IPSJ SIG Technical Report

Vol.2009-MPS-74 No.2
2009/7/13

New Solution for Resource Allocation Problem of Agent Teams Governed by a Leader

Asaki Nishikawa†1 and Masahiko Hara†1

Previously, we had proposed a multi-agent model for simulating the differentiation between a leader and its followers. However, in higher population settings, this model suffers from the leader's loss of its functionality. To solve this problem, we designed a new method that helps the leader to survive in such situations. By making use of psychological facts, this method employed two teams of agents assigning the adequate resources to another team's leader each other. In noiseless settings, we found that this method leads to some deadlock. Adding noise prevented this deadlock, although it collapsed the leader of the other team alternately. Another negotiation model was exploited to cope with this situation. This model succeeded in solving the first problem by providing a different mechanism to maintain the leader's reputation.

1. Introduction

Embodiment has been said to be a key concept for developing a new type of A.I having capabilities that go beyond those of symbolic A.I[11]. However, hardware bodies that incorporate A.I. software appear to come up against a brick wall. For example, self-organized robot teams without a leader have not been able to go beyond insect's intelligence[2]. In the case of humanoid robots with a software brain, it is necessary to manually program almost all possible sequences of actions beforehand[15]. In both cases, it remains unclear why robots with a body are more functional than symbolic A.I. For example, we do not know why living creatures have only one brain, as opposed to two or three brains. We already know biological causalities concerning this question[3]. However, a theory is required to explain the differentiation between one brain (or a leader) and a body (or followers) in more abstract layer.

To clarify the reasons why we require A.I with the body, in a more abstract layer, we refer to developmental psychology because it is one of the theories that explains the causalities of the above-mentioned differentiation in a higher layer than biology. In the context of the developmental psychology, it appears that the human body image has the following functional meanings[14)4]. Human infants cannot control their own body well in the early stages of their life because their nervous systems are not completely formed. In this stage, human infants appear to require images of a unified body (mirror image of their own or another person's body) to learn how to control their own non-cooperative bodies. Infants map their body parts onto a body image and learn to control their body with the help of this image. Probably, the mirror-neuron systems are biological bases for this learning[12)5].

The well-known "theory of mind" is another developmental psychological concept allowing children to interpret others' actions as an expression of one pivotal intention[1]. The children with "theory of mind" become capable of deceiving others. In addition to the above interpretation, the deception requires children to be capable of distinguishing their own intention from that of others. This is an ability that autistic children lack.

From the viewpoint of computer science, the body image appears to be a method to integrate distributed systems that hardly coordinate with each other. The body image helps infants to integrate their own body parts of their body into one governed system by using the similarity of self and others in mirrors. "Theory of mind" helps the children interpreting another person as a system controlled by one intention (or mind) that is different from themselves. We consider whether it is possible to simulate a system having such properties in an abstract manner.

2. Problem and Model

First, we consider the conditions that an abstract, bottom-up, model needs to satisfy for the differentiation of body parts and a brain (or a control/intention). (1) It has to simulate distributed systems controlled by a pivotal element. (2) (1) has to be achieved by self-organization without an external observer or designer. (3) Elements of the systems are agents, each of which have their own desire or purpose and cannot coordinate without (1).

†1 Flucto-Order Functions Asian Collaboration Team, RIKEN Advanced Science Institute

2.1 Our Previous Work

We have already designed a model that satisfies the above-mentioned conditions[8].

The following is a brief description of our model that simulates the emergence of a pivotal or leader agent. In our model, one agent team is composed of N agents. Each agent has one ability and one demand. The ability of each agent can be executed when it is demanded by another agent. "The demand" is a request for executing the ability of another agent. We modeled a situation in which each agent demands another agent's ability. We call this exchange of abilities "trial of coordination".

When the simulation begins, the system randomly selects agent A, and searches for agent B having an ability demanded by agent A . If agent A coincidentally has the ability demanded by agent B (we call this situation "double coincidence of demands", as described in[6]), these agents successfully exchange their abilities (a success of coordination). However, coordinations hardly occur because double coincidences of demands are formed rarely. To resolve this, we added the concept of "authority" to our model.

The concept of authority is similar to the concept of reputation in game theory[10)13]; however, it has a different meaning in our model. First, the authority of agent A from the viewpoint of B increases when the following condition is met. Assume that agent A is the executor of B's demand (i.e., agent A is "server") and B is the requester (i.e., agent B is a "client"). If server agent A refuses to execute the demand of client B, the authority of the server from the viewpoint of the client increases. In other words, the authority increases when the server is able to execute the client's demand but the client is not able to execute the server's demand.

Furthermore, the steps described below were added to the trial of coordination. If the authority of client Y for server X is higher than x (a constant real number shared by all agents), then server X executes client Y's demand, but server X's demand is suspended. Agent X delegates its demand and ability to client Y, expecting that in the future, the demand will be executed. In this case, the authority can be considered to express the expectation of the execution of a delegated demand. Delegation can be considered as a caching operation that makes a temporary copy of the server's demand and ability. We call a delegated demand "suspended demand (SD)" and the ability "proxy ability (PA)."

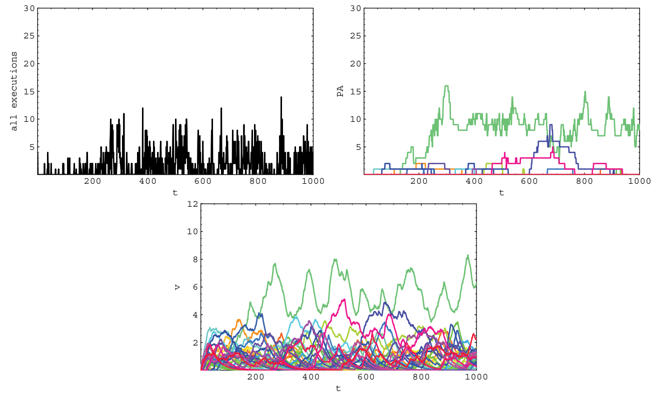

Fig. 1 We plotted the time series of *execution*, pa, and v for all agents in the system. The parameters are $N = 30$ and $x = 6$. The value of the execution indicates the number of successful coordinations. Here, the plotted pa represents $\sum_j^N pa_{ij}$, i.e., the amount of information (PAs) gathered in each agent i. v represents the intensity of the authority that an agent has; $\frac{\sum_j^N v_{ij}}{N}$, the average of v_i from all agents.

A PA is used as follows. A client Y having some PAs can execute a server's demand by using one of its PAs instead of its own ability. Conversely, agent Y can use an SD when it is a server. The agent Y that has SD can choose a demand from among its SDs instead of its own demand when a client can execute the SD. When an SD is consumed, a notification is sent to the agent that has delegated the SD, and its demand is reset because its demand has now been executed.

Before one trial of coordination, both agents negotiate their own authorities. We call each trial of coordination *step*, and N steps a *turn*. Other variables and more details of its dynamics are explained in the Appendix and in[7].

Fig. 1 shows a summary of the main results of our previous model. These figures show a time series of *execution*, pa, and v for all agents in the system. The value of execution indicates the total number of successful coordinations. Here, the plotted pa represents $\sum_j^N pa_{ij}$, i.e., the amount of system information (PAs) gathered in each agent i. v represents the intensity of the authority that an agent has; $\frac{\sum_j^N v_{ij}}{N}$, the average of v_i from all agents. After 200 turns, the execution amount observed to increase. This increase occurs in sync with increase in the pa and v. This implies that one pivotal agent owns a large amount of information about the other agents. The leader agent has a strong authority forcing other agents to be subjected to its order. This self-organization of agents' relationships leads to an increase in the execution amount.

However, our model has one disadvantage. If we increase the population in this model, the effect of the leadership agent disappears. This is because while the leadership agent is an agent that possesses an information map of other agents and a strong authority, it is simultaneously a mere agent having limited resources equal to those of other agents for performing actions. When the population is large and client selection is random, the probability of a leader being selected as a client decreases. The leader is engulfed in a swarm of other agents. If the leader agent is selected as a server, there is little opportunity for a client agent to meet the leader's demand.

This problem was solved in an ad hoc manner by introducing an external observer for preferentially selecting the pivotal agents[8)]. However, this solution has conflicts with the condition of the bottom-up model used for simulating the body.

2.2 New Solution: Selection like Mirrors

In this study, we attempt to find a solution for solving the selection problem in a bottom-up manner. In the abstract layer, our selection problem has a structure with some type of resource allocation problems. (1) To enable a complex organized systems to work efficiently, allocate limited resources for elements in them. (2) Allocation failure leads to a collapse in the organization. (3) Information on the organization is limited, and it may be noisy.

From the viewpoint of psychology, an infant requires a mirror (or others person's body image) to obtain the image of an integrated body. In the context of our study, the "theory of mind" enables one to find the pivotal element in another person's body image. Human infants appear to make use of this technique by using the mirror images of others and functional similarities between themselves and others.

Considering this factor, we introduce a fourth component to the selection problem. (4) The organization of the resource allocator and the target for the allocation are homological. In[8)], the resource allocator is an external system having no relation to the organization, whereas in this paper, the allocator is also an organization.

However, there is no established theory that explains how mirrors can be used to integrate body parts and how another person's mind/intention can be inferred.

We designed our solution by referring to the above facts about infants and on the basis of the following assumption. The "mind/intention" is similar to something that governs its own body (or has the right to decide). However, subjectively, the "mind/intention" is feeling governed by present desires from one's own body. We assume that children infer that other's "mind/intention" is similar to an owner of an object for the present desire.

In our model's framework, this logic is mapped as follows. In the present organization, a server with an ability demanded by a client has the right to decide. The server is the owner of the present desired object. An agent similar to the server must be a pivotal agent of another organization. This team selects this agent as the next client of another team.

Thus, the "mind/intention" selects the partner's "mind/intention." The selected "mind/intention" conversely selects a partner's "mind/intention" by the same logic. If the selected "mind/intention" is actually a pivotal agent, this selection method constitutes logic in which two agent teams specify a mutual leader. Because of this structure of continuous selection of each other, we call this algorithm "selection like mirror."

Our new agent selection algorithm is described below. We assume two teams of agents, called A and B.

(1) Randomly select $client^0_A$ that is a client agent of team A at step 0.
(2) Determine an server agent for $client^0_A$. This is $server^0_A$.
(3) Assume that an attribute vector of $server^0_A$ used for this selection is ι. ι may be pa, sd, or v. In this study, we used pa. For the definitions of these variables, see the Appendix. A sufficient condition for ι is that its norm can be calculated. We call this norm $n(\iota^0_A)$. $n(x)$ is the Euclid norm; $\sqrt{\sum_i^N |x_i^2|}$

4) Calculate the distances between the norms of all agents in team B and the norm of $server^0_A$. We call these distances $d(n(\iota^0_A), n(\iota^0_{B_k}))$ (k is the index of agents in team B). Here, $d(x, y) = |x - y|$.[⋆1]
5) Select an agent k^* in B that has the minimum distance to the norm of $server^0_A$'s ι.
6) Agent k^* of (5) is a new client of team B. We call it $client^1_B$.
7) Repeat steps (1) to (6) by using $client^1_B$ instead of $client^0_A$.

We required a the random selection at step 0 in order to bootstrap this algorithm. After the bootstrap, we can continue this selection algorithm for an arbitrary number of steps.

3. Results and Some Modifications

Fig. 2 shows a time series of the two agent teams with the selection like mirrors. Each of the teams is the same as the agent group shown in Fig. 1. From Fig. 2, it is observed that the mutual selections lead to an increase in the coordination level of both teams. However, in the figures in the middle row, many gaps appeared. These reflect a type of deadlock involved in the selections. During these gaps, only four agents forming a closed loop were calling each other.

These loops are formed as follows. Assume that client i in team A demanded server j's ability. Server j was similar to agent k in B. Agent k was selected as a client of B. Agent k demanded an ability of server agent l. Agent l was occasionally similar to the first client agent i in A, and coordination was not successful. Then, the loop was formed. The two teams cannot escape from this loop without an help of a random change in demands in the loop. If N increases, the probability of the simulation "detecting" this loop also increases.

Fig. 3 shows the time series shown in Fig. 2 with noise. 5% bit flips to information vector ι were added. We observe that the gaps seen in Fig. 2 disappear. The coordination level of this system was slightly greater than the one shown in Fig 2. The noise helped the teams to break the loops. In a situations with a larger value of N, the noise is necessary because more loops occur.

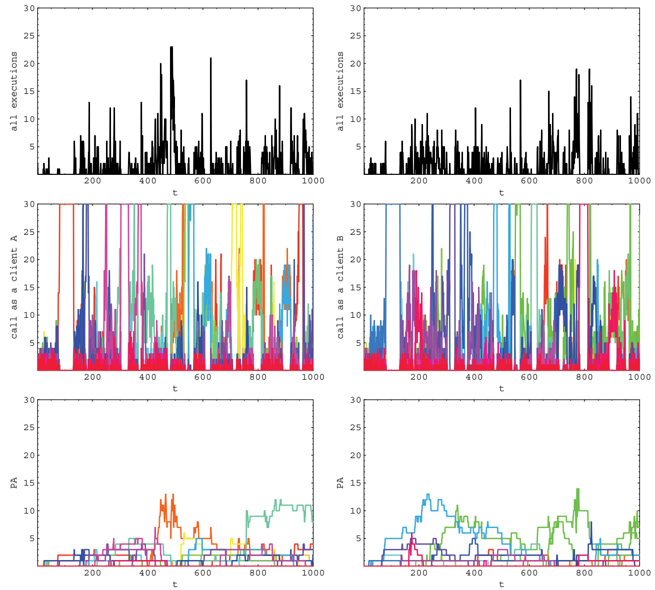

Fig. 2 Time series with selections like mirrors. The time series on the left-hand side shows a plot of the variables of team A. The one on the right-hand side shows a plot of those of B. The other parameter settings are the same as those in Fig. 1. The figures in the upper row show the execution amount. Those in the middle shows values of cac called as a client. This value counts the number of calls to an agent i as a client in one turn. The figures in the bottom row show pa.

⋆1 We added a very small amount of noise to these norms to avoid the appearance of equal distances. We used a normal distribution that has average 0 and standard deviation $\frac{1}{10^4}$.

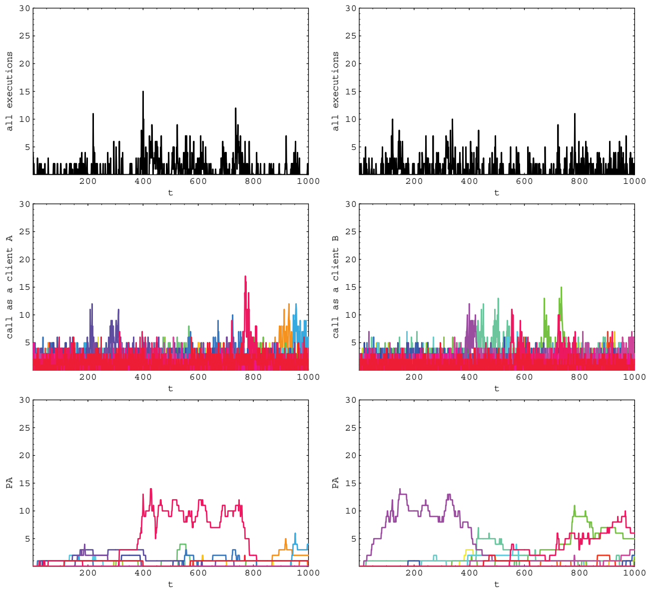

Fig. 3 Time series shown in Fig. 2 with noise. 5% bit flips to information vector ι were added.

Fig. 4 shows a simulation with higher value of N (N = 66). In this situation, the execution levels are the same as those of the one shown in Fig. 3. The imbalances of the two teams' organizations involving pivotal agents were magnified as compared to that shown in Fig. 3.

We already have a solution for solving this problem. This solution involves the simulation of some type of indeterminacy in the negotiations. This model originally dealt with the origin of money without common metrics measuring values, and in the context of this paper, a situation with only one team was considered[9]. Originally, our model for the simulation of pivotal agents simplified negotiation about authority by averaging them. An authority vector v was normalized to sum N. This implied that all authority values in vector v are relative to the shared image of wholeness, N. Negotiation by averaging could be calculated in the condition where the same value of N and the same interpretation of the meanings of symbols was shared. Our new method described in[9] altered

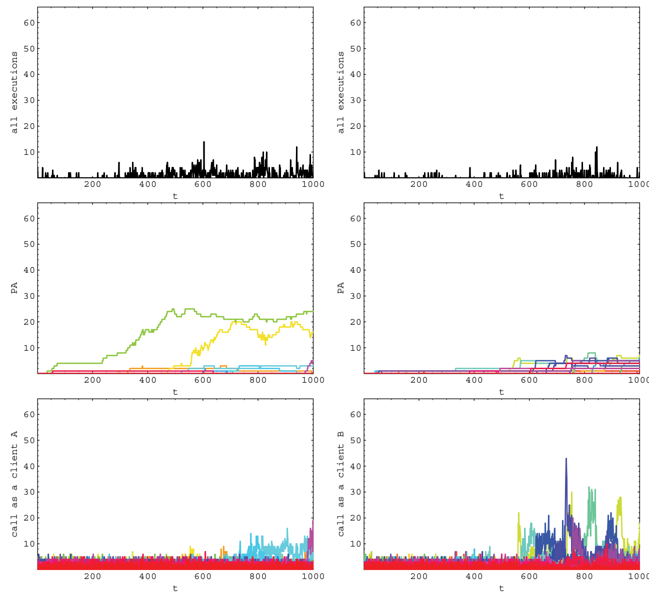

Fig. 4 Time series shown in Fig. 3 with a higher value of N (N = 66).

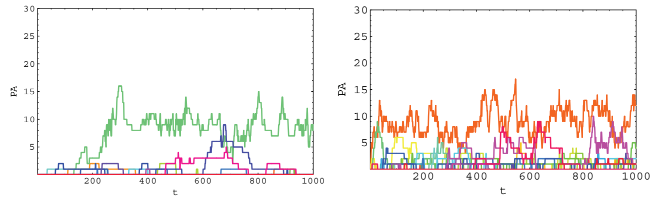

Fig. 5 The left figure is the same as Fig. 1. The right is a time series of the model equipped with relative authorities.

these conditions of simple negotiation. We introduced another negotiation regarding the context/wholeness (each agent's totalities N_i) of authority and the interpretation of meanings of symbols, before the averaging of authority vectors. Authority vectors with these negotiations are called "relative authority". For more details, see the Appendix ⋆1.

Fig. 5 shows the comparison of the time series of a model with relative authorities and one with simple authority. Both have a pivotal/leader agent. However, the differences between the pivotal agent and other agents are relatively minor in the relative authority model as compared to the original one. The former has many local leaders. The coordination levels of both are similar. The relative authority model is more robust than the original one with regard to the range of x, i.e., for almost all value of x value lesser than N, it can generate a leader. The original authority model requires a small limited range of x for generating a leader. The effects of these properties of the relative authority model have not yet tested in a two-team settings.

Fig. 6 shows the same time series as that in Fig. 4 with relative authorities. The organizations of the two teams are very similar like mirror images. The execution levels were approximately 10 times higher than that in the case of the ordinary authority mechanism. The robust leaders in both teams select each other, along with appropriate fluctuations. The selection like mirrors effectively functioned to allocate resources to the team by distinguishing the pivotal agent and other agents.

As shown in Fig. 5, a pivotal/leader agent in the relative authority model has many local leaders surrounding it. In comparison, a leader in the original model has almost no local leader around itself. This is the main difference between the two models and leads to the difference between the time series. However, it is incorrect. As seen from figures in the middle row of Fig. 6, with the two-team settings, the local leaders disappear.

We compared the execution processes of the original and relative models. When the alternations of the leaders occurred (Fig. 4) , there were short periods in

⋆1 We have already submitted details of this method for another model in Japanese[9]. An international journal submission is being prepared.

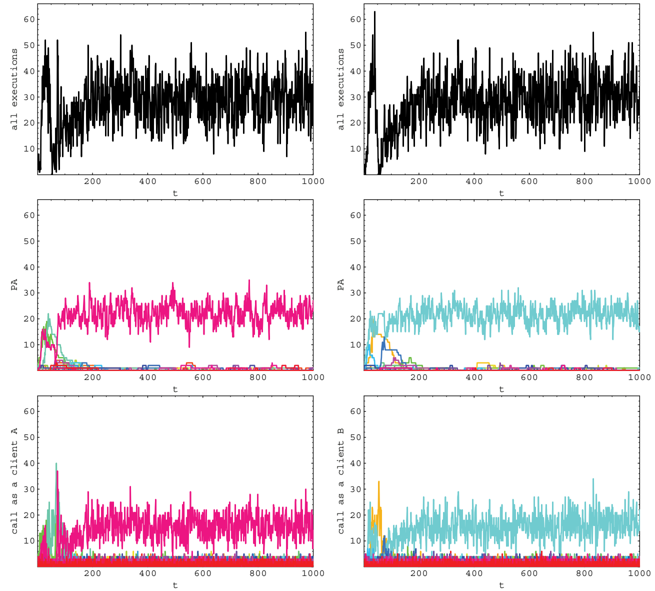

Fig. 6 The same time series as that in Fig. 4 with relative authorities.

which each of the two teams had a leader. A closer study of the execution in this period revealed that the mutual selections occurred as in the case of the relative authority model. However, one of the two leaders did not have sufficient authority to subject others to its orders. This leader cannot change its demands and therefore it gradually loses its PAs and collapses. Fig. 7 shows that the differences of authorities in the original model are smaller than the relative one.

The leader in the relative authority model has many local leaders in the PA space, whereas in the v space, its authority is highly centered. This is counterintuitive. The appearance of the local centers in the relative authority model might be explained as follows. In the relative authority model, each agent has a different image of the whole, N_i. This difference allows many leaders to coexist with different levels of authority. When they are generated, these local leaders have a shorter time scale than the main leader, and therefore, they collapse. The absolute values of their authority are smaller than those of the main leader; however,

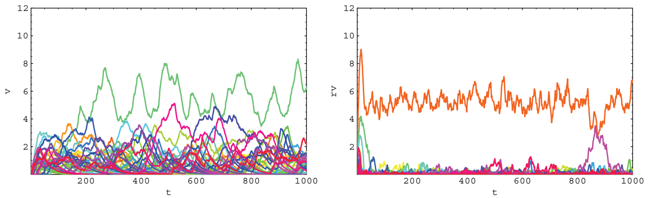

Fig. 7 The time series of authorities with the same configuration as that shown in Fig. 5. The left-hand side shows the original model, and the right-hand side shows the relative authority model.

these local leaders have their own PAs in small N_i communities. This explains the coexistence of the large difference in v and the existence of local leaders. In the original model, all agents share a common scale measure for authority, N. This makes the difference relatively small.

In the two-team setting, two leaders select each other as a client. This inhibits the appearance of local leaders because coordinations are established by the main leaders before local ones can be generated. Therefore, in the relative authority model, there exist only large differences.

4. Conclusions

As an unsolved problem from our previous work, we considered the problem of allocating resources to an agent's team in the situation in which information about the team is limited. A solution that exploits the organizational similarity of the teams was proposed. However, this solution encountered deadlocks. The noise induced fluctuations of the selections helped the system to avoid these deadlocks. However, in the presence of noise, the stability of the pivotal agents could not be maintained. By using a new negotiation model called "relative authority" we recovered the stability of the pivotal agents and solved the above-mentioned problem.

Appendix

We describe some details of our model that are required for understanding this paper.

4.1 Variables

Our model consists of N agents. An agent i has several attributes.

- Demand $\mathbf{d}_i = (d_{i1}, d_{i2}, \ldots, d_{in})$: $d_{ij} = 0$ or 1. $\sum_j^N d_{ij} = 1$
  Here, j is the index of the ability that an agent i demands. If $d_{ij} = 1$, then agent i demands ability j, and if 0, it does not. In this model, an agent only demands one ability at a time, and therefore, the sum of d_i is one. The demand vector is randomly initialized, i.e., $d_{ij'} = 1$ for random j'.

- Ability $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$: $a_{ij} = 0$ (for $i \neq j$) $a_{ij} = 1$ (for $i = j$)
  The ability vector represents an operation that an agent i can execute. If $a_{ij} = 1$, then agent i can reply to the demand that requires index j. In our model, each agent has one ability. Here, for simplicity, agent i has ability i. An ability is fixed and it does not decrease, and therefore, this vector does not change in a time series.

- Authority $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{in})$: $0 \leq v_{ij} \leq N$, $0 \leq \sum_j^N v_{ij} \leq N$
  This vector represents the authority of agent j for i. In other words, if $v_{ij} = r$, then agent i has an authority with intensity r (positive real number) from j. Because parameter x that we define below has value relative to N and we compare v with x, we impose a summation limitation on v. The initial values are all zero.

- Suspended Demand $\mathbf{sd}_i = (sd_{i1}, sd_{i2}, \ldots, sd_{in})$: $sd_{ij} = 0$ or 1
  This vector represents the suspended demands delegated to an agent i. If $sd_{ij} = 1$, then the agent i has a suspended demand that requires ability index j. Suspended demand j is consumed when agent i selects sd_{ij} as an alternative to its own demand. The initial values are all zero.

- Proxy Ability $\mathbf{pa}_i = (pa_{i1}, pa_{i2}, \ldots, pa_{in})$: $pa_{ij} = 0$ or 1
  This vector represents the proxied abilities that an agent i has. If $pa_{ij} = 1$, then agent i has a proxied ability from j. In our model because agent i has ability i in our model, it is not necessary to consider where this ability acquired from. Proxy abilities are a temporary copy of an ability from other

agents, and therefore if agent i uses pa_{ij}, then its value is reset to zero. The initial values are all zero.

4.2 Dynamics

4.2.0.1 One Step of Model

One trial of coordination (exchange of demands) is modeled as below.

(1) Selection of Client

The system selects agent c randomly. We call this agent a "client".

(2) Selection of Server

The client selects agent s that can respond to its demand. We call this agent a "server." The client selects a server from among all agents in the system based on the criterion given below. The client searches for agents that have an ability or a proxy ability w that it wants, i.e., the client selects an agent s that satisfies $d_{cw} = 1$ and ($a_{sw} = 1$ or $pa_{sw} = 1$).

(3) Negotiation about Authority Vectors

Before a trial of coordination, the client and the server negotiate their views of authority. This is simply modeled by averaging their authority vectors, i.e.,

$$v'_{cj} = v'_{sj} = \frac{v_{cj} + v_{sj}}{2}, \text{ for each j}$$

. We normalize the authority vectors, every time they change as given below.

$$\text{if } \sum_j^N v'_{cj} \geq N, \text{ then } v''_{cj} = v'_{cj}\frac{N}{\sum_j^N v'_{cj}}, \text{ for each j}$$

We also normalize v_s in the same manner.

(4) Judging Cooperativity and Execution of Demand

From the definition, a server has the ability to satisfy a client's demand; however, the opposite condition has not been tested.

In the first case, we consider that the client can satisfy a server's demand or a suspended demand, i.e., the client's ability can satisfy the server's demand (or the suspended demand) by coincidence or the client's proxy abilities satisfy the demanded ability (($d_{sw'} = 1$ or $sd_{sw'} = 1$) and ($a_{cw'} = 1$ or $pa_{cw'} = 1$)). In this case, the exchange of demands succeeds, and the agents execute their abilities. "Real" executions do not occur in our model because there is no actual implementation of abilities. We simply count the number of executions. We call this quantity $e(t)$ or "all executions." t denotes a "turn." It unites N exchanges.

(a) Using Ability

When an agent executes its ability, the system increments $e(t)$ by 2.

(b) Using Proxied Ability

When a client uses its proxy ability, the system resets $pa_{cw'}$ to 0 and increments $e(t)$ by 2.

In this case, the agent that delegated this proxied ability receives a notification and some procedures begin in order to inhibit the abuse of proxy abilities. For more details about this procedure, see[7].

The client satisfies its demand and its demand is reset to zero. The server requires more complicated procedures. If the server's suspended demand is satisfied, its own demand still persists. Of course, if the server's demand was also satisfied, both are reset to zero ($d_{cw} = d_{sw'} = 0$).

If the server used a suspended demand, then additional procedures notify the relevant agents of the use of it[7].

Next, we consider a case in which the client cannot meet both the server's demand and the suspended demand.

(a) Delegating Demand and Ability

If the server, finds authority of the client to be greater than x (i.e., $v''_{sc} \geq x$), it suspends and delegates its demand to the client (i.e., $sd_{cw'} = 1$). It also delegates its ability to client c, i.e., $pa_{cs} = 1$. Note that x is a constant shared by all agents. In this case, only the server executes the client's demand, and therefore, the system increments $e(t)$ by 1. The client's demand is reset to zero (i.e., $d_{cw} = 0$).

(b) **Declining Demand**
If the server finds the authority of the client to be less than x (i.e., $v''_{sc}x$), it declines the client's demand. In this case, the authority of the server in the client increases by 1.0 (i.e., $v''_{cs} + 1.0$). Because no executions occur, $e(t)$ does not change.

(5) **Updating Demands**
Because of these processes, d_c or d_s is reset to 0, and then, for a randomly selected index k, d_{ck} or d_{sk} is updated to 1. When the update of the demand of agent u occurs (i.e., d_{uo} becames $d_{uo'}$), all suspended demands of u also change simultaneously, as described in[7].

4.2.0.2 One Turn of Model

We call these five processes one "step". As mentioned above, one turn comprises N steps. At each turn, the system selects one agent k and changes its demand randomly. This prevents the system from falling into equilibrium with no exchange/cooperation.

Details about the model that have been omitted here are explained completely in[7].

4.3 Relative Authority

4.3.1 The Relative Whole and Units

Most of the conceptual description is omitted here. However, if required, it can be referred from[9] or a forthcoming paper.

As the relative whole, N_i^t for each agent i is used. Each agent calculates N_i^t at the end of a turn using memories of a latest T turn as follows.

n_i^t is the total number of different agents encountered with i in the past T turn. In this study, $T = 10$. An "encountered agent" may be one among the following. (1) It has the ability to satisfy the demand of i. (2) It requires i's ability. (3) It was included in the information offered at the time of a negotiation. For the same agent, it is counted only once. N_i^t is updated as $N_i^{t+1} = n_i^t$. N_i^0 is randomly determined such that it is between $[2, \frac{N}{5}]$. We determined 2 as the minimum value of the relative whole. The initial maximum was set lower than N in order to exclude very large value of N_i^0.

On an unit 1_i^t for each agent, it is updated as follows at the end of a turn. $1_i^{t+1} = \frac{1}{N_i^{t+1}}$

4.3.2 Dialogues

The negotiation is divided into the following steps.

- By dialogues about the relative whole, agents update their interpretations for symbols ($= \psi_i^{t,s}$).
- Agent j communicates a numeric value $v_{jm}^{t,s}$ to agent i as a symbol of an opinion on agent m.
- i interprets the symbol by $\psi_i^{t,s}$.
- j performs the same processes as i.
- Each agent averages the authority vectors.

An opinion is sent to a negotiation partner only when it satisfies the condition $v_{ik}^{t,s} \geq 1_i^t$.

4.3.3 Updates for Interpretations

Assume N_j^t is a relative whole sent by j. Agent i calculates $R_{ij}^{t,s,0}$ by the interpretation $\psi_i^{t,s,0}$.

$$R_{ij}^{t,s,0} = \frac{\psi_i^{t,s,0}(N_j^t)}{N_i^t \times \psi_i^{t,s,0}(N_j^t)} + \frac{N_i^t}{\psi_i^{t,s,0}(N_j^t) \times N_i^t} \tag{1}$$

$$= \frac{1}{N_i^t} + \frac{1}{\psi_i^{t,s,\tau}(N_j^t)}. \tag{2}$$

Agent j also calculates $R_{ji}^{t,s,0}$. The upper third index of $\psi_i^{t,s,0}$ represents the total number of updates in the negotiation.

Next, as a criterion the a termination of updates, $d_{ij}^{t,s,0} = d_{ij}^{t,s,0}$ is calculated. $d_{ij}^{t,s,0}$ is defined as $|R_{ij}^{t,s,0} - R_{ji}^{t,s,0}|$. The updates terminate when $1_i^t \geq d_{ij}^{t,s,\tau}$. If both agents terminate the updates, they proceed to the next procedure. If both fail, by an algorithm mentioned below, each agent updates its interpretation to $\psi_i^{t,s,\tau+1}$ and repeats the calculation of $d_{ij}^{t,s,\tau+1}$.

When only agent i among the two agents terminates the interpretation, with updated $\psi_j^{t,s,\tau+1}$ of j, both agents re-calculate $d_{ij}^{t,s,\tau+1}$. If $d_{ij}^{t,s,\tau+1} \leq d_{ij}^{t,s,\tau}$, then two agents restart from calculating $\psi_i^{t,s,\tau+2}$. Otherwise, the interpretation updates are forced to terminate.

4.3.4 Implementation of the Interpretation

In this study, the update method for $\psi_i^{t,s,\tau}$ is implemented along with the following conditions.

- Each agent has a minimum learning ability that allows it to memorize past random decisions and use these results until a problem is encountered.
- The value below 1_i^t is not discriminable.

$\psi_i^{t,s,\tau}(n)$ is implemented as a table that holds entries corresponding to the value for each n. The algorithm for calculating one return value for it is as follows.

- A set of entries corresponding to the index contained in the domain $(n \pm \frac{1}{2}1_i^t)$ is selected.
- If there is no entry in the set, a value corresponding to n is randomly determined in the range of $[2, N_i^t]$. The obtained result is returned as an answer, and its value is memorized as a new entry.
- If the entry set contains elements, one of them is selected at random and it is returned.
- When updating the table, a value for the index is changed to a randomly selected value in the range of $[2, N_i^t]$.

In addition, $\psi_i^{t,0,0}$ is a table holding no entries.

4.3.5 Transfer of Opinions

After the dialog on the relative whole, opinions are exchanged. An opinion is averaged as follows.

$$v_{ik}^{t,s} = \frac{1}{2}\left\{\frac{v_{ik}^{t,s} \times \psi_i^{t,s,and\tau}(N_j^t)}{N_i^t \times \psi_i^{t,s,\tau}(N_j^t)} + \frac{v_{jk}^{t,s} \times N_i^t}{\psi_i^{t,s,\tau}(N_j^t) \times N_i^t}\right\} \quad (3)$$

$$= \frac{1}{2}\left\{\frac{v_{ik}^{t,s}}{N_i^t} + \frac{v_{jk}^{t,s}}{\psi_i^{t,s,\tau}(N_j^t)}\right\} \quad (4)$$

If $v_{jk}^{t,s}$ is not sent, it is treated as zero.

Threshold of Authority and the Relative Unit

1_i^t is used as a unit for the change in authority. Moreover, when the authority changes, it is normalized as $\sum_k^N v_{ik}^{t,s} = N_i^t$.

In addition, during a trial of coordination, the delegating condition is altered as

$$v_{ik}^{t,s} \geq \frac{x}{N} \times N_i^t \quad (5)$$

References

1) Baron-Cohen, S., Tager-Flusberg, H. and Cohen, D. J.: *Understanding Other Minds: Perspectives from Developmental Cognitive Neuroscience*, Oxford University Press (2000).
2) Brooks, R.: A robust layered control system for a mobile robot, *Robotics. and. Automation,. IEEE. Journal. of.*, Vol.2, No.1, pp.14–23 (1986).
3) Cebria, F., Kobayashi, C., Umesono, Y., Nakazawa, M., Mineta, K., Ikeo, K., Gojobori, T., Itoh, M., Taira, M. and Alvarado, A. S.: FGFR-related gene nou-darake restricts brain tissues to the head region of planarians, *Nature*, Vol.419, No.6907, pp.620–624 (2002).
4) Gallagher, S. and Meltzoff, A.: The earliest sense of self and others: Merleau-Ponty and recent developmental studies, *Philosophical Psychology*, Vol.9, No.2, pp. 211–233 (1996).
5) Ishida, H.: BSI Forum: Shared mapping of own and others ’ bodies in visuo-tactile bimodal area of the monkey parietal cortex (2008).
6) Menger, C.: *Principles of Economics*, New York University Press (1981).
7) Nishikawa, A.: The Emergence and Collapse of the Self Monitoring Center in Multi-agent Systems., *KES-AMSTA* (Nguyen, N. T., Jo, G., Howlett, R. J. and Jain, L.C., eds.), Lecture Notes in Computer Science, Vol.4953, Springer, pp.803–812 (2008).
8) Nishikawa, A.: Mandate Game : Model Of Anticipation Exchange And Decision-Making, *International Journal of Computing Anticipatory Systems* (Dubois, D., ed.), Vol.20-22 (2008).
9) Nishikawa, A.: On the Origin of Money without a Unit of Measure, *The 13 th JAFEE Congress in Okayama "Coevolution and the Diversity of Evolutionary Path"*, http://www.e.okayama-u.ac.jp/jafee/paper/c33.pdf (2009). (In Japanese).
10) Ohtsuki, H., Iwasa, Y. and Nowak, M. A.: Indirect reciprocity provides only a narrow margin of efficiency for costly punishment, *Nature* (2009).
11) Pfeifer, R. and Scheier, C.: *Understanding Intelligence*, MIT Press (1999).
12) Rizzolatti, G. and Craighero, L.: The Mirror-Neuron System, *Annual Review of Neuroscience*, Vol.27, No.1, pp.169–192 (2004).
13) Santos, F. C., Santos, M. D. and Pacheco, J. M.: Social diversity promotes the

emergence of cooperation in public goods games, *Nature*, Vol. 454, No. 7201, pp. 213–216 (2008).

14) Vyt, A.: *Self-Recognition in Infants: Some Reflections Beyond the Mirror*, Lawrence Erlbaum Associates (1994).

15) Yamashita, Y. and Tani, J.: Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment, *PLoS Computational Biology*, Vol.4, No.11 (2008).