Switching Behavioral Strategies for Effective Team Formation by Autonomous Agent Organization

Masashi Hayano, Yuki Miyashita and Toshiharu Sugawara

Dept. of Computer Science and Communications Engineering, Waseda University, 169-8555, Tokyo, Japan

Keywords: Allocation Problem, Agent Network, Bottom-up Organization, Team Formation, Reciprocity.

Abstract: In this work, we propose agents that switch their behavioral strategy between rationality and reciprocity depending on their internal states to achieve efficient team formation. With the recent advances in computer science, mechanics, and electronics, there are an increasing number of applications with services/goals that are achieved by teams of different agents. To efficiently provide these services, the tasks to achieve a service must be allocated to agents that have the required capabilities and the agents must not be overloaded. Conventional distributed allocation methods often lead to conflicts in large and busy environments because high-capability agents are likely to be identified as the best team member by many agents, resulting in inefficiency of the entire system due to concentration of task allocation. Our proposed agents switch their strategies in accordance with their local evaluation to avoid conflicts occurring in busy environments. They also establish an organization in which a number of groups are autonomously generated in a bottom-up manner on the basis of dependability in order to avoid the conflict in advance while ignoring tasks allocated by undependable/unreliable agents. We experimentally evaluate our proposed method and analyze the structure of the organization that the agents established.

1 INTRODUCTION

Recent advances in information technologies such as the Internet have enabled computerized systems to achieve on-demand and real-time controls/services using timely data captured in the real world. Examples of applications based on such viewpoints include the Internet of Things (IoT) (Stankovic, 2014), the Internet of Services (IoS) (Nain et al., 2010), sensor networks (Glinton et al., 2008), and grid/cloud computing (Foster, 2002). Tasks to achieve required services in these applications are realized within teams, i.e., by combining a number of different software and hardware nodes or agents that have specialized functions. In these systems, the nodes are massive, are located in a variety of positions, and operate in the Internet autonomously since they are usually created by different developers. Even so, they have to be appropriately identified by their functions and performance and then be allocated the suitable and executable components of a task (called a *subtask* hereafter) to achieve it. Mismatching or excessive allocations of subtasks to agents, which are autonomous programs to control hardware/software and/or to execute the allocated subtasks, result in the delay or failure of services. The teams of agents for the required tasks need to simultaneously be formed in a timely manner for timely service provision.

The task allocation problem described above is a fundamental problem in computer science and many studies have been conducted in the multi-agent systems (MAS) context to examine it. For example, coalitional formation is a theoretical approach in which (rational) agents find the optimal coalitional structure (the set of agent groups) that provides the maximal utilities for a given set of tasks (Dunin-Keplicz and Verbrugge, 2010; Sheholy and Kraus, 1998). However, this approach assumes that the systems are static, relatively small and unbusy because it assumes the (static) characteristic function to calculate the utility of an agent set. It also requires high computational costs to find (semi-)optimal solutions, making it impractical when the systems are large and busy. Another approach that is more closely related to our method is team formation by rational agents. In this framework, a number of leaders that commit to form teams for tasks first select the agent appropriate for executing each of the subtasks on the basis of the learning of past interaction and solicit them to form a team to execute the task. Agents that receive a num-

Hayano, M., Miyashita, Y. and Sugawara, T.
Switching Behavioral Strategies for Effective Team Formation by Autonomous Agent Organization.
DOI: 10.5220/0005748200560065
In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016) - Volume 1*, pages 56-65
ISBN: 978-989-758-172-4

ber of solicitations accept one or a few of them depending on their local viewpoints. When a sufficient number of agents has accepted the solicitations, the team can succeed in executing the target task. However, conflicts occur if many solicitations by leaders are concentrated to only a few capable agents, especially in large and busy MAS, so the success rate of team formation decreases in busy environments.

In the real world, people also often form teams to execute tasks. Of course, we usually behave rationally, i.e., we decide who will provide the most utility. However, if conflicts in forming teams are expected to occur and if no prior negotiation is possible, we often try to find reliable people with whom to work in advance. Reliable people are usually identified through past success with cooperative work (Fehr and Fischbacher, 2002). Furthermore, if the opportunities for group work are frequent, we try to form implicit or explicit collaborative structures based on (mutual) reliability. In an extreme case where team work with unreliable people is required, we may ignore or understate offers for the sake of possible future proposals with more reliable people. Such behavior based on reciprocity may be irrational because offers from non-reciprocal persons are expected to be rewarding in at least some way. However, it can stabilize collaborative relationships and reduce the possibility of conflicts in team formations. Thus, we can expect steady benefits in the future through working based on reciprocity. To avoid conflicts and improve efficiency in group work in computerized systems, we believe that agents should identify which agents are cooperative and build the agent network on the basis of mutual reliability that is appropriate for the patterns and structures of the service requirements.

To avoid conflicts in team formation in large and busy MAS, we propose a computational method of enabling efficient team formations that have fewer conflicts (thereby ensuring stability) by autonomously generating reliability from reciprocity. The proposed agents switch between two behavioral strategies, rationality and reciprocity: they initially form teams rationally and identify reliable agents through the success of past team works and then identify a number of reliable agents that behave reciprocally. Of course, they return to the rational strategy if the reliable relationships are dissolved. The concept behind this proposal is that many conflicts occur in the regime of only rational agents because such agents always pursue their own utilities. Conversely, the regime of only reciprocal agents experiences less conflict but seems to constrain the behavior of some agents in the cooperative structure without avail. We believe that the appropriate ratio between rationality and reciprocity will result in better performances. However, the relationship between the ratios and performance, and which agents in an agent network should behave rationally, remains to be clarified. Thus, we propose agents that switch strategies in a bottom-up manner by directly observing the reciprocal behavior of others and the success rates of team formation.

This paper is organized as follows. In the next section, we describe related work in task allocation and reciprocity in human society. Section 3 presents the model of our problem and framework and Section 4 describes the proposed agents that adaptively switch their behavioral strategies. Then, in Section 5 we experimentally evaluate the performance of team formation with the regime of the proposed agents by comparing it with those with regimes of only rational agents and only cooperative agents with static collaborative relationships. We also investigate how the ratios between rational and reciprocal agents vary according to the workload. We conclude in Section 6 with a brief summary and mention of future work.

2 RELATED WORK

Achieving allocations using a certain negotiation method or protocol is a fundamental approach in the MAS research. The conventional *contract net protocol* (CNP) (Smith, 1980) approach and its extensions has been studied by many researchers. For example, Sandholm (Sandholm and Lesser, 1995) extended CNP by introducing *levels of commitment* to make a commitment breakable with some penalty. One of the key problems in negotiation protocols is that the number of messages exchanged for agreement increases as the number of agents increases. Thus, several studies have attempted to reduce the number of messages and thereby improve performance (Parunak, 1987; Sandholm and Lesser, 1995). Although recent broadband networks have eased this problem at the link level, agents (nodes) are now overloaded by excessive messages, instead. Furthermore, it has been pointed out that the eager-bidder problem, where a number of tasks are announced concurrently, occurs in large-scale MAS , in which case CNP with levels of commitment does not work well (Schillo et al., 2002). Ishida (Gu and Ishida, 1996) also reported that busy environments negatively affect the performance of CNP. Thus, these methods cannot be used in large-scale, busy environments.

Coalitional formation is a theoretical approach based on an abstraction in which agents find the optimal coalitional structure that provides the maximal utilities for a given set of tasks (Dunin-Keplicz and

Verbrugge, 2010; Sheholy and Kraus, 1998; Sims et al., 2003; Sless et al., 2014). Although this technique has many applications, it assumes static and relatively small environments because high computational costs to find (semi-)optimal solutions are required and the static characteristic function for providing utilities of agent groups is assumed to be given. Market-based allocation is another theoretical approach based on game theory and auction protocol. In this approach, information concerning allocations is gathered by auction-like bidding. Although it can allocate tasks/resources optimally in the sense of maximizing social welfare, it cannot be applied to dynamic environments where optimal solutions frequently vary. Team formation (or coalitional formation in a task-oriented model) is another approach in which individual agents identify the most appropriate member agent for each subtask on the basis of the learning of functionality and the capabilities of other agents (Coviello and Franceschetti, 2012; Hayano et al., 2014; Genin and Aknine, 2010; Abdallah and Lesser, 2004). However, this may cause conflicts in large-scale and busy MAS, as mentioned in Section 1. Our aim is to reduce such conflicts by building upon our previous work (Hayano et al., 2014) and introducing agents that switch from rational behavior to reciprocal behavior (and vise versa) on the basis of results of past collaboration.

Many studies in computational biology, sociology, and economics have focused on the groups that have been organized in human societies (Smith, 1976). For example, many studies have tried to explain irrational behaviors for collaboration in group work using reciprocity. The simplified findings of these studies are that people do not engage in selfish actions toward others and do not betray those who are reciprocal and cooperative, even if selfish/betraying actions could result in higher utilities (Gintis, 2000; Fehr et al., 2002; Panchanathan and Boyd, 2004). For example, Panchanathan and Boyd (Panchanathan and Boyd, 2004) stated that cooperation could be established from indirect reciprocity (Fehr and Fischbacher, 2002), while the authors of (Fehr et al., 2002; Gintis, 2000) insisted that fairness in cooperation may produce irrational behavior because rational agents prefer a higher payoff even though it may reduce the payoff to others. However, agents do not betray relevant reciprocal agents because such a betrayal would be unfair. Fehr and Fischbacher (Fehr and Fischbacher, 2002) demonstrated how payoffs shared among collaborators affected strategies and found that punishment towards those who distribute unfair payoffs is frequently observed, although administering the punishment can be costly (Fehr and Fischbacher, 2004). In this paper, we attempt to introduce the findings above into the behaviors of computational agents.

3 MODEL

3.1 Agent and Tasks

We use a simpler model for representing tasks and the associated utilities than that used in (Hayano et al., 2014) because our focus is more on identifying which learning parameters and mechanisms contribute to the self-organization of groups for team formation.

Let $\mathcal{A} = \{1, \ldots, n\}$ be a set of agents. Agent $i \in \mathcal{A}$ has its associated resources (corresponding to functions or capabilities) $H_i = (h_i^1, \ldots, h_i^p)$, where h_i^k is 1 or 0 and p is the number of resource types. Parameter $h_i^k = 1$ means that i has the capability for the k-th resources. Task T consists of a number of subtasks $S_T = \{s_1, \ldots, s_{l(T)}\}$, where $l(T)$ is a positive integer. If there is no confusion, we denote $l = l(T)$ simply. Subtask s_j requires some amount of resources, which is denoted by $(r_{s_j}^1, \ldots, r_{s_j}^p)$, where $r_{s_j}^k = 0$ or 1 and $r_{s_j}^k = 1$ means that k-th resource is required to execute s_j. Agent i can execute s_j only when

$$h_i^k \geq r_j^k \quad \text{for } 1 \leq \forall k \leq p$$

is satisfied. We often identify subtask s and its associated resource $s = (r_s^1, \ldots, r_s^p)$. We can say that task T is executed when all the associated subtasks are executed.

3.2 Execution by a Team

Task T is executed by a set of agents by appropriately allocating each subtask to an agent. A *team* for executing task T is defined as (G, σ, T), where G is the set of agents. Surjective function

$$\sigma : S_T \longrightarrow G$$

describes the assignment of S_T where subtask $s \in S_T$ is allocated to $\sigma(s) \in G$. We assume that σ is a one-to-one function for simplicity, but we can omit this assumption in the discussion below. The team for executing T has been successfully formed when the conditions

$$h_{\sigma(s)}^k \geq r_s^k \tag{1}$$

hold for $\forall s \in S_T$ and $1 \leq \forall k \leq p$.

After the success of team formation for task T, the team receives the associated utility $u_T \geq 0$. In general, the utility value may be correlated with, for example, the required resources and/or the priority.

However, here we focus on improving the success rate of team formation by autonomously establishing reliable groups, so we simplify the utility calculation and distributions; hence, all agents involved in forming the team receive $u_T = 1$ equally when they have succeeded but receive $u_T = 0$ otherwise. Note that agents are confined to one team and cannot join another team simultaneously. This assumption is reasonable in some applications: for example, agents in a team are often required to be synchronized with other agents. Another example is in robotics applications where the physical entities are not sharable due to spatial restrictions (Zhang and Parker, 2013). Even in a computer system that can schedule multiple subtasks, selecting one team corresponds to the decision on which subtasks should be done first.

3.3 Forming Teams

For a positive number λ, λ tasks per tick are requested by the environment probabilistically and stored in the system's task queue $Q = \langle T_1, T_2, \dots \rangle$, where Q is an ordered set and *tick* is the unit of time used in our model. Parameter λ is called the *workload* of the system. Agents in our model are in either an *inactive* or *active* state, where an agent in the active state is involved in forming a team and otherwise is inactive. Inactive agents first decide to play a role, *leader* or *member*; how they select the role is discussed later.

Inactive agent i playing a leader role picks up task T from the head of Q, and becomes active. If i cannot find any task, it stays inactive. Active agent i then finds subtask $s \in S_T$ that i can execute. Then, i identifies $|S_T| - 1$ agents to allocate subtasks in $S_T \setminus \{s\}$. (If i cannot find any executable subtask in S_T, it must identify $|S_T|$ agents. In our explanation below, we assume that i can execute one of the subtasks, but we can omit this assumption if need.) How these agents are identified will be discussed in Section 4. The set of i and the identified agents is called the *pre-team* and is denoted by G_T^p. Agent i sends the agents in G_T^p messages soliciting them to join the team and then waits for the response. If the agents that accept the solicitations satisfy condition (1), the team (G, σ, T) is successfully formed, where G is the set of agents to which the subtask in S_T is allocated, and the assignment σ is canonically defined on the basis of the acceptances. Then, i notifies $G \setminus \{i\}$ of the successful team formation and all agents in G continue to be active for d_T ticks for task execution. At this point, i (and agents in G) return to inactive. However, if an insufficient number of agents for T accept the solicitation, the team formation by i fails and i discards T and notifies the agents of the failure. The agents in G then return to inactive.

When an agent i that decides to play a member, it looks at the solicitation messages from leaders and selects the message whose allocated subtask is executable in i. The strategy for selecting the solicitation message is described in Section 4. Note that i selects only one message, since i can join only one team at a time. Agent i enters the active state and sends an acceptance message to the leader j of the selected solicitation and rejection messages to other leaders if they exist. Then, i waits for the response to the acceptance. If it receives a failure message, it immediately returns to the inactive state. Otherwise, i joins the team formed by j and is confined for duration d_T to its execution, after which it receives $u_T = 1$ and returns to inactive. If i receives no solicitation messages, it continues in the inactive state.

Note that we set the time required for forming a team to d_G ticks, thus, the total time for executing a task is $d_G + d_T$ ticks. We also note that leader agent i can select pre-team members redundantly; for example, i selects $R \geq 1$ agents for each subtask in S_T (where R is an integer). This can increase the success rate of team formation but may restrain other agents redundantly. We make our model simpler by setting $R = 2$, as our purpose is to improve efficiency by changing behavioral strategies.

4 PROPOSED METHOD

Our agents have three learning parameters. The first is called the *degree of expectation for cooperation* (DEC) and is used to decide which agents they should work with again. The other two are called the *degree of success as a leader* (DSL) and the *degree of success as a member* (DSM) and are used to identify which role is likely to be successful for forming teams. We define these parameters and explain how agents learn and use them in this section.

4.1 Learning for Cooperation

Agent i has the DEC parameter c_{ij} for $\forall j \ (\in \mathcal{A} \setminus \{i\})$ with which i has worked in the same team in past. The DEC parameters are used differently depending on roles. When i plays a leader, i selects pre-team members in accordance with the DEC values, i.e., agents with higher DEC values are likely to be selected. How pre-team members are selected is discussed in Section 4.3. Then, the value of c_{ij} is updated by

$$c_{ij} = (1 - \alpha_c) \cdot c_{ij} + \alpha_c \cdot \delta_c, \qquad (2)$$

where $0 \leq \alpha_c \leq 1$ is the learning rate. When j replies to i's solicitation message with an acceptance, c_{ij} is updated with $\delta_c = 1$; otherwise, it is updated with $\delta_c = 0$. Therefore, j with a high DEC value is expected to accept the solicitation by i.

After i agrees to join the team that is initiated by leader j, i also updates c_{ij} using Eq. (2), where δ_c is the associated utility u_T, i.e., $\delta_c = 1$ when the team is successfully formed and $\delta_c = 0$ otherwise. Agent i also selects the solicitation messages according to the DEC values with the ε-greedy strategy.

After the value of c_{ij} in i has increased, j may become uncooperative for some reasons. To forget the outdated cooperative behavior, the DEC values are slightly decreased in every tick by

$$c_{ij} = max(c_{ij} - \nu_F, 0), \tag{3}$$

where $0 \leq \nu_F \ll 1$.

4.2 Role Selection and Learning

Agent i learns the values of DSL and DSM to decide which role, leader or member, would result in a higher success rate of team formation. For this purpose, after the team formation trial for task T, parameters e_i^{leader} and e_i^{member} are updated by

$$\begin{aligned} e_i^{leader} &= (1-\alpha_r) \cdot e_i^{leader} + \alpha_r \cdot u_T \quad \text{and} \\ e_i^{member} &= (1-\alpha_r) \cdot e_i^{member} + \alpha_r \cdot u_T, \end{aligned}$$

where u_T is the received utility value that is 0 or 1 and $0 < \alpha_r < 1$ is the learning rate for the DSL and DSM.

When i is inactive, it compares the values of DSL and DSM: specifically, if $e_i^{leader} > e_i^{member}$, i decides to play a leader, and if $e_i^{leader} < e_i^{member}$, i plays a member. If $e_i^{leader} = e_i^{member}$, its role is randomly selected. Note that when i selects the leader as the role but can find no task in Q, i does nothing and will select its role again in the next tick.

4.3 Agent Switching Behavioral Strategies

Our main objective in this work is to propose a new type of agent that switches its behavioral strategy, rational or reciprocal, depending on its internal state. In this section, we first discuss how worthy-to-cooperate agents (called *dependable* agents) are identified and then go over the behaviors of rational and reciprocal agents. Finally, we explain how agents select their behavioral strategies.

4.3.1 Dependable Agents

Agent i has the set of dependable agents $D_i \subset \mathcal{A} \setminus \{i\}$ with the constraint $|D_i| \leq X_F$, where X_F is a positive integer and is the upper limit of dependable agents. The elements of D_i are decided as follows. For the given threshold value $T_D > 0$, after c_{ij} is updated, if $c_{ij} \geq T_D$ and $|D_i| < X_F$ are satisfied, i identifies j as dependable by setting $D_i = D_i \cup \{j\}$. Conversely, if $\exists k \in D_i$ s.t. $c_{ik} < T_D$, k is removed from D_i.

4.3.2 Behaviors of Agents with Rational and Reciprocal Strategies

Behavioral strategies mainly affect to decide collaborators. Both leader agents with rational and reciprocal behavioral strategies select the members of the pre-team based on the DEC values with ε-greedy selection. Initially, agent i sets $G_T^p = \{i\}$ and allocates itself to the subtask s^0 ($\in S_T$) executable in i.[1] Then, $\tilde{S}_T = S_T \setminus \{s^0\}$ and i sorts the elements of $\mathcal{A}$ by descending order of the DEC values. For each subtask $s^k \in \tilde{S}_T$, i seeks from the top of $\mathcal{A}$ an agent that can execute s^k and is not in G_T^p and then adds it to G_T^p with probability $1 - \varepsilon$. However, with probability ε, the agent for $s \in G_T^p$ is selected randomly. If $R = 1$, the current G_T^p is the pre-team member for T. If $R > 1$, i repeats the seek-and-add process for subtasks in $\tilde{S}_T$ $R - 1$ times.

Behavioral differences appear when agents play members. An agent with rational behavioral strategy selects the solicitation message sent by the leader whose DEC value is the highest among the received ones. An agent with reciprocal behavioral strategy selects the solicitation message in the same way but ignores any solicitation messages sent by leaders not in D_i. Note that by ignoring non-dependable agents, no solicitation messages may remain in i (i.e., all solicitations will be declined). We understand this situation in which i does not accept the messages for the sake of possible future proposals from dependable agents, and thus, we can say that this ignorance may be irrational. All agents also adopt the ε-greedy selection of solicitation messages, whereby the selected solicitation message is replaced with another message randomly selected from the received messages with probability ε.

4.4 Selection of Behavioral Strategies

When agent i decides to play a member, it also decides its behavioral strategy on the basis of the DSM values e_i^{member} and D_i. If the DSM e_i^{member} is larger than the parameter T_m (> 0), and if $D_i \neq \emptyset$, i adopts the reciprocal strategy; otherwise, it adopts rationality. The parameter T_m is a positive number and is used in the threshold for the criterion of whether or not i

[1] s^0 may be *null*, as mentioned before.

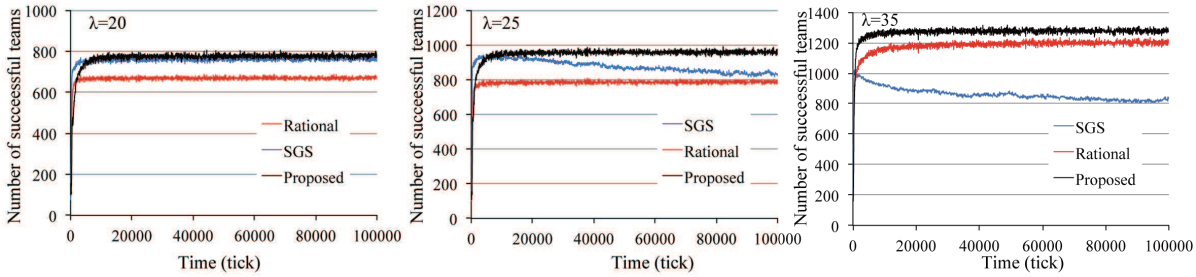

Figure 1: Team formation performance.

has had a sufficient degree of success working as a member. Thus, T_m is called the *member role threshold for reciprocity*. When i plays a leader, its strategy is not affected by how members are selected.

We have to note that i memorizes dependable agents on the basis of DEC values that reflect the success rates of team formation so that it can expect the utility after that. In this sense, the DEC values are involved in rational selections, and therefore dependable agents are identified on the basis of rational decision. In our framework, after a number of dependable agents are identified, i changes its behavior. Therefore, we can say that at first, i pursues only the utilities, but when it has identified a number of dependable agents that may bring utilities, i tries not only to work with them preferentially but also to reduce chances of unexpected uncooperative behaviors.

5 EXPERIMENTS

5.1 Experimental Setting

We investigated the performance (the success rates) of team formation in the society of the proposed agents, the structure of behavioral strategies, and the networks of dependability and team formation achievement. We also experimentally compared it with the performance of those in the society of *rational agents* and that of the proposed agents of the static group regime whose structures are initially given and fixed. A rational agent always behaves on the basis of rationality, thus corresponding to the case where $X_F = 0$. The agents with the static group regime are initially grouped into teams of six random agents, and any agent that initiates a task always allocates the associated subtasks to other agents in the same team. Thus, this type of agent corresponds to the case where D_i is fixed to the members of the same group and $R = 1$. We call this type of agent the *static group-structured agents* or the SGS agents.

Let the number of agents $|\mathcal{A}|$ be 500 and the number of resource types p be six. The amount of k-th resource of agent i, h_i^k, and the amount of the k-th resource required for task s, r_s^k, is 0 or 1. We assume that at least one resource in H_i is set to 1 to avoid null-capability agents. On the other hand, only one resource is required in s, so $\exists k, r_s^k = 1$, and $r_s^{k'} = 0$ if $k' \neq k$. A task consists of three to six subtasks, so $l(T)$ $(= |S_T|)$ is the integer between three and six. The duration for forming a team, d_G, is set to two and the duration for executing a task, d_T, is one. Other parameters used in Q-learning for agent behaviors are listed in Table 1. Note that while ε-greedy selection and Q-learning often used for parameter learning, where we use the shared learning rate α and random selection rate ε. The experimental data shown below are the mean values of ten independent trials.

Table 1: Parameter values in experiments.

Parameter	Value
Initial value of DEC c_i	0.1
Initial value of DSL e_i^{leader}	0.5
Initial value of DSM e_i^{member}	0.5
Learning rate α $(= \alpha_c, \alpha_r)$	0.05
Epsilon in ε-greedy selection ε	0.01
Decrement number γ_F	0.00005
Threshold for dependability T_D	0.5
Max. number of dependable agents X_F	5
Member role threshold for reciprocity T_m	0.5

5.2 Performance Results

Figure 1 plots the number of successful teams every 50 ticks in societies consisting of the SGS, rational, and proposed agents when workload λ is 20, 25, and 35. Note that since all agents individually adopted ε-greedy selection with $\varepsilon = 0.01$ when selecting member roles and solicitation messages, approximately four to five tasks per tick according to the value of workload λ (so, 200 to 250 tasks every 50 ticks) were used for challenges to find new solutions, but in these situations, forming teams was likely to fail. We also note that $\lambda = 20$, 25, and 35

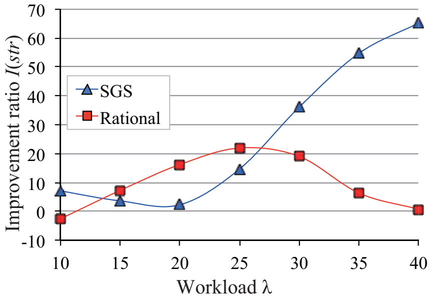

Figure 2: Performance improvement ratios.

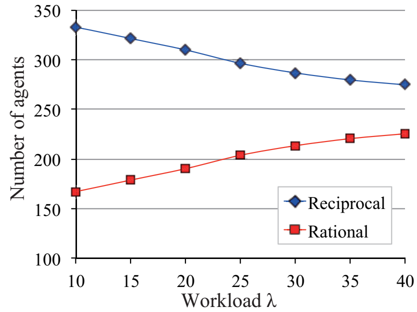

Figure 3: Selected Behavioral Strategies at 100000 ticks.

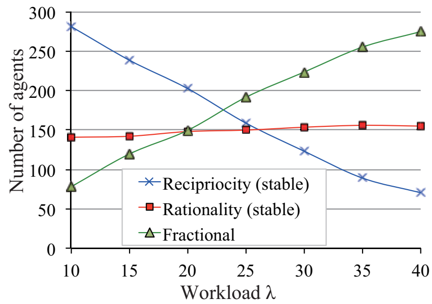

Figure 4: Stability of Behavioral Strategies.

corresponds to the environment where work is low-loaded, balanced (slightly lower than the system's limit of performance when λ is around 25 and 30), and overloaded, respectively. Figure 1 shows that the performance with the proposed agents always outperformed those with other strategies. When the system load was low, the performance with the SGS agents was (slightly lower but) almost identical to that with the proposed agents. However, when $\lambda = 25$ and 35, their performance gradually decreased. In the busy environment, an agent that learned to play a leader in a group encountered many team formation failures, thereby starting to learn that it was ineffective as a leader. In such cases, other agents started to play the leader roles instead, but among the SGS agents, groups are static and no leaders existed in a number of groups. Thus, the number of team formation failures increased. Because many conflicts occurred in the society of only the rational agents, their performance was always lower than that with the proposed agents. However, in a busy environment ($\lambda = 35$), the performance by the proposed agents also reached a ceiling and their difference became smaller.

As Fig. 1 suggests that the improvement ratios might vary depending on the work load, we plotted the ratios in Fig. 2, where the improvement ratio $I(str)$ was calculated as

$$I(str) = \frac{N(proposed) - N(str)}{N(proposed)} \times 100,$$

where $N(str)$ is the number of successful teams per 50 ticks with agents whose behavioral strategy is *str* (*"proposed," "SGS,"* or *"rational."*)

Figure 2 indicates that the performance improvement ratio of the society of the SGS agents, $I(SGS)$, was small when the workload was low ($\lambda \leq 20$) but that it monotonically increased in accordance with the system's workload when $\lambda > 20$. The improvement ratios to the rational agents $I(rational)$ depict a characteristic curve, becoming maximal around $\lambda = 25$ and 30, which is near but below the system's limit, as mentioned above. We think this is the effect of autonomous organization in the society of the proposed method, as reported in (Corkill et al., 2015); we will discuss this topic later. Finally, when the workload was low (λ less than ten), we could not observe any clear difference in their performances since conflict in team formation rarely occurred.

5.3 Behavioral Analysis

To understand why teams were effectively formed in the society of the proposed agents, we first analyzed the characteristics of the behavioral strategy and role selections. Table 2 lists the numbers of leader agents in which $e_i^{leader} > e_i^{member}$ were satisfied at the time of 100,000 ticks (so they played leaders) when λ was varied. As shown, we found that the number of leader agents slightly decreased when the workload increased, but were almost invariant around 100, and thus, 400 agents were likely to play member roles. The number of subtasks required to complete a single task fixed between three and six with uniform probability and the structures of the task distribution did not change in our experiments. Hence, the number of leaders that initiate the team formation also seems to be unchanged.

On the other hand, behavioral strategies were selected differently depending on the workload. The relationships between the workload and the structures of behavioral strategies at the end of the experiment are plotted in Fig. 3. It indicates that reciprocity was selected by over half of the agents, but this number

Table 2: Number of leaders.

Workload (λ)	10	15	20	25	30	35	40
Number of leaders	107.8	104.7	102.4	99.5	97.9	97.3	97.6

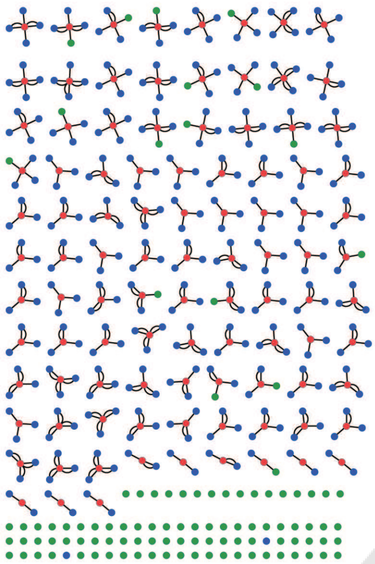

Figure 5: Structure of dependability.

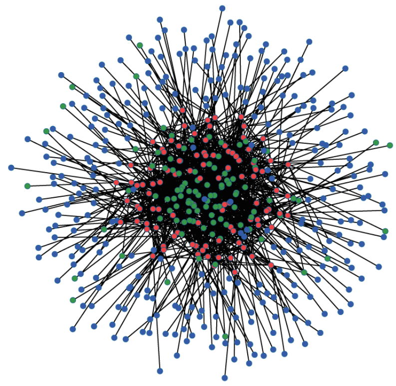

Figure 6: Structure based on formed teams (all agents).

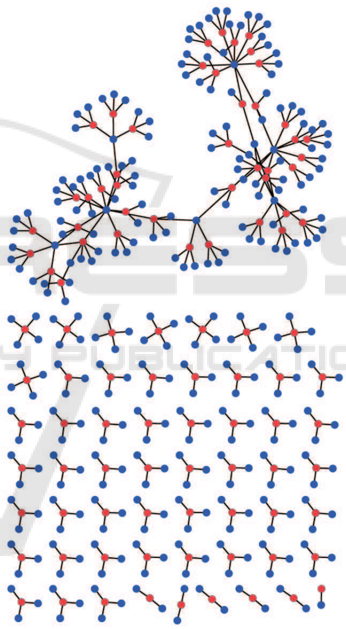

Figure 7: Structure based on formed teams (reciprocal).

gradually decreased with the increase of the workload. Furthermore, in Fig. 4 we plot the number of selected behavioral strategies during 95,000 to 100,000 ticks, where the agents that stably selected, for example, reciprocity mean that they always selected reciprocity during 95,000 to 100,000 ticks and "fractional strategy" means agents that changed their strategy at least once during this period.

First, we can observe that the number of agents stably selecting reciprocity as their behavioral strategy decreased. This is expected because they have a greater chance of forming teams as the workload increases, and thus they may have more chances to change their strategies. Nevertheless, the number of agents stably selecting rationality barely changed around 30% of the agent population and if anything slightly increased in accordance with the workload. Hence, we can say that a number of the reciprocal agents occasionally become rational agents and worked like *freelancers*. We discuss this further in Section 5.5.

5.4 Structural Analysis

We investigated the structure of dependability based on the elements of D_i for $\forall i \in \mathcal{A}$. The structure of the proposed agents in a certain trial at the time of 100,000 ticks is shown in Fig. 5, where $\lambda = 25$ and green nodes are agents with rationality that were isolated. Blue and red nodes correspond to agents with reciprocity that played members (blue) and leaders (red). We can see that, although the upper limit of dependable agents X_F is set to five, all member agents with reciprocity have only one dependable agent. This is because agents can belong to only one team at the same time.

We believe that all agents formed their team on the basis of the network of dependability. Figure 6 shows the network based on teams actually formed during the period from 95,000 to 100,000 ticks. A link between agents is generated only when leader-member relationships were established more than or equal to ten times. Figure 6 appears dense and complicated because agents with rationality (green nodes) pursued their utilities and formed with any agents. Thus, in Fig. 7 we omit the green nodes from Fig. 6. This figure shows when compared with Fig. 5 that almost all reciprocal agents formed teams with only dependable agents having higher priority. However, they were sometimes required to form larger teams, and therefore solicited non-dependable agents that were not in D_i. The agents selected in this situation behaved rationally. Note that there is a single but relatively larger connected component in the upper part of Fig 7. A number of the reciprocal agents in this connected component changed their behavioral strategies during the observation period. However, their selected leaders were limited, so it was neither dense nor complicated like the network in Fig. 6, where a majority of rational agents were connected with more than ten leader agents during the period.

5.5 Discussion

We believe that the mixture of reciprocity and rationality produces an efficient and effective society. The appropriate ratio between these behavioral strategies is still unknown and probably depends on a variety of factors such as task structure, workload, and topology of the agent network. Our study is the first attempt to pursue this ratio by introducing autonomous strategy decision making through social and local efficiency. We also believe that bottom-up construction of organization, such as the group/association structures based on dependability discussed in this paper, is another important issue to achieve a truly efficient society of computer agents like a human society. Thus, another aim of this study is to clarify the mechanism to establish such an organization in a bottom-up manner. Our experimental results suggest that reciprocity is probably what generates the organization, but further experimentation is required to clarify this.

As shown in Fig. 2, if we look at the curve of $I(rational)$ from the society of the proposed agents, it peaked around $\lambda = 25$ to 30, which is near but below the system's limit of task execution. This peak, called the *sweet spot* in (Corkill et al., 2015), is caused by the appropriate organizational structure of the agent society. In our case, the proposed agents established their groups on the basis of dependability through their experience of cooperation. We want to emphasize that this curve indicates an important feature of the organization: namely, that its benefit rises up to the surface when the efficiency is really required. When the system is not busy, any simple method works well, and when the system is beyond the limit of the theoretical performance, no method can help the situation. When the workload is near the system's limit, the potential capabilities of agents must be maximally elicited. The experimental results suggest that the organization generated by the proposed agents partly elicited their capabilities in situation where it was really required.

Figures 5 and 7 indicate that agents generate groups of mostly four or five members on the basis of their dependability, and actually they form teams from only within their groups if the number of subtasks is less than or equal to four or five. Even if they generated larger groups for larger tasks, only one or two agents were solicited from outside of the groups. Because these agents were not beneficial enough for them to stay in the groups of dependability, they dropped out of the groups and behaved rationally. If the solicited agents behaved reciprocally, the solicitation messages might be ignored, and rational agents are thus likely to be solicited. Therefore, rational agents work like freelancers, compensating for the lack of member agents in larger tasks. The role of rational agents from this viewpoint is essential, especially in busy environments: when the workload is high, the rational agents can earn more utilities, thereby increasing the ratio of rational agents as shown in Fig. 3.

6 CONCLUSION

We proposed agents that switch their behavioral strategy between rationality and reciprocity in accordance with internal states on the basis of past cooperative activities and success rates of task executions in order to achieve efficient team formation. Through their cooperative activities, agents with reciprocal behavior established groups of dependable agents, thus improving the efficiency of team formation by avoiding conflicts, especially in large and busy environments. We experimentally investigated the performance of the society of the proposed agents, the structures of selected roles and behavioral strategies, and the agent network through actual cooperation in the same teams.

Our future work is to more deeply investigate the bottom-up organization by the proposed agents. For example, we have to examine how the entire performance is affected by the ratios of reciprocity to rationality. We also need to clarify the protocols to explicitly form an association based on mutual dependability.

REFERENCES

Abdallah, S. and Lesser, V. R. (2004). Organization-based cooperative coalition formation. In *2004 IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology (IAT 2004)*, pages 162–168. IEEE Computer Society.

Corkill, D., Garant, D., and Lesser, V. (2015). Exploring the Effectiveness of Agent Organizations. In *Proc. of the 19th Int. Workshop on Coordination, Organizations, Institutions, and Norms in Multiagent Systems (COIN@AAMAS 2015)*, pages 33–48.

Coviello, L. and Franceschetti, M. (2012). Distributed team formation in multi-agent systems: Stability and approximation. In *Proc. on 2012 IEEE 51st Annual Conf. on Decision and Control (CDC)*, pages 2755–2760.

Dunin-Keplicz, B. M. and Verbrugge, R. (2010). *Teamwork in Multi-Agent Systems: A Formal Approach*. Wiley Publishing.

Fehr, E. and Fischbacher, U. (2002). Why Social Preferences Matter - The Impact of Non-Selfish Motives on Competition. *The Economic Journal*, 112(478):C1–C33.

Fehr, E. and Fischbacher, U. (2004). Third-party punishment and social norms. *Evolution and Human Behavior*, 25(2):63–87.

Fehr, E., Fischbacher, U., and Gächter, S. (2002). Strong reciprocity, human cooperation, and the enforcement of social norms. *Human Nature*, 13(1):1–25.

Foster, I. (2002). What is the grid? a three point checklist. *Grid Today*, 1(6).

Genin, T. and Aknine, S. (2010). Coalition formation strategies for self-interested agents in task oriented domains. *2010 IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology*, 2:205–212.

Gintis, H. (2000). Strong reciprocity and human sociality. *Journal of Theoretical Biology*, 206(2):169 – 179.

Glinton, R., Scerri, P., and Sycara, K. (2008). Agent-based sensor coalition formation. In *2008 11th Int. Conf. on Information Fusion*, pages 1–7.

Gu, C. and Ishida, T. (1996). Analyzing the Social Behavior of Contract Net Protocol . In de Velde, W. V. and Perram, J. W., editors, *Proc. of 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW 96), LNAI 1038*, pages 116 – 127. Springer-Verlag.

Hayano, M., Hamada, D., and Sugawara, T. (2014). Role and member selection in team formation using resource estimation for large-scale multi-agent systems. *Neurocomputing*, 146(0):164 – 172.

Nain, G., Fouquet, F., Morin, B., Barais, O., and Jezequel, J. (2010). Integrating IoT and IoS with a Component-Based Approach. In *Proc. of the 36th EUROMICRO Conf. on Software Engineering and Advanced Applications (SEAA 10)*, pages 191–198.

Panchanathan, K. and Boyd, R. (2004). Indirect reciprocity can stabilize cooperation without the second-order free rider problem. *Nature*, 432(7016):499–502.

Parunak, H. V. D. (1987). Manufacturing experience with the contract net. In Huhns, M., editor, *Distributed Artificial Intelligence*, pages 285–310.

Sandholm, T. and Lesser, V. (1995). Issues in automated negotiation and electronic commerce: Extending the contract net framework. In Lesser, V., editor, *Proc. of the First Int. Conf. on Multi-Agent Systems (ICMAS'95)*, pages 328–335.

Schillo, M., Kray, C., and Fischer, K. (2002). The Eager Bidder Problem: A Fundamental Problem of DAI and Selected Solutions. In *Proc. of First Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS2002)*, pages 599 – 606.

Sheholy, O. and Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Journal of Artificial Intelligence*, 101:165–200.

Sims, M., Goldman, C. V., and Lesser, V. (2003). Self-organization through bottom-up coalition formation. In *Proc. of the Second Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 867–874, New York, NY, USA. ACM.

Sless, L., Hazon, N., Kraus, S., and Wooldridge, M. (2014). Forming coalitions and facilitating relationships for completing tasks in social networks. In *Proc. of the 2014 Int. Conf. on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 261–268.

Smith, J. M. (1976). Group Selection. *Quarterly Review of Biology*, 51(2):277–283.

Smith, R. G. (1980). The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, C-29(12):1104–1113.

Stankovic, J. (2014). Research Directions for the Internet of Things. *IEEE Internet of Things Journal*, 1(1):3–9.

Zhang, Y. and Parker, L. (2013). Considering inter-task resource constraints in task allocation. *Autonomous Agents and Multi-Agent Systems*, 26(3):389–419.