2013

Self-adaptation-based dynamic coalition formation in a distributed agent network: a mechanism and a brief survey

Dayong Ye
University of Wollongong, dayong@uow.edu.au

Minjie Zhang
University of Wollongong, minjie@uow.edu.au

Danny Sutanto
University of Wollongong, soetanto@uow.edu.au

Publication Details

Ye, D., Zhang, M. & Sutanto, D. (2013). Self-adaptation based dynamic coalition formation in a distributed agent network: a mechanism and a brief survey. IEEE Transactions on Parallel and Distributed Systems, 24 (5), 1042-1051.

Self-adaptation-based dynamic coalition formation in a distributed agent network: a mechanism and a brief survey

Abstract
In some real systems, e.g., distributed sensor networks, individual agents often need to form coalitions to accomplish complex tasks. Due to communication and computation constraints, it is infeasible for agents to directly interact with all other agents to form coalitions. Most previous coalition formation studies, however, overlooked this aspect. Those studies did not provide an explicitly modeled agent network or assumed that agents were in a fully connected network, where an agent can directly communicate with all other agents. Thus, to alleviate this problem, it is necessary to provide a neighborhood network structure, within which agents can directly interact only with their neighbors. Toward this end, in this paper, a self-adaptation-based dynamic coalition formation mechanism is proposed. The proposed mechanism operates in a neighborhood agent network. Based on self-adaptation principles, this mechanism enables agents to dynamically adjust their degrees of involvement in multiple coalitions and to join new coalitions at any time. The self-adaptation process, i.e., agents adjusting their degrees of involvement in multiple coalitions, is realized by exploiting a negotiation protocol. The proposed mechanism is evaluated through a comparison with a centralized mechanism (CM) and three other coalition formation mechanisms. Experimental results demonstrate the good performance of the proposed mechanism in terms of the entire network profit and time consumption. Additionally, a brief survey of current coalition formation research is also provided. From this survey, readers can have a general understanding of the focuses and progress of current research. This survey provides a classification of the primary emphasis of each related work in coalition formation, so readers can conveniently find the most related studies.

Keywords
era2015

Disciplines
Engineering | Science and Technology Studies

Publication Details
Ye, D., Zhang, M. & Sutanto, D. (2013). Self-adaptation based dynamic coalition formation in a distributed agent network: a mechanism and a brief survey. IEEE Transactions on Parallel and Distributed Systems, 24 (5), 1042-1051.

Self-Adaptation-Based Dynamic Coalition Formation in a Distributed Agent Network: A Mechanism and a Brief Survey

Dayong Ye, Minjie Zhang, *Member*, *IEEE*, and Danny Sutanto, *Senior Member*, *IEEE*

Abstract—In some real systems, e.g., distributed sensor networks, individual agents often need to form coalitions to accomplish complex tasks. Due to communication and computation constraints, it is infeasible for agents to directly interact with all other agents to form coalitions. Most previous coalition formation studies, however, overlooked this aspect. Those studies did not provide an explicitly modeled agent network or assumed that agents were in a fully connected network, where an agent can directly communicate with all other agents. Thus, to alleviate this problem, it is necessary to provide a neighborhood network structure, within which agents can directly interact only with their neighbors. Toward this end, in this paper, a self-adaptation-based dynamic coalition formation mechanism is proposed. The proposed mechanism operates in a neighborhood agent network. Based on self-adaptation principles, this mechanism enables agents to dynamically adjust their degrees of involvement in multiple coalitions and to join new coalitions at any time. The self-adaptation process, i.e., agents adjusting their degrees of involvement in multiple coalitions, is realized by exploiting a negotiation protocol. The proposed mechanism is evaluated through a comparison with a centralized mechanism (CM) and three other coalition formation mechanisms. Experimental results demonstrate the good performance of the proposed mechanism in terms of the entire network profit and time consumption. Additionally, a brief survey of current coalition formation research is also provided. From this survey, readers can have a general understanding of the focuses and progress of current research. This survey provides a classification of the primary emphasis of each related work in coalition formation, so readers can conveniently find the most related studies.

Index Terms—Distributed multiagent systems, coalition formation, self-adaptation

1 Introduction

As agent technology becomes more capable and more reliable, multiagent systems have been widely utilized to model real-world applications, such as distributed robotic systems [1], distributed sensor networks [2], supply chain management [3], cloud computing [4], and grid computing [5]. In many applications of multiagent systems, groups of agents need to dynamically join together in a coalition to complete a complex task, which none of them can complete independently. For example, in a distributed vehicle-tracking sensor network, to track a vehicle efficiently, at least three sensor agents are required to triangulate the position of a vehicle moving through the region [6].

Recently, many efforts have been done on coalition formation in multiagent systems, e.g., [7], [8], [9], [10]. There is a common assumption in these studies that an agent network structure is not explicitly modeled or such a structure is based on a fully connected network, namely that an agent can directly communicate with all the agents in the network.[1] However, in many real circumstances, particularly in large and distributed environments, it is infeasible for each individual agent to consider all the other agents to form coalitions due to time, communication and computation constraints, such as wireless sensor networks where each node has a limited communication range. To overcome this limitation, several researchers, e.g., [11], [12], imposed a neighborhood network structure among agents and required that agents directly communicate only with their neighbors. It should be noted that the introduction of such a network structure does not imply advantages or disadvantages. Instead, it just demonstrates the reality and constraints in the real world. For example, in some real systems, e.g., wireless sensor networks, due to the communication constraint, sensor nodes can directly communicate only with their neighbors, i.e., one hop nodes. Thus, in such sensor networks, if a node wants to form a coalition with other nodes to execute a task, e.g., several nodes collaboratively monitor a moving target, it is unlikely for the node to find coalition members directly in the whole sensor network but only from its neighbors (or neighbors' neighbors if necessary). Obviously, there are several new challenges, when coalition formation mechanisms are

- *D. Ye and M. Zhang are with the School of Computer Science and Software Engineering, University of Wollongong, Wollongong NSW 2522, Australia. E-mail: dy721@uowmail.edu.au, minjie@uow.edu.au.*
- *D. Sutanto is with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong NSW 2522, Australia. E-mail: danny@elec.uow.edu.au.*

Manuscript received 21 Sept. 2011; revised 24 June 2012; accepted 4 July 2012; published online 11 July 2012.
Recommended for acceptance by D. Turgut.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-12-0923.
Digital Object Identifier no. 10.1109/TPDS.2012.213.

1. Here, the phrase "network structure" denotes some sort of communication topologies of a network. If in a network, two agents can directly communicate with each other, they are considered to be a neighbor of each other, i.e., neighborhood. "Neighborhood network structure" has the same meaning with "network structure" in this paper. An "agent network" indicates a network, which is connected by a number of agents and which is formed in a communication topology as well.

designed in a structured agent network, such as direct communication of agents with their neighbors to find potential coalition members and uncertainty of distant activities. Gaston and desJardins [11], [13], and Glinton et al. [12] have made great efforts in this way. They investigated the impact of diverse network structures on coalition formation among agents and found that an underlying network structure does have a crucial effect on the performance of agents coalition formation. The studies done in [11], [13], and [12] initiated a new research field in the study of coalition formation, i.e., designing coalition formation mechanisms within explicitly modeled network structures. However, since their research focused on the effect of network structures on coalition formation, the coalition formation mechanisms developed in their research were relatively simple. In their mechanisms [11], [13], [12], an agent can join only one coalition and once a coalition is formed for a task, the coalition is fixed and agents cannot leave the coalition before the task is finished.

Against this background, in this paper, we design a coalition formation mechanism in a structured agent network, which is claimed as a main contribution of this paper. The proposed mechanism assumes that there is a network with explicit links between the agents, such that only agents that are linked to each other (directly or indirectly) can form coalitions. Additionally, our coalition formation mechanism incorporates the self-adaptation concept, which enables agents to dynamically adjust their degrees of involvement in coalitions and to join new coalitions, via negotiation, at any time if necessary. The process of self-adaptation in a large-scale and distributed system is of key importance to the performance of the system as a whole and it can be employed in agent networks to improve the cooperative behaviors of agents [14]. Compared with most related studies, which do not take a network structure into account, we consider the existence of an underlying network structure. Compared with those related studies, which do consider network structures, our mechanism, by integrating the self-adaptation notion, enables agents to have autonomy and flexibility when agents execute tasks. Our mechanism is elucidated by using distributed task allocation. By employing a general application area, i.e., distributed task allocation, instead of a particular existing system, we can develop a general mechanism that can be potentially applied to a wide variety of applications.

The remainder of the paper is organized as follows: Section 2 introduces the agent network model. Section 3 proposes the dynamic coalition formation mechanism. Experimental results and analysis are presented in Section 4 and the paper is concluded in Section 5. The brief survey of current coalition formation studies and the discussion about the difference between our work and these studies are given in the supplementary file, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2012.213.

2 Agent Network Model

To cope with the issue of coalition formation in a structured agent network, we first give the formal definition of the agent network.

Definition 1. (Agent Network). *An* agent network *consists of a set of interdependent agents, namely,* $A = \{a_1, \ldots, a_n\}$, *and a* Compatible Relation R, $R \subseteq A \times A$. *The meaning of* R *is "*a neighbor of,*" so we denote an ordered couple* $\langle a_i, a_j \rangle \in R$ *if and only if* a_j *is a neighbor of* a_i. *Since* R *is a* Compatible Relation, *namely that* R *is* reflexive *and* symmetric, *it can be achieved that* $\forall a_i : a_i \in A \Rightarrow \langle a_i, a_i \rangle \in R$ *and* $\forall a_i, a_j \in A : \langle a_i, a_j \rangle \in R \Rightarrow \langle a_j, a_i \rangle \in R$.

The definition of the reflexive and symmetric relation R is to indicate that the connection between two agents is bidirectional. This definition is necessary, because when an agent wants another agent to join the former's coalition, there is a negotiation between them (described in Section 3).

Each agent $a \in A$ is composed of three tuples $\langle r_a, Neig(a), State(a) \rangle$, where, r_a is the resource, which agent a contains, $Neig(a)$ is a set, which contains the neighbors of agent a, and $State(a)$ demonstrates the state of agent a. Each agent $a \in A$ has a single fixed resource,[2] $r_a \in [1, \varepsilon]$, where ε is the number of different resources that are present in the agent network. Before describing the states of an agent, we provide the definition of three roles used throughout this paper, i.e., *Initiator*, *Participant*, and *Mediator*.

Definition 2. ((Initiator), (Participant), and (Mediator)). *In an agent network, the agent, which initializes a task allocation is called* Initiator*; the agent, which accepts the announced task is called* Participant*; and the agent that receives another agent's commitments for assistance to find* Participants *is called* Mediator.

It should be noted that the roles, i.e., *Initiator*, *Participant*, and *Mediator*, do not imply any architecture in a coalition. Instead, different roles are used only to distinguish different responsibilities of agents.

Definition 3. (States). *There are two states of agents in the network, i.e.,* $\text{States} = \{IDLE, BUSY\}$, *and an agent can be only in one of the two states at any time step. When an agent is an* Initiator, Participant, *or* Mediator *the state of an agent is* BUSY. *An agent in* IDLE *state is available and has not been assigned or committed to any task.*

In this paper, it is supposed that only an *IDLE* agent can be assigned to a new task as an *Initiator*; both *IDLE* and *BUSY* agents can join partially fulfilled tasks as *Participant*s or be committed to partially fulfilled tasks as *Mediator*s. A partially fulfilled task is a task, for which a full coalition is in the formation procedure and has not yet formed.

Suppose there is a set of tasks $\Theta = \{\theta_1, \ldots, \theta_m\}$ arriving at the agent network. Each task $\theta \in \Theta$ consists of four tuples, namely $\langle AT(\theta), DL(\theta), PD(\theta), R(\theta) \rangle$. $AT(\theta)$ is the arrival time of task θ, $DL(\theta)$ indicates the latest start time of task θ, $PD(\theta)$ indicates that how long it will take to finish the task θ and $R(\theta) = \{r_\theta^1, \ldots, r_\theta^j\}$ is a set that dictates the resources, which are needed for successfully completing the task. Therefore, for each task, the *Initiator* has to find appropriate agents, which have the required resources, to form a coalition to handle the task. The term *coalition* is defined as follows.

2. This assumption will be relaxed in Section 4.

Definition 4. (Coalition). *A* coalition c *is a set of agents, i.e.,* $c \subseteq A$*, which cooperate together to complete a complex task* $\theta \in \Theta$*. Each coalition is associated with a task, and then there is a set of coalitions* $C = \{c_1, \ldots, c_h\}$*, which are associated with tasks,* $\theta_1, \ldots, \theta_h$*, respectively. A valid coalition should satisfy the situation that the resources of agents in the coalition should cover the required resources of the associated task, i.e.,* $\bigcup_{a \in c_j} r_a \supseteq R(\theta_j)$.

To form a coalition, the *Initiator* of a task needs to make a contract with each *Participant*. The details of contracts will be described later in Section 3.1. It should be noted that, in this paper, each agent is allowed to join multiple coalitions. Thus, the term, "degree of involvement," is also needed.

Definition 5. (Degree of Involvement (DoI)[3]). *A degree of involvement indicates the extent that an agent belongs to a coalition. The value of* DoI *is ranged in* $(0, 1]$*, where a higher value means that an agent belongs to a coalition in a larger extent. The sum of DoI values of an agent in different coalitions should be equal to or less than 1.*

A DoI in a coalition has an effect on the cost of an agent to complete a task. The initial *DoI* value of an agent joining a coalition is determined by a contract, which is achieved through negotiation between the *Initiator* and the *Participant*. The *DoI* value of an agent can be adjusted at any time in the future by paying penalty to the counterpart. Thus, it can be seen that a coalition is formed via making contracts between the *Initiator* and *Participant*s.

After formalizing the agent network, the principle of our coalition formation mechanism will be depicted.

3 Coalition Formation Mechanism

In an agent network, agents make decisions based only on local information about the system, and the decision making process of agents is autonomous without external control. Hence, we need to define another set $P = \{P_1, \ldots, P_n\}$. P is defined as a *partition* of the *Compatible Relation* R, which has been described in *Definition 1*. Accordingly, it can be obtained that $\bigcup_{1 \le i \le n} P_i = R$ and $\forall P_i, P_j \in P : i \neq j \Rightarrow P_i \cap P_j = \emptyset$. The set P can be generated by using *Algorithm 1*.

Algorithm 1. Create a partition P on relation R
begin:
1: **for each** a_i, $a_i \in A$, in sequential order
2: **if** $\exists a_j \in A : \langle a_i, a_j \rangle \in R$ **then**
3: $P_i \leftarrow P_i \cup \{\langle a_i, a_j \rangle\}$;
end

From *Algorithm 1*, it can be found that P_i is composed of ordered couples, and each ordered couple dictates an agent, which is a neighbor of a_i. P_i and $Neig(a_i)$ look like having the same meaning, but, actually, they are used for different purposes. P_i represents not only neighbors of agent a_i but also other agents, which have indirect connections with a_i established during future task allocation processes, although P_i initially denotes only neighbors of agent a_i. On the other hand, $Neig(a_i)$ stores only directly linked neighboring agents of agent a_i. The coalition formation mechanism, used by an *Initiator* to find appropriate agents to form a coalition, is illustrated in *Algorithm 2* as follows.

Algorithm 2. Coalition Formation Mechanism
begin:
1: **Call Algorithm 1** to generate P;
2: **for each** θ_i, $\theta_i \in \Theta$, in sequential order
3: randomly select an *IDLE* agent, $a_i \in A$, as *Initiator*;
4: $State(a_i) \leftarrow BUSY$;
5: **while** $t < DL(\theta_i)$ **do** * t is the real time * \
6: **for each** $a_j \in A : \langle a_i, a_j \rangle \in P_i$
7: **if** $\exists r^l_{\theta_i} \in R(\theta_i) : r^l_{\theta_i} = r_{a_j}$ **and** $r^l_{\theta_i}$ is unsatisfied **then**
8: **Negotiate**(a_i, a_j);
9: **end if**
10: **end for**
11: **if** $\forall r^l_{\theta_i} \in R(\theta_i) : r^l_{\theta_i}$ is satisfied **then**
12: **break**;
13: **else**
14: **select** a_k as *Mediator* based on the number of a_k's neighbors, where $\langle a_i, a_k \rangle \in P_i$;
15: $State(a_k) \leftarrow BUSY$;
16: $P_i \leftarrow P_i \circ P_k$;
17: **end if**
18: **end while**
19: **end for**
end

For each task θ_i, in Line 3, an *IDLE* agent is randomly selected to be an *Initiator*, denoted as a_i. In this paper, tasks come from outside of the agent network. A new task can arrive at each time step with a specific probability, and is given to a randomly selected *IDLE* agent. Then, before the deadline of task θ_i (Line 5), a_i checks whether the resources of its neighbors, including itself, can satisfy any resource requirements of task θ_i. If any neighboring agent can satisfy any one of the resource requirements, a_i will negotiate with this agent regarding the DoI of this agent in a_i's coalition (Lines 6-10). The negotiation protocol (Line 8), adopted to make a contract between an *Initiator* and a *Participant*, will be described later. It should be noted that, in Line 6, a_i selects coalition members only from the set P_i, which consists of a_i's neighbors (and mediators' neighbors if any). Thus, our coalition formation mechanism takes the communication constraint into account. Thereafter, in Lines 11-13, if all the resource requirements of task θ_i are satisfied, the coalition has been formed and the task can be started. It should be noted that, for a single resource requirement, a_i has to make several temporary agreements, i.e., temporary contracts, with several different agents, (one temporary agreement with one distinct agent), because agents may cancel temporary agreements before final agreements are made and leave the *Initiator* a_i very little time to find an alternative. Thus, multiple temporary agreements are used to reduce the possibility of an *Initiator*'s failure to form a coalition. However, making too many temporary agreements for a single resource will incur much communication cost to the

3. The concept of "degree of involvement" was originally mentioned in operation research field, e.g., [15], [16], where such a concept described the participation level of a person to an organization, e.g., how much time or how many resources a person uses in an organization. In this paper, this concept is borrowed for our problem.

Initiator. Hence, there is a tradeoff with regard to the number of temporary agreements for a single resource. In this paper, the number of temporary agreements for a single resource is set to two, which is obtained through experiments to achieve the best results. Here, a temporary agreement means that when an agent, say a_j, decides to join the *Initiator* a_i's partial formed coalition, where some task resource requirements have not been satisfied, the agreement between a_i and a_j is a temporary agreement. A temporary agreement can be canceled by either the *Initiator* or the *Participant* at any time without paying penalty. Once the coalition becomes a full coalition, namely that all the task resource requirements are satisfied, those uncanceled temporary agreements will automatically become final agreements and cannot be canceled by either party of the agreements. For the clarity purpose, such details of temporary agreements are not demonstrated in *Algorithm 2*. On the other hand, if any resource requirement is not satisfied, in Line 14, a_i selects one of the agents in the set P_i as *Mediator*, say a_k, and commits task θ_i to a_k. Then, a_k will be responsible for a_i to find available coalition members from a_k's neighbors. The *Mediator* selection is based on the number of neighbors that each neighbor of a_i has. The more neighbors an agent has, the higher probability that agent could be selected as a *Mediator*. In the case that a_k's neighbors still cannot fulfill the resource requirements of task θ_i, a_k chooses one of its neighbors as a new *Mediator* and commits task θ_i to the new *Mediator*. This process will continue until the deadline of task θ_i is reached. In Line 16, the notation "$\circ$" is *relational composition*. The meaning of this notation is that $\forall\langle x_i, y_i\rangle \in X, \langle y_j, z_j\rangle \in Y : y_i = y_j \Rightarrow \langle x_i, z_j\rangle \in Z$. In this paper, *relational composition* is utilized to model indirect connections between *Initiator*s and *Participant*s via *Mediator*s. Through this way, an *Initiator* can request not only its directly linked neighbors but also other indirectly connected agents for help.

3.1 The Negotiation Protocol

To operate the coalition formation mechanism, we need another important component, i.e., a negotiation protocol. The coalition formation problem can be modeled as a negotiation process between an *Initiator* and a *Participant*, where an *Initiator* acts as a *buyer* and a *Participant* plays as a *seller*. The negotiation focuses on a single issue, i.e., the DoI of a *Participant* into a coalition, which is being formed by an *Initiator*. Two constraints are listed as follows, with which each agent should comply:

1. The DoI of an *Initiator* in its initiated coalition is postulated to be 1 and cannot be adapted.[4] Other agents, which are not *Initiator*s, can dynamically join multiple coalitions with different degrees of involvement.
2. Both *Initiator*s and *Participant*s cannot cancel final agreements. *Participant*s can adapt the degrees of involvement in their joined coalitions by paying penalty to *Initiator*s, and *Participant*s can join other new coalitions if necessary. Temporary agreements can be canceled by either *Initiator*s or *Participant*s without paying penalty.

The negotiation protocol employed in this paper extends the alternating offers protocol [17] by allowing an agent to make multiple agreements with other agents and to cancel temporary agreements without paying penalty. The alternating offers protocol [17] has been widely used for bilateral bargaining (e.g., An et al. [18]). Other more complex negotiation protocols may be also available for our problem, e.g., [19], [20], but based on our investigation, the alternating offers protocol is powerful enough for our problem and it is easy to implement. It should be noted that the main contribution of this paper is the idea of integrating the self-adaptation notion into dynamic coalition formation rather than this negotiation protocol used only for realizing the self-adaptation notion.

There are four possible actions that a *buyer* (*Initiator*) and a *seller* (*Participant*) can take are as follows:

- $offer[o]$, where o is *buyer*'s offer to a *seller*. An offer is determined by four factors, which are the pressure of deadline, the payment of the resource paid by the *buyer* to the *seller*, the duration of using the resource, and the demand/supply ratio of the *buyer*'s required resource.
- $accept[o]$. When a *seller* receives an offer o, it can accept the offer, which results in a temporary agreement made with the *buyer*.
- $counter\_offer[o']$. If a *seller* is not happy with an offer o, it can send back a counter-offer o' for its available resource. A counter-offer o' is determined by three aspects, which include the current state of the *seller*, e.g., whether it has joined other coalitions and the degrees of involvement into those coalitions, the payment received by the *seller* from the *buyer*, and the demand/supply ratio of the *seller*'s available resource.
- $cancel[o]$. After a temporary agreement is achieved by a *buyer* and a *seller*, any one of them can cancel the temporary agreement without paying penalty. A final agreement, however, cannot be canceled by either a *buyer* or a *seller*.

The negotiation protocol, displayed in Line 8 of *Algorithm 2*, is shown in *Algorithm 3* as follows.

Algorithm 3: **Negotiate**(a_i, a_j)
* a_i is the *buyer* and a_j is the *seller* * \
begin:
1: **while** $t <$ predefined period **do** * t is the real time * \
2: a_i generates an offer o to a_j;
3: **if** a_j accepts o **then**
4: $\mathcal{A}^{\mathcal{T}}(a_i) \leftarrow \mathcal{A}^{\mathcal{T}}(a_i) \cup \{o\}$;
5: $\mathcal{A}^{\mathcal{T}}(a_j) \leftarrow \mathcal{A}^{\mathcal{T}}(a_j) \cup \{o\}$
6: $State(a_j) \leftarrow BUSY$;
7: **return**;
8: **else**
9: a_j generates a counter-offer o' to a_i;
10: **if** a_i accepts o' **then**
11: $\mathcal{A}^{\mathcal{T}}(a_i) \leftarrow \mathcal{A}^{\mathcal{T}}(a_i) \cup \{o'\}$;
12: $\mathcal{A}^{\mathcal{T}}(a_j) \leftarrow \mathcal{A}^{\mathcal{T}}(a_j) \cup \{o'\}$;

4. This postulation is set, because it is conceived that an *Initiator* usually has to negotiate with many other agents to form a coalition, and thus, it usually does not have enough energy and time to finish other *Initiators*' tasks.

13: $State(a_j) \leftarrow BUSY$;
14: **return;**
15: **else**
16: **continue**;
17: **end if**
18: **end if**
19: **end while**
end

During a predefined negotiation period, in Line 2, the *buyer* a_i first generates an offer o and sends the offer to the *seller* a_j, where $o = \langle pay, DoI_u, DoI_l, pe, pd \rangle$. Each element in o is calculated by using the following equations:

$$pay = \begin{cases} p(r_{a_j}) \dfrac{t - AT(\theta)}{DL(\theta) - AT(\theta)}, & \text{if } \mathcal{A}^{\mathcal{T}}(a_i) = \emptyset, \\ \dfrac{p(r_{a_j})}{|\mathcal{A}^{\mathcal{T}}(a_i)|} \dfrac{t - AT(\theta)}{DL(\theta) - AT(\theta)}, & \text{otherwise,} \end{cases} \tag{1}$$

$$DoI_u = 1 - \frac{pay}{p(r_{a_j})}, \tag{2}$$

$$DoI_l = \frac{DoI_u}{\tau}, \tag{3}$$

$$pe = \alpha \cdot pay, \tag{4}$$

$$pd = PD(\theta). \tag{5}$$

In (1), pay is the intended payment made by a_i to a_j, $p(r_{a_j})$ is the maximum payment that a_i can endure for the required resource r_{a_j}, $\mathcal{A}^{\mathcal{T}}(a_i)$ is the set of temporary agreements achieved by a_i for the resource r_{a_j}, and $|\mathcal{A}^{\mathcal{T}}(a_i)|$ denotes the number of temporary agreements in the set $\mathcal{A}^{\mathcal{T}}(a_i)$. An *Initiator* can get benefit from the task if and only if all the subtasks of the task are completed, while a *Participant* can obtain the relevant payment when it finishes the assigned subtask. Here, a subtask corresponds to a resource requirement of a task.

In (2), DoI_u is the upper bound DoI in the coalition, which a_i wants a_j to join for task θ. It can be found that a_i's expected DoI_u value of a_j in a_i's coalition decreases with the increase of payment from a_i to a_j, which seems somewhat weird. This is caused by a_i's concession strategy, as a_i's deadline is approaching. Such time-dependent concession strategies have been broadly used in the literature (e.g., Faratin et al. [21]).

In (3), DoI_l is the lower bound DoI about a_j in a_i's coalition, which means that a_j cannot reduce its DoI value in a_i's coalition less than DoI_l. τ is a coefficient, which is a positive integer. Thus, if a_j accepts an offer from a_i, its original DoI in a_i'coalition is DoI_u. a_j is able to decrease its DoI value in a_i' coalition later, but the DoI value must not be less than DoI_l.

In (4), α, where $0 < \alpha < 1$, is a coefficient and pe is the total penalty if a_j wants to reduce the DoI in a_i's coalition from the upper bound to the lower bound, i.e., from DoI_u to DoI_l. The exact penalty a_j should pay to a_i, recorded as $pe_{j \to i}$, is based on the extent that a_j wants to lessen the DoI in a_i's coalition. Specifically, $pe_{j \to i}$ can be calculated by using (6), where DoI' is the current DoI and DoI'' is the intended DoI to which a_j wants to adjust. For example, a_j's original DoI value in a_i's coalition is 1, i.e., $DoI_u = 1$ and the lower bound is 0.33, i.e., $DoI_l = 0.33$ (suppose $\tau = 3$). Its current DoI value in a_i's coalition is 0.8, i.e., $DoI' = 0.8$ (suppose that a_j has changed its DoI value in a_i's coalition to join another coalition with DoI value 0.2). Now, an agent, say a_k, also wants a_j to join its coalition with DoI value 0.3. Then, a_j may reduce the DoI value in a_i's coalition again from 0.8 to 0.5, i.e., $DoI'' = 0.5$, by paying the penalty, $pe_{j \to i} = \frac{pe}{1 - 0.33} \cdot (0.8 - 0.5)$, to a_i, and simultaneously join a_k's coalition with DoI value 0.3, (recall that the sum of DoI values of a_j should not be greater than 1). a_j never wants to increase its DoI value in any coalition, as this will not bring any revenue to a_j but will even incur an extra cost to a_j

$$pe_{j \to i} = \frac{pe}{DoI_u - DoI_l} \cdot (DoI' - DoI''). \tag{6}$$

In (5), pd indicates the period, during which the required resource is needed.

After receiving an offer o from a_i, in Line 3, a_j evaluates whether the offer o is acceptable. This evaluation is based on how much revenue a_j could get (7). In (7), the $cost$ of a_j depends on its DoI in a_i's coalition and how long its resource will be used by a_i. The notation $pe_{j \to k}$ means the penalty that a_j has to pay other *Initiator*s if a_j wants to lower its DoI values in their coalitions.

$$rv = \begin{cases} pay - cost, & \text{if } \mathcal{A}^{\mathcal{T}}(a_j) = \emptyset, \\ pay - cost - pe_{j \to k}, & \text{otherwise.} \end{cases} \tag{7}$$

If rv is greater than a predefined threshold, a_j will accept the offer o and a temporary agreement is achieved (Lines 4 and 5). Otherwise, a_j generates a counter-offer o' to a_i (Line 9). The elements, which consist of a counter-offer o', are the same as those in an offer o. Since the negotiation issue is only the DoI as described earlier, a_j will adjust only its DoI value in o to meet its predefined threshold revenue via (7) and will create a counter-offer o' with the newly calculated DoI value. a_i then will evaluate the counter-offer o' by comparing the DoI value in o' with its reserved DoI value. If the DoI value in o' is greater than its reserved DoI value, o' is acceptable and a temporary agreement is achieved (Lines 11 and 12). Otherwise, a_i continues to go to the next negotiation round (Line 16) until the predefined negotiation period is reached.

When all the resource requirements are satisfied, the *Initiator*, i.e., the *buyer*, a_i will select the most valuable (the least payment) temporary agreement and cancel other temporary agreements. For the *Participant*, i.e., the *seller*, a_j can execute several tasks simultaneously with the summation of DoI values equal to or less than 1.

3.2 How Does an Agent Adjust DoI Values?

There is another problem in the coalition formation process, which has to be solved, i.e., how can an agent adjust its DoI values in current coalitions to minimize its penalty when it joins a new coalition? Here is an example to illustrate the problem. An agent has joined three coalitions with DoI values 0.2, 0.3, and 0.5, respectively. After negotiation, this agent decides to join a new coalition with DoI value 0.4. Now, the agent has to decide how to adjust the DoI values

in the three current coalitions, as the sum of the DoI values in the current three coalitions should be at most 0.6 (derived from $1-0.4$) and this agent has to minimize its penalty (remember that reducing DoI value in a coalition will incur penalty). We first formulate the problem and then provide an algorithm to solve this problem.

Let us suppose that an agent, say a_j, has joined n coalitions. The sum of DoI values in these n coalitions is 1, i.e., $\sum_{1\le i\le n} DoI_i = 1$. Now another agent also wants a_j to join its coalition with the DoI DoI_{n+1}. If a_j is interested in the new offer, i.e., DoI_{n+1}, it has to calculate the minimal penalty it will pay to the n coalition leaders. It is assumed that a_j reduces ΔDoI_i in ith coalition, where $0 \le \Delta DoI_i \le DoI_i - DoI_l^i$. The question for a_j is how to select each ΔDoI_i to minimize $\sum_{1\le i\le n} \frac{pe_i}{DoI_u^i - DoI_l^i}\Delta DoI_i$, given $\sum_{1\le i\le n}(DoI_i - \Delta DoI_i) + DoI_{n+1} = 1$, where pe_i is the total penalty if a_j wants to reduce the DoI in a_i's coalition from the upper bound to the lower bound. For convenience, we denote $peRate_i = \frac{pe_i}{DoI_u^i - DoI_l^i}$, whose value is known by a_j. Likewise, the given condition can be written as another form $\sum_{1\le i\le n}\Delta DoI_i = C$ and $C = \sum_{1\le i\le n+1} DoI_i - 1$. Since each DoI_i $(1 \le i \le n+1)$ is known by a_j, the value of C is known by a_j as well. The question now can be represented as how to select each ΔDoI_i to minimize $\sum_{1\le i\le n} peRate_i \cdot \Delta DoI_i$, given $\sum_{1\le i\le n}\Delta DoI_i = C$. *Algorithm 4* can be used to derive the optimal ΔDoI_i.

Algorithm 4: Adjusting DoI values
* Suppose that agent a_j attempts to adjust n DoI values * \
begin:
1: **if** $\sum_{1\le i\le n}(DoI_i - DoI_l^i) < C$ **then**
2: **break**; * no solution in this case, the offer is unacceptable * \
3: **else if** $\sum_{1\le i\le n}(DoI_i - DoI_l^i) = C$ **then**
4: **for each** ΔDoI_i; * $1 \le i \le n$ * \
5: $\Delta DoI_i \leftarrow DoI_i - DoI_l^i$;
6: **else** * $\sum_{1\le i\le n}(DoI_i - DoI_l^i) > C$ * \
7: ranking $peRate_i$, such that $peRate_1 \le \cdots \le peRate_n$;
8: find an integer k, $1 \le k < n$, such that
9: $\sum_{1\le i<k}(DoI_i - DoI_l^i) < C$ **and** $\sum_{1\le i\le k}(DoI_i - DoI_l^i) \ge C$;
10: **for** $1 \le i < k$
11: $\Delta DoI_i \leftarrow DoI_i - DoI_l^i$;
12: **for** $i = k$
13: $\Delta DoI_i \leftarrow C - \sum_{1\le i<k}(DoI_i - DoI_l^i)$;
14: **for** $k < i \le n$
15: $\Delta DoI_i \leftarrow 0$;
16: **end if**
end

The proof of validity of *Algorithm 4* is provided as follows.

Proposition 3.1. *The result derived from Algorithm 4 can minimize* $\sum_{1\le i\le n} peRate_i \cdot \Delta DoI_i$, *given* $\sum_{1\le i\le n}\Delta DoI_i = C$.

Proof. For cases 1 and 2, i.e., $\sum_{1\le i\le n}(DoI_i - DoI_l^i) < C$ and $\sum_{1\le i\le n}(DoI_i - DoI_l^i) = C$, the validity of the algorithm is obvious. Thus, our proof focuses on case 3, i.e., $\sum_{1\le i\le n}(DoI_i - DoI_l^i) > C$.

According to *Algorithm 4*, in case 3, it is easy to deduce that $\sum_{1\le i\le n}\Delta DoI_i = C$, so the condition can be satisfied. We arbitrarily select $\Delta DoI'_1, \ldots, \Delta DoI'_n$, where each $\Delta DoI'_i \in [0, DoI_i - DoI_l^i]$ and $\sum_{1\le i\le n}\Delta DoI'_i = C$. What we need to do now is to prove $\sum_{1\le i\le n} peRate_i \cdot \Delta DoI'_i \ge \sum_{1\le i\le n} peRate_i \cdot \Delta DoI_i$. It is easy to achieve that

$$\begin{cases} \Delta DoI'_i \le \Delta DoI_i = DoI_i - DoI_l^i, 1 \le i < k; \\ \Delta DoI'_i \ge \Delta DoI_i = 0, k < i \le n. \end{cases}$$

If $\Delta DoI'_k \ge \Delta DoI_k$,[5] because $\sum_{1\le i\le n}(\Delta DoI'_k - \Delta DoI_k) = \sum_{1\le i\le n}\Delta DoI'_k - \sum_{1\le i\le n}\Delta DoI_k = C - C = 0$, we, thus, have $\sum_{1\le i<k} -(\Delta DoI'_k - \Delta DoI_k) = \sum_{k\le i\le n}(\Delta DoI'_k - \Delta DoI_k)$. Therefore, from $peRate_1 \le \cdots \le peRate_n$, we can deduce that

$$\begin{aligned}
&\sum_{1\le i\le n} peRate_i \cdot \Delta DoI'_i - \sum_{1\le i\le n} peRate_i \cdot \Delta DoI_i \\
&= \sum_{1\le i\le n} peRate_i \cdot (\Delta DoI'_k - \Delta DoI_k) \\
&= \sum_{1\le i<k} -peRate_i \cdot -(\Delta DoI'_k - \Delta DoI_k) \\
&\quad + \sum_{k\le i\le n} peRate_i \cdot (\Delta DoI'_k - \Delta DoI_k) \\
&\ge \sum_{1\le i<k} -peRate_{k-1} \cdot -(\Delta DoI'_k - \Delta DoI_k) \\
&\quad + \sum_{k\le i\le n} peRate_k \cdot (\Delta DoI'_k - \Delta DoI_k) \\
&= -peRate_{k-1} \cdot \sum_{1\le i<k} -(\Delta DoI'_k - \Delta DoI_k) \\
&\quad + peRate_k \cdot \sum_{k\le i\le n} (\Delta DoI'_k - \Delta DoI_k) \\
&= -peRate_{k-1} \cdot \sum_{k\le i\le n} (\Delta DoI'_k - \Delta DoI_k) \\
&\quad + peRate_k \cdot \sum_{k\le i\le n} (\Delta DoI'_k - \Delta DoI_k) \\
&= (peRate_k - peRate_{k-1}) \cdot \sum_{k\le i\le n} (\Delta DoI'_k - \Delta DoI_k) \ge 0
\end{aligned}$$

Then, this result indicates that $\Delta DoI_1, \ldots, \Delta DoI_n$ derived from *Algorithm 4* can minimize $\sum_{1\le i\le n} peRate_i \cdot \Delta DoI_i$, given $\sum_{1\le i\le n}\Delta DoI_i = C$. □

After illustrating our self-adaptation-based coalition formation mechanism, in the next section, we will demonstrate its performance in a simulated agent network in comparison with other coalition formation mechanisms.

4 Experiments and Analysis

To evaluate the performance of the proposed coalition formation mechanism, a set of simulation experiments are conducted. In these experiments, an agent network is initialized by using the random geometric graphs [22], which can be constructed to have a wide range of densities, i.e., the varying average number of neighbors. We first describe the experimental setup and thereafter present the experimental results and analysis.

5. For the case that $\Delta DoI'_k < \Delta DoI_k$, the deducing process is the same.

TABLE 1
Parameters Setting

Para.	Values	Explanations
n	$200 \sim 400$	The number of agents
ϵ	10	The number of resource types
deg	$4 \sim 12$	The average number of neighbours
m	50000	The number of time steps
α	0.2	The ratio to calculate penalty from payment
τ	3	The ratio of upper bound to lower bound DoI values
λ	$0.1 \sim 0.9$	The probability with which tasks are introduced

4.1 Experimental Setup

To objectively exhibit the effectiveness of our coalition formation mechanism, named *self-adaptation coalition formation* (*SACF*), we compare our mechanism with three other mechanisms, i.e., *Centralized* mechanism (CM), *classic coalition formation* (*CCF*) [12] mechanism, and *flexible coalition formation* (*FCF*) mechanism.

1. *CM*. This is an ideal centralized coalition formation mechanism, in which there is an external omniscient central manager that maintains information about all the agents and tasks. The central manager is able to interact with all the agents in the network without cost. When an agent has a complex task to be completed, it simply requests the central manager to seek for the most suitable agents in the network, which could fulfill the task and form coalitions with those agents. This method is not practical or robust, but it can be used as an upper bound of the performance in our experiment.
2. *CCF mechanism*. This mechanism was proposed by Glinton et al. [12], which does not enable agents to either dynamically adjust their degrees of involvement in a coalition or autonomously join multiple coalitions. These two behaviors, i.e., adjusting degrees of involvement and joining multiple coalitions, are self-organizing behaviors. Thus, it can be conceived that CCF does not integrate self-adaptation.[6] Through the comparison with *CCF*, the significance of integrating self-adaptation into coalition formation could be revealed.
3. *FCF mechanism*. This mechanism is created by us, which is a simplified version of *SACF*. This mechanism allows agents to breach a contract and leave a coalition by paying penalty to the coalition leader, i.e., *Initiator*. Agents, however, cannot partially breach a contract. Therefore, agents can join only one coalition at any time step. By comparing with this mechanism, the importance of the notion, i.e., the DoI, can be exposed.

In the agent network, each agent is randomly assigned a single resource $r_a \in [1, \varepsilon]$. For simplicity, tasks are created by randomly generating resource requirements ($R(\theta)$). The number of resources required for a given task is chosen uniformly from $[1, \varepsilon]$, as a result $|R(\theta)| \leq \varepsilon$. Then, each $r^l_\theta \in R(\theta)$ is selected randomly from $[1, \varepsilon]$. In addition, at each time step, a task arrives at the agent network with a probability λ. The required time to complete a task ($PD(\theta)$) is a random positive integer and the latest start time of a task ($DL(\theta)$) is also a random positive integer, which must be greater than the task arrival time ($AT(\theta)$). The task is then randomly assigned to an *IDLE* agent for allocation as described in Section 3. Finally, the evaluation criteria consist of $Profit_{Net}$, which is the summation of each individual agent's profit, and time consumed by these mechanisms. $Profit_{Net}$ can be calculated by using

$$Profit_{Net} = \sum_{i=1}^{|A|}(reward_i + recPenalty_i - penalty_i). \quad (8)$$

In (8), each agent's profit is the difference between its income and payout. An agent's, say a_i's, income consists of the reward, which is obtained when it completes tasks ($reward_i$), and the penalty, which is received from other agents when they change their degrees of involvement in that agent's coalition ($recPenalty_i$). An agent's payout is the penalty it pays to the coalition leaders, i.e., the *Initiator*s, when it changes its degrees of involvement in their coalitions ($penalty_i$). $|A|$ denotes the number of agents in the network. As this paper mainly focuses on theoretical study, we just use a general programming language, i.e., JAVA, to simulate the proposed mechanism. In the experiment, "agent" is initially programmed as a class and then all agents are the objects of this class, and resources are simply represented as integers. Moreover, the experiment is run on Windows XP SP3 operating system with Intel Core 2 Duo 3 GHz CPU and 2 GB RAM. The experimental results are obtained by averaging 100 runs.

For clarity, the values of parameters, which are exploited in these experiments and their meanings are listed in Table 1. These values were chosen experimentally to provide the best results.

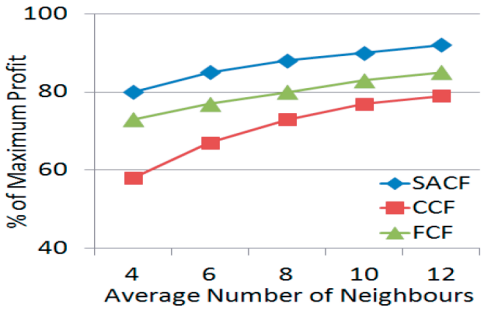

Fig. 1. Profit on different average degrees relative to *CM*.

4.2 Experimental Results

Figs. 1, 2, and 3 demonstrate the relative percentage profits obtained by *SACF*, *CCF*, and *FCF*, respectively, compared with the maximum profit, which is gained by the *CM*. It can be seen that *SACF* performs consistently better than *CCF* and *FCF* in all situations. This is because that *SACF* allows an agent to dynamically join more than one coalitions by adapting its DoI in each coalition and this mechanism can potentially make each agent get more profit.

6. Glinton et al. enabled the agent network structure to be changed, namely that each agent can remove its current neighbors and establish connections with other agents as its new neighbors. This agent network structure adaptation process was called self-adaptation in their paper. The aim of their paper is to investigate whether limiting the number of neighbors of each agent has impact on the coalition formation result. For the *CCF* algorithm itself, i.e., the coalition formation algorithm employed in their paper, no self-adaptation concept was introduced.

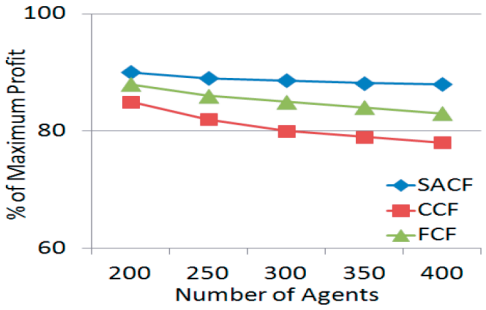

Fig. 2. Profit on different number of agents relative to *CM*.

In Fig. 1, (where the number of agents is 300 and task arrival probability is 0.5), with the increasing average number of neighbors, the profits obtained by agents under *SACF*, *CCF*, and *FCF* are gradually ascending. This is because that when each agent has more neighbors, an *Initiator* can form coalitions for more tasks in a predefined period because it does not always need to commit tasks to other agents. In other words, each agent having more neighbors can accelerate coalition formation process and make more tasks to be completed in the predefined period. It should also be noted in Fig. 1 that the profit obtained by agents under *CCF* rises much more sharply than that under *SACF* and *FCF*, which means that *CCF* depends heavily on the density (average number of neighbors) of a network.

In Fig. 2, (where the average number of neighbors is 8 and task arrival probability is 0.5), when more agents form a network, the profit obtained by agents under *CCF* decreases steeply but that under both *SACF* and *FCF* keeps relatively steady. This can be explained by the fact that when the number of agents increases but the average number of neighbors of each agent is fixed, each task is potentially to be allocated through a longer commitment chain and thus, in a predefined period, less tasks can be allocated. Under *SACF*, however, an agent can dynamically join multiple coalitions to execute more than one tasks and, therefore, can obtain extra profit. Likewise, under *FCF*, an agent has the flexibility to leave a coalition and join another coalition. Thereby, this flexibility encourages agents to join more profitable coalitions, and this flexibility can bring extra profits to agents.

In Fig. 3, (where the number of agents is 300 and the average number of neighbors is 8), when, at each time step, a task arrives more possibly, the profit obtained by agents under *CCF* reduces remarkably, while, in comparison, the profit obtained by agents under *SACF* and *FCF* drops only a little. This can be ascribed to the fact that when tasks arrive

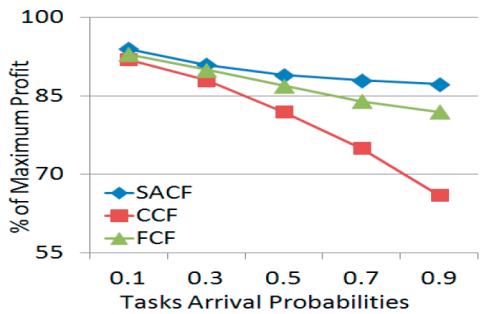

Fig. 3. Profit on different task arrival probabilities relative to *CM*.

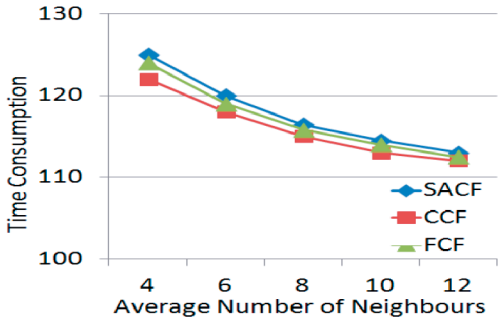

Fig. 4. Time consumption of different mechanisms.

with a low probability, which results in a small number of tasks in the agent network, *Initiator*s relatively have low possibilities to encounter *BUSY* agents during their task allocation processes, so all the three mechanisms, *SACF*, *CCF*, and *FCF*, can obtain decent performance. Nevertheless, once tasks arrive with a very high probability, i.e., a large number of tasks in the network, *Initiator*s are very likely to meet *BUSY* agents during their task allocation processes. *Initiator*s under the *CCF* mechanism then have to give up the *BUSY* agents and try to find other *IDLE* agents, which decreases the possibility of successful coalition formation. Under *SACF* and *FCF*, even if *Initiator*s meet *BUSY* agents, these *Initiator*s still have the chance to establish coalitions with those *BUSY* agents. Thus, agents are more possible to achieve more profits. The acute readers might consider that, under the *FCF* mechanism, once an agent breaches a contract, i.e., leaving a coalition, and joining another coalition, the number of successfully formed coalitions does not increase compared with the *CCF* mechanism. This consideration is reasonable. However, under the *FCF* mechanism, when an agent leaves a coalition, it has to pay penalty to the coalition leader, i.e., the *Initiator*. This penalty is counted as a profit of the *Initiator*, so the overall profit of the entire agent network still rises.

In Fig. 4, (where the number of agents is 300 and task arrival probability is 0.5), the average time consumption of each task is displayed, which exhibits that the three mechanisms spent almost the same time length to finish each simulation run (which consists of 50,000 time steps). This is due to the fact that when an *Initiator*, under *CCF*, meets a *BUSY* agent, the *Initiator* has to bypass this *BUSY* agent and tries to find an *IDLE* agent, but an *Initiator*, under *SACF* and *FCF*, can also negotiate with the *BUSY* agent and the *BUSY* agent may join the *Initiator*'s coalition. Thereby, agents under *SACF* and *FCF* need to negotiate with *Initiators* regarding their participation and this negotiation obviously costs some extra time. On the other hand, agents under *CCF* have to bypass some *BUSY* agents when allocating tasks and bypassing *BUSY* agents to find suitable *IDLE* agents also needs some additional time. Furthermore, it can be found that the time consumption of *FCF* is a little less than that of *SACF*. This is because that under *FCF*, when an agent decides to breach a contract, it does not need to adjust its *DoI* values, since it can join only one coalition. However, under *SACF*, an agent, which joins a new coalition, has to use *Algorithm 4* to adjust the DoI values in its currently participating coalitions.

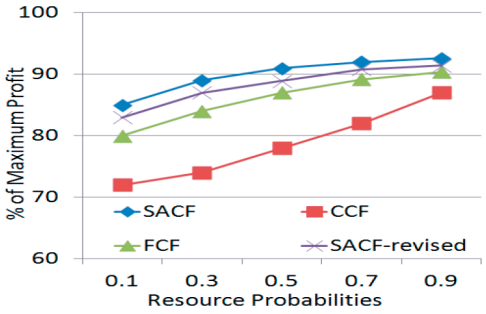

Fig. 5. Profit on different resource probabilities relative to *CM*.

In the previous sections, we assumed that an agent has only one type of resources and an agent cannot leave a coalition once that coalition is formed. Now, we relax these two assumptions and test whether the assumptions have an impact on the performance of the mechanism. To relax the first assumption, we introduce another notion, *Resource Probability* (*RP*), in range $(0, 1)$, which means that an agent is assigned each type of resources with the probability *RP*.[7] For the second assumption, we just need to set each $DoI_l^i = 0$, namely that an agent's lower bound of the DoI in each coalition is 0. This revised *SACF* mechanism is recorded as *SACF-revised*. In addition, the number of agents in the network is set to 300, the average number of neighbors is fixed at 8 and the task arrival probability is 0.5.

It can be seen from Fig. 5 that *SACF*, *SACF-revised*, and *FCF* have the same trend of performance that the profit grows steadily with the increase of the RP. However, under *CCF*, the profit soars up steeply with the increase of the RP. This is owing to the fact that the profit obtained by agents under *CCF* heavily depends on how many resources each agent has. If each agent averagely has more resources, each agent has the potential to complete more subtasks of a task and then the task allocation chain would be shortened, which implies that a coalition for a task is more likely to be successfully formed. Therefore, more profits can be obtained by agents overall. Under *SACF*, *SACF-revised*, and *FCF*, the RP does not have much impact on them, because these mechanisms enable an agent to simultaneously join multiple coalitions, which means that an agent's resources can be utilized by several coalitions simultaneously. Specifically, it can be found that the profit obtained by agents under *SACF* is consistently greater than that under *SACF-revised*, which means that restricting agents to leave coalitions is more profitable for the entire network. This is due to the fact that when an agent leaves a coalition, the coalition leader has to find another agent to replace the leaving agent. Nevertheless, it is difficult to find another suitable agent for an uncompleted task. With time elapse, both the benefit of a task and the rest time of completing the task are reducing, so obviously, it is difficult to find an agent to finish a task in a limited time with a relatively low benefit. The *Initiator* will then lose the benefit of the task, since one of its subtasks cannot be completed, (recall the statement in Section 3.1 that an *Initiator* can get benefit from the task if and only if all the subtasks of a task are completed).

In summary, the performance of our *SACF* is around 85-95 percent of the upper bound, i.e., the performance of the CM, and, on average, 10-15 percent better than *CCF* with nearly the same time consumption. Compared to the other two revised version of *SACF*, i.e., *FCF* and *SACF-revised*, *SACF* outperforms them as well. This result demonstrates that the effect of incorperating self-adaptation into coalition formation is significant and the assumption, not allowing agents to leave formed coalitions, is reasonable and profitable.

7. It has to be guaranteed that an agent has at least one type of resource.

5 Conclusion

This paper introduced a self-adaptation-based dynamic coalition formation mechanism, which enables agents to dynamically adjust their degrees of involvement in different coalitions to achieve efficient task allocation. The proposed mechanism considers the existence of a neighborhood network structure among agents and integrates the self-adaptation concept. To realize the self-adaptation concept, a negotiation protocol was employed. The integration of self-adaptation results in good performance compared with another famous coalition formation mechanism [12] and other two revised versions of our mechanism. In addition, this paper focuses on using self-adaptation concept to solve coalition formation problem, instead of developing a self-adaptation system. That is why some unexpected behaviors, which are likely to happen in self-adaptation systems, e.g., conflicts among components, did not occur in our design and experimental phases. This research can be exploited in many distributed systems, where resources are distributed and agents are highly autonomous, such as supply chain management, resource management in distributed agent-based grid systems and service-oriented cloud computing, which involve resource discovery, resource negotiation and resource composition.

In the proposed coalition formation mechanism, a negotiation protocol is employed to make contracts between *Initiator*s and *Participant*s. It is conceived that a more efficient negotiation protocol would bring a better result of coalition formation. Thereby, we intend to develop a more efficient negotiation protocol in the future. Additionally, as the main contribution of this paper is the proposed coalition formation mechanism instead of the survey, thorough analysis and comparison with current related studies are not presented. Such a thorough survey is one of our future studies as well.

We are also interested in developing a network reorganization mechanism, also known as network structure adaptation mechanism, to incorporate with the proposed coalition formation mechanism to achieve better results of coalition formation. For example, agent a_i connects to agent a_j, and a_j also links with another agent a_k. If a_i often needs help from a_k, then a_i may want to directly connect to a_k to save communication overhead and time consumption. On the other hand, if a_i rarely communicates with its neighbor a_j, then a_i may want to disconnect with a_j to save computation cost associated with taking a_j into account whenever a_i initializes a coalition formation process. According to this example, it can be found that a network reorganization mechanism is helpful for a coalition formation mechanism to achieve better results compared with the one, which does not use a reorganization mechanism. Finally, we plan to extend this mechanism to an open and mobile agent network, where new agents can join as well as existing agents can leave the network, and agents can also have mobility to travel in the network. To realize this scheme, another network reorganization mechanism is necessary, because whenever new agents join or existing agents leave or agents travel in the network, the network structure is inevitably changed.

REFERENCES

[1] C.L. Ortiz, R. Vincent, and B. Morisset, "Task Inference and Distributed Task Management in the Centibots Robotic System," *Proc. Fourth Int'l Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS '05),* pp. 860-867, July 2005.

[2] V. Lesser, C.L. Ortiz, and M. Tambe, *Distributed Sensor Networks: A Multiagent Perspective.* Springer, 2003.

[3] D. Pardoe and P. Stone, "An Autonomous Agent for Supply Chain Management," *Handbooks in Information Systems Series: Business Computing,* vol. 3, pp. 141-173, 2009.

[4] K.M. Sim, "Agent-Based Cloud Computing," *IEEE Trans. Services Computing,* vol. 5, no. 4, pp. 564-577, Oct.-Dec. 2012.

[5] J. Cao, S.A. Jarvis, S. Saini, D.J. Kerbyson, and G.R. Nudd, "Arms: An Agent-Based Resource Management System for Grid Computing," *Scientific Programming,* vol. 10, pp. 135-148, 2002.

[6] M. Sims, C.V. Goldman, and V. Lesser, "Self-organization through bottom-up coalition formation," *Proc. Int'l Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS '03),* pp. 867-874, July 2003.

[7] O. Shehory and S. Kraus, "Methods for Task Allocation via Agent Coalition Formation," *Artificial Intelligence,* vol. 101, nos. 1/2, pp. 165-200, May 1998.

[8] T. Rahwan, S.D. Ramchurn, A. Giovannucci, and N.R. Jennings, "An Anytime Algorithm for Optimal Coalition Structure Generation," *J. Artificial Intelligence Research,* vol. 34, pp. 521-567, 2009.

[9] T. Michalak, J. Sroka, T. Rahwan, M. Wooldridge, P. McBurney, and N.R. Jennings, "A Distributed Algorithm for Anytime Coalition Structure Generation," *Proc. Int'l Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS '10),* pp. 1007-1014, May 2010.

[10] S.D. Ramchurn, M. Polukarov, A. Farinelli, C. Truong, and N.R. Jennings, "Coalition Formation with Spatial and Temporal Constraints," *Proc. Int'l Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS '10),* pp. 1181-1188, May 2010.

[11] M.E. Gaston and M. desJardins, "Agent-Organized Networks for Dynamic Team Formation," *Proc. Int'l Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS '05),* pp. 230-237, July 2005.

[12] R. Glinton, K. Sycara, and P. Scerri, "Agent Organized Networks Redux," *Proc. 23rd Nat'l Conf. Artificial Intelligence (AAAI '08),* pp. 83-88, July 2008.

[13] M.E. Gaston and M. desJardins, "The Effect of Network Structure on Dynamic Team Formation in Multi-Agent Systems," *Computational Intelligence,* vol. 24, no. 2, pp. 122-157, 2008.

[14] G.D.M. Serugendo and M.-P. Gleizes, "Self-Organisation and Emergence in MAS: An Overview," *Informatica,* vol. 30, no. 1, pp. 45-54, 2006.

[15] J. Stringer, "Operational Research for 'Multi-Organizations'," *Operational Research Quarterly,* vol. 18, no. 2, pp. 105-120, 1967.

[16] J.P. Aubin, "Cooperative Fuzzy Games," *Math. Operations Research,* vol. 6, no. 1, pp. 1-13, 1981.

[17] A. Rubinstein, "Prefect Equilibrium in a Bargaining Model," *Econometrica,* vol. 50, no. 1, pp. 97-109, 1982.

[18] B. An, V. Lesser, D. Irwin, and M. Zink, "Automated Negotiation with Decommitment for Dynamic Resource Allocation in Cloud Computing," *Proc. Int'l Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS '10),* pp. 981-988, May 2010.

[19] B. An and V. Lesser, "Negotiation over Decommitment Penalty," *Proc. Int'l Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS '11),* pp. 1101-1102, May 2011.

[20] B. An, V. Lesser, and K.M. Sim, "Strategic Agents for Multi-Resource Negotiation," *J. Autonomous Agents and Multi-Agent Systems,* vol. 23, pp. 114-153, 2011.

[21] P. Faratin, C. Sierra, and N. Jennings, "Negotiation Decision Functions for Autonomous Agents," *J. Robotics and Autonomous Systems,* vol. 24, nos. 3/4, pp. 159-182, 1998.

[22] J. Dall and M. Christensen, "Random Geometric Graphs," *Phys. Rev. E.,* vol. 66, p. 016121, 2002.

Dayong Ye received the BEng degree from Hefei University of Technology, P.R. China, in 2003 and the MSc degree from the University of Wollongong, Australia, in 2009. Now, he is working toward the PhD degree in computer science at the University of Wollongong, Australia. His research interests focus on self-organizing multiagent systems and their applications.

Minjie Zhang received the BSc degree from Fudan University, P.R. China, in 1982 and the PhD degree in computer science degree from the University of New England, Australia, in 1996. Now, she is an associate professor of computer science at the University of Wollongong, Australia. Her research interests include multiagent systems, agent-based simulation, and modeling in complex domains. She is a member of the IEEE.

Danny Sutanto received the BEng and PhD degrees from the University of Western Australia, in 1978 and 1981, respectively. Now, he is a professor of power engineering at the University of Wollongong, Australia. His research interests include energy storage systems and voltage stability. He is a senior member of the IEEE.