Proposal and Evaluation of Reward Sharing Method Based on Safety Level

Naoki KODAMA *, Kazuteru MIYAZAKI **, and Hiroaki KOBAYASHI *

Abstract: Profit sharing guarantees the rationality for a specific class by the rationality theorem. However it has a problem that the final result of learning is strongly influenced by the early learning phase. A reward sharing method based on safety level (RSMSL) solved this problem by introducing safety level and increasing/decreasing reward sharing rate. In this paper, we will make two improvements to the RSMSL, modification of the reward sharing function and introduction of a safety discount rate. The improved efficiency is shown in an inverted pendulum problem and a keepaway task.

Key Words: reinforcement learning, exploitation-oriented learning, profit sharing, multi-agent learning.

1. Introduction

Reinforcement learning is a type of machine learning that adapts to environments with rewards as a clue. Traditional reinforcement learning methods are often based on dynamic programming (DP). A reinforcement learning problem mainly comes down to an optimization problem to obtain the optimal policy in Markov decision processes (MDPs) by assuming Markov property for the environment. There are a temporal difference (TD) method, Q-learning (QL) [1], and a sarsa method (SARSA) [2],[3] as representative methods of such a DP-based method. However, there are many problems where Markov property cannot be assumed. Examples are perceptual aliasing problems and multi agent problems. The DP-based method does not guarantee the optimality and its rationality under these environments.

On the other hand, exploitation-oriented learning (XoL) [4] was proposed as an approach to reduce the number of trial and error searches by strengthening the obtained experience. XoL aims firstly to guarantee the rationality than the optimality with fewer trial and error searches. Profit sharing (PS), a kind of XoL, guarantees the rationality for a specific class by the rationality theorem [12]. The rationality theorem does not require the condition of MDPs and can be applied naturally to other classes. Actually, the PS-based method shows the effectiveness also in many non-MDPs [5]–[7].

So far PSwithEFP [5] using expected failure probability (EFP) [8] was proposed as a PS method that can handle reward and penalty at the same time. PSwithEFP has succeeded in granting PS the ability to learn penalty avoidance policy quickly. The study [5] shows the effectiveness in the keepaway task [9] where penalty avoidance policy is important for learning.

However, the essential problem for PS still remains. That is, the final result of learning is strongly influenced by the early learning phase. For example, suppose that if a not so good trial managed to end successfully and got a large reward in the early learning phase, then those rules will be reinforced strongly. This means that those rules will be selected with large possibility in the subsequent learning and result in an inefficient strategy. This is a problem caused by PS reward acquisition, and it is a difficult problem to solve for PSwithEFP.

In this paper, in order to solve the problem, we propose the reward sharing method based on safety level (RSMSL) [10] and make two improvements to it. RSMSL solved this problem by introducing safety level and increasing/decreasing reward sharing rate. In [10], it was shown for a keepaway task that although the learning speed in the early learning phase is slower than PSwithEFP, the final result is better than that of PSwithEFP. However, the average value was better but some results were very bad. Therefore, we aim to realize a method that enables more stable learning of penalty avoidance policy by re-examining the process of decreasing the safety level and the reward sharing function. To show the effectiveness, we use an inverted pendulum problem as a single agent problem and a keepaway task as a multi agent problem.

2. Reinforcement Learning

2.1 Definition of Terms

After obtaining a sensory input from the environment, a learner selects an action and executes it. For a series of actions, a reward or a penalty is given from the environment. The time is discretized with the cycle of sensory input and action execution as one unit. A sensory input from the environment is a state. Due to incompleteness of sensory input, it may be necessary to select different actions at the same state. This is commonly called perceptual aliasing problems [11].

A state-action pair is called a rule. When an action a is selected in a certain state x, the rule is described as rule(x,a). A rule sequence from the initial state or immediately after rewarded (punished) rule to the next rewarded (punished) rule is called an episode. When different actions are selected at the same state in a certain episode, the rule sequence between them is called a detour sequence, and a rule always existing on the
208

SICE JCMSI, Vol. 11, No. 3, May 2018

detour sequence in all episodes where the rule was selected up to the present is called an ineffective rule, and otherwise it is called an effective rule. Obviously ineffective rules should not be selected if effective rules exist in the same state [12]. If a rule determined to be an effective rule at one time is subsequently determined to be an ineffective rule, the rule is called a type 2 confusion [7]. A function for determining an action from state input is called a policy. If a policy gives a positive expected value of the reward per unit action, it is called a rational policy, and a policy that maximizes it is called an optimal policy. A policy that minimizes an expected value of the penalty per unit action is called a penalty avoidance policy. Note that in order to ignore the reward in the penalty avoidance policy, rules that cannot be given a penalty and a reward (such as the case when the episode length is infinite) are included.

2.2 Profit Sharing

Profit sharing (PS) is a method to collectively reinforce evaluation values given to all the rules in the episode when rewarded. A function that returns the reinforcement value that varies according to the distance from the rewarded is called a reinforcement function. Let \( f_i \) be a reinforcement value for the \( i \)-th step rule before the rewarded rule. For the episode of length \( l(\text{rule}_1, \ldots, \text{rule}_i, \ldots, \text{rule}_2, \text{rule}_1) \), the weight \( \omega_{\text{rule}_i} \), of rule \( i \) is reinforced as Eq. (1):

\[
\omega_{\text{rule}_i} \leftarrow \omega_{\text{rule}_i} + f_i, \\
f_i = \lambda^{-i}R,
\]

where \( \lambda \) is the discount rate \( 0 < \lambda \leq 1 \), \( R \) is the reward.

A necessary and sufficient condition for a policy that always selects the rule with the highest evaluation value at each state to be a rational policy is given by the rationality theorem of PS as Eq. (2) [12]:

\[
\forall i=1, 2, \ldots, W, \quad L \sum_{j=i}^{W} f_j < f_{i-1},
\]

where \( W \) is the maximum length of the episode, and \( L \) the maximum number of effective rules existing at the same state. In general, the value of \( L \) is not known, but as the simplest reinforcement function satisfying the theorem is a geometric decreasing function with a common ratio \( 1/(\text{Number of action alternatives}) \).

PS is a learning method that strongly strengthens experience. Therefore, PS does not guarantee that the learned policy necessarily maximizes the reward, but it can learn rational policies with fewer trial and error searches. In a DP-based method such as QL, acquisition of the optimal policy is guaranteed in MDPs that is a one of problems without type 2 confusion, but even the rationality is not guaranteed with other classes. On the other hand, the rationality theorem of PS guarantees the acquisition of a rational policy for the classes without type 2 confusion and does not require the condition of MDPs.

3. PS Dealing with Rewards and Penalties

3.1 Purpose of a Penalty in PS

A penalty in PS has two purposes. The first is to accelerate learning speed of a rational policy by learning a penalty avoidance policy. For example, in a keepaway task, the acquisition frequency of penalty is overwhelmingly higher than the acquisition frequency of reward, then learning speed is mainly improved by the penalty avoidance policy. Actually, in [5], the introduction of penalty into PS has resulted in improved learning speed compared to the only reward case.

Second, in an environment where the state transition is probabilistic, by limiting the selection of rules with the penalty, learning of a policy with the expected value of the reward per unit action \( (\text{ERpA}) \) becomes higher. This is one of the important factors because it directly leads to improvement of learning performance. For example, consider two same-length paths in a maze with pitfalls where tasks fail with a certain probability. In that case, it is better trivially to go through a route with lower probability of falling into the pitfall, and the ERpA also becomes larger. However, PS that learns only with rewards has insufficient ability to learn policies with a higher ERpA. In fact, in [5], it was shown that by introducing penalty into PS, the final learning result was greatly improved compared with only reward.

However, the use of penalty [13] may deteriorate the learning performance depending on the environment. For example, the learning performance greatly deteriorated due to the concurrent learning problem [14].

In recent years, PS with EFP was proposed as a method to efficiently handle both reward and penalty. It can quickly learn the penalty avoidance policy by introducing expected failure probability (EFP).

3.2 PS with EFP

In [8], EFP was proposed to efficiently propagate penalty. EFP is “the expected failure probability (that is, obtaining penalty) in the future as a result of using an action in a certain state.”

EFP is updated online as follows:

(1) When rule \((x,a)\) is selected and received penalty immediately,

\[
p_i(E|x,a) = (1 - \eta) \times p_{i-1}(E|x,a) + \eta.
\]

(2) When rule \((x,a)\) is selected and the state transited from the state \( x \) to the state \( k \),

\[
p_i(E|x,a) = (1 - \eta) \times p_{i-1}(E|x,a) + \eta \times p(E|k)
\]

\[
p(E|k) = \sum_{i} p(i|k)p(E|rule(k,i)),
\]

where \( x \) is the state, \( a \) the action, \( N_a \) the number of actions which can be selected in state \( k \), \( E \) the event of failure in the future, \( t \) the number of EFP updates for that rule, \( p_i(E|x,a) \) the EFP when rule \((x,a)\) is used at \( t \)-th step, \( \eta \) the propagation rate of failure probability \( (0 < \eta < 1) \), \( p(E|k) \) the EFP when passing through state \( k \) and \( p(i|k) \) is the probability of action selection of rule \((k,i)\).

PS with EFP [5] introduces the EFP into a roulette wheel selection. The calculation formula of the probability of action selection with EFP introduced for roulette wheel selection is as follows:

\[
pe(a|x) = \frac{(1 - p(E|x,a))\omega(x,a)}{\sum_{i=1}^{N_a}(1 - p(E|x,i))\omega(x,i)},
\]

where \( pe(a|x) \) is the probability of action selection of rule \((x,a)\) in PS with EFP, and \( N_a \) is the type of action.
From Eq. (5), it does not select a rule with a higher EFP, but a rule that will not fail more. For example, considering two rules of rule\( (x, a) \) and rule\( (x, b) \), when \( \omega(x, a) = 200, p(E|rule(x, a)) = 0.8, \omega(x, b) = 100, \) and \( p(E|rule(x, b)) = 0.4 \), the probability \( p(\omega(x)) = 40\% \) and \( p(\omega(x)) = 60\% \).

Since EFP is incorporated into action selection, experience of penalty is immediately reflected, and learning of penalty avoidance policy is done quickly.

4. RSMSL and Improvement RSMSL

4.1 Introduction of Safety Level

As mentioned above, PS has a problem that the final result of learning is strongly influenced by the early learning phase due to the characteristic that it strongly strengthens the obtained experience. If inefficient rules are strengthened strongly at the early learning phase, there is a possibility of converging to inefficient policies. For example, let us consider two rules of rule\( (x, a) \) and rule\( (x, b) \). Suppose that the initial values of weights are \( \omega(x, a) = 10 \) and \( \omega(x, b) = 10 \), and as a result of a certain random action, reinforcement value 80 is given to the rule\( (x, a) \). Then, the weight of rule\( (x, a) \) becomes \( \omega(x, a) = 90 \), and in selecting an action by roulette wheel selection, the selection probability of rule\( (x, a) \) is 90\% while the selection probability of rule\( (x, b) \) is 10\%. PS generally uses a relatively large reward compared to the initial values of the weight to reduce the number of trial and error searches. Therefore, at the early learning phase, the action selection probability of the rule that the reward obtained for the first time is greatly increased, and even if the rule is an inefficient rule, PS may learn the rule. Since EFP is a calculated expected value, enough trial and error searches are necessary for correct penalty avoidance. Therefore, it cannot fully cope with the initial behavior of learning, and it may converge to wrong rules in some cases. Actually in [5], although that, some experiments with EFP were not able to learn a good policy in comparison with other experiments. Since this is a problem caused by reward acquisition, it is difficult to solve by merely introducing penalty into action selection probability.

In order to solve the problem, [10] introduces the safety level \( S(x, a) \) for rule\( (x, a) \) and updates the weight using safety level as reward sharing method based on safety level (RSMSL) is proposed. Here, \( S(x, a) \) is increased/decreased when reward/punishment is given to all rules in the episode at the same time. The share of reward to the weight \( \omega_{rule} \) of Eq. (1) is also increased or decreased depending on \( S_{rule} \) (the safety level of the i-th step rule before the rewarded rule).

First, \( S_{rule} \) is updated as Eqs. (6) and (7):

\[
S_{rule} \leftarrow S_{rule} + \sigma,
\]

\[
S_{rule} \leftarrow S_{rule} - \phi,
\]

where \( \sigma \) is the increase of safety level, \( \phi \) the decrease of safety level. The initial value of \( S(x, a) \) is zero, and if \( S(x, a) \) becomes negative, it is reset to zero. Since these values are increment/decrement values of safety level performed in one update, these can be determined appropriately according to learning speed and accuracy.

Next, using the safety level, the reward sharing formula of Eq. (1) is changed as follows:

\[
\omega_{rule} \leftarrow \omega_{rule} + (1 - e^{-S_{rule}}) \phi_t.
\]

4.2 Introduction of Lowest Success Probability \( X \) and Selection of \( \sigma \) and \( \phi \)

We introduce a lowest success probability \( X \). The probability \( p(R|rule(x, a)) \) that when selecting rule\( (x, a) \), the reward will be obtained at the end of the episode in the future is considered. Further following, we determine \( \sigma \) and \( \phi \) such that \( S(x, a) \) will increase if \( p(R|rule(x, a)) > X \). This means that if the rule\( (x, a) \) is selected, the probability of obtaining a penalty at the end of episode in the future is \( (1 - p(R|rule(x, a))) \). At this time, \( E(x, a)(\)the expected value of acquisition safety level per unit action of the rule\( (x, a) \)\) is calculated as follows:

\[
E(x, a) = \sigma p(R|rule(x, a)) - \phi (1 - p(R|rule(x, a))),
\]

Here, \( E(x, a) \) is positive if \( p(R|rule(x, a)) \) satisfies Eq. (10), and we expect that the safety level will increase.

\[
p(R|rule(x, a)) > \frac{\phi}{\sigma + \phi}.
\]

The objective is to increase the safety level of only rules with \( p(R|rule(x, a)) \) higher than \( X \). Therefore, Eq. (11) is obtained by Eq. (10):

\[
X = \frac{\phi}{\sigma + \phi}.
\]

From Eq. (11), the conditions of \( \sigma \) and \( \phi \) are given by Eq. (12):

\[
\sigma = \frac{\phi(1 - X)}{X}, \quad \phi = \frac{\sigma X}{1 - X}.
\]

From Eq. (12), the parameters \( \sigma \) and \( \phi \) are in a proportional relationship. Therefore, if one is decided, the other is determined automatically. In this paper, we decide \( \phi \) then calculate \( \sigma \) using Eq. (12).

Replacing \( \sigma \) or \( \phi \) by \( X \) makes parameter setting easier for a designer. To use \( X \) means that the designer wishes to learn rules that can earn rewards with the probability more than \( X \).

In this method, only the safety level of the rule\( (x, a) \) such that \( p(R|rule(x, a)) \) larger than \( X \) increases, and the rule with the safety level is rewarded with Eq. (8). Since this method suppresses reward acquisition until the safety level increases enough, it can prevent convergence before learning the penalty avoidance policy, and the problems listed above can be resolved.

4.3 Improvement of RSMSL

In this section, we make two improvements to the RSMSL. The first improvement is to the reward sharing function. Although the safety level increases only for the rule that satisfies the condition of Eq. (10), it can happen that the safety level of the inefficient rule also increases caused by the randomness. In the RSMSL reward sharing formula Eq. (8), the smaller the safety level is, the larger the reward gradient becomes. Therefore, due to accidental results, a large reward may be given to a wrong rule. In order to solve this, we change Eq. (8) to Eq. (13):

\[
\omega_{rule} \leftarrow \omega_{rule} + \min(1, S_{rule}) \phi_t.
\]

For Eq. (13), a small safety level gives the small reward gradient, so the above problem can be solved.

The second improvement is to introduce a discount rate for safety level. In the conventional RSMSL, the period from the current state to punished state was not considered. However,
Since the function when it exceeds this range, it is relocated to the initial position. In fact, it should avoid a short-term penalty more severely than long-term penalty. So to achieve this, we introduce a discount rate to the reduction of safety level. Specifically, for the episode of length $l$ ($\text{rule}_1, \ldots, \text{rule}_r, \rightarrow x_r$), an update formula on the reduction of the safety level of the $i$-th rule $S_{\text{rule}_i}$ is changed as follows:

$$ S_{\text{rule}_i} \leftarrow S_{\text{rule}_i} - \mu \phi^i, $$

where $\mu$ is the safety discount rate ($0 < \mu < 1$). As a result, a shorter term penalty will decrease safety level than a long term, so a short term penalty will be avoided with a priority.

When using Eq. (14), Eq. (9) becomes as follows:

$$ E(x, a) = \sigma p(R|\text{rule}_{(x,a)}) - \mu^s \phi (1 - p(R|\text{rule}_{(x,a)})),$$

where $\text{rule}_{(x,a)}$ is the $n$-th rule from the end of the episode. Therefore, Eq. (10) becomes as follows:

$$ p(R|\text{rule}_{(x,a)}) > \frac{\mu^s \phi}{\sigma + \mu^s \phi}. $$

Since the function $f(x) = \frac{1}{\sigma x^2}$, $(x,a > 0)$ is monotonically increasing with respect to $x$, we have

$$ 0 < \frac{\mu^s \phi}{\sigma + \mu^s \phi} \leq \phi \frac{\sigma}{\sigma + \phi}, $$

where $(0 < \mu^s \leq 1)$ and $(n = 0, 1, \ldots)$. From Eq. (16), if $p(R|\text{rule}_{(x,a)}) > X$, we have $E(x, a) > 0$ for all $0 < \mu^s \leq 1$ and $n = 0, 1, \ldots$. Therefore Eqs. (11) and (12) are still useful to design parameters.

5. Application to Inverted Pendulum Problem

5.1 Inverted Pendulum Problem

We apply the proposed method to the inverted pendulum system shown in Fig. 1. Here, $m$ is the mass of the pendulum $(m = 0.1 \text{kg})$, $M$ is the sum of the mass of the pendulum and the cart $(M = 1.1 \text{kg})$, $2L$ is the length of the pendulum $(2L = 1.0 \text{m})$, and $F$ is the power that acts on the cart. The initial position is the state in which the pendulum is just above and the cart is at the center. As a sensory input, a four-dimensional continuous value of the position of the cart($x$), the velocity of the cart($\dot{x}$), the angle of the pendulum($\theta$), and the angular velocity of the pendulum($\dot{\theta}$) are given. The agents discretize the sensory input into 12, 20, 4, and 20 sections respectively.

The movable range of the cart is $-2.4 \text{m} < x < 2.4 \text{m}$, and when it exceeds this range, it is relocated to the initial position.

The action output $F$ takes $+10N$ or $-10N$, and the noise subject to “uniform distribution between $-0.1$ and $0.1$” is added to $F$. The one sampling time of the simulation is 0.02 seconds. If the cart remains within the movable range and the angle of the inverted pendulum remains within “$-18^\circ < \theta < 18^\circ$” for 10 seconds, the trial is “SUCCESS”. If not so, it is “FAILURE”. Then the cart is rearranged at the initial position. The reward is given everytime the sign of $\theta$ changes twice. The penalty is given when the trial is failure.

5.2 Results and Discussion

One experiment consists of 10,000,000 actions, and such experiments were performed 100 times. In the simulation, we compare the improved RMSMSL (ImpRMSMSL), PS, PSwithEFP (PwE), RMSMSL, and SARSA. The $\epsilon$–greedy method is adopted as action selection of SARSA. The value of the learning rate and $\epsilon$ were linearly attenuated so that the value changes initial 1.0 to 0.05 at 80,000,000 actions. For SARSA, the values of reward and penalty were set to 1.0 and $-1.0$, respectively. In methods based on PS, roulette wheel selection is adopted as action selection, the initial value of the weight was set to 10, and the reward was set to 100. As other parameters, the propagation of failure probability $\eta$ was 0.415, the decrease of safety level $\phi$ was 0.1, the safety discount rate $\mu$ was 0.6, and the lowest success probability $X$ was 0.6. The discount rate $\gamma$ and $\lambda$ were set to 0.95 and 0.80 for SARSA and PS, respectively.

The result is obtained by the evaluation mode using the weight at that time for each 10,000 actions. In the evaluation mode, the action is selected by the $\epsilon$-greedy strategy with $\epsilon$ is set to 0.05. Figure 2 shows the average number of SUCCESS’s per 10,000 actions where the horizontal axis is the number of actions. Figure 3 and Fig. 4 show the trajectory of cart for 10 experiments when inverted by greedy strategy, using the weight after learning. In this experiment, most of FAILURE were cases where the cart went outside the movable range. Therefore, it is considered that the behavior of the cart using the policy acquired by each method shows the performance of the method.

In Fig. 2, two RMSMSL methods give better results than the other methods. PwE has the same performance as PS. In this simulation, since rewards are given frequently, it is considered that convergence of weight is overwhelmingly earlier than convergence of EFP. On the other hand, since the RMSMSL suppresses reward acquisition until the safety level was leaned sufficiently, the penalty avoidance policy could be learned even in such cases. Furthermore, the ImpRMSMSL got better results.
than the original RSMSL, and this shows the importance of avoiding penalty for this problem. From Fig. 3, PS and PwE have not successfully avoided the penalty that the cart goes out of movable range. On the other hand, from Fig. 4, the two RSMSL methods have acquired a policy to prevent cart from leaving the movable range compared with PS and PwE. From this, the RSMSL methods are getting more penalty avoidance policy than the PS methods. In particular, the ImpRSMSL succeeded in inverting for 10 seconds in all of these experiments, and the effectiveness was confirmed. Therefore as described in Section 4, by changing the reward sharing formula and introducing the safety discount rate, more stable learning of penalty avoidance policy becomes possible. From the above, it is clear that the proposed method is effective in the single agent problem.

6. Application to Keepaway Task

6.1 Keepaway Task

We apply the proposed method to the keepaway task [9] and show its effectiveness under multi-agent environment. The keepaway task is a Robocup soccer sub-task, and the keeper team keeps the ball by passing and moving so that the taker team does not rob the ball within the limited area.

In this paper, we consider 3-vs-2 keepaway task learning experiments with three keeper agents that perform ball keeping and two taker agents that rob the ball. Each agent is placed in the initial position when a taker robs the ball or when the ball goes out of the field. When keeper performs a pass action and the pass passes by a friendly keeper, give rewards to the agent who gave out the pass and the agent who took the pass and continue the trial. One trial finishes when the ball is taken by a taker or the ball goes out of the field. When the ball is taken by a taker or goes out of the field, give penalty to the agent who had the ball. Three keeper agents independently learn each other. The initial position of each agent and the field of the keepaway task are shown in Fig. 5.

6.2 Design of State, Action, Reward and Penalty

The state set perceived from the environment consists of the angular relationship between agent that perceives state and Keeper 1, between agent that perceives state and Keeper 2, between agent that perceives state and the nearest taker, and between agent that perceives state and the ball in each four types from $-20^\circ$ to $20^\circ$, from $20^\circ$ to $90^\circ$, from $90^\circ$ to $270^\circ$, and the remaining angle with the angle facing $0^\circ$. Discretize the distance to the ball in three types (up to 30 (cm), from 30 (cm) to 60 (cm), and above 60 (cm)). Show the distance relationship between agents in 6 types as shown in Table 1. There are 18,432
states in total.

Keeper 1 is on the right side and Keeper 2 is on the left side as seen from the agent who perceives the state. In the multi-agent environment, the number of states has a large influence on learning performance. Therefore, in this paper, a position of a Taker considers only one Taker closest to an agent that perceives the state.

A keeper who perceives state uses a roulette wheel selection and selects one action from an action set. There are Stop, Dribble, Kick, Goto Ball, Go Left, and Go Right as the action set. Stop indicates that the agent is stationary. Dribble means that the agent kicks out the ball weakly in one of the three directions (ahead, right, and left). Kick makes the agent kicks out the ball strongly in one of the three directions (ahead, right, and left). Goto Ball makes the agent approaches the ball. Go Left makes the agent turns to the ball, turns left at 45°, and then goes straight ahead. Go Right makes the agent turn to the ball, turn right at 45°, and then go straight ahead. When a keeper who perceives state holds a ball, an action is selected from Stop, Dribble, and Kick, and when he doesn’t hold the ball, an action is selected from Stop, Goto Ball, Go Left, and Go Right. A taker turns and runs straight behind the ball.

In this keepaway task in this paper, we focus on the pass action and strengthen the rule. Therefore, it is regarded as a success when a pass is received by another agent. When a keeper passes a ball successfully to another keeper, both keepers are given rewards and the trial continues. When the ball is taken by a taker or goes out of the field, the keeper who holds the ball is given a penalty, the trial ends, and new trial starts from the initial state.

### 6.3 Results and Discussion

Here, we compare the ImpRSMSL, PS method, PwE method, RSMSL, and SARSA. The reward $R$ was 100, the discount rate $\lambda$ was 0.8, the propagation of failure probability $\eta$ was 0.415, the decrease of safety level $\phi$ was 0.1, and the lowest success probability $X$ was 0.6. One experiment consisted of 100,000 trials, and such experiments were performed 30 times for each method.

The results are shown in Figs. 6 and 7, and Tables 2 and 3. Figure 6 shows the transition of the number of successful passes. Fig. 7 shows the transition of the number of actions per trial, Table 2 shows the maximum value, the minimum value, the mean, and the standard deviation of the average number of successful passes during 30 experiments. Circles are the best result among five methods.

### Table 1 Distance level.

| Flag | Keeper 1 | Keeper 2 | Taker |
|------|----------|----------|-------|
| 1    | short    | middle   | long  |
| 2    | short    | long     | middle|
| 3    | middle   | short    | long  |
| 4    | long     | short    | middle|
| 5    | middle   | long     | short |
| 6    | long     | middle   | short |

Fig. 6 Average number of successful passes.

Fig. 7 Average number of actions per trial.

Table 2 The maximum value, the minimum value, the mean, and the standard deviation of the average number of successful passes during 30 experiments. Circles are the best result among five methods.

| Method      | Maximum value | Minimum value | Mean     | Standard deviation |
|-------------|---------------|---------------|---------|-------------------|
| ImpRSMSL   | 2.705         | 1.867         | 2.394   | 1.406             |
| PS          | 1.892         | 0.419         | 0.932   | 0.000             |
| PwE         | 2.326         | 1.095         | 1.763   | 0.001             |
| RSMSL       | 2.169         | 1.169         | 1.520   | 0.000             |
| SARSA       | 387           | 27.201        | 42.778  | 45.644            |

Table 3 The maximum value, the minimum value, the mean, and the standard deviation of the average number of actions per trial during 30 experiments. Circles are the best result among five methods.

| Method      | Maximum value | Minimum value | Mean     | Standard deviation |
|-------------|---------------|---------------|---------|-------------------|
| ImpRSMSL   | 67.387        | 42.940        | 51.676  | 45.241            |
| PS          | 44.169        | 13.444        | 22.971  | 40.061            |
| PwE         | 52.610        | 27.201        | 42.778  | 45.644            |
| RSMSL       | 4.796         | 8.935         | 4.842   | 3.612             |
| SARSA       | 53.012        | 37.201        | 42.778  | 45.644            |

The ImpRSMSL overwhelms the RSMSL in terms of the minim
imum value and the standard deviation.

As described in Section 4, in the RSMSL reward sharing formula, for small safety level, the gradient of reward becomes large, so due to some accidental results, learning may result in an inefficient rule by acquiring many rewards before learning the penalty avoidance policy. Also, because there was no discount rate in the reduction of the safety level, the avoidance of a penalty with a high priority cannot be done and the penalty avoidance performance becomes deteriorated since the early learning phase depends on the penalty avoidance performance mainly the total learning results become instable.

ImpRSMSL seems to have achieved stable penalty avoidance performance by improving such points and could obtain better results. It can be confirmed from Fig. 7. In Table 3, although the standard deviation is inferior to that of the RSMSL, the maximum value, the minimum value, and the mean overwhelm those of the existing methods. This also shows that the ImpRSMSL can learn penalty avoidance policy more efficiently than the RSMSL and PSWithEFP. From these results, the effectiveness of the ImpRSMSL proposed in this paper was confirmed under multi-agent environment.

7. Conclusion

We made two improvements to the RSMSL and applied the ImpRSMSL to an inverted pendulum problem and a keepaway task. The ImpRSMSL showed the highest performance among SARSA, PS, PSWithEFP, and the RSMSL.

As a result, it was shown that the problem that the early learning phase has a large influence on the final learning results, which is weak point of XoL, was improved.

In the future, in order to apply the proposed method to the real environment, we will introduce deep reinforcement learning [15],[16] to deal with continuous input and apply it to the real soccer robot to discover other real environmental problems.

References

[1] C.J.H. Watkins and P. Dayan: Technical note: Q-learning, Machine Learning, Vol. 8, No. 3-4, pp. 279–292, 1992.
[2] R.S. Sutton and A.G. Barto: Reinforcement Learning: An introduction, A Bradford Book, MIT Press, 1998.
[3] G.A. Rummery and M. Niranjan: On-line Q-learning using connectionist systems, Technical Report CUED/F-INFENG/ TR-166, Department of Engineering, Cambridge University, 1994.
[4] K. Miyazaki and S. Kobayashi: Exploitation-oriented learning PS-r*, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 13, No. 6, pp. 624–630, 2009.
[5] K. Miyazaki, K. Furukawa, and H. Kobayashi: Proposal of PSWithEFP and its evaluation in multi-agent reinforcement learning, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 21, No. 5, pp. 930–938, 2017.
[6] K. Miyazaki and S. Kobayashi: An extention of profit sharing to partially observable Markov decision processes: Proposition of PS-r* and its evaluation, Machine Learning, Vol. 18, No. 5, pp. 286–296, 2003.
[7] K. Miyazaki, S. Arai, and S. Kobayashi: Learning deterministic policies in partially observable Markov decision process, Transactions of the Japanese Society for Artificial Intelligence, Vol. 14, No. 1, pp. 148–156, 1997.
[8] H. Muraoka, K. Miyazaki, and H. Kobayashi: Proposal of a propagation algorithm of the expected failure probability and the effectiveness on multi agent environments, IEEE Transactions on Electronics, Vol. 136, No. 3, pp. 273–281, 2016 (in Japanese).
[9] P. Stone, R.S. Sutton, and G. Kuhlmann: Reinforcement learning toward RoboCup soccer keepaway, Adaptive Behavior, Vol. 13, No. 3, pp. 165–188, 2005.
[10] N. Kodama, K. Miyazaki, and H. Kobayashi: Proposal of reward sharing method based on safety level and verification of its effectiveness in multi-agent environment, Proc. SICE Annual Conference 2017, pp. 1159–1164, 2017.
[11] L. Chrisman: Reinforcement learning with perceptual aliasing: The perceptual distinctions approach, Proc. of the 10th National Conf. on Artificial Intelligence, pp. 183–188, 1992.
[12] K. Miyazaki, M. Yamamura, and S. Kobayashi: A theory of profit sharing in reinforcement learning, Journal of Japanese Society for Artificial Intelligence, Vol. 9, No. 4, pp. 580–587, 1994 (in Japanese).
[13] K. Miyazaki, S. Tsuboi, and S. Kobayashi: Reinforcement learning for penalty avoiding rational policy making, Transactions of the Japanese Society for Artificial Intelligence, Vol. 16, No. 2, pp. 185–191, 2001 (in Japanese).
[14] S. Arai, K. Miyazaki, and S. Kobayashi: Methodology in multi-agent reinforcement learning approaches by Q-learning and profit sharing, Transactions of the Japanese Society for Artificial Intelligence, Vol. 13, No. 4, pp. 609–618, 1998 (in Japanese).
[15] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis: Human-level control through deep reinforcement learning, Nature, Vol. 518, pp. 529–533, 2015.
[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller: Playing atari with deep reinforcement learning, NIPS Deep Learning Workshop 2013, arXiv:1312.5602, 2013.