2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

An efficient reinforcement learning method for dynamic environments using short term adjustment

Hidehiro Nakano†, Satoko Takada‡, Shuichi Arai† and Arata Miyauchi†

†Musashi Institute of Technology,
1-28-1, Tamazutsumi, Setagaya-ku, Tokyo, 158-8557 Japan
Email: nakano@ic.cs.musashi-tech.ac.jp
‡Software Engineering Center, TOSHIBA CORPORATION,
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, Kanagawa, 212-8582 Japan

Abstract—This paper proposes a novel reinforcement learning method for dynamic environments. A learning agent estimates changing environments by comparing rule sequence with each action selection probability. If the change is estimated, action selection probabilities are temporarily adjusted. We derive the condition for the amount of adjustment to be flexibly adaptive for dynamic environments. Our method provides better learning performances in various dynamic environments than conventional methods. We present some numerical results for our method applied to dynamic maze problems.

1. Introduction

Reinforcement learning is the framework of learning methods to adapt unknown environments through trial-and-error [1]-[3]. Recently, reinforcement learning for realistic problems has been studied intensively with a great interest. As an approach to such problems, consideration of efficient learning methods for dynamic environments is one of important subjects. However, if conventional learning methods are applied to dynamic environments, the learning performances tend to decrease severely. Because, learning agents can not recognize change of environments, and hold learning results although an environment changes into a different environment. Therefore, we should consider a learning method by which learning agents can adapt changing environments flexibly. Preliminary results along this line can be found in Ref. [4].

This paper proposes a novel reinforcement learning method for dynamic environments. The proposed method is based on Dynamic Profit Sharing (DPS, [2][3]), and operates as almost same as DPS if an environment does not change. A learning agent estimates changing environments by comparing rule sequence with each action selection probability. If the change is estimated, action selection probabilities are temporarily adjusted. We derive the condition for the amount of adjustment to be flexibly adaptive for dynamic environments. Our method provides better learning performances in various dynamic environments than conventional methods. We present some numerical results for our method applied to dynamic maze problems.

2. Profit Sharing

Profit Sharing (PS) is known as one of reinforcement learning methods. A learning agent(s) in an environment selects an action at each state based on action selection probabilities. Then, a pair of the action and state is memorized as a rule. If the agent achieves a goal state, the agent can obtain a reward from the environment. At every episode[1], the obtained reward is shared to each rule. Generally, in PS, the following geometric decreasing function is used as the reinforcement function.

$$f_i = \frac{1}{S} f_{i-1}, \quad i = 1, 2, \cdots, W-1, \tag{1}$$

where W is the length of an episode, and i is the number of steps from a goal state to the ith state. f_i is reward value to the ith rule, and $1/S$ is a decreasing ratio in the reinforcement function. If the parameter S is set to satisfy the rationality theorem, PS can realize learning with rationality [1].

3. Dynamic Profit Sharing

In Eq. (1), f_i decreases exponentially if i increases; the states which are far from a goal state can not be reinforced sufficiently. Therefore, it is difficult to apply the conventional PS to large scale environments. In order to overcome such a problem, we have proposed Dynamic Profit Sharing (DPS) such that the parameter S in Eq. (1) is set to different values in each state, and changes dynamically in the learning [2][3]. In DPS, the following function is used as the reinforcement function.

$$f_i = \frac{1}{S(i)} f_{i-1}, \quad i = 1, 2, \cdots, W-1, \tag{2}$$

where $1/S(i)$ is a decreasing ratio to the ith state. Let $R_{eff}(i)$ (respectively, $R_{ine}(i)$) be the maximum reward of an effective rule (respectively, ineffective rule) in the ith state. Then, $S(i)$ for each state is decided by

$$\frac{P_{eff}(i)}{1 - P_{ine}(i)} > \sum_{j=1}^{W_{ine}(i)} \left(\frac{1}{S(i)} \right)^j, \tag{3}$$

[1] The sequence from an initial state to a goal state is said to an episode.

where $P_{eff}(i)$ and $P_{ine}(i)$ are action selection probabilities of the rules having rewards $R_{eff}(i)$ and $R_{ine}(i)$, respectively. $W_{ine}(i)$ is the number of times at which the ineffective rule is selected continuously [3]. DPS can realize more efficient learning than the conventional PS. In the proposed method presented in the next section, DPS is used as a basic learning module.

4. Proposed Method

4.1. Short Term Adjustment (STA)

The proposed method uses roulette selection for action selection. Let us consider that the number of rules in each state are n. In the proposed method, the action selection probability of the kth rule in an state is decided by

$$P_k = \frac{R_k + \Delta r_k}{\sum_{l=1}^{n} R_l + \Delta r_k}, \qquad (1 \le k \le n) \tag{4}$$

where P_k (respectively, R_k) is action selection probability (respectively, amount of reward) of the kth rule. Δr_k is virtual reward to the kth rule. That is, action selection probability of the kth rule is adjusted by giving virtual reward Δr_k. We refer to this method as Short Term Adjustment (STA). Let $\mathbf{a} = \{a_1, a_2, \cdots, a_n\}$. Let us consider the case where $a_p \in \mathbf{a}$ (respectively, $a_q \in \mathbf{a}$) is an effective rule (respectively, ineffective rule) in a state, and the rule a_q is the target to apply STA. Next, we derive the condition for the amount of adjustment Δr_q to the rule a_q to be flexibly adaptive for dynamic environments.

Case 1. (see Figure 1(a))
If the effective rule a_p (respectively, ineffective rule a_q) changes into the ineffective rule (respectively, effective rule) at an episode, expected value of reward of the rule a_q must be greater than that of the rule a_p in order to efficiently reinforce the new effective rule a_q. Let r_{ine} be the maximum reward given to an ineffective rule at a state, let r_{eff} be the reward given to an effective rule at the state, and let $1/S$ be the decreasing ratio in the reinforcement function. Then, r_{ine} can be written by

$$r_{ine} = r_{eff} \cdot \sum_{l=1}^{\infty} \left(\frac{1}{S}\right)^{l} = \frac{r_{eff}}{S-1}. \tag{5}$$

Let r_p and r_q be the reward given at an episode for the rules a_p and a_q, respectively. Then, from Eqs. (4) and (5), the condition for Δr_q should be

$$\begin{aligned} P_p \cdot r_p &\le P_q \cdot r_q, \\ \frac{R_p}{\sum_l R_l + \Delta r_q} \cdot \frac{r_{eff}}{S-1} &\le \frac{R_q + \Delta r_q}{\sum_l R_l + \Delta r_q} \cdot r_{eff}, \\ \Delta r_q &\ge \frac{R_p}{S-1} - R_q. \end{aligned} \tag{6}$$

Case 2. (see Figure 1(b))
If an effective rule a_p (respectively, ineffective rule a_q) hold to be an effective rule (respectively, ineffective rule), expected value of reward of the rule a_q must be less than that of the rule a_p in order to inhibit reinforcement of the ineffective rule a_q. From Eqs. (4) and (5), the condition for Δr_q should be

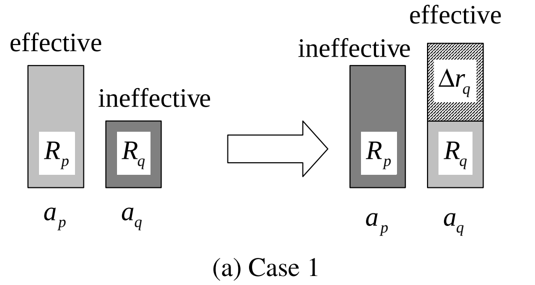
$$\begin{aligned} P_p \cdot r_p &\ge P_q \cdot r_q, \\ \frac{R_p}{\sum_l R_l + \Delta r_q} \cdot r_{eff} &\ge \frac{R_q + \Delta r_q}{\sum_l R_l + \Delta r_q} \cdot \frac{r_{eff}}{S-1}, \\ \Delta r_q &\le (S-1)R_p - R_q. \end{aligned} \tag{7}$$

4.2. Algorithm of proposed method

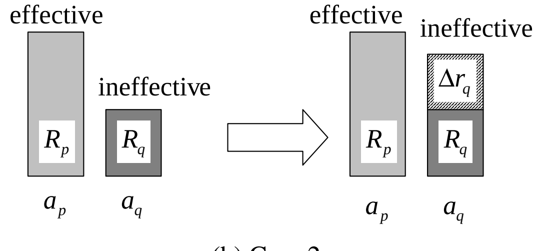
Next, we consider the timing to apply STA. From the property of the reinforcement learning, a last selected rule in a state must be an effective rule. On the other hand, if learning with rationality is carried out, an effective rule in a state can obtain more reward than the other ineffective rules. Therefore, by comparing a last selected rule $a_{last}(i)$ in the ith state with a rule $a_{max}(i)$ having maximum reward value in the state, the change of the environment can be estimated. Such a comparison is executed for each state at the end of every episode. Based on the comparison results, STA is applied in the next episode. Considering the conditions (6) and (7), the amount of adjustment $\Delta r_{last}(i)$ to the rule a_{last} should be

$$\begin{aligned} \frac{R_{max}(i)}{S(i)-1} - R_{last}(i) &\le \Delta r_{last}(i) \\ &\le (S(i)-1)R_{max}(i) - R_{last}(i), \end{aligned} \tag{8}$$

where $R_{max}(i)$ and $R_{last}(i)$ are the amount of reward of the rules $a_{max}(i)$ and $a_{last}(i)$, respectively.



(a) Case 1



(b) Case 2

Figure 1: Change of effective and ineffective rules.

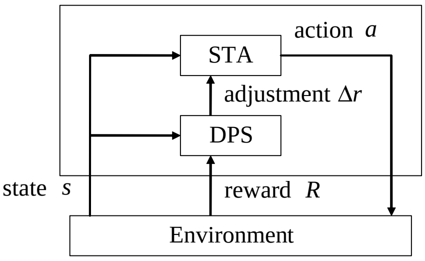


Figure 2: Flow graph of proposed method.

The overall algorithm of the proposed method is as follows (see Figure 2).

1. Select an action based on action selection probabilities
2. Repeat Step 1 until an agent achieves a goal state
3. Learn by using DPS
4. Compare rule $a_{max}(i)$ with $a_{last}(i)$ for each state
   If $a_{max}(i) = a_{last}(i)$, then $\Delta r_{last}(i) = 0$
   If $a_{max}(i) \neq a_{last}(i)$, then apply to STA
5. Return to Step 1

5. Numerical Experiments

We apply Profit Sharing (PS, [1]), Dynamic Profit Sharing (DPS, [2][3]) and Short Term Adjust (STA, proposed method) to simple maze problems as example environments, and compare these learning performances. At each grid of the maze, a learning agent can select 4 kinds of actions, “UP”, “DOWN”, “LEFT” and “RIGHT”. In PS, the value of S in Eq. (1) is set to the number of actions at each state ($S = 4$) in order to satisfy the rationality theorem [1]. In DPS, the value of $S(i)$ in Eq. (2) is decided dynamically based on action selection probability [2][3]. In STA, the amount of adjustment $\Delta r_{last}(i)$ is set to the maximum value which is satisfy the condition (8).

First, we perform experiments for the static maze environment as shown in Figure 3. Minimum steps to the goal state are 16. Number of episodes are 5000 for each method. Figure 4 shows learning curves for the static environments. The learning performances of DPS and STA are almost the same to each other, and are better than that of PS. This means that STA does not decrease learning performance of the original DPS.

Next, we performs experiments for the dynamic maze environment as shown in Figure 5. From the initial grid to the 4th grid, optimal actions in the left and right mazes are the same. Minimum steps to the goal state are 10 for both mazes. Number of episodes are 10000 for each method. We investigate two kinds of cases where (1) the left maze changes into the right maze at once (at 5000 episode), and (2) the left and right mazes changes alternately at three times (at 2500, 5000 and 7500 episodes).

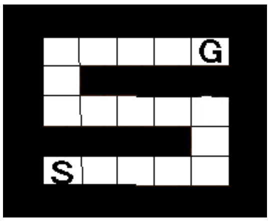


Figure 3: Static maze environment.

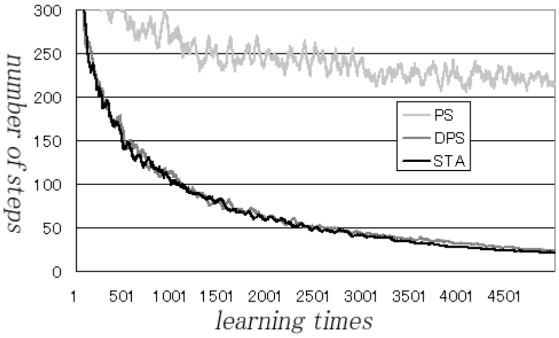


Figure 4: Learning curves for the static environment.

Figure 6 shows learning curves for the environment shown in Figure 5. As shown in Figure 6(a), the learning performances of DPS and STA are almost the same until the environment changes. Just after 5000 episode, the number of steps in DPS increases. Then, the number of steps in STA does not increase significantly and converges quickly to minimum steps. On the other hand, in Figure 6(b), similar characteristics can be found just after 2500 episode. However, just after 5000 and 7500 episodes, the number of steps in DPS and STA do not increase at all. This is reason why learning results for the previous environments are held, and the learning agent uses these results.

Next, we performs experiments for the dynamic maze environment as shown in Figure 7. Minimum steps to the goal state are 16 for both mazes. As compared with the environment shown in Figure 5, minimum steps are longer and the number of common grids between two mazes are less. Number of episodes are 20000 for each method. We investigate two kinds of cases where (1) the left maze changes into the right maze at once (at 10000 episode), and (2) the left and right mazes changes alternately at three times (at 5000, 10000 and 15000 episodes). Figure 8 shows learning curves for the environment shown in Figure 7. The number of steps in PS do not converge and increase at every timing of the change of environment. By contrast, DPS and STA can learn the dynamic environment much more efficiently than PS. In Figure 8(a), convergent values of DPS and STA at 20000 episodes approximate 26 and 16, respectively. In Figure 8(b), convergent values of DPS and STA at 20000 episodes approximate 21 and 16, respectively. These results show that STA provides better learning performance for the dynamic environment than the original DPS.

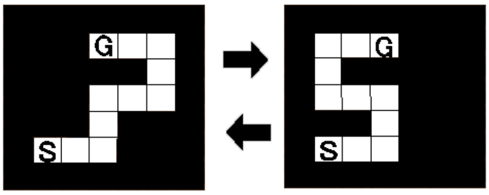
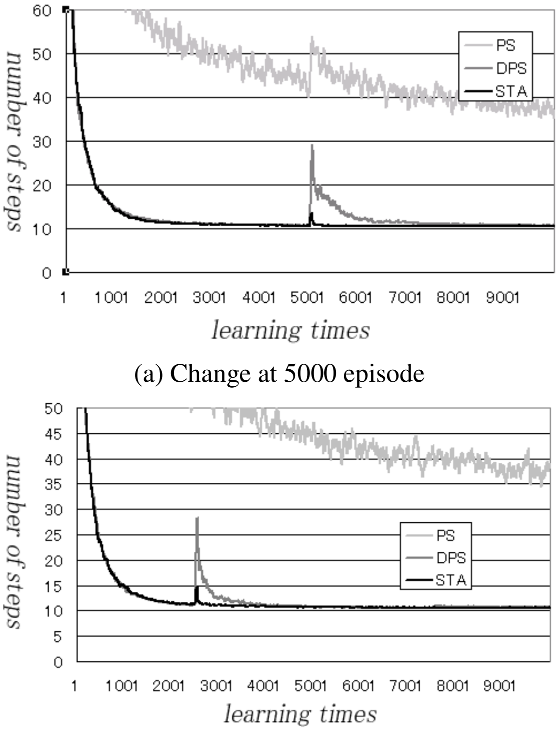


Figure 5: Dynamic maze environment A.



(a) Change at 5000 episode

(b) Change at 2500, 5000 and 7500 episodes

Figure 6: Learning curves for the dynamic environment A.

6. Conclusions

We have proposed a novel reinforcement learning method for dynamic environments. A learning agent estimates changing environments by comparing rule sequence with each action selection probability. As the change in an environment is estimated, action selection probabilities are temporarily adjusted by giving virtual reward to each rule. We have derived the condition for the amount of the virtual reward by considering estimated reward values. We have applied the proposed method to maze problems where shapes of maze change dynamically. The proposed method does not decrease learning performance of the original DPS for static environment, and provides better learning performances than the original DPS for dynamic environments.

Future problems include (1) Application to more frequently changing environment, (2) Analysis of rationality in the proposed method, (3) Consideration for environments including plural effective rules in the same state.

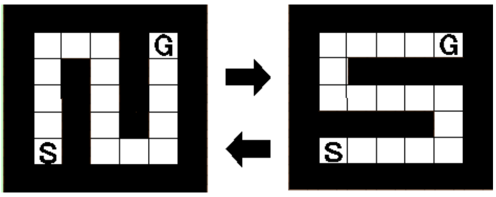
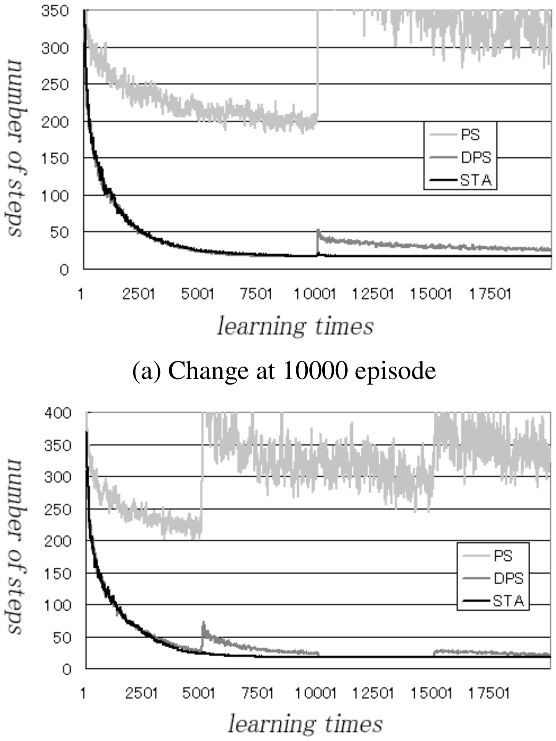


Figure 7: Dynamic environment B.



(a) Change at 10000 episode

(b) Change at 5000, 10000 and 15000 episodes

Figure 8: Learning curves for the dynamic environment B.

References

[1] K. Miyazaki, M. Yamashita and S. Kobayashi, "A theory of profit sharing in reinforcement learning" *J. JSAI*, vol. 9, no. 4, pp. 580-587, 1994. (in Japanese)

[2] S. Takada, Y. Hasegawa, T. Akimoto, A. Miyauchi and S. Arai, "The efficient learning method of reinforcement learning," *Proc. ISITA*, pp. 815-818, 2002.

[3] Y. Hasegawa, S. Takada, H. Nakano, S. Arai and A. Miyauchi, "Reinforcement learning method using dynamic reinforcement function based on action select probability," *IEICE Trans. Funds.* (submitted, in Japanese)

[4] S. Takada, H. Nakano, S. Arai and A. Miyauchi, "Reinforcement learning method considered dynamic environment," *Proc. ISITA*, pp. 83-86, 2004.