The Exploitation Reinforcement Learning Method on POMDPs

*Wataru Uemura, Atsushi Ueno, Shoji Tatsumi
Osaka City University, Japan
558–8585 Sugimoto 3–3–138, Sumiyoshi-ki, Osaka city, Japan
Email: wataru@kdel.info.eng.osaka-cu.ac.jp
URL: http://www.kdel.info.eng.osaka-cu.ac.jp/˜ wataru

***Abstract*— In this paper, we discuss the reinforcement learning system on POMDPs. Learning agents receive rewards by a process of trial and error and estimate the episode from the information of the rewards. Profit Sharing distributes the rewards to the rules of an episode and estimates the cumulative amount of each rule. It is known that the conventional distribution can not cumulate correctly on POMDPs. We show that the reason of the incorrect cumulation is caused by the difference reinforcement values at the same state. And we propose the Episode-based Profit Sharing (EPS) that distributes the value corresponding to the episode length at the same state. EPS reinforces the first rule of the episode by r/L^W and the reinforcement values of otherwise rules are 0, where W is the length of the episode, r is received reward, and L is the number of rules that must be suppressed, so the sufficient value of L is the number of rules at the state. EPS can reinforce the rule with the part of the episode that can be reused for POMDPs. We show that EPS suppresses the reinforcement of a detour rule. The experiments show that EPS can get the good performance on both MDPs and POMDPs.**

I. Introduction

Reinforcement learning is one of machine learning methods based on trial and error. An agent who is a learner, finds the solution by the interaction with the environment. The agent can receive the reward if and only if it reaches the goal state, then it estimates from the information of rewards the episode that consists of the selected rules. All you have to do is to make the condition you want the agent to learn, and the agent learn of itself by a process of trial and error. This paper is based on the Profit Sharing which distributes the rewards to the rules of episode and estimates the cumulative amount of each rule.

Profit Sharing distributes the rewards by a reinforcement function. It is fatal for an agent to select rules not to lead to a goal state because the agent learns if and only if it reaches the goal state. So agents must not reinforce the rule to make a loop. The set of rules which make a loop in an episode is called a *loop sequence*. The reinforcement value of a rule into a loop must be less than that of a rule out of a loop in order that an agent should not reinforce the loop sequence. We call a *suppression of rule reinforcement* as a such size relation between reinforcement values of two rules. A reinforcement function should form a geometrical decreasing function because of a loop suppression on Markov decision processes (MDPs)[12]. But on partially observable Markov decision processes (POMDPs) where observations of agents are limited, some rules should not be suppressed. There are such rules if the value of rules are confused, and agents have to reinforce those rules of each state equally. This paper proposes an Episode-based Profit Sharing (EPS) that reinforces rules at each state equally. We show that EPS can use the conventional reinforcement function on MDPs.

Section 2 introduces the reinforcement learning and Profit Sharing. Section 3 describes the POMDPs, and the problem of Profit Sharing. And Section 4 provides EPS as a novel reward distribution method, shows that EPS does not make a loop, and experiments with EPS to show the performance. Last Section 5 discusses the relation between the length of an episode and the received rewards, and compares EPS with the conventional method on POMDPs.

II. Reinforcement Learning

On the model of a reinforcement learning, an agent takes center stage. At the time t, an agent gets a state s_t as a sensory input from the environment. And the agent selects and outputs an action a_t from the set of actions which the agent can do at that state. Selection of the action a_t at the state s_t is known as a rule (s_t, a_t), and a rule selection is known as a policy. An agent moves to the next state s_{t+1} by the action selection, and the agent receives a reward r_t if the state s_{t+1} is a goal state. If the state s_{t+1} is not a goal state, the agent does not receive a reward ($r_t = 0$). The set of rules between a reward and a reward is called as an episode. The agent can estimate the rules from the information of this reward.

A. MDPs

If the state transition probability P_t from a state s_t to a state s_{t+1} is depended on only both the state s_t and the action a_t — $P_t = P(s_{t+1}|s_t, a_t)$, this stochastic process is called as Markov Process. The sequence decision process of Markov process is called as Markov decision processes (MDPs). On MDPs, if the state transition probability is deterministic — all state transition probability is equal to 1, then an agent can calculate the optimum solution when the agent gets all state transition. If the state transition probability is non-deterministic, then an agent has to estimate the expected reward values of rules by many trials.

B. estimation of rules values

There are two kind of reinforcement learning systems [11]. One is the method that estimates the values compared with the other estimating values. This method is known as a bootstrap learning. The other is the way that estimates the values by oneself. This is known as a non-bootstrap learning.

On a bootstrap learning, the estimate values mean how much to receive rewards if the agent selects its rule. Temporal Difference (TD) Learning[10], Q-Learning[13], and Sarsa[7] are the typical bootstrap learning methods. It is shown that Q-Learning can get the optimum solution because the estimate values become nearly equal to the real reward estimate value at the infinite trials on some conditions. On a non-bootstrap learning, the estimate values mean the amount of received rewards. These estimate values are empirical values because they are based on the past-rewards. An agent might learn the local optimum solution because this method does not think about rules the agent has not selected yet. This paper is based on the Profit Sharing that is one of the non-bootstrap learning methods.

C. Profit Sharing

Profit Sharing distributes the rewards to the rules of episode and estimates the cumulative amount of each rule. The distribution function is called as a reinforcement functions $f(x)$, where $x = 0$ means the distribution value for the last rule(s_t, a_t) of the episode because Profit Sharing distributes the reward from the goal state to the start state. Profit Sharing reinforces by one episode because it uses the set of rules in an episode, and distributes the reward r to the $\omega(s_x, a_x)$,

$$\omega(s_x, a_x) \leftarrow \omega(s_x, a_x) + r \times f(x). \tag{1}$$

where the estimate value of rule (s_x, a_x) is noted as $\omega(s_x, a_x)$.

Profit Sharing can not learn without the reward information. So it is fatal for an agent to continue selecting rules not to lead to a goal state. An agent must not reinforce the rules to make a loop. At a state with a loop, the rules into a loop are called *detour rules* and the other rules are called *non-detour rules*[12]. Now a detour rule has to be suppressed of rule reinforcement by a non-detour rule in order to avoid the reinforcement of the detour rule. On MDPs a loop sequence of an episode means that an agent passed through the loop. So the non-detour rule is closer to the goal state than the detour rule. And we should suppress the rule than the closer one to the goal state. A reinforcement function should form a geometrical decreasing function when the rules are suppressed.

III. POMDPs

Some states are not correctly observed by an agent if the observation capability of the agent is limited. For example, when an agent can not get the position in a maze, the agent sees no difference between one state and another state in the maze using the observation of walls around itself. Such perceptual aliasing problem[14] is called as POMDPs.

A. conventional approaches for POMDPs

Memory based method[3] separates the non-aliasing states from the aliasing observation by using the history. This method needs a lot of history in order to separate from aliasing.

Next, we discuss the case that the aliasing observation can not separate. In this case, the different actions are needed at each observation. The deterministic policy can not solve this case. So the Stochastic Gradient Ascent[2] that takes a probability policy is proposed. But the case that the Stochastic Gradient Ascent takes worse policy is noticed[6].

B. problem in Profit Sharing

Profit Sharing uses the estimate value of rules in selecting rules. The estimate value does not be correct value when an aliasing observation confuses the agent observation capability. The conventional reinforcement function of Profit Sharing has this problem. The reinforcement of Profit Sharing is expressed by

$$\omega(o_x, a_x) \leftarrow \omega(o_x, a_x) + r \times f_x, \tag{2}$$

where o_x is the observation from the state s_x. On this equation, there is no problem because Profit Sharing does not use the relationship between observations. Profit Sharing does not correctly estimate rules if a rule(o_x, a_x) = a rule$(o_{x'}, a_{x'})$, a state s_x is not equal to a state $s_{x'}$ and an action a_x is not equal to an action $a_{x'}$. We discuss about this case. The problem in Profit Sharing is the case that an agent confuses between a reinforcement rule and a non-reinforcement one. For example, at Figure 1-(a) an agent has to suppress the rule (s_1, a_2) than the rule (s_1, a_1). At Figure 1-(b) an agent can not separate the state s_1 and the state s_2 from observation o. If the agent suppresses the rule (o, a_2) than the rule (o, a_1) at the state s_1, its suppression will reinforce the rule (o, a_1) to make a loop at the state (s_2). Both the rule (o, a_1) and the rule (o, a_2) at Figure 1-(b) are needed to receive a reward and must not be suppressed. None of needed rules for a reward must be suppressed. On MDPs it is needless for an agent to think of the rule suppression because there is not aliasing state (like Figure 1-(b)). On POMDPs it is need for an agent to think of the rule suppression. All rule for a reward should be reinforced equally. All rule in an episode should be reinforced equally at each state, because an agent can see no difference between Figure 1-(a) and Figure 1-(b) with one episode.

thorem 1: On POMDPs the condition to distribute correctly the reward is

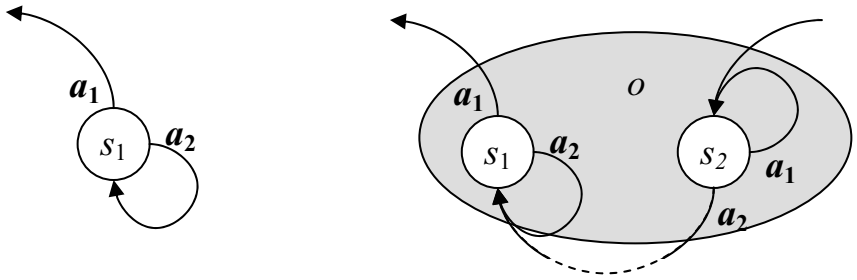
$$f_x = \begin{cases} \alpha_{o_x} & \text{first reinforcement of rule } x \\ 0 & otherwise, \end{cases} \tag{3}$$

where rule x is reinforced by the function f_x. α_{o_x} has to take the constant value at each observation s_x.

IV. EPS: A NOVEL REWARD DISTRIBUTION

This paper proposes the Episode-based Profit Sharing (EPS) that fills the need for the correct distribution on POMDPs. The reinforcement function of EPS is

$$f_x = \begin{cases} 1/L^W & \text{first reinforcement of rule } x \\ 0 & otherwise, \end{cases} \tag{4}$$



(a): A non-aliasing state **(b): Aliasing states.**

Fig. 1. Aliasing states and a non-aliasing state.

where L is the number of non-detour rules at a state, then the number of rule-1 is sufficient for L. We show that EPS can suppress the reinforcement of rules that make a loop.

If the environment has aliasing states, then the reinforcement function to distribute correctly rewards needs *theorem* 1. The perceptual aliasing problem does not affect EPS because EPS can fill the needs from *theorem* 1. So we have no need to think about the affectable of the aliasing states.

We show the two case, one is that only one state makes a loop, and the other case is that multiple states make a loop. Next we propose the sub-episode method that reinforces rules with part of an episode. When part of an episode can be used always, the reinforcement function matches a geometrical decreasing function, that is the conventional function. Last, the experiments show that EPS can get the good performance on both MDPs and POMDPs.

A. a loop consisting of single state

Now we discuss the case that one observation makes a loop. The reinforcement value is written as Δ. The difference of reinforcement values between a non-detour rule and a detour rule is

$$\Delta(o, \text{non-detour rule}) > \Delta(o, \text{detour rule}). \tag{5}$$

Proof is shown in Appendix-B. So EPS can suppress the reinforcement of rules that make a loop in the case of single state.

B. a loop consisting of multiple states

Now we discuss the case that has two or more observations in order to make a loop. The difference of reinforcement values between a non-detour rule and a detour rule is

$$\Delta(o_l, \text{non-detour rule}) > \Delta(o_l, \text{detour rule}). \tag{6}$$

Proof is shown in Appendix-B.EPS can suppress the reinforcement of rules that make a loop in the case of multiple states. So we can show the suppression proof of EPS.

C. same rules in an episode

EPS reinforces the rule only one time even if the episode has the same rule. So the length W of an episode with the same rule is replaced by the length W' of an episode without the same rules. The existence of same rules in an episode means that the episode has a loop. The suppression proof of EPS with the length W' is correct because the value i that is the number of loop is the same between the proof on single state and multiple states. So we use the length W' of an episode that does not have the same rules.

D. using part of an episode

We discuss about the sub-episodes (o_i, a_i), (o_{i+1}, a_{i+1}), ..., (o_t, a_t) $(i = 1, 2, \ldots, t-1)$ which are the parts of an episode (o_1, a_1), (o_2, a_2), ..., (o_t, a_t). An agent can learn from the sub-episodes which start at the time i. To use Sub-episodes has to fill the needs for *theorem* 1 in order to distribute correctly rewards on POMDPs. When an agent can see no difference between the observation o_{k_1} and the observation o_{k_2} affected by perceptual aliasing, there may be some difference between the state s_{k_1} and the state s_{k_2}. In this case, the agent can not use the sub-episode which has the rule (o_k, a_k) is the start rule in order to fill the needs for *theorem* 1 $(k_1 < k \leq k_2)$. That is to say that the agent can use the sub-episode starting at the rule (o_k, a_k) $(k \leq k_1$ or $k_2 < k)$. It is the same when two or many observations are affected by perceptual aliasing. The rules between the observation o_{k_1} and the observation o_{k_2} are defined as rules *on an observation loop*. The flag to mean whether the rule (o_k, a_k) is on an observation loop or not is d_k which is defined as

$$d_k = \begin{cases} 0 & o_k \text{ is on an observation loop.} \\ & (k_1 < k \leq k_2, o_{k_1} = o_{k_2}). \\ 1 & otherwise. \end{cases} \tag{7}$$

An agent can reinforce rules using the length $t - i + 1$ of the sub-episode (o_i, a_i), (o_{i+1}, a_{i+1}), ..., (o_t, a_t). Now the amount f_x of reinforcement for rule (o_x, a_x) is

$$f_x = \sum_{k=x}^{W} \frac{1}{L^k} d_k. \tag{8}$$

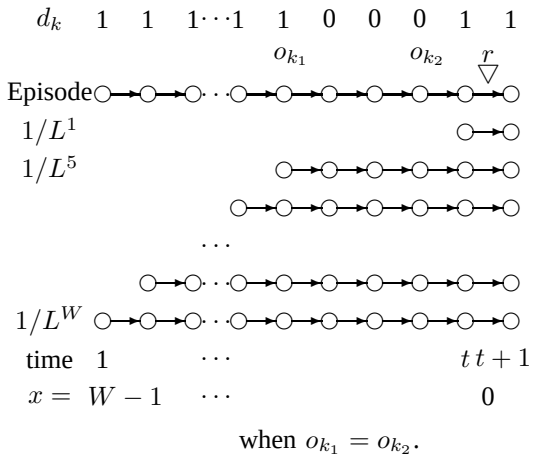


Fig. 2. Reuse sub-episodes

Figure 2 shows this reuse sub-episodes. So the reinforcement function of EPS with sub-episodes is

$$f_x = \begin{cases} \sum_{k=x}^{W} \frac{1}{L^k} d_k & \text{first reinforcement of rule } x \\ 0 & otherwise. \end{cases} \quad (9)$$

The reinforcement function on MDPs that is $d_k = 1$ ($\forall k = 1, 2, \ldots, W$) and has no same rules in an episode is

$$f_x = \sum_{k=x}^{W} \frac{1}{L^k}. \quad (10)$$

Given $W \to \infty$, the reinforcement function f_x becomes the geometrical decreasing function with a common ratio $1/L$. This function matches the conventional function.

E. experiments

An agent can not know how many states affected perceptual aliasing on POMDPs. We prepare two environments, one has aliasing states by half of all and the other has no aliasing states. On first experiment, the environment has aliasing states by half of all (Firgure 3). Agent can select one action from four actions Left, Right, Up, Down at each state. If the direction of the selected action is equal to one of the arrow in the figure, then the agent moves to next state. An agent observes the observation o_1 at the state s_1, state s_2, and state s_3, and the agent should select randomly one action from three actions except for the Left action because the agent has to select the action Right, Down, and Up at each state . At the state s_4, state s_5, and state s_6, the agent has to learn the action moving to next state because the observation is equal to the state. If the agent moves to the goal state, the agent receives the reward 10 and moves to the start state. We do not count the state transition from the state s_{start} and the state s_{goal}, so the optimum policy can move to the goal state in 12 average steps. The performance means received rewards per number of the selected action, and the performance of the optimum policy is $10/12 \simeq 0.833$.

The result is shown at Figure 4. The action selection of Q-Learning is ϵ-greedy which selects the maximum estimate value in 90% probability and random actions 10% probability. The Profit Sharing with the geometric decreasing function as the conventional reinforcement function is written as PS(Decrease). The performance of PS(Decrease) becomes worse because of the influence of the perceptual aliasing. Only EPS can distribute the reward because the perceptual aliasing has no effect on it. On next experiment, the environment has no aliasing states. The environment belongs to the class of MDPs, so we use the maze of Sutton[9] as a famous maze task (Figure 5). The state S is the start state and the state G is the goal state. The state is not changed if the agent selects the action moving to the wall. The reward is 10, and the optimum steps is 14 steps. So the performance of the optimum policy is $10/14 \simeq 0.714$. The result is shown at Figure 6. EPS with the sub-episode is EPS(Sigma). The reinforcement function of EPS(Sigma) is about the same function as the conventional function. We can see little difference between EPS(Sigma) and the conventional PS because EPS(Sigma) does not use the sub-episode if the start rule of sub-episode is on a observation loop. EPS without the sub-episode is EPS(One). EPS(Sigma) learns faster than EPS(One) because EPS(Sigma) has more information of episodes.

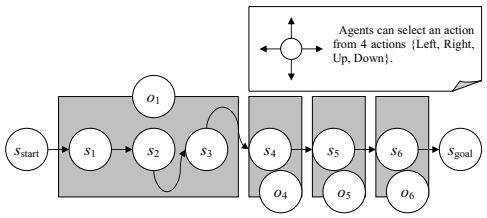


Fig. 3. An experimental environment.

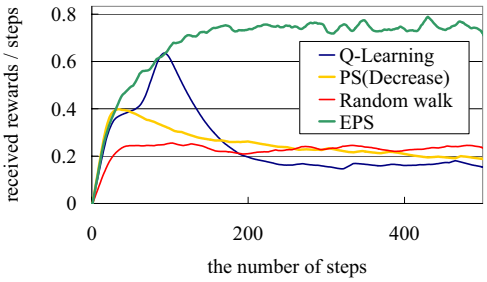


Fig. 4. The result on POMDPs.

V. ABOUT THE REWARD DISTRIBUTION OF EPS

We discuss the difference about using sub-episode or not. And we compare EPS with the conventional approaches to the POMDPs.

A. the relationship between the reward and the length of an episode

We discuss about the relationship between the reward distribution of EPS and the length of an episode. The length of an episode means the number of state transition to the reward. On reinforcement learning system, an agent can not know the

5	11	14	20	26	31	37		G
4	10		19	25	30	36		45
S	9		18	24	29	35		44
3	8		17	23	28	34	40	43
2	7	13	16	22		33	39	42
1	6	12	15	21	27	32	38	41

Fig. 5. The maze of Sutton.

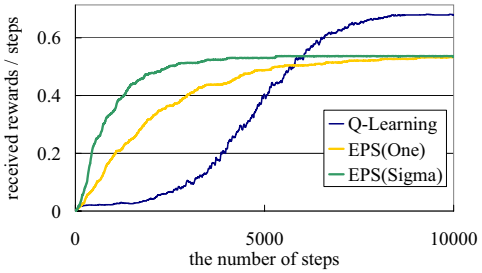


Fig. 6. The result on MDPs.

absolute value of received reward. The larger reward may be on the environment, or this reward may be the largest. The agent can know an episode of the small length is better episode than of the large length with the same rewards. But the agent can not know which episode is better for the different rewards. So there is a trade-off between rewards and the length of an episode. We consider the behavior of EPS to this trade-off. We assume that the received reward r of the action a_1 and the action a_2 are the same. The length of an episode selecting the action a_1 is W, and the length of $a2$ is $W+1$. We discuss about the reinforcement with the sub-episodes. The reinforcement function becomes a geometrical function with the common ratio $1/L$. Now the reinforcement value for the L-th selection of the action a_2 is equal to one for the one selection of the action a_1. So this reinforcement function estimates the values by a factor of L per one different length of episodes. We discuss about the reinforcement without the sub-episodes. Now the reinforcement value for the L-th selection of the action a_2 is equal to one for the one selection of the action a_1. But in contrast to the geometrical function, the number of action selection before the rule reinforcement affects the reinforcement value. For example, the reinforcement value in the case that the number of action selection before the action a_1 is W_1 and W_2 after the action a_1, is equal to one in the case that $W_1 - 1$ before the action a_2 and $W_2 + 1$ after the action a_2. So an agent can reinforce with the sub-episodes faster than without the sub-episodes. The result of experiment 6 shows about this difference.

Next we discuss the case that the sub-episodes can not be given. For example, the agent can not know the sequence of rules in an episode. EPS can distribute correctly rewards in this case. In this case, we usually use multi-agents instead of a single agent. So EPS may apply to the multi-agent systems.

B. compare EPS with the conventional approach on POMDPs

It is needless for the reinforcement of EPS to know the ratio between the aliasing states and the non-aliasing states. All rules selected in an episode are reinforced equally by EPS. At the aliasing state, two or more rules are needed to make an episode, so EPS reinforced these rules equally. At the aliasing state, two or more rules are needed to reach a goal state, so EPS reinforces these rules equally.

The ratio between the aliasing states and the non-aliasing states is needed by the memory based method. EPS does not need such information.

The statistical method is proposed [6][8] that estimates statistically the state is aliasing. This method selects actions randomly at the aliasing state, so it can not learn the better solution. EPS can learn the better solution at the aliasing state. And the statistical method needs the many trial to estimate statistically. EPS does not need so many trial, so we can expect the faster learning.

VI. CONCLUSION

The conventional reinforcement function that distributes the reward to the rules is a geometrical decreasing function in order to suppress a loop on Markov decision processes (MDPs). On POMDPs the geometrical decreasing function can not be used because there are rules which should not be suppressed. We show the condition to distribute correctly rewards for POMDPs is that rules at the same state in an episode are reinforced equally.

We propose a novel reward distribution, EPS, that fills the needs for this condition. EPS reinforces rules in an episode equally, and the reinforcement value of each rule is $1/L^W$, where L is the number of actions at each state and W is the length of the episode. If the sub-episode which is parts of the episode is used, its reinforcement can fill the needs for the correct distribution condition on POMDPs. We show EPS can suppress rules to make a loop with a single state and two or more states. On MDPs the reinforcement function with sub-episodes becomes a geometrical decreasing function that is the conventional function. An agent can not know the ratio aliasing states on POMDPs, so we prepare the two experiments. First experiment has aliasing state in the half of environments. Second experiment has no aliasing state. We can see the correct distribution of rewards both on POMDPs and on MDPs. And we discuss the sub-episode, and show the possibility of EPS for the multi-agent systems. In the future, we must discuss what condition is needed for the multi-agent systems.

REFERENCES

[1] Grefenstette, J.J., Credit assignment in rule discovery systems based on genetic algorithms, Machine Learning, Vol.3, pp.225–245 (1988).
[2] Kimura, H., Yamamura, M. and Kobayashi, S., Reinforcement learning in partially observable Markov decision processes, Journal of Japanese Society for Artificial Intelligence, Vol.11, No.5, pp.761–768 (1996).

[3] McCallum, R.A., Instance-based utile distinctions for reinforcement learning with hidden state, Proc. of the 12th International Conference on Machine Learning, pp.387–395 (1995).
[4] Miyazaki, K., Yamamura, M. and Kobayashi, S., A theory of profit sharing in reinforcement learning, Journal of Japanese Society for Artificial Intelligence, Vol.9, No.4, pp.580–587 (1994).
[5] Miyazaki, K., Arai, S. and Kobayashi, S., Learning deterministic policies in partially observable Markov decision processes, Journal of Japanese Society for Artificial Intelligence, Vol.14, No.1, pp.148–156 (1999).
[6] Miyazaki, K., and Kobayashi, S., An extention of Profit Sharing to partially observable Markov decision processes: proposition of PS-r* and its evaluation, Transactions of the Japanese Society for Artificial Intelligence, Vol.18, No.5, pp.286–296 (2003).
[7] Rummery, G.A. and Niranjan, M., On-line Q-learning using connectionist systems, Technical Report CUED/F-INFENG/TR 166, Engineering Department, Cambridge University (1994).
[8] Saito, K., and Masuda, S., Profit Sharing introducing the judgement of incomplete perception, Transactions of the Japanese Society for Artificial Intelligence, Vol.19, No.5, pp.379–388 (2004).
[9] Sutton, R.S., Integrated architecture for learning, planning, and reacting based on approximating dynamic programing, Proc. of 7th International Conference on Machine Learning, pp.216–224 (1990).
[10] Sutton, R.S., Learning to predict by method of temporal differences, Machine Learning, Vol.4, pp.9–44 (1988).
[11] Sutton, R.S. and Barto, A.G., Reinforcement learning – an introduction –, the MIT Press (1998).
[12] Uemura, W., and Tatsumi, S., About the reinforcement function for Profit Sharing, Transactions of the Japanese Society for Artificial Intelligence, Vol.19, No.4, pp.197–203 (2004).
[13] Watkins, C.J.C.H. and Dayan, P., Technical note:Q-Learning, Machine Learning, Vol.8, pp.279–292 (1992).
[14] Whitehead, S.D. and Balland, D.H., Active perception and reinforcement learning, The Seventh International Conference on Machine Learning (ICML '90), pp.162–169 (1990)

VII. Appendix A: A loop consisting of a single state

We discuss a loop consisting of a single state. The state has two kinds of rules, a detour rule and a non-detour rule. It is more difficult to suppress the reinforcement of a detour rule if the number of detour rules is lower and the number of non-detour rules is higher. So it is the most difficult case to suppress the reinforcement of a detour rule when the state has Lth non-detour rules and one detour rule. There are n_1th action selections before observation o and n_2th action selections after the observation ($N = n_1 + n_2$). The probability to select the detour rule is assumed to be p. And we calculate the expected reinforcement value $Delta$ of each rule.

$$\Delta(o, \text{non-detour rule}) = \frac{1}{L}\sum_{i=0}^{\infty} p^i(1-p)\frac{r}{L^{N+i}} \tag{11}$$

$$\Delta(o, \text{deour rule}) = \sum_{i=0}^{\infty} p^i p(1-p)\frac{r}{L^{N+i+1}} \tag{12}$$

So the difference the expected reinforcement values between a non-detour rule and a detour rule is

$$\begin{aligned}
&\Delta(o, \text{non-detour rule}) - \Delta(o, \text{detour rule}) \\
&= \sum_{i=0}^{\infty} p^i(1-p)\frac{r}{L^{N+i}}\left(\frac{1}{L} - p\frac{1}{L}\right) \\
&= \sum_{i=0}^{\infty} p^i(1-p)\frac{r}{L^{N+i+1}}(1-p) \\
&> 0,
\end{aligned} \tag{13}$$

because $1 < L$ and $0 < p < 1$. So we can show that a loop consisting of a single state can be suppressed. ■

VIII. Appendix B: A loop consisting of two or many states

We discuss a loop consisting of many states which are observed as o_l ($l = 1, 2, \ldots, M$). At the state observed as o_i, the most difficult case to suppress the reinforcement of a detour rule is the same as a single state loop. And it is difficult if the probability to select a detour rule is the same p at each state. Then we calculate the expected reinforcement value $Delta$ of each rule.

$$\Delta(o_l, \text{non-detour rule}) = \frac{1}{L}\sum_{j=0}^{M-1}\sum_{i=0}^{\infty}(p^M)^i p^j(1-p)\frac{r}{L^{N+Mi+j}} \tag{14}$$

$$\Delta(o_l, \text{detour rule}) = \sum_{k=0}^{M-1}\sum_{j=0}^{M-1}\sum_{i=0}^{\infty}(p^M)^i p^j p p^k(1-p) \times\frac{r}{L^{N+Mi+j+k+1}} \tag{15}$$

So the difference the expected reinforcement values between a non-detour rule and a detour rule is

$$\begin{aligned}
&\Delta(o_l, \text{non-detour rule}) - \Delta(o_l, \text{detour rule}) \\
&= \sum_{j=0}^{M-1}\sum_{i=0}^{\infty}(p^M)^i p^j(1-p)\frac{r}{L^{N+Mi+j}} \times\left(\frac{1}{L} - \sum_{k=0}^{M-1}\frac{p^{k+1}}{L^{k+1}}\right) \\
&> \sum_{j=0}^{M-1}\sum_{i=0}^{\infty}(p^M)^i p^j(1-p)\frac{r}{L^{N+Mi+j}} \times\left(\frac{1}{L} - \frac{p}{L}\frac{1}{1-\frac{p}{L}}\right) \\
&= \sum_{j=0}^{M-1}\sum_{i=0}^{\infty}(p^M)^i p^j(1-p)\frac{r}{L^{N+Mi+j}} \times\frac{L-1}{\frac{L}{p}-1} \\
&> 0
\end{aligned} \tag{16}$$

because $1 < L$ and $0 < p < 1$. So we can show that a loop consisting of Mth states can be suppressed. ■