Weighted Opinion Sharing Model for Cutting Link and Changing Information among Agents as Dynamic Environment

Fumito UWANO *, Rei SAITO *, and Keiki TAKADAMA *

Abstract: This paper proposes a weighted opinion-sharing method called conformity-autonomous adaptive tuning (C-AAT) that enables agents to communicate and share correct information in a small-world network even when the links and information change dynamically. Concretely, each agent estimates weights for each of its neighbors by comparing their opinions with its own, increasing the weight if both are the same and decreasing it otherwise. To investigate the proposed method’s effectiveness, experiments were conducted for three scenarios: (1) a static network with sensor agents that were almost equally likely to share incorrect environment information; (2) a static network with sensor agents whose probability of sharing incorrect information changed over time; (3) a dynamic network where some agent links were randomly cut over time. The experimental results led to three conclusions about C-AAT: (i) it can make the agents’ opinions robust against incorrect sensor agent opinions by decreasing the weights; (ii) it can decrease the weights of agents conveying incorrect opinions with varying probabilities to prevent incorrect opinions being shared; and (iii) it can help agents share correct opinions by increasing the weights of their neighbors even if the agents receive fewer opinions due to links being cut.

Key Words: opinion-sharing, agent, weight, link, information.

1. Introduction

In society, people often communicate with others in order to form their own opinions: they discover others’ opinions through communication, then use them to form their own opinions. To simulate such decision-making processes, Glinton et al. proposed a multi-agent opinion-sharing model [1] that regards the agents as people who are communicating with others in order to form their own opinions. In this model, there are a small number of so-called sensor agents and many normal agents. The sensor agents can receive correct information from the environment, while the normal agents can only receive opinions from their neighboring agents. The agents share their opinions as follows.

(1) The sensor agents receive correct information and form their opinions based on it.

(2) The sensor agents convey their opinions to neighboring normal agents.

(3) Normal agents that have received opinions form their own opinions and convey them to neighboring agents.

(4) Neighboring agents that have received opinions also form their own opinions and convey them to others.

It should be noted here that the opinions received need not be correct, leading to contributions of agents that wrongly share incorrect opinions. To encourage the agents to form correct opinions despite conveying both correct and incorrect ones, Pryymak proposed the autonomous adaptive tuning (AAT) algorithm [2]. This can improve the proportion of correct opinions shared in networks of various sizes where the agents might convey incorrect opinions. However, it cannot be applied to dynamic networks because it assumes that all agents’ opinions have the same influence, sensor agents acquire correct information with constant probability, and agent links are never cut (i.e., the network is static). Dynamic networks are important because real-world networks are not static and real people build relationships based on kindness, trust, social standing, or family, and generally weight others’ opinions based on their relationships with them. To handle such relationships, this paper proposes an improved AAT algorithm where the agents form opinions by considering their relationships with neighboring agents. Using this improved algorithm, we aim to investigate how relationships can help us (or the agents) share correct opinions. Note that the aim of this research is not to consider the new algorithm’s implications for human society or investigate how it causes the agents develop human-like behaviors, but simply to develop agents that can provide correct information to users even when encountering incorrect information. This includes, for example, agents that can provide correct information (e.g., this road is safe) in disaster situations. Our previous research has confirmed that our proposed agents can provide correct information even when encountering incorrect information in static environments (where the inter-agent weights are constant and the agent links are always connected), so we now investigate whether our improved AAT agents can cope with the same issues in more complex dynamic environments (where the inter-agent weights change as neighboring agents become unreliable and agent links can become disconnected if the network goes down). The agent relationships are implemented using a weighted network where the weights influence the agents’ decision-making processes. To investigate our new algorithm’s effectiveness, we compare the proportion of correct opinions.
opinions it shares with that for the original AAT applied to the same weighted networks.

This paper is organized as follows. Section 2 introduces the opinion-sharing model, and Section 3 describes the AAT algorithm. Section 4 presents the improved AAT algorithm. Section 5 discusses the experiments and examines the results. Finally, Section 7 presents our conclusions.

2. Opinion-Sharing Model

The idea of opinion-sharing was formulated to capture dynamics of the decision-making processes of a network of cooperating agents. In this model, the agents can share their opinions by communicating with their neighbors, while some agents have noisy sensors and can only receive information related to the environment. All agents aim to form correct opinions based on information from their sensors and their neighbors’ opinions, and eventually form a consensus.

The agents aim to propagate correct opinions, subject to the following limitations [1].

• Only the few agents with sensors (i.e., the sensor agents) can observe the environment.

• The sensor agents may form incorrect opinions since the sensors can receive incorrect information.

• The agents can only communicate with their neighbors in the network.

2.1 Overview of the Opinion-Sharing Model

In this model, the network \(G(A, E)\) consists of a large set of agents \(A = \{i_1 : l = 1, \ldots , N\}\), \(N \gg 100\) connected by edges in the edge set \(E\), where \(l\) is the number of neighboring agents connected to agent \(l\) and \(N\) is the total number of agents. Each agent \(i \in A\) can only communicate with its neighbors \(D_i = \{j : \exists (i, j) \in E\}\), and the average number of neighbors is defined as the degree \(d = \sum_{i \in A} |D_i|/N\). The network is sparse because this degree is small for all agents, i.e., \(d \ll N\). The environment state \(b\) involves only binary values, for example \(b \in B\) where \(B = \{\text{white, black}\}\). Here, \(\text{white and black}\) are regarded as correct and incorrect information/opinions, respectively, and all agents have correct opinions if they all form the opinion \(\text{white}\). The set \(B\) is used because it has been argued that binary choices can be applied to a wide range of real-world situations [1]. The aim of the agent community is to find the true state \(b\) observed by some of the sensor agents. In particular, each agent’s aim is to form an opinion \(o_i\) that represents the real state of the environment, i.e., \(o_i = b\). Note that \(o_i = \text{white}\) indicates that agent \(i\) has formed a correct opinion. Each agent forms its opinion by relying on its neighbors’ opinions, while the sensor agents also rely on their noisy sensors. The agents form opinions by having private beliefs \(P_i(b = \text{white})\), corresponding to the probability that \(b = \text{white}\) (denoted as \(P_i\) from now on). Likewise, \(1 - P_i\) corresponds to the probability that \(b = \text{black}\). The agents’ beliefs are updated by starting from some initial prior \(P_i^0\) and then defining later belief states \(P_i^k\), where \(k\) is the current belief update step. A small number of sensor agents \(S \subset A, |S| \ll N\) have noisy sensors and can observe the environment state \(b\). Each sensor agent \(i \in S\) periodically receives an observation \(s_i \in B\) that is of low accuracy \(r\) (\(0.5 < r \leq 1\)). These agents incorporate their sensor observations using a formal update process based on Bayes’ theorem [1]:

\[
P_i^k = \frac{C_{upd} P_i^{k-1}}{(1 - C_{upd})(1 - P_i^{k-1}) + C_{upd} P_i^{k-1}},
\]

where

\[
C_{upd} = r \quad \text{if} \quad s_i = \text{white},
\]

\[
C_{upd} = 1 - r \quad \text{if} \quad s_i = \text{black}.
\]

In these equations, \(C_{upd}\) indicates the degree to which the agent believes the neighbors’ opinions or the information from the environment. Note that \(C_{upd}\) is also utilized for normal agents in Eq. (4).

The thresholds \(1 - r\) and \(r\) are the confidence bounds, and the range is \(0.5 < r < 1\). Figure 1, taken from Pryymak et al. [2], shows that the opinion update function has a sharp hysteresis loop. An agent will only change its opinion from \(\text{white to black}\) if \(P_i^k\) decreases below not \(b\) but \(1 - b\) and vice versa; we call this a hysteresis loop. If a new observation supports the opposite state, an agent may or may not change its opinion, because the received opinion may be incorrect. The agents send new opinions to their neighbors only when their own opinion changes, and their neighbors then update their own beliefs and opinions. The agents incorporate their neighbors’ opinions using a formal update rule similar to the one for the sensor agents. When an agent receives new opinions from its neighbors \(o_j, j \in D_i\), it uses the same belief update rule for each opinion \(o_j\),

\[
O_i^k = \begin{cases} 
\text{undetermined,} & \text{if the opinion is initialized or } k = 0, \\
\text{white} & \text{if } P_i^k \geq \sigma, \\
\text{black} & \text{if } P_i^k \leq 1 - \sigma, \\
{O}_i^{k-1} & \text{otherwise}.
\end{cases}
\]

Fig. 1 Update rule of opinion.

The thresholds \(1 - \sigma\) and \(\sigma\) are the confidence bounds, and the range is \(0.5 < \sigma < 1\). Figure 1, taken from Pryymak et al. [2], shows that the opinion update function has a sharp hysteresis loop. An agent will only change its opinion from \(\text{white to black}\) if \(P_i^k\) decreases below not \(b\) but \(1 - b\) and vice versa; we call this a hysteresis loop. If a new observation supports the opposite state, an agent may or may not change its opinion, because the received opinion may be incorrect. The agents send new opinions to their neighbors only when their own opinion changes, and their neighbors then update their own beliefs and opinions. The agents incorporate their neighbors’ opinions using a formal update rule similar to the one for the sensor agents. When an agent receives new opinions from its neighbors \(o_j, j \in D_i\), it uses the same belief update rule for each opinion \(o_j\),

\[
\text{Eq. (1), where}
\]

\[
C_{upd} = t_i \quad \text{if} \quad o_j = \text{white},
\]

\[
C_{upd} = 1 - t_i \quad \text{if} \quad o_j = \text{black},
\]

where \(t_i \in [0, 1]\) is the importance level, and indicates how much weight agent \(i\) gives to its neighbors’ opinions in forming its own opinion. Note that \(t_i\) does not change the neighbor links because it only affects the number of neighboring agents required to form an opinion. This is a measure of the influence of its neighbors’ opinions and is a conditional probability. The importance level is similar to the accuracy in Eq. (1), but unlike the sensor accuracy, each agent must find its own importance level \(t_i\) because it is initially unknown. The algorithm for determining \(t_i\) is described in Section 3. The agents only consider importance levels \(t_i\) in the range \(t_i \in [0.5, 1]\), with \(t_i = 0.5\) indicating that the agents should ignore all opinions received opinions and \(t_i = 1\) indicating that it should change its own belief to \(P_i^k = (1, 0)\) and ignore its previous belief \(P_i^{k-1}\).
This model may converge to a false state. Accordingly, the agents are identified with these neighbors in themselves. In regard to this model, we consider that the agents are not equated with these neighbors since it may be quite natural.

2.2 Model Performance Metrics

The model is simulated for \( M = \{m_l : l \in 1, \ldots, |M|\} \) rounds. At the start of each round, the new true state \( b^m \in B \) of the environment is randomly selected and the agents reset their opinions and beliefs. At the end of the round, the agents’ conclusions are observed and the current true state expires. During the round, the agents are given a limited number of steps for their opinions to converge (e.g., 2000 steps per round). At each step, the agents observe the current state and receive opinions, using them to calculate their awareness rates (see below), set their importance levels, and store their accuracies.

As a measure of the average accuracy of the agents’ opinions at the end of each round, Glinton et al. proposed using the proportion of agents that formed a correct opinion [3]:

\[
R = \frac{1}{N|M|} \sum_{i \in A} \left| \{m \in M : \alpha_i^m = b^m\} \right|.
\]

(5)

In addition, a performance index was proposed by Pryymak et al. [2] for individual agents. Since an agent cannot determine when it has formed a correct opinion, it measures how often it forms a correct opinion, called the awareness rate:

\[
h_i = \frac{\left| \{m \in M : \alpha_i^m \neq \text{undetermined}\} \right|}{|M|}.
\]

(6)

The value given by Eq. (6) is divided by the number of rounds to calculate the average number of correct opinions formed over the course of all rounds. Each agent calculates its awareness rate at the end of every round and sets its importance level so that its awareness rate is as close to the target as possible. Since this is repeated for each round, the awareness rate will steadily increase over time, which is why it is divided by the number of rounds.

This myopic metric can be calculated locally by each agent and is important for the AAT algorithm described in Section 3.

3. Autonomous Adaptive Tuning (AAT) Algorithm

In this section, we introduce the AAT algorithm, which is designed to improve the accuracy \( R \) for a variety of complex networks by allowing the agents to share their opinions with each other. In this algorithm, the agents automatically update their beliefs by relying only on local information. In particular, it is based on observing that the accuracy \( R \) increases when the opinion-sharing dynamics change phase from a stable state (where opinions are shared only within small communities \( \forall i \in A : h_i \ll 1 \)) to an unstable one (where the opinions are propagated widely, \( h_i = 1 \)). Note that each agent forms its opinion based on only a few other opinions, and may form incorrect opinions due to the unstable sensors when \( h_i = 1 \). As a result, it is essential for the agents to share their opinions with smaller groups before the large cascade occurs so that they do not react to large numbers of incorrect opinions. In order to optimize the parameters to deal with this issue, the algorithm regulates the agents’ importance levels in three stages.

- Each agent creates a list of candidate importance levels to reduce the search space for the following stages. This step happens only once, at the start of the experiment.
- After each round, each agent estimates awareness rates (as described in Section 2.2) for each candidate level.
- Each agent sets its importance level by considering how close each of these awareness rates is to the target. It is essential that it tunes the importance levels gradually when considering the influence of its neighbors.

Note that the importance levels are discrete values, and each agent selects the importance level candidate whose associate awareness rate is closest to the target awareness rate. In the following sections, we describe these three AAT algorithm stages in detail.

3.1 Importance Level Candidates

In this section, we describe how the agents estimate the importance level candidates \( T_i \). Using a set of importance level candidates allows the agents to reduce the problem of selecting an importance level from a continuous range to choosing from a set of consecutive values in the range \([0, 1, 1] \). Since there is only a small number of sensor agents, we focus on agents that update their beliefs using only the opinions of neighbors without sensors. Pryymak et al. described the sampling dynamics of the agents’ beliefs, where an agent \( i \) can change its opinion from black to white after receiving more white opinions [2]. Starting from a prior belief of \( P_i^0 \) (black), an agent may update its opinion to white, due to an increase in belief after receiving a series of white opinions. The most important aspect of these dynamics is the update step when an agent has just changed its opinion, because this is the only time when it sends a new opinion to its neighbors. Consequently, we focus on how many times an agent must update its belief before it changes its opinion. According to the opinion update rule given in Section 2.1, we consider the case where the agent’s belief matches one of the confidence bounds \( P_i^l \in [\sigma, 1 - \sigma] \). Given that the maximum number of opinions that the agent can receive is limited by its number of neighbors \( |D_i| \), we can pare down the potential candidate importance levels. The agent should find the importance levels where its belief coincides with one of the confidence bounds \( P_i^l \in [\sigma, 1 - \sigma] \) in \( l \in 1, \ldots, |D_i| \) updates (see Eq. (3)). After solving this problem, it can obtain the set of importance level candidates that lead to forming an opinion after receiving \( 1, \ldots, |D_i| \) opinions, as follows:

\[
T_i = \{t_0 : P_i(t_0) = \sigma, l \in 1, \ldots, |D_i|\}
\]

\[
\cup \{t_0 : P_i(t_0) = 1 - \sigma, l \in 1, \ldots, |D_i|\}.
\]

(7)

The set of importance level candidates is thus limited to twice the number of neighbors, \( |T_i| = 2|D_i| \). This is the necessary and sufficient set of importance levels for which the agent forms an opinion after different number update steps, and it needs to be initialized only once. The agent then only has to estimate the optimal importance level from its set of importance level candidates.

3.2 Estimating Agent Awareness Rates

In this section, we describe the criteria used to select an importance level candidate. As mentioned above, the AAT algorithm is based on the observation that, the community accuracy
R improves when the opinion-sharing dynamics transitions between stable and unstable phases. In order to estimate optimal parameters, each agent has to determine the lowest importance level that allows it to form an opinion. In this opinion-sharing model, the following two conditions help to maximize the accuracy $R$ [4].

- Each agent has to form an opinion, so it should aim for a high awareness rate $h_{i}$, because agents without correct opinions decrease the community accuracy.
- Each agent has to form its opinion as late as possible with only local information after it has gathered as many opinions from its neighbors as possible.

To satisfy these requirements, the agent must select the importance level $t_{i} \in T_{i}$ from the candidates that allows it to form an opinion ($h_{i} = 1$). However, since the sensors' values are influenced by random noise, the opinion-sharing dynamics, such as phase transitions, are stochastic. The agents cannot form opinions until opinions are shared on a large scale, due to their awareness rates. The agents should thus select the lowest importance level $t_{i}$ from the candidates $T_{i}$ such that the awareness rate approaches the target $h_{trg}$, which is slightly lower than maximum, $h_{i} = 1$. Each agent solves the following optimization problem:

$$t_{i} = \arg \min_{t_{i} \in T_{i}} [h_{i}(t_{i}) - h_{trg}].$$

In this problem, $h_{i}(t_{i})$ is the awareness rate corresponding to the importance level $t_{i}$ that the agent selects. The optimal parameter value for achieving versatile network dynamics is $h_{trg} = 0.9$ [2].

### 3.3 Importance Level Selection Strategy

Each agent affects the dynamics and awareness rates of all the other agents, due to the dependence of the agents’ opinions on those of their neighbors. If an agent selects its optimal importance level greedily, based on the definition of its optimization problem (see Eq. (8)), it may significantly affect the local community dynamics. Instead, the agent should select a strategy that does not dramatically change the dynamics, so that it can estimate community awareness rates accurately and solve the problem more quickly. To select such a strategy, the agent should focus on the following inference. The agents’ awareness rates increase monotonically with the importance level. The agents should thus select the lowest importance level greedily, based on the definition of its importance level.

With the hill-climbing strategy, the agent should use the hill-climbing strategy rather than the greedy strategy allows the agents to achieve higher accuracy [2].

### 4. AAT with a Weighted Agent Network

This paper proposes an agent model for dynamic environments and a weighted AAT-based opinion-sharing method called conformity-AAT (C-AAT).

#### 4.1 Agent Model and Bidirectional Weighted Network

Figure 2 shows the agent architecture, which consists of a memory and a processor. The memory stores the agent’s opinion, belief, importance level, and weights. Unlike the standard AAT approach, the proposed agents have importance levels and weights, and they share their opinions using the three-step process shown in Fig. 2 (i.e., the proposed method). All agents have weights for all neighbors. This type of network, where the link between two agents has different weights for each agent’s opinion of the other is called a “bidirectional weighted network.” Concretely, the agents have weights $W$ for all their neighbors in the network:

$$w_{i}[j] \in W \quad (0 \leq w \leq 1.0),$$

where $j$ is the neighboring agent $D_{i} = \{ j : \exists (i, j) \in E \}$ (i.e., $j$, and $E$ indicate the two agents and edge set, respectively). The number of weights is the same as the number of neighboring agents (i.e., $|W_{i}| = |D_{i}|$). Figure 3 shows a bidirectional weighted network with four agents. In this example, agent A has weights $w_{A}$, $w_{C}$, and $w_{D}$, and their values are represented by the line widths; i.e., $w_{B} > w_{C} > w_{D}$, so agent A is most influenced by agent B’s opinion. Figure 4 shows the weights for two linked agents, A and D. Since each agent has weights for all neighboring agents, the weight of the link between agents A and D is different from that of the one between agents D and A. In this example, agent D is more influenced by A than A is by D.

#### 4.2 Conformity-AAT (C-AAT)

The C-AAT algorithm executes the following processes based on the importance levels determined by AAT.

- Updating the weighted importance levels. This is done when an agent receives an opinion from a neighboring agent (see Section 4.2.1).
- Estimating the weights for neighboring agents. This is
4.2.1 Updating the weighted importance levels
The weighted network represents the agent relationships. Each agent updates its importance level based on its neighbor weights using the following equation and uses this to form an opinion by C-AAT:

\[
t_k^i = \frac{w_{tk}^{k-1}}{(1 - w)(1 - r_k^i)} + w_k^{k-1}.
\]  
(10)

Here, \( r_k^i \) is the weighted importance level for agent \( i \) at the current step \( k \), and \( w \) is the weight. Although \( r_k^i \) can take values between 0 and 1.0 (0 \( \leq r_k^i \leq 1.0 \)), \( r_k^i \) cannot be less than 0.5 for opinion-sharing problems. If this happens, it is set to 0.5. In addition, the weight \( w \) has the following effects.

- If \( w = 0 \), the agent’s importance level is unaffected by the neighboring agents’ opinions (i.e., \( r_k^i = 0.5 \)).
- If \( w = 0.5 \), the agent’s importance level remains the same (i.e., \( r_k^i = r_k^{i-1} \)).
- If \( w = 1.0 \), the agent is strongly influenced by the neighboring agents’ opinions (i.e., \( r_k^i = 1.0 \)).

Figure 5 shows how each agent updates its weighted importance level. Here, the gauges are importance level meters, indicating the change caused by one neighbor’s opinion, and one of the radial lines on the gauge becomes the current opinion. For example, the line will swing right (left) by one unit if the agent receives the opinion white (black). If the line swings to the right-hand (left-hand) end, the agent will form the opinion white (black). If the meter’s scale is finer, the agent will be less influenced by its neighbor. Agent A updates its weighted importance level based on the current importance level and all the neighbor weights. Remember that the importance level represents how much the agent needs the opinions of its neighboring agents to form its own opinion, while the weights indicate how much the agent is influenced by the opinions received from each neighboring agent. Agent with weighted importance levels can have different importance levels for each neighboring agent, while agents with only one importance level cannot treat neighboring agents differently.

4.2.2 Updating the weights
After an agent receives an opinion from a neighbor, it updates the weight \( w_k^i[j] \) for the neighbor as follows:

\[
w_k^i[j] = \frac{aw_k^{i-1}[j]}{(1 - a)(1 - w_k^{i-1}[j]) + aw_k^{i-1}[j]}.
\]  
(11)

Here, \( j \) is the neighboring agent, while \( a \) is called the influence rate and is defined as

\[
\alpha = \begin{cases} 
0.01 & \text{if } \alpha_i = \alpha_j, \\
0.01 & \text{if } \alpha_i \neq \alpha_j, \\
0.5 & \text{if } \alpha = 0.5.
\end{cases}
\]  
(12)

The influence rate is initialized as \( \alpha = 0.5 \). The agent then updates \( \alpha \) as follows.

1. If the agent’s opinion was already the same as the received opinion, it adds 0.01 to \( \alpha \) for that neighbor.
2. If the agent’s opinion is different from the received opinion, it subtracts 0.01 from \( \alpha \) for that neighbor.

Through this process, the agent confirms neighbors that share its opinion. Figure 6 shows an example of when an agent confirms its neighbors. In this example agent A has the opinions [true, false, true], while the neighboring agents B, C, and D have the opinions [true, true, true], [false, false, true], and [false, false, false], respectively. In this case, the agent confirms agents B and C, which have similar opinions more than agent D.

5. Experiment
5.1 Experimental Setup
To investigate the effectiveness of our C-AAT algorithm, we now compare C-AAT with AAT for the opinion-sharing problem. The experimental setup was as follows.

- We adopted a small world network topology for the community, since we wanted to simulate a case that was similar to the real world.
- Only five sensor agents could observe the true state \( \beta \) and the information they received was noisy (\( r = 0.55 \)).

During the experiments, the network topology and the probability \( r \) of the sensor agents acquiring correct information was changed to create 10 seeds and we measured the average success rate. We used the following three agent networks.

- **Experiment A**
  In Experiment A, all sensor agents had the same probability of acquiring correct information (\( r = 0.55 \)).

- **Experiment B**
  In Experiment B, the probability of the sensor agents acquiring correct information changed over time. Concretely, testing began with Experiment A and changed to another experiment after 150 rounds. Two of the sensor agents in this experiment could acquire correct information with probabilities of 0.1 and 0.9 (Case 1), or 0.3 and 0.7 (Case 2).
• Experiment C
In Experiment C, the links between agents could be cut. Concretely, randomly-selected links were cut after 100 rounds.

5.2 Parameters Used
Table 1 shows the parameters used. We employed small world networks (line 1) composed of randomized connections (line 2) with $p_{revise} = 0.12$. There were 100 agents in the network (line 3) and the average path-length ($L$) and clustering coefficient ($C$) values were 3.995758 and 0.455693, respectively (lines 4 and 5). Each agent was linked to eight neighboring agents (line 6), and there were five sensor agents (line 7).

5.3 Results
5.3.1 Experiment A
Figure 7 presents the average success rate of both methods. The vertical axis gives the success rate, while the horizontal axis gives the seed. This shows that AAT’s success rate was only 34%.

Figure 8 presents the success rates for each of the 10 seeds. The vertical axis again gives the success rate, while the horizontal axis gives the seed. This shows that AAT’s success rate was around 63.2%, while that of C-AAT was around 89.0%. It is worth noting that, in Experiment A, AAT showed better performance than in Experiment A for both cases. In small world networks, AAT agents are easily influenced by noise and might share incorrect opinions. The accuracy of the sensor agents in Experiment A was 0.55, while that in Experiment B was 0.9 or 0.7 (i.e., much less noisy). Since the agents shared the opinions from the more accurate sensor agents, AAT’s result were better for Experiment B.

Figures 9 and 10 present the average success rates for C-AAT and AAT in Cases 1 and 2, respectively, with the axes as before. Figure 9 shows that AAT’s success rate for Case 1 was around 88.5%, while that of C-AAT was around 98.6%. Likewise, Figure 10 shows that, AAT’s success rate for Case 2 was around 72.8%, while that of C-AAT was around 89.0%.

5.3.2 Experiment B
Figures 11 and 12 present the success rates for each of the 10 seeds in Cases 1 and 2, respectively, with axes as before. Figures 11 shows that AAT’s success rate was around 100% for all seeds except 7 and 9, while that of C-AAT was always around 100%. Figure 12 shows that AAT’s success rate was high for all seeds except 2, 7, and 8, while C-AAT’s success rate was around 100% for all seeds except 7.

5.3.3 Experiment C
Figure 13 presents the average success rates for C-AAT and AAT, with the axes as before. This shows that AAT’s success rates was around 63.2%, while that of C-AAT was around 70.9%.

Figure 14 presents the success rates for each of the 10 seeds, with the axes as before. This shows that AAT’s success rate was only high for seeds 1, 3, and 5, while C-AAT’s success rate was always around 100%.

5.3.2 Experiment B
Figures 9 and 10 present the average success rates for C-AAT and AAT in Cases 1 and 2, respectively, with the axes as before. Figure 9 shows that AAT’s success rate for Case 1 was around 88.5%, while that of C-AAT was around 98.6%. Likewise, Figure 10 shows that, AAT’s success rate for Case 2 was around 72.8%, while that of C-AAT was around 89.0%.

It is worth noting that, in Experiment B, AAT showed better performance than in Experiment A for both cases. In small world networks, AAT agents are easily influenced by noise and might share incorrect opinions. The accuracy of the sensor agents in Experiment A was 0.55, while that in Experiment B was 0.9 or 0.7 (i.e., much less noisy). Since the agents shared the opinions from the more accurate sensor agents, AAT’s result were better for Experiment B.

Figures 11 and 12 present the success rates for each of the 10 seeds in Cases 1 and 2, respectively, with axes as before. Figure 11 shows that AAT’s success rate was around 100% for all seeds except 7 and 9, while that of C-AAT was always around 100%. Figure 12 shows that AAT’s success rate was high for all seeds except 2, 7, and 8, while C-AAT’s success rate was around 100% for all seeds except 7.

5.3.3 Experiment C
Figure 13 presents the average success rates for C-AAT and AAT, with the axes as before. This shows that AAT’s success rates was around 63.2%, while that of C-AAT was around 70.9%.

Figure 14 presents the success rates for each of the 10 seeds, with the axes as before. This shows that AAT’s success rate was only high for seeds 1, 3, and 5, while C-AAT’s success rate was always around 100%.

5.3.2 Experiment B
Figures 9 and 10 present the average success rates for C-AAT and AAT in Cases 1 and 2, respectively, with the axes as before. Figure 9 shows that AAT’s success rate for Case 1 was around 88.5%, while that of C-AAT was around 98.6%. Likewise, Figure 10 shows that, AAT’s success rate for Case 2 was around 72.8%, while that of C-AAT was around 89.0%.

It is worth noting that, in Experiment B, AAT showed better performance than in Experiment A for both cases. In small world networks, AAT agents are easily influenced by noise and might share incorrect opinions. The accuracy of the sensor agents in Experiment A was 0.55, while that in Experiment B was 0.9 or 0.7 (i.e., much less noisy). Since the agents shared the opinions from the more accurate sensor agents, AAT’s result were better for Experiment B.

Figures 11 and 12 present the success rates for each of the 10 seeds in Cases 1 and 2, respectively, with axes as before. Figure 11 shows that AAT’s success rate was around 100% for all seeds except 7 and 9, while that of C-AAT was always around 100%. Figure 12 shows that AAT’s success rate was high for all seeds except 2, 7, and 8, while C-AAT’s success rate was around 100% for all seeds except 7.

5.3.3 Experiment C
Figure 13 presents the average success rates for C-AAT and AAT, with the axes as before. This shows that AAT’s success rates was around 63.2%, while that of C-AAT was around 70.9%.

Figure 14 presents the success rates for each of the 10 seeds, with the axes as before. This shows that AAT’s success rate was only high for seeds 1, 3, and 5, while that of C-AAT was high for all seeds except 0, 2, and 6.
6. Discussion

6.1 Experiment A

Figures 15 and 16 show the distributions of agent opinions for AAT and C-AAT, respectively. The vertical axis gives the awareness rate of each opinion, while the horizontal axis gives the time step. Three agent states are shown: (1) agents who have the correct opinion (white); (2) agents who have the incorrect opinion (gray); and (3) agents who do not have an opinion yet (black). In Fig. 16, a phase transition occurs at 226 steps. In the opinion-sharing model, the agents repeat the following two steps: (1) agent forming an opinion, the agents disseminate it, and (2) agents receive others’ opinions and use them to form their own opinions. In addition, since each agent was linked to an average of eight neighbors, the number of agents that receive other opinions grew exponentially. These figures show that, almost all the AAT agents shared an incorrect opinion early on, while almost all the C-AAT agents shared correct opinions because they were less influenced by the sensor agents. Table 2 indicates the weights agent 23 gave to its neighboring agents. Here, AgentID and ID identify the agent and eight neighboring agents, respectively, and Importance Level and Awareness Rate indicate the parameters used for this round. In addition, the Weight column shows the weights given to each neighboring agent, the New_Importance Level column gives the new belief values, and Sensor indicates whether the neighboring agent has a sensor (i.e., is a sensor agent). (In this case, only agent 19 is a sensor agent.) In this table, the neighbor agent weights are around 0.6, but the weight for the sensor agent is 0.56. This
difference is important, because this causes a difference of 0.1 in the importance levels. In contrast, AAT agents accept other agents’ opinions with 100% probability because they do not use weights. In addition, the target awareness rate $h_{targ}$ was 0.9, suggesting that the AAT agents formed opinions easily (i.e., the importance level readily became low). In small world networks, AAT agents are easily influenced by noise (i.e., noisy information from the environment and incorrect opinions from agents), and might share incorrect opinions. Since the sensor agents’ opinions were noisy, if they passed on incorrect opinions early on, then the agents might start sharing them. The table suggests that the C-AAT agent paid less attention to the sensor agent’s opinion than to those of the other neighboring agents, indicating that C-AAT, which considers the agents’ opinions carefully, can give better performance than AAT.

6.2 Experiment B

Figures 17 and 18 give distributions of agent opinions for AAT and C-AAT, respectively, for seed 7 in Case 1, with the axes and states shown as before. These show that almost all the AAT agents shared incorrect opinions early on, while almost all the C-AAT agents shared correct opinions, for the same reasons as in Experiment A.

Figures 19 and 20 give distributions of agent opinions for AAT and C-AAT, respectively, for seed 7 in Case 2. In contrast with Case 1, 60% of the AAT agents shared incorrect opinions and the rest were undetermined, while all the C-AAT agents shared incorrect opinions. Table 3 indicates the weights agent 71 gave to its neighboring agents for seed 7, expressed as in Table 2. This agent was linked to two sensor agents, and formed an opinion if it received opinions from both of them. Without input from the sensor agents, it conveyed incorrect information. Here, C-AAT made the agent decrease both sensor agent weights. However, the weights for both sensor agents should be small in order to ensure correct opinions are shared. To tackle this issue, other agents should decrease their weights for agents linked to sensor agents. Note that the agents did not know how accurate the sensors were, and simply adjusted their weights automatically. In this case, the agents shared incorrect opinions, suggesting that they should not have formed their own opinions before the sensor had sent them the correct information with constant probability. To tackle this problem, the agents would have to form their opinions more carefully, and hence would have to decrease the weights given to other agents.

6.3 Experiment C

Figures 21 and 22 give distributions of agent opinions for AAT and C-AAT, respectively, with the axes and states shown as before. These show that the C-AAT agents were less influenced by the sensor agents’ opinions. On the other hand, C-AAT could not always make the agents share correct opinions, even when AAT could (e.g., for seeds 0 and 2).

Figure 23 shows the C-AAT awareness rates for seed 0,
showing that its weights were not effective in this case. Since, in Experiment C, the links were cut randomly for every round, the agents were less able to receive opinions than in the other experiments. In this case, AAT set the importance levels so as to acquire a large number of opinions.

Tables 4 and 5 focus on the agent that first formed an incorrect opinion when the link to the sensor agent was cut in Experiment C for seed 0. They indicate this agent’s weights for Experiments A and C, respectively. These results show that the importance level was higher for Experiment C (0.66) than for Experiment A (0.60). In this case, the importance level was so large that the weights were ineffective. For seeds 0 and 9, the weighted importance level promoted the sharing of incorrect opinions. To tackle this issue, C-AAT should limit the sensor agent weights.

### 7. Conclusion

This paper focused on the opinion-sharing problem of how to share correct information in a society where there is incorrect information. For this, the AAT method has been proposed, based on a multi-agent system. This has demonstrated good performance for a variety of networks, provided that all the agents’ opinions have the same weight (i.e., they have the same relationships with each other) and the agent links cannot change. However, in real life, people’s relationships, e.g., with friends, colleagues, and family, are not all the same, and it is necessary to cope with people sharing incorrect information. To tackle this issue, we have proposed C-AAT. This adds weights that indicate how much the agents are influenced by each of their neighbors, and introduces weighted importance levels to help them share correct information. To investigate C-AAT’s effectiveness, we carried out three sets of experiments: (a) a static network with sensor agents that are almost equally likely to convey incorrect information; (b) a static network with sensor agents whose probability of conveying incorrect information changed over time; and (c) a dynamic network where the agent links were randomly cut over time. These experiments led us to the following conclusions.

1. C-AAT can make the agents’ opinions robust against incorrect sensor agent opinions by decreasing the weights.
2. C-AAT can decrease the weights of agents conveying incorrect opinions with varying probabilities to prevent incorrect opinions being shared.
3. C-AAT can help agents share correct opinions by increasing the weights of their neighbors, even if the agents receive fewer opinions due to links being cut.

Our results show that C-AAT, our weighted opinion-sharing method, effectively enables agents to share correct opinions in networks which simulate the real world.

To solve more difficult real-world problems, we should establish a method for sharing correct opinions in networks with different issues in future work. In particular, we should consider networks where the topology and number of agents can change.
References

[1] R.T. Glinton, P. Scerri, and K. Sycara: Towards the understanding of information dynamics in large scale networked systems, 2009 12th International Conference on Information Fusion, pp. 794–801, July 2009.

[2] O. Pryymak, A. Rogers, and N.R. Jennings: Efficient opinion sharing in large decentralised teams, Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, Vol. 1, AAMAS ’12, pp. 543–550, Richland, SC, 2012.

[3] R. Glinton, P. Scerri, and K. Sycara: An investigation of the vulnerabilities of scale invariant dynamics in large teams, Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, Vol. 2, AAMAS ’11, pp. 677–684, Richland, SC, 2011.

[4] R. Saito, N. Tatebe, R. Takano, and K. Takadama: Network construction for correct opinion sharing by selecting a curator agent, Proceedings of the SICE Annual Conference 2015, pp. 363–368, July 2015.

Fumito UWANO
He received the Master’s degree from the University of Electro-Communications, Japan, in 2017. He is currently a doctoral student at the University of Electro-Communications, and a research fellow of Japan Society for the Promotion of Science. His research interests include multi-agent systems and reinforcement learning.

Rei SAITO
He received the Master’s degree from the University of Electro-Communications, Japan, in 2017. His research interests include multi-agent systems.

Keiki TAKADAMA (Member)
He received the Doctor of Engineering degree from the University of Tokyo, Japan, in 1998. He worked as a visiting researcher at Advanced Telecommunications Research Institute International (ATR) between 1998 and 2002 and then worked at the Tokyo Institute of Technology as a lecturer between 2002 and 2006. He became an associate professor at the University of Electro-Communications in 2006 and has been a professor there since 2011. His research interests include multi-agent systems, reinforcement learning, and evolutionary computing.