Reinforcement Learning with Internal Reward for Multi-Agent Cooperation: A Theoretical Approach

Fumito Uwano
The University of Electro-Communications
W6-309, 1-5-1 Chofugaoka, Chofu, Tokyo, Japan
uwano@cas.hc.uec.ac.jp

Naoki Tatebe
The University of Electro-Communications
W6-309, 1-5-1 Chofugaoka, Chofu, Tokyo, Japan
tatebe@cas.hc.uec.ac.jp

Masaya Nakata
The University of Electro-Communications
W6-309, 1-5-1 Chofugaoka, Chofu, Tokyo, Japan
m.nakata@cas.hc.uec.ac.jp

Keiki Takadama
The University of Electro-Communications
W6-309, 1-5-1 Chofugaoka, Chofu, Tokyo, Japan
keiki@inf.uec.ac.jp

Tim Kovacs
The University of Bristol
Merchant Venturers Building, Woodland Road, Clifton BS8 1UB, United Kingdom
tim.kovacs@bristol.ac.uk

ABSTRACT

This paper focuses on a multi-agent cooperation which is generally difficult to be achieved without sufficient information of other agents, and proposes the reinforcement learning method that introduces an internal reward for a multi-agent cooperation without sufficient information. To guarantee to achieve such a cooperation, this paper theoretically derives the condition of selecting appropriate actions by changing internal rewards given to the agents, and extends the reinforcement learning methods (Q-learning and Profit Sharing) to enable the agents to acquire the appropriate Q-values updated according to the derived condition. Concretely, the internal rewards change when the agents can only find better solution than the current one. The intensive simulations on the maze problems as one of testbeds have revealed the following implications:(1) our proposed method successfully enables the agents to select their own appropriate cooperating actions which contribute to acquiring the minimum steps towards to their goals, while the conventional methods (i.e., Q-learning and Profit Sharing) cannot always acquire the minimum steps; and (2) the proposed method based on Profit Sharing provides the same good performance as the proposed method based on Q-learning.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—Multiagent systems

General Terms

Performance

Keywords

Multi-Agent System, Analysis, Q-learning, Internal Reward

1. INTRODUCTION

Multi-agent reinforcement learning is suitable to tackle the many problems such as multi-robot cooperation and cars navigation. However, it is generally difficult to derive the good performance because the agents have to cooperate with each other. For this issue, some previous works, for example, have proposed swarm reinforcement learning[1] and fast adaptive learning in stochastic games[2]. Concretely, swarm reinforcement learning anables agents to choose appropriate actions from agents' various actions to generate some formation of agents. Fast adaptive learning, on the other hand, enables agents to choose the optimal action for stochastic games by observing actions of other agents, which promotes other agents to choose the optimal actions by showing the action of the agent. However, swarm reinforcement learning is heuristic, meaning that the agent cooperation cannot be guaranteed due to insufficient information of the other agents. Fast adaptive learning is theoretic but the complete information is needed which is generally difficult to acquire such an information. Even if we assume to acquire the complete information, a huge amount of communication would be needed but we generally cannot guarantee no communication failure. To tackle the above issues, this paper proposes the theoretic method with a (very) small amount of information. Concretely, this paper theoretically derives the condition of selecting appropriate actions by changing internal rewards given to the agents, and extends the reinforcement learning methods (i.e. Q-learning and Profit Sharing) to enable the agents to acquire the appropriate Q-values updated according to the derived condition. Note that Q-learning agents are employed because of the mathematical proof in Q-learning (i.e., the convergence of Q-value is proofed in the single agent environment.), and profit sharing agents are also employed for a comparison with Q-learning ones. As the first step towards our purpose in this paper, we start to investigate the proposed method in the simple maze problem where two agents learn the actions to minimize the steps towards their goals through the cooperation with each other by a

BICT 2015, December 03-05, New York City, United States

DOI 10.4108/eai.3-12-2015.2262878

small amount of communication. This paper is organized as follows. Section 2 explains reinforcement learning and Section 3 describes the multi-agent cooperation task addressed in this paper. Our method is proposed in Section 4. Section 5 conducts the experiment and analyzed the obtained results. Finally, our conclusion is given in Section 6.

2. REINFORCEMENT LEARNING

Before we add a mechanism using the internal reward to two reinforcement learning techniques Q-learning and Profit Sharing, this section gives their brief descriptions.

2.1 Q-learning

Q-learning[3] is a very popular reinforcement learning technique which is originally designed for a single-agent task. As the general framework of reinforcement learning, an agent interacts with an environment; the agent observes a state from an environment, takes an action then receives a reward from it.

In Q-learning, the agent calculates a state-action value (called Q-value $Q(s,a)$) for each possible state-action pair in the environment, which estimates a future reward the agent will eventually receive when its action a is executed in its state s. The agent acquires a policy $\pi(s,a)$ to decide which action should be executed to maximize the received reward. This results in finding the minimum step to a proper goal returning the maximum reward. Technically the policy π can be a probability in selecting the action on the state s and is calculated on the basis of Q-value $Q(s,a), a \in A$ where A is the action space that defines possible actions the agent can take at the state s.

The agent which is powered by Q-learning aims at estimating all possible Q-values accurately in order to find the minimum step, thorough the interaction with the environment. $Q(s,a)$ is updated as follows;

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a' \in A'} Q(s',a') - Q(s,a)], \quad (1)$$

where $\max Q(s',a')$ is the largest Q-value in state s' after the transition from state s to s' with executing action a; r is the reward received from the environment; α is the learning rate and γ is the discount factor. α is the real number from 0 to 1, and expresses what percentage of new value(reward, etc) incorporated to Q-value. γ is the real number from 0 to 1, and presents how much incorporate the Q-value calculated before to new Q-value.

2.2 Profit Sharing

Profit Sharing[4] is also a popular reinforcement learning technique. Similar to Q learning, an agent interacts with the environment and calculates the stat-action values on the framework of Profit Sharing. Different from Q-learning, Profit Sharing is designed to calculate Q-values when the agent when the agent receives the reward; Q-learning calculates Q-values every after executing the action. The agent stores a history of state-action pairs the agent sensed and executed. Then, a set of Q-values for all stored state-action pairs is calculated as follows;

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + C_{bid}\left[r(t) - Q(s_t,a_t)\right], \quad (2)$$
$$\text{while } t = 1, \cdots, \text{episode} - 1,$$

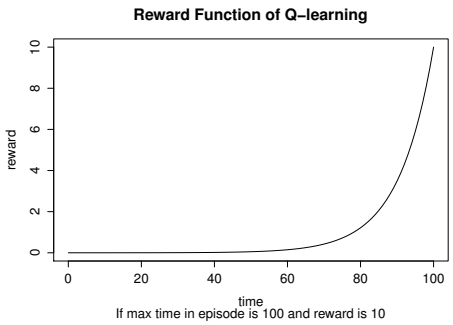

Figure 1: Reward function for Profit Sharing this paper employs

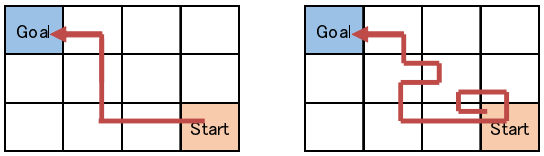

Figure 2: Actions putted value (left) and all actions (right)

where (s_t, a_t) is the state-action pair the agent was placed at time t; $r(t)$ is a reward function decides a value of reward assigned to $Q(s_t,a_t)$; C_{bid} is the coefficient. The reward function $r(t)$ can be defined in several ways. Here, we employ the following reward function as shown in Figure 1;

$$r(t) = r\gamma^{\text{episode}-t} \quad (3)$$

In this study, A Profit Sharing agent put Q-value on the actions of the path made by the agent as left of figure 2. In figure 2, the agent reaches the goal by passing through the path same as right of figure 2. then, the agent makes the path left of figure 2 from right of figure 2, and updates actions on the path left of figure 2. the way of making the pass is according to the following three rules.

1. The agent follows a path, and stores the path it already pass.
2. When the agent goes back to the same state, the path between same states from the path already stored by the agent is reset.
3. If the agent reaches the goal, the way is finished.

The reason why the agent makes this path is that Q-values of all actions become like Q-value of common Q-learning and the situation becomes like that of proposed method.

3. MULTI-AGENT COOPERATION TASK

This section introduces a multi-agent cooperation task using a 3x8 grid maze problem. Figure 3 shows an example of 3x8 grid maze. On the maze this paper uses, as shown in the figure, we define there are two possible start states (A, B) where agents will be initially placed before learning; and two goal states (S, L) where the agents attempts to reach. A difficulty of multi-agent cooperation task on the maze, is

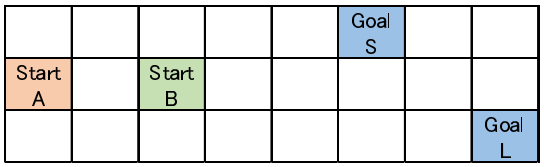

Figure 3: Maze Problem

each agent attempts to selfishly learn the minimum step for him thus does not find a cooperating behavior with other agents. For instance, on the maze problem shown by Figure 3, when two agents are placed at states A and B respectively, they attempt to reach the same goal state S since their minimum steps to goal can be achieved by reaching it. In this case, the agent who is placed at state B can potentially reach goal state S faster than another agent at state A. Thus, the agent at state A should reach the goal state L. This is the best (but selfish) solution for the agent at state B while the worst solution for agent at state A since he should take the longest step to reach goal. The cooperating behavior can be determined as; the agents at state A and B reach the goal state S and L respectively. This difficulty is often called as a dilemma problem.

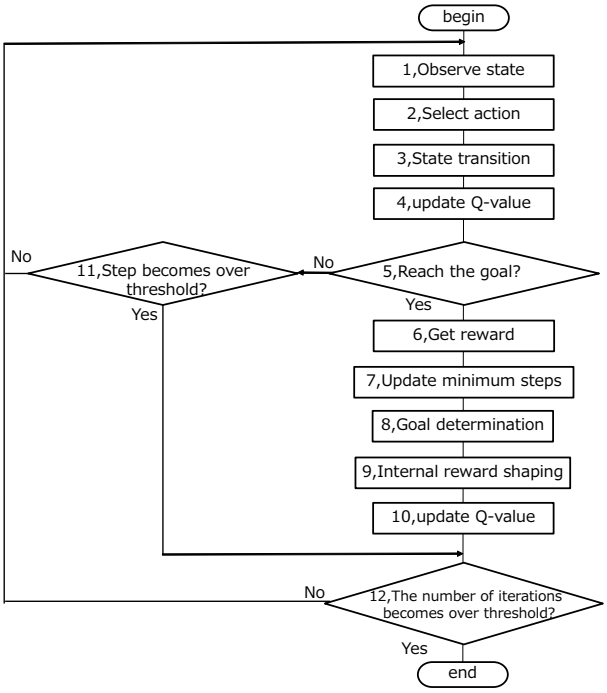

Figure 4: Flow chart of proposed method

4. PROPOSED METHOD

In the dilemma maze problem like Figure 3, in order to achieve the cooperation of agents, the proposed method mainly has the following two steps; Step 1 is a process of goal determination with communication between the agents, and step 2 is a process of internal reward shaping. The remain of this section first explains the overview of the proposed method as shown in Figure 4 and then the main 2 steps pointed at the processes 8 and 9 on the figure.

4.1 Overview of procedure

As shown in Figure 4, the agents observe a state and then choose an action as the framework of reinforcement learning (processes 1 and 2), which results in the state transition as process 3. Then, the agent updates Q-value for the executed action (process 4). These processes are the standard procedure of the reinforcement learning and the cycle from processes 1 to 4 is often called as "step".

After process 4 the proposed method determines whether each agent reached the goal (process 5). If the agent reach the goal, the receives reward (process 6); otherwise, jump to process 11. In process 7, the agent updates minimum steps; specifically, if the number of steps from a start position to goal are shorter than stored minimum steps the agent has acquired before, the agent replaced the minimum steps with the new ones the agent newly founded.

Next, in process 8, the agent determines the optimal goal by the minimum steps (detail can be founded in subsection 4.2). Then, the agent estimates an internal reward by using the minimum steps (detail can be founded in subsection 4.3). After that, the agent updates Q-value using internal reward in process 10. In process 11, the system determines whether the step count is greater than a threshold; If true, the system go to 12; otherwise, the agent goes back to process 1. In process 12, the system counts iteration of learning and determines whether this iteration count is greater than a threshold. the whole process is ended when the system meets this condition; otherwise, the system returns to process 1.

4.2 Goal determination

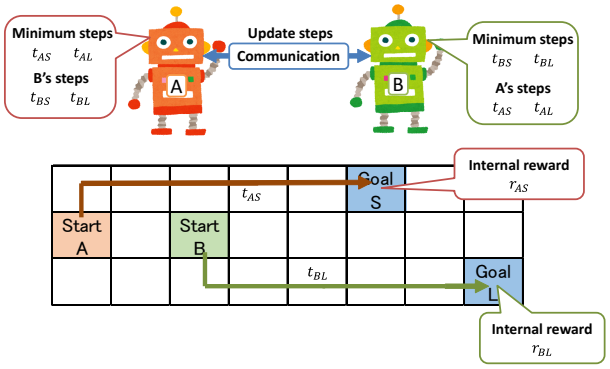

Figure 5: Goal determination

In the proposed method, agents have memorized the minimum steps between every start and every goal (because of the Q-value table). When reaching one goal by shorter steps than before, agents memorize these steps and send them to other agents, i.e., each agent share the Q-value table other agents learn. This sharing can be continued while agents' learning process to explore the maze. Different from the standard reinforcement learning where an agent explores the shortest steps to goal in order to maximize the reward he gets, the proposed method attempts agents to find goal where all agent can receive the maximum rewards per unit step. For example, Figure 5 shows these processes in the maze of figure 3. Each agent has memorized the minimum steps as t_{AS}, t_{AL}, t_{BS} and t_{BL} in the balloon of the agent.

In the figure, agent A has reached goal S and agent B has reached goal L by passing through the path indicated by the directional arrows. In this time, if the number of t_{AS} newly discovered is shorter than the number of t_{AS} which agent A already has had, the t_{AS} is replaced with new t_{AS}, and sent to agent B. Then, Agent B update t_{BL} by the way same as agent A. This results in that each agent memorizes the minimum steps between all agents and all goals. Then, each agent determines the goal where all agent can receive the maximum rewards. In this figure, goal S is the goal which agent A must reach and goal L is the goal which agent B must reach, because the step count is shortest when agent A reaches goal S and agent B reaches goal L.

4.3 Internal Reward Shaping

In this process, each agent shapes reward function to reach the goal chosen in first step. Figure 6 presents a way for two agent to cooperate each other. t_{BS} is minimum step count between start B to goal S and t_{BL} is minimum step count between start A to goal L. Red star presents the turning point to determine whether the agent reaches goal state S or L. Yellow directional arrows present the agent's mainly action for each goal states, and arrows' thickness present Q-value.

Note that each agent knows the goal with the process introduced in subsection 4.2. The agent estimates Q-value to reach the optimal goal determined in subsection 4.2 by an internal reward for Q-value described above.

In figure 6, agent A should reach goal S and agent B should reach goal L. Note, it is not necessary for agent A to set internal reward, since it reaches goal S normally (maximizing reward for the agent a). However, Agent B should set the internal reward; under the standard reinforcement learning Agent B would reach to the goal S, while Agent B need to reach the goal L for maximizing the reward for all agents. Then, in the proposed method, the internal reward is added to reform the reward shaping of agent B to reach the goal L.

In the turning point on Figure 6, the Q-value of the action to reach goal S eventually converges to a value r and thus the Q-value of the action to reach goal L is $r\gamma^2$. If Agent B uses the internal reward r_S, r_L, the Q-value of the action to reach goal S is r_S and the Q-value of the action to reach goal L is $r_L\gamma^2$. Since $r_L\gamma^2 > r_S$ is satisfied, if r_L is $\frac{r}{\gamma^2}+1$ and r_S is r, agent B will reach goal L finally and be able to cooperate. We explains the general way to shape internal reward in the following.

4.3.1 *Mathematical Analysis*

In this section, there is mathematical description for internal reward shaping in preceding section. Therefore, we describe generalized proposed method on the case of the maze of figure 6 in this paper.

Agents estimate r_{AS}, r_{AL}, r_{BS} and r_{BL} for agents to get cooperative action. r_{AS} is agent A's internal reward for goal S, r_{AL} is agent A's internal reward for goal L, r_{BS} is agent B's internal reward for goal S and r_{BL} is agent B's internal reward for goal L. Whether to reach goal S or not is determined by Q-value in the turning point, if there are

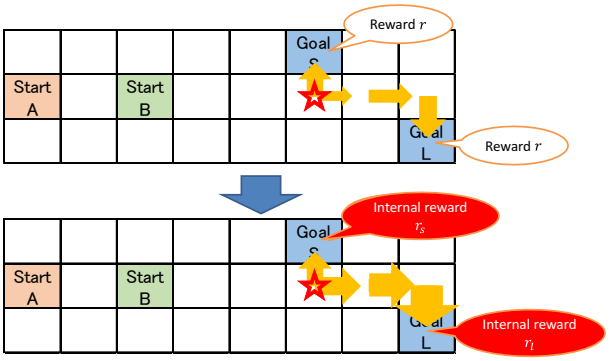

Figure 6: Internal reward

same rewards in both goal in figure 6. Because agent A must reach goal S from the situation in figure 6, its Q-value had by action toward goal S is largest in the state in front of goal S. Also in the same manner for agent B, its Q-value of the action toward goal L is largest in the turning point. If t_{AS} represents minimum step count when agent A reaches goal S, t_{AL} represents minimum step count when agent A reaches goal L, t_{BS} represents minimum step count when agent B reaches goal s and if t_{BL} represents minimum step count when agent B reaches goal L, each Q-value of the action toward goal S is as follow:

$$\gamma^{t_{AS}} r_{AS}\,, \gamma^{t_{BS}} r_{BS} \tag{4}$$

Each Q-value of the action toward goal L is as follow:

$$\gamma^{t_{AL}-t_{AS}} r_{AL}\,, \gamma^{t_{BL}-t_{BS}} r_{BL} \tag{5}$$

Because agent A must reach to goal S the expression is satisfied as follows:

$$r_{AS} > \gamma^{t_{AL}-t_{AS}} r_{AL} \tag{6}$$

Also in the same manner for agent B, the expression is satisfied as follows:

$$r_{BL} > \frac{r_{BS}}{\gamma^{t_{BL}-t_{BS}}} \tag{7}$$

In figure 6, equation 7 is equal to $r_L\gamma^2 > r_S$ (2 of $r_L\gamma^2 > r_S$ in figure 6 is $t_{BL} - t_{BS}$). As for agent A, it is not necessary to set internal reward, since $r_{AS} = r > \gamma^2 r = \gamma^2 r_{AL}$ is established while $r_{AS} = r$ and $r_{AL} = r$. Therefore, generalization from figure 6 is succeeded. On implementation, the system must consider the quantity of the difference to meet equation 6 and 7. Proposed method in this paper set parameter δ. r_{AS} and r_{BL} using δ are equation 8 and 9.

$$r_{AS} = \gamma^{t_{AL}-t_{AS}} r_{AL} + \delta \tag{8}$$

$$r_{BL} = \frac{r_{BS}}{\gamma^{t_{BL}-t_{BS}}} + \delta \tag{9}$$

Furthermore, we show the expression obtained by modifying the above equations below.

$$\gamma^{t_{AS}} r_{AS} > \gamma^{t_{AL}} r_{AL} \tag{10}$$

$$\gamma^{t_{BS}} r_{BS} < \gamma^{t_{BL}} r_{BL} \tag{11}$$

$\gamma^{t_{AS}} r_{AS}$ presents the Q-value the action to reach goal S in agent A's start point. $\gamma^{t_{AL}} r_{AL}$ presents the Q-value the

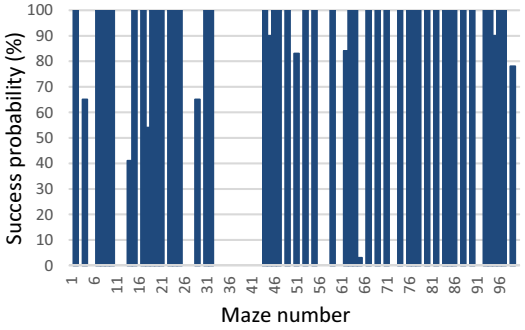

Figure 7: Q-learning case 1

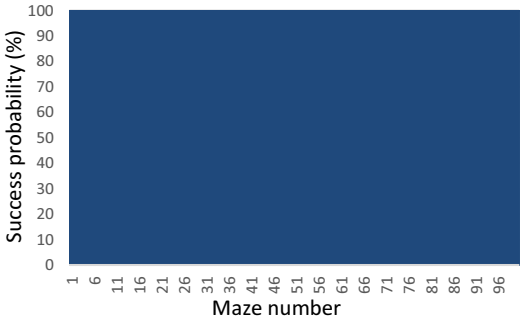

Figure 8: Proposed method(Q-learning, gap10) case 1

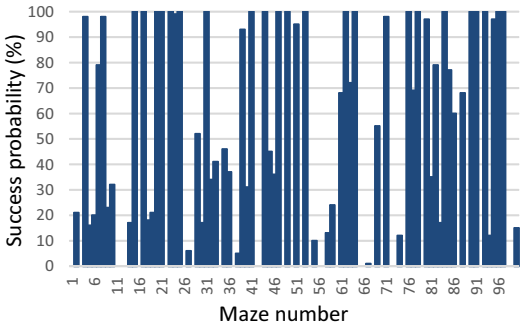

Figure 9: Q-learning case 2

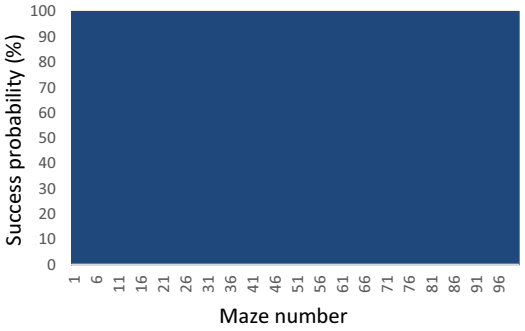

Figure 10: Proposed method(Q-learning, gap10) case 2

Figure 11: Common Q-learning vs Proposed method

action to reach goal L in agent A's start point. $\gamma^{t_{BS}} r_{BS}$ presents the Q-value the action to reach goal S in agent B's start point. $\gamma^{t_{BL}} r_{BL}$ presents the Q-value the action to reach goal L in agent B's start point. Therefore, whether to reach the goal or another goal, is decided by whether to choose action towards the goal or another goal, at the start state.

5. EXPERIMENT

5.1 Experimental setting

Here, we test our mechanism on two reinforcement learning techniques Q-learning and Profit Sharing, i.e., both are extended with the proposed method to be applicable to the multi-cooperation task on the maze problem. Specifically, we apply both extended techniques to 100 different types of 3x8 grid mazes where two start states and two goal states are differently placed in the maze. Note this paper deals with 2 agents cooperation task. We consider the following four cases as possible configurations on multi-agent cooperation task;

- case 1 : ideal and easy case

  In this case, an agent cannot reach the goal another agent already reached, and all agents already knows minimum steps between every start and every goal at first. In addition, if each agent observes the same state three times and Q-value is decreased over 0.1, Q-value is not updated in that iteration.

- case 2 : ideal and difficult case

  In this case, each agent reaches the goal even if other agents reached same goal and all agents already knows minimum steps between every start and every goal at first.

- case 3 : practical and easy case

  In this case, an agent cannot reach the goal another agent already reached, and all agents do not know minimum steps between every start and every goal at first. In addition, if each agent observe same state three times and Q-value is decreased over 0.1, Q-value is not updated in that iteration, and when minimum steps were updated, agents initialize all Q-value.

- case 4 : practical and difficult case

  In this case, each agent reaches the goal even if other agents reached the same goal and all agents do not know minimum steps between every start and every goal at first. In addition, when minimum steps were updated, agents initialize all Q-value.

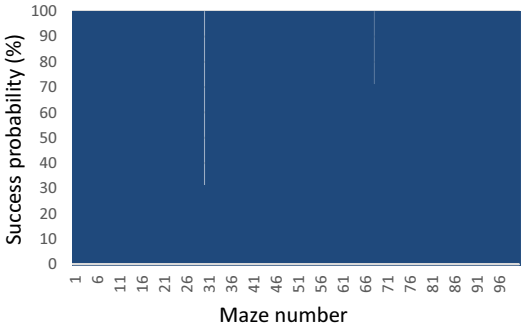

Figure 12: Proposed method(Q-learning) case 3

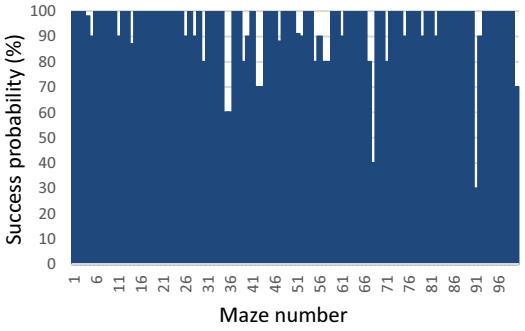

Figure 13: Proposed method(Profit Sharing) case 3

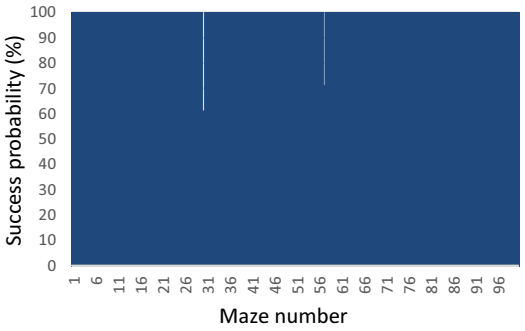

Figure 14: Proposed method(Q-learning) case 4

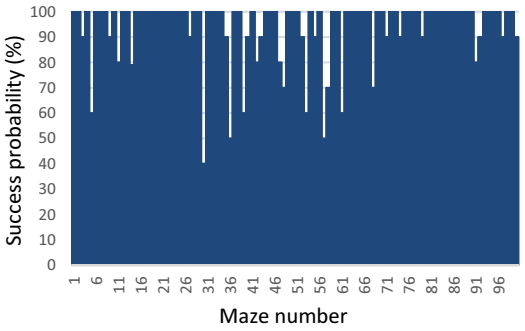

Figure 15: Proposed method(Profit Sharing) case 4

Figure 16: Q-learning vs Profit Sharing

We use the parameters for Q learning and Profit sharing as summarized in Table 1. To evaluate effectiveness of our method, as an evaluation criterion, we calculate a success rate of two agents took the cooperating behaviors at the end of iterations (among 9991∼10000 iterations). We calculate the average of the success rate of 10 experiments.

Table 1: Parameters

	Q-learning	Profit Sharing
iterations	10000	
active frequency	100	
initial Q-value	0.1	0
learning rateα	0.1	not exist
discount rateγ	0.9	not exist
coefficientC_{bid}	N/A	0.1
random seed	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	
action selection	ϵ-greedy	
	$\epsilon = 0.7$	$\epsilon = 0.5$
reward	10	
constantδ	0.1, 0.5, 1, 10	1

5.2 Result

Figures 9-15 show the success rate. The vertical axis shows success times of cooperation and horizontal axis presents 100 kinds of mazes. Agents could always cooperate each other in case 1 and 2, but agents could not always cooperate each other in case 3 and 4. Common Q-learning agents failed to cooperate each other and proposed method agents succeeded in cooperating each other in case 3 and 4. At the result of compare Q-learning agents and Profit Sharing agents, Q-learning agents were better than Profit Sharing agents.

5.2.1 Common Q-learning vs Proposed Method

The comparison result of common Q-learning and proposed method is as follow figure 11. Success rate of proposed method is 100 percent, and the proposed method is seen as the performance is good compared to common Q-learning.

5.2.2 Q-learning vs Profit Sharing

The comparison result of Q-learning applied the proposed method and Profit Sharing applied the proposed method is as follow figure 16. The performance of Profit Sharing is good. However, the performance of Q-learning is better than that of profit sharing.

5.2.3 The results of each reward difference

The result of internal reward difference is as follow figure 23. When the reward difference is large, there is the high probability of success in each case.

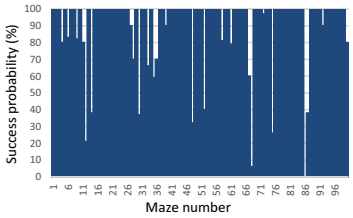

Figure 17: gap0.1, case 1

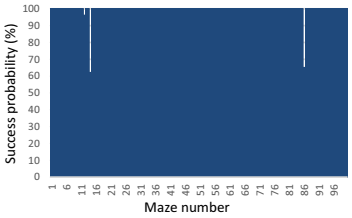

Figure 18: gap0.5, case 1

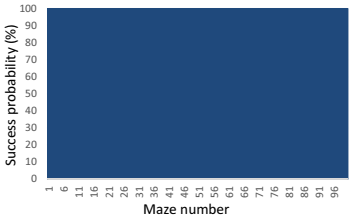

Figure 19: gap1, case 1

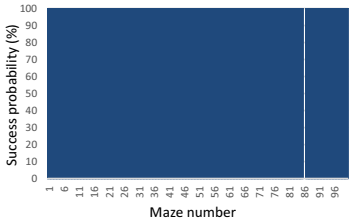

Figure 20: gap0.1, case 2

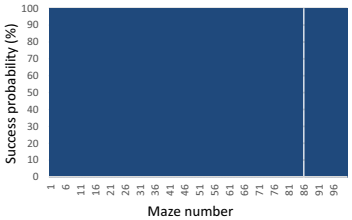

Figure 21: gap0.5, case 2

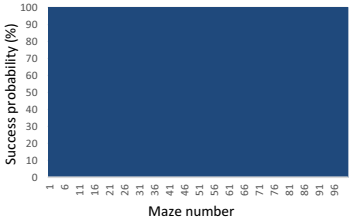

Figure 22: gap1, case 2

Figure 23: Result of reward difference for proposed method(Q-learning)

5.3 Discussion

From the result in the cases 1 and 2, we can think the mathematical analysis in Section 4 is validated, since our analysis argues that the internal reward can be theoretically determined if agents know the minimum steps to possible goal states. However, as shown in the results of the cases 3 and 4, for some mazes, our method fails to enable agents to cooperate with each other. Let us discuss why the agent sometimes fail to cooperate.

In case 3 and 4, proposed method agents using Q-learning fail to cooperate each other in same two mazes. Thus, there are mazes that agents cannot cooperate easily. The reason of this is that agents cannot reach every goal enough times. In addition, a gap between case 1,2 and 3,4 is whether agents know minimum steps or not. Therefore, the reason of error is that agents cannot search minimum steps. Against this problem, if we increase learning iterations, the errors are decreased. For example, figure 25 shows the result of 30000 iterations in case 4. However, solution for many learning iterations is not practical. From the above, a searchability of agents decides whether an agent succeeds in cooperating another agent. We put 0.7 to epsilon in order to improve the searchability in experiment of this paper. Therefore, agents search many times to large areas of maze. In this paper experiment, epsilon is larger than normal.

5.3.1 *Profit Sharing*

When reaching a goal, Profit Sharing agents update Q-values of actions between from start to goal. Therefore, Profit Sharing agents are better efficiency than Q-learning agent in terms of searching areas. However, experiment results of proposed method using Profit Sharing are worse than those of proposed method using Q-learning. The reason of this is the difference of ability of searching minimum steps between Q-learning and Profit Sharing. Q-learning is better in terms of searching minimum steps, because Q-learning ensures optimality of Q-value. However, the result of Profit Sharing didn't have dependence of each maze. For that reason, Profit Sharing agents can search larger areas than Q-learning agents in terms of Q-value.

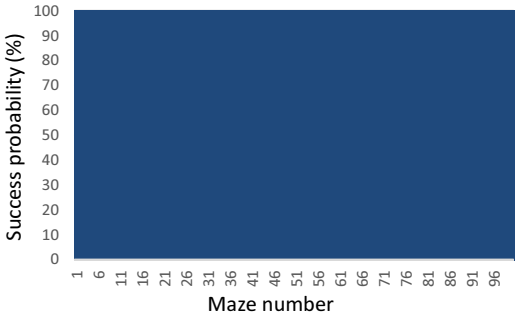

Figure 25: Result of 30000 iterations

5.3.2 *Gap between internal rewards*

Agents can cooperate when there is even a little gap between internal rewards from analysis of section 4. However, from the result of experiment, there is some error of successful probability when the gap is under 1. Therefore, in practical case, this is dependent on the ability of searching for agents. Then, successful probability is 100 percent when the gap is over 1. From the above, a gap of internal reward can cover the difference between section 4's case and practical case. For example, figure 25 presents distribution of the Q-value. There are four mazes in figure 24. Some squares in mazes are trout, and orange square is goal position, a blue square is start position and green square is common position. Some

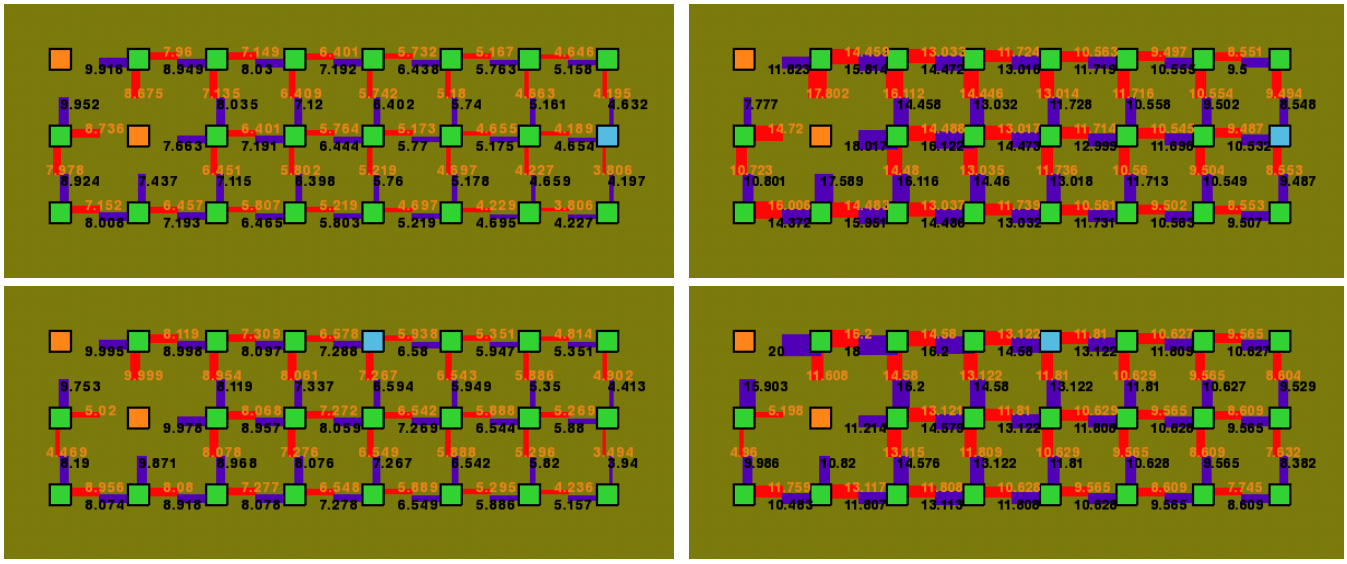

Figure 24: Q-value in maze 14 (upper left: agent A and internal reward gap is 0.1, lower left: agent B and internal reward gap is 0.1, upper right: agent A and internal reward gap is 10, lower right: agent B and internal reward gap is 10)

bar extending from squares presents action that agents can choose in this position. Red bar presents right and down action, and blue bar presents left and up action. The number near these bars presents Q-value. Upper mazes shows about Q-value of agent A, and other mazes shows about Q-value of agent B. Left mazes shows Q-value when internal reward gap is 0.1, and right mazes shows Q-value when internal reward gap is 10. From figure 24, left agents could not cooperate, but right agents had cooperated. In addition, If the gap of internal reward is increased, the gap between Q-value of actions in same position is increased. From this thing, it becomes difficult for agents to choose different action in each position than before, because the magnitude relation of Q-value does not change hardly. From the above, agents have a robustness to incorrect learning by using the gap of internal reward. Therefore, if the gap of internal reward is increased, agents can reach the goal for cooperation between agents. However, if agents make mistake to choose the goal, in other words, if agents learn by mistake without cognition, agents cannot correct learning hardly by same reason.

6. CONCLUSION

This paper focused on a multi-agent cooperation which is generally difficult to be achieved without sufficient information of other agents, and proposed the reinforcement learning method that introduced an internal reward for a multi-agent cooperation without sufficient information. To guarantee to achieve such a cooperation, this paper theoretically derived the condition of selecting appropriate actions by changing internal rewards given to the agents, and extends the reinforcement learning methods (i.e., Q-learning and Profit Sharing) to enable the agents to acquire the appropriate Q-values updated according to the derived condition. Through the intensive simulations on the 3x8 grid maze problems where two agents are required to cooperate with each other, the following implications were revealed: (1) our proposed method successfully enables the agents to select their own appropriate cooperating actions which contribute to acquiring the minimum steps towards to their goals, while the conventional methods (i.e., Q-learning and Profit Sharing) cannot always acquire the minimum steps. Note that the same tendency is obtained even in the different parameter settings such as the epsilon and the gap of internal reward; and (2) the proposed method based on Profit Sharing provides the same good performance as the proposed method based on Q-learning. What should be noticed here is that the results have only been obtained from one simple gird maze problem with two agents. Therefore, further careful qualifications and justifications, such as an analysis of results using other maze problems or an increase of the number of agents, are needed to generalize our results. Such important directions must be pursued in the near future in addition to the following future research: (1) reducing the number of sending the information to other agents; and (2) applying proposed method to other tasks.

7. REFERENCES

[1] H. Iima and Y. Kuroe, Swarm Reinforcement Learning Methods Improving Certainty of Learning for a Multi-Robot Formation Problem, CEC Conference, pp.3026-3033, 2015.

[2] Mohamed Elidrisi, Nicholas Johnson, Maria Gini and Jacob Crandall, Fast Adaptive Learning in Repeated Stochastic Games by Game Abstraction, AAMAS Conference, pp.1141-1148, 2014.

[3] Christopher JCH Watkins, Learning from Delayed Rewards, King's College, 1989.

[4] John J. Grefenstette, Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms, Machine Learning, 3, pp.225-245, 1988.

[5] Christopher JCH Watkins and Peter Dayan, Technical Note: Q-Learning, Machine Learning, 8, pp55-68, 1992.

[6] 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.