot g(a_i, t_j) \quad (6)$$

where T_{a_i} is a set of all tasks that agent a_i participates during the whole process. And based on its individual utility definition, we can rewrite the approximating global utility function in equation (3) as

$$\begin{aligned} u_g(CS^*) &= \sum_{t_j \in T} \sum_{C_j \in CS_{[\tau, \tau + \xi]}^*} v(C_j, t_j) \\ &= \sum_{t_j \in T} \sum_{C_j \in CS_{[\tau, \tau + \xi]}^*} \sum_{a_i \in C_j} u_{t_j}(a_i) \end{aligned} \quad (7)$$

With equation (7), we can design a distributed algorithm with agent undertaking computation and making

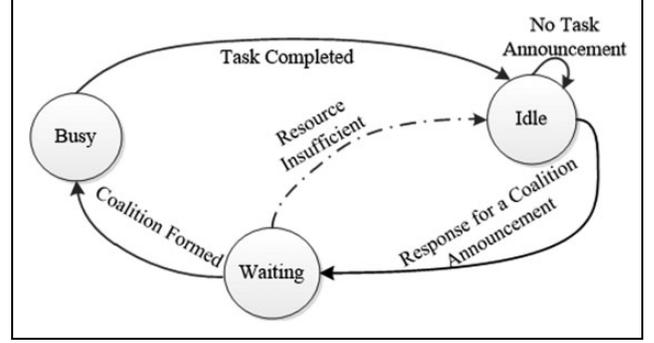


Figure 2. Agent state transition.

self-decision to figure out an optimal or suboptimal solution.

A distributed mechanism for DCF

In this section, we design a distributed mechanism to find an optimal or a suboptimal solution for our problem. First, we introduce a kind of network adopted in our study, social network, and give some formal definitions of related concepts. Then, we present the proposed algorithm, MSMA (mutual-selecting market-based algorithm), introducing its framework, working process, and so on. And finally, we give the pseudocodes of our main algorithms for better comprehension of the proposed mechanism.

Social networks

To deal with the issue of DCF in a structured network, we first give the definition of social networks, as in definition 1.

Definition 1 [Social networks]. An agent network consists of a set of independent agents, namely $A = \{a_1, a_2, \dots, a_n\}$ and a set of compatible relations $R \subseteq A \times A$. R presents a neighbor relation between agents. So, $\langle a_i, a_j \rangle \in R$ means if and only if a_j is a neighbor of a_i . Since R is a compatible relation, which means R is reflexive and symmetric, it can be achieved in two forms,

- $\forall a_i : a_i \in A \Rightarrow \langle a_i, a_j \rangle \in R$;
- and $\forall a_i, a_j \in A : \langle a_i, a_j \rangle \in R \Rightarrow \langle a_j, a_i \rangle \in R$.

Agents in a social network transport information via their neighbors, which are within their communication range, to those out of their neighborhoods. Because the working environment changes dynamically, once agents are allocated certain tasks, they start moving and their neighbors keep changing during the movement.

Based on the definition of social networks, we give out some definitions used in our mechanisms as follows.

Definition 2 [Sub-neighbors]. For each agent, sub-neighbors are a set of agents which are not in that agent's communication range but are neighbors of their neighbors. Namely,

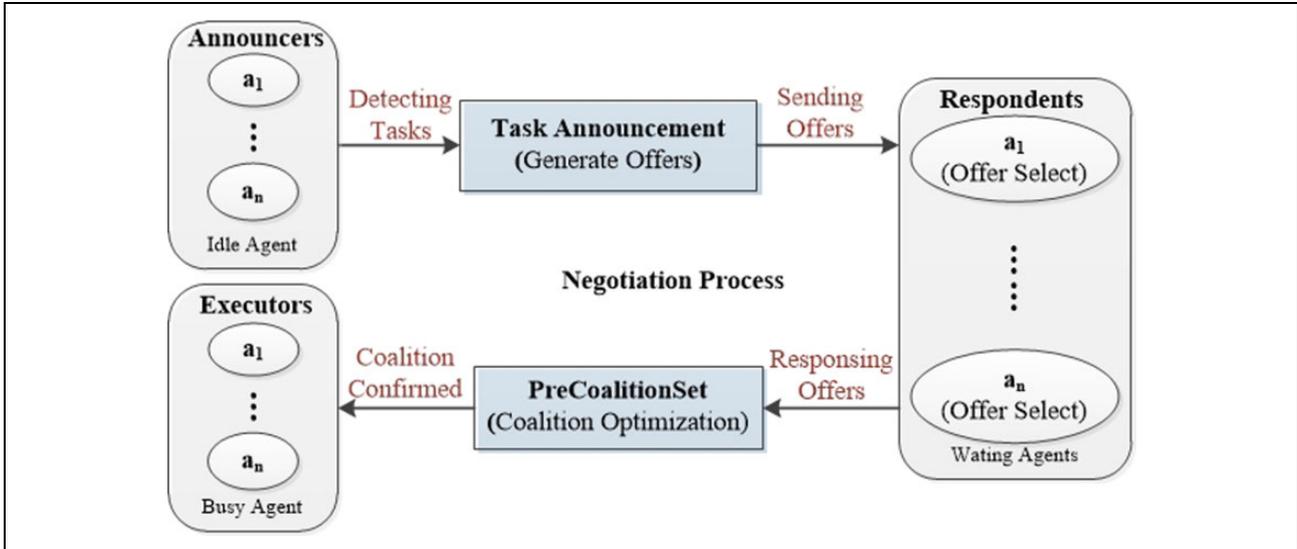


Figure 3. The framework of mechanism for DCF. DCF: dynamic coalition formation.

sub-neighbors are a set of agents that can communicate with an agent indirectly through their neighbors once.

As shown in Figure 1, for agent a_1, a_2, a_3 , and a_9 are its neighbors, and a_4, a_5 , and a_6 are its sub-neighbors. Apparently, an agent communicates with the neighbors of its neighbors by utilizing its neighbors as communication relay stations. Whether information can be passed over the system's social network via sub-neighbors depends on how the social network is constructed, or how the social network is connected.

Definition 3 [Announcer, respondent, executor]. The agent which finds a task based on its location initializing the coalition formation process is called an announcer; the agent which accepts the call of announced coalition formation is called a respondent; and the agent which is selected to form the final corresponding coalition for task execution is called an executor.

As in the DCF-S scheme,⁸ the coalition leading agent (CLA) is designed for leading a coalition and acting on behalf of its members.⁸ Here, agent announcers are designed for optimizing the coalition formation process. It should be noted that the roles do not imply any architecture in a coalition and are only used to distinguish different responsibilities of agents.

Definition 4 [Agent status]. There are three states of agents in the network, $s = \{idle, waiting, busy\}$. An agent can only be in one of the three states in any time step. When an agent is in a formed coalition, the state of the agent is busy; when an agent is in the negotiation its state is waiting; an agent in idle state has not been assigned any task nor any negotiation process for forming coalitions and an agent state transition is shown in figure 2.

Definition 5 [Task status]. There are three kinds of task states, $s = \{waiting, assigned, completed\}$. A task that is

waiting for its coalition before its deadline is in waiting status. A task is in assigned status when a qualified coalition responds to it. And a task is completed after the finishing time that it requires.

Note that when tasks are completed, responding coalitions are released at once and agents in those coalitions are labeled as idle. And after formalizing the social network, the principle of our coalition formation mechanism will be depicted.

Distributed mechanism design

In a social network, agents are distributed for responding tasks, thus they make decisions on their local information about the system. Based on this cognition, we design a distributed mechanism for DCF and the framework of our mechanism is designed as in Figure 3.

Figure 3 illustrates how the negotiation mechanism works. Its main idea comes from a mutual-selecting market principle, with which sellers and buyers both do selections on their provided choices. And we give notion of PreCoalitionSet in definition 6.

Definition 6 [PreCoalitionSet]. A set of agents which accept offers from a coalition formation announcer of task t_j is denoted as PreCoalitionSet of task t_j .

As mentioned above, agents responding to the coalition forming announcement of task t_j form a PreCoalitionSet, total resources of which might be over t_j 's requirement. Therefore, announcers implement the selection process to form an optimal coalition for executing task t_j and generating a greater utility.

Definition 7 [The negotiation protocol]. The agreement of the negotiation mechanism consists of the following three parts:

- announcers of each task generate and send offers to their neighbors;
- then, agents that receive offers hold auctions to select task offers and return their response to respective announcers;
- finally, announcers form the optimal coalition for tasks they receive from the respective responding agents, which could be defined as PreCoalitionSet in definition 6.

Announcements sent from announcers are denoted as Offers, which includes information of tasks, and a responding utility the receiver will get if participating in the task coalition. Thus, we can use a tuple $\langle t_j, a_i, nb_{a_i}^k \rangle$ to describe Offer. Here $nb_{a_i}^k$ is a receiver of announcer a_i . Evidently, offers are related to agent a_i 's individual utility; moreover, the time when an agent arrives at the location of task t_j also impacts the offer. It is obvious that a coalition gains a greater utility when its members arrive earlier. Hereafter, we take the two factors mentioned above into account, and give the definition of Offer as

$$\text{Offer}(t_j, nb_{a_i}^k) = u(nb_{a_i}^k) = \text{value}(C_j, t_j) \cdot g(nb_{a_i}^k, t_j) \quad (8)$$

where $g(nb_{a_i}^k, t_j)$ is a contribution function about how coalition utility is allocated among coalition members, as mentioned in equation (5). The definition of $g(nb_{a_i}^k, t_j)$ is given as

$$g(nb_{a_i}^k, t_j) = \omega \cdot \frac{r^{a_i}}{r^{t_j}} \quad (9)$$

where ω is a weight vector of resources with each item representing a weight on the corresponding resource. The main process of our mechanism is given in detail in algorithm 1. And algorithm 1 is our MSMA.

As introduced in algorithm 1, we design a two-sided market-based algorithm, which calculate distributedly for solve the DCF. For task t_j in time step t in line 2, a random idle agent a_i is selected as the announcer for coalition formation process of t_j . Then, the announcer updates its status to waiting, and generates offers for its receivers. Considering the constraint of physical distance and the requirement on arriving time, announcers are set to send offers to a set of agents composed of its neighbors and sub-neighbors, which helps on search pruning. Note that, offers for sub-neighbors are transported via its neighbors. Receivers select their optimal offer from those they receive, and reply corresponding announcers to form a PreCoalitionSet. Once the PreCoalitionSet has been formed, announcers do the process of coalition optimization to maximize the coalition utility. When final coalitions are formed, members in those coalitions change their status to busy and start to proceed tasks.

When the task requirement can be satisfied by the PreCoalitionSet, the announcer optimizes coalition through forming a greatest coalition utility out of the

Algorithm 1. Coalition formation mechanism at time τ .

```

1 For  $\forall a_i \in A$  call Algorithm 1
2 For task  $t_j$  in time step  $t$ 
3   Update(Neighbors); //update neighbor sets of agents
4   Update(Sub-Neighbors); //update sub-neighbor sets
5   Select a random idle agent  $a_i$  to be the announcer;
6   State( $a_i$ )  $\leftarrow$  Waiting;
7   While  $t < \text{DeadLine}(t_j)$ 
8     if  $r(t_j)$  is not fulfilled do
9       For  $\forall a_j \in N_{a_i}$ 
10        SendOffers( $a_i, a_j$ ) // from  $a_i$  to  $a_j$ 
11      End for
12    End if
13  End For
14  For each  $a_i \in A$  in State Waiting
15    OfferSelect( $a_i$ ) // offer selection
16  End for
17  For each announcer
18    If  $t < \text{DeadLine}(t_j)$ , do
19      PreCoalitionSetConstruct() //waiting for response
20      If  $r(\text{PreCoalitionSet}_{t_j})$  satisfies  $t_j$ 's requirement, do
21        CoalitionOpt( $t_j$ ) //optimize coalitions
22      End if
23    End if
24  End For

```

Algorithm 2. Coalition optimization for task t_j at time τ .

```

1 Start
2 Check the PreCoalitionSet  $PCS_{t_j}$ 
3 if  $\text{resource}(PCS_{t_j}) \geq \text{requirement}(t_j)$ 
4   For each coalition  $C_j \in 2^{PCS_{t_j}}$ 
5     if  $\text{resource}(C_j) \geq \text{requirement}(t_j)$ 
6        $C_j^* = \text{maxvalue}(C_j)$ 
7     End for
8   End if
9 End

```

PreCoalitionSet to approximate the global utility. Or else, the coalition for task t_j forms unsuccessfully and will be announced in next time step until its deadline.

As Gerding et al.^{22,23} and Gerkey and Mataric²⁴ present, the auction algorithm preforms well in optimizing utility functions of discrete parameters. Thus, we use the auction algorithm for the negotiation process. Responders choose the offer with maximum individual utility, and announcers choose those responders providing the maximum utility of the corresponding coalitions. Thus, we design the coalition formation algorithm with auctions, as in algorithm 2.

Algorithm 2 is an algorithm for coalition optimization, which is a sub-processing in algorithm 1. Each announcer proceed algorithm 2 after receiving responding during a certain time.

Exprimment analysis

We now discuss the evaluation of our MSMA based on a comparison with two established algorithms, namely, the

Table 1. Simulation settings in experiment.

Parameter	Value	
Number of agents	Senario 1	100
	Senario 2	70
	Senario 3	50
Number of resource types	2	
Ranges of agent resource	Resource 1	[0, 6]
	Resource 2	[0, 6]
Ranges of task resource	Resource 1	[0, 20]
	Resource 2	[0, 20]

OPGAA (overlapping potential game approximation algorithm)⁵ and the CGCFA (centralized greedy coalition formation algorithm)¹⁷ in the same simulation environment. In this section, we first present simulation settings in our experiment, including agent number and resource type. Then, we define two evaluation functions to compare performances of the three algorithms. Finally, we give out the evaluation function results for comparison and the analysis on the difference.

Simulation settings

For simplification, we assume that each agent possesses two kinds of resources and the quantity of its carried resources varies randomly, and that tasks have different requirements on resources, and their locations are generated randomly.

When allocating heterogeneous agents to dynamic heterogeneous task flows, task requirements usually could not be satisfied exactly due to the heterogeneity and the distribution of agents. Therefore, the total resources a coalition possessed for a task might be over the requirement, which would impact task completion directly. To achieve statistical significance, each experiment was run 20 times with different task flow simulations in three scenarios.

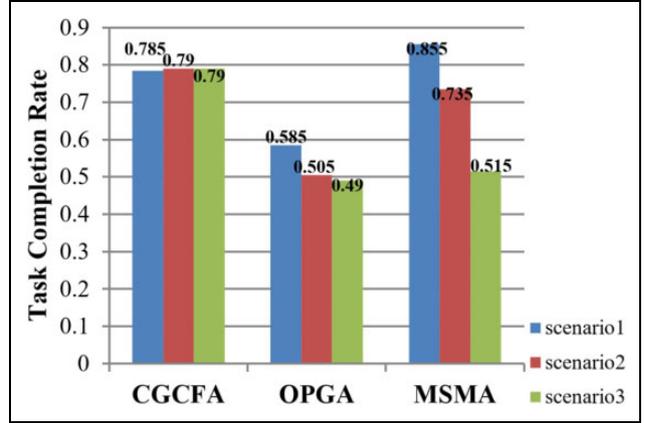
We give the detailed settings in our experiments in Table 1.

Evaluation functions

In scenario 1, there were sufficient agents; and in scenarios 2 and 3, the agents were reduced by 30% and 50%, respectively. Also, we adopted standard scores, like task completion and resource utilization, to review the performances of these three algorithms. In the two distributed algorithms, the communication range was restricted to the same fixed radius. In the OPGAA, its stochastic parameter was set as 0.7.

Here, we use task completion performance R_{tc} and resource utilization rate R_{ru} to evaluate the preformation of DCF algorithms. They are defined as

$$R_{tc} = \frac{N_c}{N_t} \quad (10)$$

**Figure 4.** Task completion rate.

where N_t is the total number of tasks during calculating time, and N_c is the completing number of tasks during that time.

$$R_{ru} = \frac{N_{rd}}{N_{rr}} \quad (11)$$

where N_{rd} is the total number of resources required in a task, and N_{rr} is the real number of resources in the corresponding task coalition.

Result analysis

To begin with, we discuss the results of the mean task completion rate obtained in our experiments. Figure 4 indicates the mean task completion performance of MSMA compared with CGCFA and OPGAA.

Although, as seen in Figure 4, the task completion rate of the MSMA fluctuates significantly in three scenarios compared with the other two algorithms, the task completion rate is only 11% worse, on average, than the CGCFA. And when agents are sufficient, the MSMA performs obviously better than the CGCFA. Furthermore, the MSMA performs magnificently better than the OPGAA in all the three scenarios. When taken together, these results show that our algorithm, based on the market mechanism, is a good model for the task allocation method in DCF. In detail, the performance of MSMA varies mainly because communication objects are reduced remarkably when agents are insufficient and task information is restricted in a smaller scale, which obviously impact the coalition generation, compared with the scenario provided with sufficient agents.

Figure 5 illustrates the specific resource utilization rate of each algorithm. Apparently, the MSMA performs almost comparably to the CGCFA, with only 1% worse, while the OPGAA performs worst on this score. Taking Figures 4 and 5 together, it is evident that high resource utility rate benefits the task completion rate, because it does not occupy agents beyond necessary numbers, which helps to keep more idle agents for other emerging tasks.

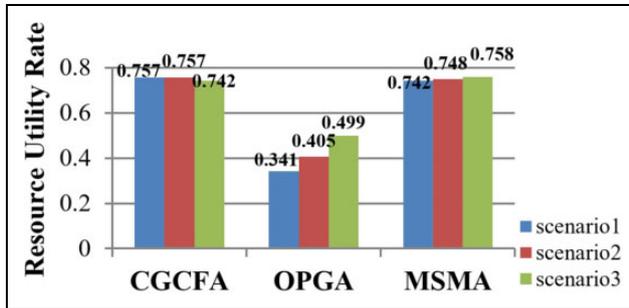


Figure 5. Resource utility rate.

It is clear that MSMA performs better when agents are sufficient. This is mainly because the social network that MSMA adopts can obtain a better network structure for more efficiently transferring information when agents are sufficient. While the OPGAA takes task completing time on top consideration besides optimization in coalition formation, it causes a task coalition obtaining agents much more than the task demands. Thus, the OPGAA presents a poor performance compared with MSMA.

Also, we can see that when agents are insufficient, the performance of MSMA drops quickly, which is because there are not enough agents to construct a good network for information transmission.

Conclusion

This article introduces a market mechanism for distributed computing to address a DCF problem. There are three main aspects of this problem. First, agents are heterogeneous, possessing more than one kind of resources with different quantities. Each agent has a sequence of tasks with different requirement to respond, which means tasks may need more than one agent to execute them successfully. Second, the task flow comes dynamically as a new one emerging timeously. This leads to a Markov planning. However, as stochastic tasks are normally intractable, an approximate global utility is proposed for approaching an optimal solution. Third, the structure of communication network keeps changing due to agents' movement within the target scope. Therefore, we propose a negotiation protocol for the changing network. Finally, experiments are designed to evaluate the efficiency of our approach, which is compared with other two algorithms proposed in other articles. In doing so, we find it almost comparable to a centralized coalition formation algorithm, especially when agents are sufficient.

However, some factors of the problem are simplified so that our study focuses on the proposed mechanism. In the future, we will extend our model to capture those factors, for example, by allowing agents subject to different execution costs for performing the same task (e.g. fuel cost or their own costs), and also allowing agents capable of reasoning their current circumstance and predicting the coming tasks to adjust their strategies for a better global utility.

The former needs agents to make their decisions for reasoning about the type and decisions of other agents, while the latter requires agents to learn from historical data and adjust their decision-making models.

As we analyzed in the “Distributed mechanism for DCF” section, our algorithm's performance drops quickly when agents are insufficient due to a poor communication social network under such situation. Thus, designing a self-adaptation mechanism for agents to construct a better network in structure is a future work to do.

Authors' note

The authors are all from the National University of Defense Technology (NUDT, People's Republic of China). All authors have read this version of the manuscript, and approved to submit to your journal.

Acknowledgments

The authors thank the editor and the anonymous referees for their suggestions and comments which were helpful in improving the article.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (grant nos. 61603406 and 61702528).

References

- Whitten AK, Choi HL, Johnso LB, et al. Decentralized task allocation with coupled constraints in complex missions. *Am Contr Conf*, San Francisco, USA, 29 June -1 July 2011, pp. 1642–1649.
- Bayram H and Isil Bozma H. Coalition formation games for dynamic multirobot tasks. *Int J Robot Res* 2015; 107: 37–54.
- Ponda S, Redding J, Choi HL, et al. Decentralized planning for complex missions with dynamic communication constraints. *Am Contr Conf* 2010; 58(8): 3998–4003.
- Choi HL, Whitten AK and How JP. Decentralized task allocation for heterogeneous teams with cooperation constraints. *Am Contr Conf* 2010; 58(8): 3057–3062.
- Voulkidis AC, Anastasopoulos MP, and Cottis PG. Energy efficiency in wireless sensor networks: a game-theoretic approach based on coalition formation. *ACM Trans Sens Netwk* 2013; 9(4): 710–717.
- Farinelli A, Rogers A, and Jennings NR. Agent-based decentralized coordination for sensor networks using the max-sum algorithm. *Auton Agents Multi Agent Syst* 2014; 28(3): 337–380.
- Stein S, Williamson SA, and Jennings NR. Decentralized channel allocation and information sharing for teams of cooperative agents. *Int Conf Auton Agents Multi Syst* 2012; 13(5): 231–238.

8. Choi HL, Brunet L, and How JP. Consensus-based decentralized auctions for robust task allocation. *IEEE Trans Robot* 2009; 25(4): 912–926.
9. Chapman AC, Micillo RA, Kota R, et al. Decentralised dynamic task allocation: a practical game-theoretic approach. *Int Joint Conf Auton Agents Multi Syst* 2009; 2: 915–922.
10. Bachrach Y, Meir R, Jun K, et al. Coalitional structure generation in skill games. In: *Twenty-fourth AAAI conference on artificial intelligence*, Atlanta, Georgia, 11-15 July 2010, pp. 703–708.
11. Bachrach Y and Rosenschein JS. Coalitional skill games. *Int Joint Conf Auton Agents Mult Syst* 2008; 204(9): 1023–1030.
12. Rahwan T, Michalak T, Wooldridge M, et al. Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *Artif Int* 2012; 186(283): 95–122.
13. Shehory O and Kraus S. Methods for task allocation via agent coalition formation. *Artif Int* 1998; 101(1–2): 165–200.
14. Garey MR and Johnson DS. *Computers and intractability: a guide to the theory of NP-completeness*. New York: Freeman, 1979.
15. Weerd MM, Zhang Y, and Klos T. Multiagent task allocation in social networks. *Auton Agent Multi Agent Syst Kluwer Acad Pub* 2012; 25(1): 46–86.
16. Klusch M and Gerber A. Dynamic coalition formation among rational agents. *IEEE Int Syst* 2002; 17(3): 42–47.
17. Gaston ME and desJardins M. Agent-organized networks for dynamic team formation. In: *International joint conference on autonomous agents and multiagent systems*, the University of Utrecht, the Netherlands, 25-29 July 2005, pp. 230–237.
18. Grinton R, Scerri P, and Sycara K. Agent organized networks redux. In: *Proceedings of the AAAI*, Chicago, IL, July 2008.
19. Ye D and Zhang M. Self-adaptation-based dynamic coalition formation in a distributed agent network: a mechanism and a brief survey. *IEEE Trans Parallel Distrib Syst* 2013; 24(5): 1042–1051.
20. Shapley LS. A value for n-person games. In: Kuhn HW and Tucker AW (eds) *Contributions to the theory of games. Annals of mathematical studies*, 28. Princeton: Princeton University Press, 1953, pp. 307–317.
21. Michalak T, Sroka J, and Rahwan T. A distributed algorithm for anytime coalition structure generation. *Int Conf Auton Agents Mult Syst* 2010; 1: 1007–1014.
22. Gerding EH, Robu V, and Stein S. Online mechanism design for electric vehicle charging. *Int Conf Auton Agents Mult Syst* 2011; 2: 811–818.
23. Gerding EH, Stein S, Robu V, et al. Two-sided online markets for electric vehicle charging. In: *Proceedings of 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*, Saint Paul, USA, 6-10 May 2013, pp. 989–996.
24. Gerkey BP and Mataric MJ. Auctions for methods for multi-robot coordination. *Int Conf Robot Autom* 2002; 18: 758–768.