Multi-Agent Cooperation Based on Reinforcement Learning with Internal Reward in Maze Problem

Fumito Uwano *, Naoki Tatebe *, Yusuke Taima *, Masaya Nakata *, Tim Kovacs **, and Keiki Takadama *

Abstract: This paper introduces a reinforcement learning technique with an internal reward for a multi-agent cooperation task. The proposed methods is an extension of Q-learning which changes the ordinary (external) reward to the internal reward for agent-cooperation. Specifically, we propose here two Q-learning methods, both of which employ the internal reward for the less or no communication. To guarantee the effectiveness of the proposed methods, we theoretically derived the mechanisms that solve the following questions: (1) how the internal rewards should be set to guarantee the cooperation among the agents under the condition of less and no communication; and (2) how the values of the cooperative behaviors types (i.e., the varieties of the cooperative behaviors of the agents) should be updated under the condition of no communication. The intensive simulations on the maze problem for the agent-cooperation task have been revealed that our two proposed methods successfully enable the agents to acquire their cooperative behaviors even in less or no communication, while the conventional method (Q-learning) always fails to acquire such behaviors.

Key Words: theoretical method, multi-agent system, reinforcement learning, internal reward, cooperation.

1. Introduction

Multi-agent reinforcement learning (MARL) is a useful approach to tackle multi-agent cooperation tasks, such as multi-robot cooperation and traffic signal control [1]–[6]. However, MARL has a difficulty of deriving good performance without communication among the agents because the agents do not know how they cooperate with each other [7],[8] due to the fact that the behavior of agents affect the behavior of other agents and vice versa. To address this issue, most of the conventional methods such as Tan’s research [9] promote agents to cooperate with each other by using other agents’ information through their communication. Such information is very useful for the cooperation among the agents, but it goes without saying that the agents cannot guarantee to acquire all information required to cooperate with each other through communication. From this fact, it is important to explore methods based on less or no communication for the cooperation among the agents.

For this issue, the conventional methods based on less or no communication can be classified four classes from the following two viewpoints as shown in Table 1: (1) the agents learn their behaviors empirically or theoretically; (2) information of other agents can be (partially) shared or not. As the empirical approach employing information of other agents, Ono et al. proposed the modular Q-learning that combined the modular architecture with Q-learning [13] and showed that the modular Q-learning agents succeeded to synthesize decision policies for the cooperation among the agents by using information of other agents [10]. However, the cooperative behavior cannot be guaranteed because the decision policies are determined heuristically. As the empirically approach without sharing information of other agents, Lima et al. proposed the swarm reinforcement learning for the collective agents [3]. In this method, the agents do not need the information of other agents, but the optimal behavior of the agents cannot be also guaranteed to be acquired. As the theoretical approach employing information of other agents, on the other hand, Elidrissi et al. proposed the fast adaptive learning in the stochastic game (FAL-SG) [5], which enabled the agents to select the optimal action by observing their actions each other. However, this method requires the complete information of all agents for the cooperation among them. Compared with these methods, Table 1 suggests that there is no method based on the theoretical approach without sharing information of other agents, even though such methods are really needed. To tackle this issue, this paper proposes the “internal reward” which is changed from the ordinary (external) reward. Specifically, the agents in our approach theoretically learn their behaviors according to the internal reward instead of the ordinary (external) reward through the less or no communication among the agents. For this issue, we employ Q-learning agents in this paper because Q-learning is well studied and analyzed mathematically (i.e., the convergence of Q-value is proved in the single agent environment [13]).

This paper is organized as follows. Section 2 explains Q-learning as reinforcement learning and proposes the internal reward for the cooperation among agents. Section 3 describes the multi-agent cooperation task addressed in this paper. Our method is proposed in Section 4. Section 5 conducts experiments and analyzes the obtained results. Finally, our conclusion is given in Section 6.

Table 1 The classified previous methods.

|                  | None              | Information sharing |
|------------------|-------------------|---------------------|
| Empirical        | Lima [3]          | Ono [10], Ichikawa [4] |
| Theoretical      | Elidrissi [5], Littman [11], Hu [12] |

* The University of Electro-Communications, Tokyo, Japan
** University of Bristol, UK
E-mail: uwano@cas.hc.uec.ac.jp
(Received April 14, 2016)
(Revised October 31, 2016)
2. Reinforcement Learning and Internal Reward

2.1 Q-Learning

Reinforcement Learning (RL) [14] is a try-and-error method which aims at maximizing an acquired reward per a unit time. As its general framework, an RL agent interacts with an environment; observes a state from an environment, executes an action, and then receives a reward from it.

Among many RL methods, Q-learning [13] is a very popular RL method which is originally designed for a single-agent task. In Q-learning, the agent estimates state-action values (called Q-value $Q(s, a)$) for the possible state-action pairs in the environment, i.e., the agent estimates an expected reward that will be received when its action $a$ is executed in its state $s$. The aim of the agent is to learn a policy $\pi(s, a)$ to decide which action should be executed to maximize the received reward. Technically, the policy $\pi$ can be the probability in selecting the action $a$ in the state $s$ and is calculated by $Q(s, a), a \in A$ where $A$ is a set of actions that includes the possible actions. To maximize the received reward, $Q(s, a)$ is updated as follows:

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a' \in A} Q(s', a') - Q(s, a)],$$

(1)

Where, $s'$ is the next-state, $a'$ is the next-action executed in the state $s'$, $r$ is the reward received from the environment, $\max_{a' \in A} Q(s', a')$ is the largest Q-value when executing the action $a' \in A$ in the state $s'$. In addition to these variables, $\alpha$ is the learning rate, while $\gamma$ is the discount factor. Precisely, $\alpha$ is the real number from 0 to 1 which indicates the learning speed, while $\gamma$ is the real number from 0 to 1 which indicates how much the future rewards should be considered as important.

2.2 Internal Reward

Since behaviors of the RL agent are directly affected by the reward, the reward design is a critical issue. In particular, it is generally difficult to derive the appropriate reward design for multi-agent cooperation. For this issue, Ichikawa et al. proposed the method based on the internal reward which enabled the agents to avoid their conflicts in a dilemma problem [15]. What should be noted here is that the internal reward is designed to be changed from the ordinary (external) reward for the cooperation among the agents. The above research suggested that the multi-agent cooperation can be achieved when designing the appropriate internal reward. More importantly, there is a possibility that the agents do not need to share the information among the agents or just need to share a few of such information if the internal rewards are designed by considering the cooperation among the agents. From this potential, this paper focuses on the internal reward and designs it for the multi-agent cooperation with less or no communication among the agents.

3. Multi-Agent Cooperation Task

3.1 Cooperation in Maze Problem

In the grid maze problem as a multi-agent cooperation task, let us focus on Fig. 1, which has the two start positions as the initial states of the agents (A, B) and the two goal positions (S, L) as the target states for the agents. A difficulty of multi-agent cooperation task in the maze problem is to find cooperative behaviors because the agents selfishly learn their minimum steps.

For instance, the two agents A and B tend to reach the same goal S because their minimum steps to the goal can be achieved by reaching it. Since the agent B can potentially reach the goal S faster than the other agent A due to the short distance from the start to the goal, the agent A has to reach the goal L. This is the best solution for the agent B while the worst solution for the agent A since he should take the longest step to reach goal. In this case, the agents A and B are respectively required to reach the goals S and L as the cooperative behaviors. This difficulty is often called as a dilemma problem.

3.2 Mathematical Definition of Dilemma Problem

To theoretically analyze the dilemma problem in the maze problem, this section focuses on the two agents (A and B) and the two goals (X and Y) by assuming that all minimum steps from start to goal are different from each other. Table 2 shows the minimum steps from starts of all agents to all goals. In this table, $t_{LX}$, $t_{LY}$, $t_{BX}$ and $t_{BY}$ indicate the minimum steps. Precisely, the minimum step of the agent A to the goal X indicates $t_{LX}$: When the goal X (or Y) is near from the agent A while the goal Y (or X) is near from the agent B, this situation indicates a non-dilemma situation. Dilemma situations occur when the goal X (or Y) is near from “both” agents (i.e., another goal Y (or X) is far from them), and these situations are represented by $t_{LX} < t_{AX} \land t_{BX} < t_{BY}$ (or $t_{AX} > t_{AY} \land t_{BX} > t_{BY}$). In addition, when both goals are nearer from the agent B (or A) than the agent A (or B), it is represented $t_{AX} > t_{AX} \land t_{AY} > t_{BY}$ (or $t_{AX} < t_{AX} \land t_{AY} < t_{BY}$). This suggests that the situation satisfies with $t_{AX} < t_{AY} \land t_{BX} < t_{BY} \land t_{AX} > t_{BY} \land t_{AX} > t_{BY}$ indicates a dilemma situation. The above explanation is summarized as Table 3 (for example, the above dilemma situation is indicated 4th line).
4. Proposed Method

Toward the cooperation among the agents, for the dilemma maze problem in Fig. 1, we propose the learning mechanism based on Q-learning as shown in Fig. 2.

4.1 Learning Procedure

In the proposed method, the agent observes its state and selects its action (processes 1 and 2), which results in the state transition (process 3). These processes are the same as standard procedure of RL, and the cycle from processes 1 to 3 is generally called as “step”. After process 3, the proposed method check whether the agent reaches the goal (process 4). If the agent reaches the goal, it receives a reward (process 5); otherwise, updates the Q-value (process 6) and jumps to the process 11. In the case of receiving a reward in the process 5, the agent updates the minimum steps if the number of the steps from the start to the goal is shorter than the current minimum steps that the agent acquired before (process 6). After that, the agent selects the appropriate goal according to the minimum steps (process 7) (described in Section 4.2). The agent estimates the internal reward according to the minimum steps (process 8) (described in Section 4.3), and updates Q-value according to the internal reward (process 9). The system selects whether the number of the step is larger than the threshold MaxStep; If true, the system goes to process 12; otherwise, the system returns to process 1 (process 11). Finally, the system counts the learning iteration and the whole process ends when this iteration count is larger than the threshold MaxIteration; otherwise, the system returns to process 1 (process 12).

4.2 Goal Selection

In order for the agents to select their appropriate goals towards the cooperation among them, we propose two versions for this, and each version is mentioned as follows.

4.2.1 Goal selection based on goal value (Version 1)

In the version 1, the agents store the information on the minimum steps from their starts to all goals and calculate their goal values which are real numbers and indicate priority for selecting one goal from all goals. Table 4 show the information of the agent A. In this table, \( t_{AS} \) and \( t_{AL} \) respectively indicate the minimum step from the start A to the goals S and L, while \( bid_{AS} \) and \( bid_{AL} \) respectively indicate the goal values for the goals S and L for the agent A. The agent B has the same kind of information \((t_{BS}, t_{BL}, bid_{BS} \text{ and } bid_{BL})\) in Table 4. The following procedure indicates how the agents select their own goals according to their own goal values and update the goal values from the minimum step.

1. Selecting Goal

The agent selects its own goal which has the largest goal value. For example, the agent A selects the goal L if \( bid_{AS} < bid_{AL} \). After selecting the goal, the agent estimates the internal reward to reach the selected goal. Note that, the agent generally selects the goal with the largest goal value but sometime selects it randomly with a small probability to update the goal values evenly.

2. Updating Goal Value

The agent updates its own goal values by the following Eqs. (2) and (3), where \( n \) indicates the current learning iteration. Note that the following equations are employed for the agent A:

\[
bid_{AS} = \frac{n - 1}{n} bid_{AS} + \frac{t_{AS}}{n} \quad (t_{AS} < t_{BS}),
\]

\[
bid_{AS} = \frac{n - 1}{n} bid_{AS} + \frac{0}{n} \quad (t_{AS} > t_{BS}).
\]

If the agents A and B select the same goal S, the agent A updates the goal value for the goal S \((bid_{AS})\) according to Eq. (2) when reaching the goal S faster than other agents; otherwise the agent updates the goal value for the goal S \((bid_{AS})\) according to Eq. (3) because the agent A cannot reach the goal S due to the fact that the agent B reaches the goal S faster than the agent A. The agent B also updates \( bid_{BS} \) in the same manner. Here, we show how the above goal selection mechanism can solve the dilemma problem when the agents update their goal values in the infinite time. Equations (2) and (3) indicate the average of the minimum steps and the average of those added 0, respectively. In the infinite time, the following Eqs. (4) and (5) are derived from the above equations:

\[
\lim_{n\to\infty} bid_{AS} = t_{AS} \quad (t_{AS} < t_{BS}), \quad (4)
\]

\[
\lim_{n\to\infty} bid_{AS} = 0 \quad (t_{AS} > t_{BS}). \quad (5)
\]

In the dilemma problem shown in Fig. 3, the goal values converge as shown in Table 5. In detail, the goal value of the goal L for the agent A becomes 0 by Eq. (3). Although \( t_{AL} > t_{BL} \) is satisfied in this situation, the goal value of the goal L for the agent A becomes 0 because the agent

![Fig. 2 Procedure of a cooperative agent.](image)

![Fig. 3 Goal selection based on goal value.](image)

| Table 4 Information of the agent from start A. |
|-----------------------------------------------|
| Goal S | Goal L |
| Minimum step | \( t_{AS} \) | \( t_{AL} \) |
| Goal value | \( bid_{AS} \) | \( bid_{AL} \) |
Table 5 Goal value table for the two agents.

|         | Goal S | Goal L |
|---------|--------|--------|
| Agent A | $t_{AS}$ | 0      |
| Agent B | $t_{BS}$ | $t_{BL}$ |

Fig. 4 Goal selection based on minimum step.

Fig. 5 Internal reward design.

B only selects the goal L. After all, from the goal values in Table 5, the agent A selects the goal S and the agent B selects the goal L by following $t_{BS} > t_{AS}$. In addition, it is clear that this selection resolves the dilemma problem in Fig. 3.

4.2.2 Goal selection based on minimum steps (Version 2)

In the version 2, the agents store the information on the minimum steps from their own starts to all goals as the same as the version 1 and also store the information on the minimum steps of the other agents. To receive the information of the other agents, the agents send it to the others when reaching one goal by shorter steps than before. This indicates that both agents share the minimum steps of other agents, i.e., both agents have the information of the minimum steps ($t_{AS}$, $t_{AL}$, $t_{BS}$, and $t_{BL}$) as shown in Fig. 4. For example, when the agents A and B respectively reach the goals S and L in Fig. 4, the agent A replaces the current $t_{AS}$ with the new $t_{AS}$ if the new $t_{AS}$ is shorter than the current $t_{AS}$ and sends it to the agent B while the agent B updates $t_{BL}$ and sends it to the agent A in the same manner. After receiving such information, the agents can select the appropriate goals. In this figure, the agent A must reach the goal S and the agent B must reach the goal L because the number of the step towards goals is shortest.

4.3 Internal Reward Design

To reach the goal selected by the method described in Section 4.2.1, the internal reward is also employed in the version 2. Figure 5 shows how the internal reward is designed for the cooperation among the agents. In this figure, $t_{BS}$ and $t_{BL}$ respectively indicate the minimum step from start B to goals S and L, the star mark represents the turning position to select whether the agent B should reach the goal S or L, and the yellow directional arrows represent the actions towards goals S and L where the thickness of the arrows indicates the size of Q-values. Considering the cooperative situation where the agents A and B respectively reach the goals S and L, it is not necessary to set the internal reward of the agent A because it reaches the goal S normally. In contrast, however, the internal reward of the agent B should be set because it also reaches the goal S (not the goal L) normally. To solve this dilemma situation, the internal reward is designed for the agent B to reach the goal L.

In the turning position in Fig. 5, Q-value of the action to reach the goal S converges to $r$ while Q-value of the action to reach the goal L converges to $y^2$ in the standard RL. Considering the internal rewards $r_{AS}$ and $r_{LS}$ provided for the final action to reach the goals S and L, $r_{LS} > r_{AS}$ should be satisfied in the turning position for the cooperative situation. Towards such situation, $r_{LS}$ and $r_{BL}$ should be set as $r$ and $\frac{r}{\gamma} + \delta(>0)$, respectively.

4.3.1 Mathematical analysis

Here, we generalize the inequality ($ir_{LS} > ir_{AS}$) described in the previous section in all states in the maze problem in Fig. 5. To acquire cooperative behaviors, the internal rewards should be designed as the following equations, where $ir_{AS}$ and $ir_{AL}$ are respectively the internal rewards of the goals S and L for the agent A, while $ir_{BS}$ and $ir_{BL}$ are respectively the internal rewards of the goals S and L for the agent B. Considering that $t_{AS}$, $t_{AL}$, $t_{BS}$, and $t_{BL}$ indicate the minimum steps of the agents A and B toward the goals S and L, Q-values of the action toward the goal S at the starts A and B are described as follows:

$$y^{ir_{AS}}ir_{AS} \cdot y^{ir_{BS}}ir_{BS}.$$  (6)

Q-values of the action toward the goal L are described as follows:

$$y^{ir_{AL}}ir_{AL} \cdot y^{ir_{BS}}ir_{BS}.$$  (7)

Since the agent A must reach the goal S, the following inequality should be satisfied in any states:

$$ir_{AS} > y^{ir_{AL}}ir_{AL}.$$  (8)
Algorithm 2 (Proposed method 2)

Require: Agents take starting positions
\[ f_s = \text{MaxStep}(i, j = 0, \text{AgentNumber}, x = 0, \text{GoalNumber}) \]
\[ T_s = \text{MaxStep}(i = 0, \text{AgentNumber}, x = 0, \text{GoalNumber}) \]
\[ g_i \in G \]

1. for iteration = 0 to MaxIteration do
2. for All agents reach the goals or step = 0 to MaxStep do
3. Agents observe their states
4. Agents choose actions
5. The agents which don’t reach the goal update Q-value
6. if Agent i has reached goal x then
7. \[ T_i = \text{step} \]
8. end if
9. end for
10. for i = 0 to AgentNumber do
11. if \( T_i < f_s \) then
12. \[ f_i = T_i \]
13. Agent i send \( f_i \) to other agents
14. end if
15. end for
16. Agents estimate internal reward
17. Agents update Q-value by the internal reward
18. end for

In the same manner, the following inequality for the agent B should also be satisfied in any states:

\[ ir_{BL} > \frac{ir_{RS}}{\gamma^{min=ir_{AS}}} \]  \[ (9) \]

In Fig. 5, Eq. (9) show the generalization of the inequality, \( ir_{BL} > ir_{RS} \) (where \( ir_{BL} - ir_{BS} = 2 \)). As mentioned above, it is not necessary for the agent A to design the internal reward because \( ir_{AS} = \gamma^2ir_{AL} \) is satisfied while \( ir_{AS} = r \) and \( ir_{AL} = r \). In implementation, Eqs. (10) and (11) with \( \delta(>0) \) are employed instead of Eqs. (8) and (9):

\[ ir_{AS} = \gamma^{ir_{BL}-ir_{AL}} + \delta \]  \[ (10) \]
\[ ir_{BL} = \frac{ir_{RS}}{\gamma^{min=ir_{AS}}} + \delta \]  \[ (11) \]

4.4 Algorithms

Before explaining Algorithms 1 and 2 for the versions 1 and 2 in our methods, we start to explain the following variables in Algorithms 1 and 2, \( t_j \) indicates the minimum step of the agent \( j \) to the goal \( x \) which the agent \( i \) does not have \( t_j \), and the agent \( j \) does not have \( t_j \), for the agent i and j. \( t_{jx}, t_{jy} \) and \( t_{jx}, t_{jy} \) for the agent \( j \). The agent \( i \) updates \( t_{ix}, t_{iy} \) or \( t_{iy} \) by himself when his minimum step towards the goal \( x \) or \( y \) changes and only updates \( t_{jx}, t_{jy} \) when receiving it (\( t_{jx}, t_{jy} \)), from the agent \( j \) (the agent \( j \) updates \( t_{jx}, t_{jy}, t_{jx}, t_{jy} \), in the same manner of the agent \( i \). In the algorithm 1, on the other hand, the agents do not have the minimum steps of the other agents, i.e., the agent \( i \) does not have \( t_{ix}, t_{iy} \) and the agent \( j \) does not have \( t_{jx}, t_{jy} \).

As the remaining variables, \( g_i \) and \( v_g \), respectively indicate the identification number of the goal which the agent \( i \) selects and the goal value of \( g_i \), both of which are only employed for the algorithm 1. Finally, \( ir_i \) indicates the internal reward of the agent \( i \) as mentioned in Section 4.3. Since the objective of this research is to explore the cooperation method with less or no communication among the agents, the agents do not know all variables of others in the algorithm 1 while they only know \( t_{jx} \) or \( t_{jy} \) in Algorithm 2 as shown in Table 6, which indicates whether each variable is known or unknown to other agents. As the variables in Algorithms 1 and 2, \( iteration, MaxIteration, MaxStep, \) AgentNumber and GoalNumber indicate the current iteration, the maximum iterations, the maximum steps, the numbers of the agents and the goals, respectively.

4.4.1 Algorithm 1 (proposed method 1)

In the algorithm 1 as the proposed method 1 (version 1), all agents observe the states, select actions and update Q-values (lines 3, 4 and 5). If the agent \( i \) reaches the goal \( x \), the current step is temporarily stored as \( T_i \) (lines 6, 7 and 8). After all agents reach their goals (or the step exceeds MaxStep), the agent \( i \) updates \( T_i \) by \( T_i \) as the minimum step if \( T_i \) is smallest (lines from 10 to 14). The agent \( i \) selects the goal \( g_i \) to set the goal value (line 15), and \( v_g \) is updated according to \( g_i \) (lines from 16 to 22). The agent \( i \) estimates the internal reward by Eq. (10) to reach the appropriate goal for the cooperation (line 23). The Q-value of the action to reach the goal is updated with the internal reward by Eq. (12) (line 24):

\[ Q(s, a) \leftarrow Q(s, a) + \alpha [ir_{AS} + \gamma ir_{AL}] \]  \[ (12) \]
5. Experiment

5.1 Experimental Design & Setting

To investigate the effectiveness of our methods, we conduct the following 3 experiments in the maze problems as follows:

- Experiment 1: The effect of the proposed methods is investigated in a small size maze (3 × 8 grid).

- Experiment 2: The robustness of the proposed method is investigated in the different types of mazes which are created by changing the start and the goal positions in the 3 × 8 grid maze.

- Experiment 3: The scalability of the proposed methods is investigated by applying them into a large size maze (i.e., 6 × 16 as 2 times larger than the 3 × 8 maze).

Other conventional methods, e.g., Tan’s approach [9], are not compared with the proposed methods which are based on no or less communication among the agents. This is because such methods are not designed for the situation with no or less communication due to the fact that they require information acquired by communication from other agents. In detail, we investigate the two proposed methods as shown in Table 7. In this table, the column indicates the methods 1 (left side) or 2 (right side) while the line indicates whether the internal reward is employed or not (upper side), whether the goal is selected according to the goal value or the minimum steps towards the goal (middle side), and whether the communication is not employed or less employed (lower side).

As the evaluation criterion, this paper evaluates the success rate for the cooperation between two agents. The success rate is calculated by (the number of the success cooperation among the agents) / (the total number of experiments), where the success cooperation is done when both agents reach their own appropriate goals. The total number of experiments is determined by the number of trials (e.g., 30 trials with 30 different seeds in one maze in experiments 1 and 3, and 300 trials with 30 different seeds in 10 kinds of mazes in the experiment 2). In addition to this criterion, this paper also investigates the effect of random goal selection and the constant $\delta$, respectively. Concretely, we conduct the experiment by employing the four ratios for random goal selection (5%, 10%, 15%, 20%) and four constant $\delta$ $(0.1, 0.5, 1, 10)$ in Eqs. (10) and (11). Note that $\eta = 0$ cannot be set because the agents are required to estimate the goal values of all combinations of the agents and the goals for the appropriate goal selection.

The parameters for Q-learning as summarized in Table 8. In this table, the learning iterations and step counts are respectively limited to 30000 and 100 as the threshold in Experiments 1 and 2, while they are respectively limited to 400000 and 100 in Experiment 3. We initialize Q-values of all states as 0. The parameters $\alpha$ and $\gamma$ are set as 0.1 and 0.9. Note that the proposed methods employ the $\epsilon$-greedy selection in the learning phase, while they employ the greedy selection in the evaluation phase. Concretely, the agents select their actions according to the $\epsilon$-greedy selection method with $\epsilon = 0.7$, and evaluate the learning result according to the greedy selection method. We set $\epsilon = 0.7$ in learning phase because the agents have to explore...
5.2 Experimental Results

5.2.1 Experiment 1

Figures from 9(a) to 9(c) show the success rates of the cooperation between two agents in the conventional method and the two proposed methods. The vertical axis indicates the success rates, while the horizontal axis indicates the learning iteration. The four lines in Fig. 9(b) respectively indicate the results of $\eta = 5\%$, $10\%$, $15\%$ and $20\%$, while the four lines in Fig. 9(c) respectively indicate the results of $\delta = 0.1$, $0.5$, $1.0$ and $10$. As shown in Figs. from 9(a)-(c), the success rate of the conventional method (Q-learning) is quite low, while that of both proposed methods increases as the iterations increases and finally becomes 1. In the dilemma maze problem, either agent is required to give up to reach the nearest goal for the cooperation among the two agents, i.e., either agent is required to yield its goal to the other agent. Since the Q-learning agent is designed to maximize the reward per a unit step, it does not have a giving up or yielding mechanism for such a situation. This derives the result that the success rate of the Q-learning agent is mostly 0. From these results, the proposed methods enable the agents to cooperate with each other.

Figures 10 and 11 show the enlarged images of Figs. 9(b) and 9(c), respectively. In Fig. 10, the success rate of $\eta = 5\%$ is high until about 3500 iterations, and slowly converges at 1. The success rate of $\eta = 10\%$ is middle among the rate on the other $\eta$ and lastly converges at 1. The success rate of $\eta = 15\%$ is the highest until about 1200 iteration and converges at 1 fast. Since the parameter $\eta$ indicates the exploration capability of the agents to correctly estimate the goal value. From the results, $\eta = 15\%$ can derive a good balance between exploration and exploitation because the success rate is the highest and converges at 1 fast in comparison with the other $\eta$. This can be also supported by the results that (1) the success rates of $\eta = 5\%$ and $\eta = 10\%$ become high at the first several iterations, but slowly converges because of a lack of exploration; and (2) the success rate of $\eta = 20\%$ converges fast but is lower at the first several iterations because of a lack of exploitation. In Fig. 11, the success rate of $\delta = 0.1$ or $\delta = 0.5$ is low, that of $\delta = 1$ is middle, and that of $\delta = 10$ is the high. These suggest that the success rate with the large $\delta$ becomes high while converges 1 slowly because the large $\delta$ is hard for the agents to estimate the Q-values more accurately than the other $\delta$.  

5.2.2 Experiment 2

In the result of Experiment 2, the success rate of Q-learning is 0, while the success rates of the two proposed methods are 1, which are averaged from 30 different seeds in 10 mazes (i.e., 300 trials). This suggests that the two proposed methods are robust to the different start and goal positions in 10 kinds of the $3 \times 8$ grid mazes.

5.2.3 Experiment 3

Figure 12 shows the success rate of the cooperation among the two agents in the two proposed methods in the large size maze. In this figure, the vertical axis indicates the number of learning iteration. From this figure, the success rate of both proposed methods increases as the number of iterations increases and finally becomes 1. Figure 13 shows the goal values in all combinations of the agents and the goals. In this figure, the vertical axis indicates the goal values averaged from 30 different seeds at the
last iteration, while the horizontal axis indicates all combinations of the agents and the goals (e.g., “A to S” means that the goal value of the agent A to the goal S). The line on each bar in Fig. 13 indicates the standard deviation, called “error bar”, of the goal values of 30 different seeds.

From this figure, the goal values of “A to S” and “B to L” (i.e., the target of the agents for their cooperation) are respectively larger than those of “A to L” and “B to S” in all trials. From Fig. 13, the error bar of “A to S” and “B to L” is smaller than those of “A to L” and “B to S”, respectively. This suggests that the proposed methods enable the agents to cooperate with each other even in the large size maze.

5.3 Experimental Analysis

5.3.1 Goal selection

In the method 1, the agents can select their appropriate goals for the cooperation between the two agents by estimating the goal value which converges to the minimum steps towards the goal that the agent reaches. This suggests that it is important for the agents to acquire their own minimum steps accurately. Note that, however, the agents do not have to acquire all accurate minimum steps in order to select the goal for the cooperation among the agents. Considering the situation in Fig. 1, the following inequalities are satisfied when the agents A and B reach the goals S and L. Note that there are the same kinds of inequalities for the agent A but the inequalities are meaningless because if the agent B can choose the appropriate goal (i.e., the goal L in this situation), the agent A does not have to choose the goal. The agent A has no choice to select the goal L.

\[
bid_{BL} > bid_{BS}, \quad (13) \\
bid_{BL} < bid_{AL}. \quad (14)
\]

From the above inequalities, the agent who is close to a goal requires to learn more accurate goal values (i.e., the minimum steps) than the agent who is far from the goal.

Regarding $\eta$ as the parameter of random degree of the goal selection, $\eta$ is needed and has to be small (but not 0) because the agents continue to select the goal selected at first if $\eta = 0$. From Fig. 10, the small $\eta$ such as $\eta = 5$ and 10 achieves higher success rate at the first several iterations, but the success rate converges late, while the large $\eta$ such as $\eta = 15$ and 20 indicates the opposite. Although $\eta$ should be large (for exploration) at the first several iterations and 0 (for exploitation) at last, $\eta = 15$ is suitable among the constants $\eta$ empirically.

In the method 2, the agents update the minimum step of the combination from all starts to all goals when the shorter minimum step is found than the one that the agents have. This suggests that the minimum steps do not have to be true numbers because the agents aim to accept the shorter minimum step to select the appropriate goals. The minimum steps satisfy the inequality $\max(t_{AS}, t_{AL}) < \max(t_{BS}, t_{BL})$ when the agents A and B respectively reach the goals S and L. We show the equation extended from the inequality as follows.

\[
t_{AL} \text{ or } t_{BS} = \max(t_{AS}, t_{AL}, t_{BS}, t_{BL}) \quad (15)
\]

In this equation, $t_{AL}$ or $t_{BS}$ should be largest among all minimum steps so that the agents cooperate with each other in this situation, and other minimum steps can be any numbers.

5.3.2 Internal reward

From the experimental results, the internal reward enable the agents to acquire their appropriate actions to cooperate with each other, and the constant $\delta$ affect the success rate. Regarding the effect of the constant $\delta$, Fig. 11 suggests that the success rate can increase by changing $\delta$, but the convergence speed cannot be improved by changing $\delta$. Concretely, a large $\delta$ contributes to an increase of the Q-values towards the goal selected by the agent before the convergence, but it cannot recover the Q-values easier if the other goal is selected. Furthermore, $\delta$ related to Q-values gives an influence to the actions of the agents during their learning. From Fig. 11, the agents can estimate Q-values soon with suppressing an exploration of other actions if $\delta$ is large, while the agents cannot estimate Q-values soon with keeping an exploration of the other actions if $\delta$ is small.

5.3.3 Scalability of the proposed method

From the results of Experiments 2 and 3, the proposed methods are robust to the different start and goal positions represented by 10 kinds of mazes, and they can be also applied in large scale maze problems. However, the proposed methods require a lot of learning iterations to converge the success rate to 1 as shown in Fig. 12. This is because it is hard for the agents to find the minimum steps as the size of the maze increases, i.e., it needs a lot of time to correctly estimate the goal value, the internal reward, and the Q-value compared with small size maze (3 × 8 grid). To tackle this issue, the exploration criterion of the agent should be changed (e.g., Boltzmann selection is employed instead of $\epsilon$-greedy selection).

5.4 Discussion

5.4.1 Verification of goal selection

The agents in the method 1 can always select the optimal goal if the agents can acquire the minimum steps towards their goals accurately. To verify this issue, Figs. 14 and 15 show the estimated goal values by the method 1 and the theoretical analysis, respectively. In these figures, the vertical axis indicates the averaged goal values of 30 different seeds at the last iteration, while the horizontal axis indicates all combinations of the agents and the goals. The bars indicates the goal value employing the different $\eta$ (i.e., 5%, 10%, 15% and 20%), and the lines on the top of the bars indicates the standard deviation of the goal values. Since the error bar of “A to S” and “B to L” is respectively smaller than those of “A to L” and “B to S”, it is clear that the goal values in the simulation are mostly estimated by the theoretical analysis. In the case of $\eta = 5$, the standard deviation of the agent A to the goal L is large but the goal value of the agent A to the goal S becomes to be close to the theoretical one. In the case of $\eta = 20$, the result shows the opposite tendency. Such an implication suggests that the lower $\eta$ promotes the agents to exploit actions, while the higher $\eta$ promotes the agents to explore actions for the goal selection. This
Figures 14 and 15 show the estimated and theoretical goal values, respectively. Specifically, these goal values are estimated at 30000 iterations in a certain seed number with $\delta = 1$ and $\eta = 5$. In these figures, the goal values are shown for the transitions from the start (S) to the goal (G), and from the goal (G) to the other states (L). The goal values are estimated based on the distribution of the Q-values, and the theoretical goal values are calculated using the proposed methods.

5.4.2 Verification of the internal reward

Figure 16 shows the converged Q-values by the methods 1 and 2 and the theoretical analysis. Specifically, these Q-values are estimated at 10000 iterations in a certain seed number with $\delta = 1$ and $\eta = 5$. In this figure, the white, black, and gray squares indicate the start, the goal, and the other state, respectively. The black arrows from squares indicate the actions that the agents can select at the state. The right and left sides show the Q-values of the agents A and B, respectively. The upper, middle, and lower figures show the Q-values by the methods 1 and 2, and theoretical analysis, respectively.

From Fig. 16, the agents can appropriately estimate the internal rewards in both methods because the Q-values of actions toward the goals closely converge to the theoretical internal rewards, meaning that the Q-values of all state-action patterns are estimated enough accurately for the cooperation among the agents. Furthermore, since the Q-values of the actions that select the correct goal are largest in every state in both methods, the proposed methods enable the agents to estimate the Q-values in all states by Eqs. (8) and (9). This implication verifies the internal rewards theoretically.

6. Conclusion

This paper focused on the multi-agent cooperation which is generally difficult to achieve without communication among the agents, and proposed the two Q-learning methods for the multi-agent cooperation under the less and no communication. For this purpose, our proposed methods change the ordinary (external) reward to the internal reward. To guarantee the effectiveness of the proposed methods, this paper theoretically derived the mechanisms that solve the following questions: (1) how the internal rewards should be set to guarantee the cooperation among the agents under the condition of less and no communication (for methods 1 and 2). For this issue, Eq. (10) was proposed, meaning that the internal reward for the action toward the correct goal should be larger than the ordinary (external) reward for the action toward the wrong goal; and (2) how the goal values should be updated for the cooperation among the agents under the condition of no communication (for method 1). For this issue, Eqs. (2) and (3) were proposed, meaning that the farthest goal value from the start position should be updated to become larger than those of the other goals. Through the intensive simulations on the cooperating task in the maze problem where the two agents are required to cooperate with each other, we revealed that both methods successfully enable the agents to select their own appropriate cooperative behaviors which contribute to minimizing the total steps towards their goals even in the less or no communication, while the conventional methods cannot always acquire such behaviors. In detail, the proposed method derived the cooperation without communication by prioritizing the goal to select the yielding action.

The proposed method would be able to be applied into the several tasks or applications such as the pursuit game problem, puddle world problem, or transportation problem. Before applying to other tasks, however, we have to check whether the tasks or applications satisfy the following two conditions designed in the maze problem: (1) the learning is terminated when all agents reach the goals; (2) when one agent reaches the goal, the agent cannot move from the goal (i.e., absorbing Markov chain). This means that the proposed methods can be applied into many tasks or applications which satisfy these conditions. In addition to this issue, further careful qualifications and justi-
fications, such as an analysis of results with an increase of the number of agents, are needed to generalize our results.

References

[1] Y.-J. Kim, K.-H. Park, and J.-H. Kim: Modular Q-learning based multi-agent cooperation for robot soccer, Robotics and Autonomous System, Vol. 35, No. 2, pp. 109–122, 2001.

[2] M. Camara, O. Bonham-Carter, and J. Jamadinova: A multi-agent system with reinforcement learning agents for biomedical text mining, Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics, BCBI ’15, pp. 634–643, New York, NY, USA, 2015.

[3] H. Iima and Y. Kuroe: Swarm reinforcement learning methods improving certainty of learning for a multi-robot formation problem, CEC, pp. 3026–3033, May 2015.

[4] Y. Ichikawa and K. Takadama: Designing internal reward of reinforcement learning agents in multi-step dilemma problem, Journal of Computational Intelligence and Intelligent Informatics, JACIII, Vol. 17, No. 6, pp. 926–931, 2013.

[5] M. Gini M. Elidrisi, N. Johnson, and J. Crandall: Fast adaptive learning in repeated stochastic games by game abstraction, AAMAS, pp. 1141–1148, May 2014.

[6] K.J. Prabuchandran, A.N. Hemanth Kumar, and S. Bhatnagar: Multi-agent reinforcement learning for traffic signal control, 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), pp. 2529–2534, Oct. 2014.

[7] K. Tuyls, K. Verbeek, and T. Lenaerts: A selection-mutation model for Q-learning in multi-agent systems, Proc. 2nd Int. Joint Conf. Autonomous Agents and Multiagent Systems, pp. 693–700, 2003.

[8] A.L.E.M. de Cote and M. Restelli: Learning to cooperate in multi-agent social dilemmas, AAMAS, pp. 783–785, May 2006.

[9] M. Tan: Multi-agent reinforcement learning: Independent vs. cooperative agents, Proceedings of the Tenth International Conference on Machine Learning, pp. 330–337, 1993.

[10] N. Ono and K. Fukimoto, Eds.: Multi-agent reinforcement learning: A modular approach, Second International Conference on Multiagent Systems, pp. 252–258, 1996.

[11] M.L. Littman: Markov games as a framework for multi-agent reinforcement learning, 11th International Conference on Machine Learning, pp. 157–163, July 1994.

[12] J. Hu and M.P. Wellman: Nash Q-learning for general-sum stochastic games, J. Mach. Learn. Res., Vol. 4, pp. 1039–1069, 2003.

[13] C.J. H. Watkins: Learning from Delayed Rewards, PhD thesis, King’s College, 1989.

[14] R.S. Sutton and A.G. Barto: Introduction to Reinforcement Learning, 1st edition, MIT Press, 1998.

[15] Y. Ichikawa and K. Takadama: Designing internal reward of reinforcement learning agents for conflict avoidance in multi-step dilemma problem, The 16th Asia Pacific Symposium on Intelligent and Evolutionary Systems, pp. 152–157, 2012.