SICE Annual Conference 2008
August 20-22, 2008, The University Electro-Communications, Japan

Swarm Reinforcement Learning Algorithms Based on Sarsa Method

Hitoshi Iima[1] and Yasuaki Kuroe[2]

Department of Information Science, Kyoto Institute of Technology, Kyoto, Japan
[1](Tel : +81-75-724-7467; E-mail: iima@kit.ac.jp)
[2](Tel : +81-75-724-7445; E-mail: kuroe@kit.ac.jp)

Abstract: We recently proposed swarm reinforcement learning algorithms in which multiple agents are prepared and they all learn concurrently with two learning strategies: individual learning and learning through exchanging information. In the proposed swarm reinforcement learning algorithms, Q-learning method was used for the individual learning. However, there have been proposed several reinforcement learning methods, and it is required to investigate how to apply these methods to swarm reinforcement learning algorithms and evaluate their performance. In this paper, we propose swarm reinforcement learning algorithms based on Sarsa method in order to obtain an optimal policy rapidly for problems with negative large rewards. The proposed algorithm is applied to a shortest path problem, and its performance is examined through numerical experiments.

Keywords: Swarm reinforcement learning, Sarsa, Swarm intelligence

1. INTRODUCTION

In ordinary reinforcement learning algorithms [1], basically a single agent learns to achieve a goal through many episodes. If a learning problem is complicated, it may take much computation time to acquire the optimal policy. Meanwhile, for optimization problems, population-based methods such as genetic algorithms and particle swarm optimization (PSO) [2] have been recognized that by using multiple candidate solutions they are able to find rapidly the global optimal solution for multi-modal functions with wide solution space. It is expected that by introducing the concept of population-based methods into reinforcement learning algorithms, optimal policies can be found rapidly.

We recently proposed reinforcement learning algorithms using multiple agents [3], and call them *swarm reinforcement learning algorithms*. In the algorithms, multiple agents are prepared and they all learn concurrently with two learning strategies: individual learning and learning through exchanging information. In the former strategy, each agent learns individually by using a usual reinforcement learning method. In the latter strategy, the agents exchange information among them regularly during the individual learning and they learn based on the exchanged information.

Learning methods called multi-agent reinforcement learning have been proposed [4]. Basically, the aim of the multi-agent reinforcement learning methods is to acquire optimal policies in tasks achieved by cooperation or competition among multiple agents. In the methods each agent regards information of other agents as a part of environments. The concept of the swarm reinforcement learning methods is different from that of the multi-agent reinforcement learning methods. Basically, the swarm reinforcement learning methods could treat both tasks achieved by a single agent and achieved by cooperation or competition among multiple agents. In the methods, multiple agents are prepared in order to learn in shorter learning time. In this paper we consider problems of single-agent tasks.

In the previously proposed swarm reinforcement learning algorithms, Q-learning method was used for the individual learning. However, there have been proposed reinforcement learning methods, and it is required to investigate how to apply these methods to swarm reinforcement learning algorithms and evaluate the performance of the swarm reinforcement learning algorithms using the methods. Sarsa method, as well as Q-learning method, is one of typical existing reinforcement learning methods. Sarsa method has been recognized to be more effective than Q-learning method especially for problems with negative large rewards. In this paper, we propose swarm reinforcement learning algorithms based on Sarsa method in order to obtain an optimal policy rapidly for such problems. The proposed algorithm is applied to a shortest path problem with negative large rewards, and its performance is examined through numerical experiments.

2. SWARM REINFORCEMENT LEARNING METHOD

This section describes the swarm reinforcement learning method [3].

2.1 Basic Framework

The swarm reinforcement learning method is motivated by population-based methods in optimization problems and its basic framework is as follows. Multiple agents are prepared and they all learn concurrently with two learning strategies: individual learning and learning through exchanging information. In the former strategy, each agent learns individually by using a usual reinforcement learning method. In this section, assuming that Q-learning method is used for the individual learning, the swarm reinforcement learning method is explained. In the latter strategy, the agents exchange information among them regularly during the individual learning and they learn based on the exchanged information.

In ordinary reinforcement learning algorithms, the

agent often takes a useless action bringing a small reward, which makes learning time longer. On the other hand, in the swarm reinforcement learning method, since the multiple agents are prepared, some of these agents could take useful actions bringing a larger reward. If each agent learns based on information of the agents who take the useful actions by the information exchange, agents can acquire the optimal policy in a shorter learning time.

In the learning through exchanging the information, each agent updates its own Q-values by referring to the Q-values which are evaluated to be more useful and superior to those of the other agents for finding rapidly the optimal Q-values. For this purpose, the Q-values of each agent are evaluated for each episode after the individual learning. To evaluate the Q-values is not necessary for usual reinforcement learning algorithms. On the other hand, in the swarm reinforcement learning method, it is essential to introduce an appropriate criterion to evaluate the Q-values, which is described in the next subsection.

2.2 Evaluation of Q-values

As stated above, in the swarm reinforcement learning method, it is necessary to appropriately evaluate the Q-values of each agent at the end of each episode. Since the objective of reinforcement learning is to maximize the return, it seems to be most suitable to evaluate the Q-values by directly calculating the return. However, this is not practical, because this calculation requires many simulations. Instead, we proposed a method of evaluating the Q-values such that the evaluation results are close approximations of the returns.

Basically, in the proposed method, the Q-values of each agent are evaluated by the sum of rewards which the agent obtains during the previous one episode. However, to simply sum up them causes the following problem. Since a Q-value is updated whenever an agent takes an action in the individual learning, Q-values at and shortly after the beginning of the episode are rather different from those at the end of the episode. Even if a new episode begins with the Q-values at the end of the previous episode, the same actions as the previous episode are not necessarily taken and the same rewards are not necessarily obtained. The rewards obtained by using the Q-values at and shortly after the beginning are considered to have only a little relation with the Q-values at the end. Thus, such rewards are discounted and the discounted results are summed up. Therefore, we define the evaluated value E for the Q-values at the end of each episode as

$$E = \sum_{k=1}^{N} d^{N-k} r_k \tag{1}$$

where N is the number of actions in the episode, r_k is the reward for the k-th action, and $d(< 1)$ is the discount parameter. The definition (1) could bring that the larger the evaluated value E for Q-values is, the superior the Q-values are.

2.3 Update Scheme of Q-value

We proposed the following three update schemes called BEST, AVE and PSM in order to update Q-values through exchanging the information among the agents.

· **BEST** (Best state-action value method)

If the superior Q-values are copied to those for each of the other agents, each agent can improve its Q-values in future episodes. According to this idea, the Q-value $Q_i(s,a)$ of agent i for action a in state s is updated by

$$Q_i(s,a) \leftarrow Q^{\mathrm{best}}(s,a) \tag{2}$$

where $Q^{\mathrm{best}}(s,a)$ is the Q-value which is evaluated superior to that of any other agent, and is called the best Q-value.

· **AVE** (Average state-action value method)

In BEST, the Q-values which are not best are discarded, and replaced with the best Q-values. However, they are not necessarily worthless, because they are updated by agents through the individual learning. Moreover, the diversity of Q-values may be lost, because the Q-values of all the agents for each state-action become the same Q-value, $Q^{\mathrm{best}}(s,a)$. Consequently, optimal Q-values may not be found.

In order to remove these drawbacks of BEST, in AVE each agent updates its Q-values by averaging its current Q-value and the best Q-value for each state-action. The update equation is given by

$$Q_i(s,a) \leftarrow \frac{Q^{\mathrm{best}}(s,a) + Q_i(s,a)}{2}. \tag{3}$$

· **PSM** (Particle swarm method)

Population-based methods are often used for finding a global optimal solution in optimization. Thus, an optimal policy could be also found rapidly by applying an update procedure used in the population-based methods to the reinforcement learning problem. In PSM, PSO [2] is adopted for an update scheme of Q-value. The personal best of each agent i and the global best are determined by evaluating E for the Q-values and are stored. Let the personal best P_i be the best Q-values found by the agent i so far, and the global best G be the best Q-values found by all the agents so far. It is noted that the global best G is not the best Q-value Q^{best}, which is the best among the Q-values of all the agents at only the previous episode. Each agent updates its Q-values by using the global best and its personal best. Following the update procedures of PSO, we give the update equations as:

$$\begin{aligned} V_i(s,a) &\leftarrow WV_i(s,a) \\ &\quad +C_1R_1(P_i(s,a) - Q_i(s,a)) \\ &\quad +C_2R_2(G(s,a) - Q_i(s,a)) \qquad (4) \\ Q_i(s,a) &\leftarrow Q_i(s,a) + V_i(s,a) \qquad (5) \end{aligned}$$

where $V_i(s,a)$ is a so-called velocity, W, C_1 and C_2 are weight parameters, and R_1 and R_2 are uniform random numbers in the range from 0 to 1.

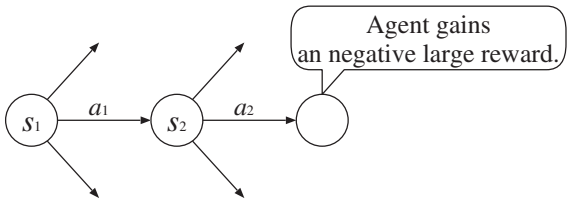


Fig. 1 Environment in which Sarsa method is effective

3. THE PROPOSED ALGORITHMS

This section proposes swarm reinforcement learning algorithms in which Sarsa method is used for the individual learning. In Sarsa method, the Q-value $Q(s^*, a^*)$ for action a^* in state s^* is updated by

$$Q(s^*, a^*) \leftarrow Q(s^*, a^*) + \alpha \left[r + \gamma Q(s_{\rm n}, a_{\rm n}) - Q(s^*, a^*) \right] \quad (6)$$

where

α : step-size parameter,
r : reward,
γ : discounted-rate parameter,
$s_{\rm n}$: next state,
$a_{\rm n}$: action taken in $s_{\rm n}$.

Sarsa method is characterized by using the next action $a_{\rm n}$ in Eq. (6). On the other hand, in Q-learning method the Q-value is updated by using the maximum Q-value $\max_a Q(s_{\rm n}, a)$ instead of $Q(s_{\rm n}, a_{\rm n})$ in Eq. (6).

Sarsa method has been recognized to be more effective than Q-learning method especially for problems with negative large rewards. Let us explain differences between Sarsa method and Q-learning method by using a simple example. Let an agent in state s_1 perceive the next state s_2 by taking action a_1, and gain a negative large reward by taking the next action a_2 in s_2, as shown in Fig. 1. In this figure, a circle and an arrow mean a state and an action, respectively. In both Sarsa method and Q-learning method, because $Q(s_2, a_2)$ becomes a negative large value by the negative large reward, the agent learns that a_2 is a bad selection. Moreover, in Sarsa method, because $Q(s_1, a_1)$ also tends to become a negative large value by Eq. (6) in future episodes, the agent can learn that a_1 is not a good selection. Consequently, it can avoid such actions and acquire a better policy rapidly. On the other hand, in Q-learning method, because other Q-values in s_2 are generally larger than $Q(s_2, a_2)$, $Q(s_1, a_1)$ is updated without using $Q(s_2, a_2)$ in the future episodes. Therefore, the agent can not learn that a_1 is not a good selection.

We propose the swarm reinforcement learning algorithms based on Sarsa method in order to obtain an optimal policy rapidly for problems with negative large rewards. In the algorithms, their basic framework is the same as the swarm reinforcement learning algorithms explained in Section 2, and Sarsa method is used for the individual learning. The flow of the algorithms is as follows.

Y : number of episodes for which the individual Sarsa method is performed between the information exchange among the agents.
I : number of agents.
T : total number of episodes.

Step 1 Set the initial values of $Q_i(s, a)$ $(\forall i, s, a)$. Set $t \leftarrow 0$. The variable t means the number of episodes.

Step 2 Set $y \leftarrow 1$. The variable y means the number of episodes after the information exchange.

Step 3 Update $Q_i(s, a)$ by applying the following Sarsa method for one episode for each agent i.

- Step 3-1 Set the initial state s^*, and set $k \leftarrow 1$. The variable k means the number of actions.
- Step 3-2 Take an action a^* according to $Q_i(s^*, a)$ by using the ε-greedy method. As a result, a reward r and the next state $s_{\rm n}$ are obtained. In the ε-greedy method, an action is taken randomly with probability ε, and the action whose value is maximum $(\max_a Q_i(s^*, a))$ is taken with probability $1 - \varepsilon$.
- Step 3-3 Select the next action $a_{\rm n}$ according to $Q_i(s_{\rm n}, a)$ by using the ε-greedy method. This action $a_{\rm n}$ will be taken in Step 3-6.
- Step 3-4 Update $Q_i(s^*, a^*)$ by Eq. (6).
- Step 3-5 If a terminate condition of episode is satisfied, go to Step 4. Otherwise, set $k \leftarrow k + 1$.
- Step 3-6 Set $s^* \leftarrow s_{\rm n}$ and $a^* \leftarrow a_{\rm n}$. Take the action a^*, and a reward r and the next state $s_{\rm n}$ are obtained. Return to Step 3-3.

Step 4 Calculate the evaluated value E for $Q_i(s, a)$ of each agent by Eq. (1).

Step 5 If $y < Y$, set $y \leftarrow y + 1$ and return to Step 3.

Step 6 Update $Q_i(s, a)$ of each agent by applying an information exchange scheme such as BEST, AVE and PSM.

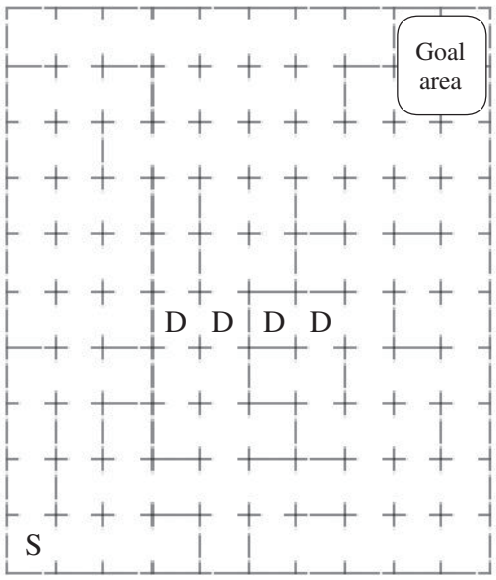
Step 7 Set $t \leftarrow t + IY$. If $t \geq T$, terminate this algorithm. Otherwise, return to Step 2.

4. NUMERICAL EXPERIMENTS

The effectiveness of the proposed algorithm is examined by applying it to a shortest path problem with negative large rewards. A usual shortest path problem can easily be solved by ordinary reinforcement learning algorithms, and we make it harder to solve. We set up a shortest path problem with the goal position being changing randomly within a specified range.

4.1 Shortest Path Problem for Experiments

The shortest path problem is that an agent finds the shortest path from the start cell to the goal cell in an n by n grid world. In this grid world, we let the coordinates at the bottom left be (1,1), and those at the top right be (n,n). While the start cell is (1,1) and fixed, the goal cell



S : Start cell D : Dangerous cell
Goal cell is determined randomly from the goal area.

Fig. 2 Example of grid world for n=10

is changing within a range and is determined at random.

The agent perceives its own coordinates (x,y), and has four possible actions to take: moving up, moving down, moving left and moving right, that is to say, it has actions to move into $(x, y+1)$, $(x, y-1)$, $(x-1, y)$ and $(x+1, y)$. Some cells have walls at the boundaries with their adjacent cells, and the movement of the agent is blocked by the walls and the edge of the grid world. In addition, there are several dangerous cells. If agent reaches one of these cells, it can not move from the cell and can not reach the goal cell.

Figure 2 shows an example of the grid world for n=10. In our experiments, we let the size n=40, and the x and y coordinates of the goal cell are determined randomly in the range from 35 to 40, respectively. Hence, they are determined randomly from 36 cells. In this grid world there are dangerous cells at (x^*,20) ($10 \leq x^* \leq 30$).

When an agent reaches the goal cell, it gains the reward $+100$. When it reaches a dangerous cell, it gains -100 for learning to avoid the dangerous cell. Otherwise, it gains -1. By using this rule of rewarding, the evaluated value E becomes larger when the number of actions to reach the goal cell is smaller. In addition, when the agent reaches a dangerous cell, E becomes smaller. The value of discount-rate parameter γ is set to be 0.999.

4.2 Experimental Set up

In this paper we consider problems of single-agent tasks, and the problem explained in the previous subsection is a single-agent task. We apply the proposed swarm reinforcement learning algorithm using Sarsa method (called Swarm-Sarsa) to the problem. Swarm-Sarsa is compared with the swarm reinforcement learning algorithm using Q-learning (Swarm-Q), one-agent Sarsa (1agent-Sarsa) and one-agent Q-learning (1agent-Q) in order to show the effectiveness of Swarm-Sarsa.

In 1agent-Sarsa and 1agent-Q, the following values are used for their parameters:

step-size parameter : $\alpha = 0.1$,
probability of random action : $\varepsilon = 0.2$,
total number of episodes : $T = 12000$.

All the initial values of $Q_i(s, a)$ are set to be zero. The coordinates of the goal cell are changed randomly at every episode. Each episode finishes when an agent reaches the goal cell or a dangerous cell, or it takes 10000 actions.

In Swarm-Sarsa and Swarm-Q, the following values are used for their parameters in addition to the values used in 1agent-Sarsa and 1agent-Q:

number of agents : $I = 4$,
number of episodes for which the individual Q-learning is performed between the information exchange : $Y = 1$,
discount parameter in (1) : $d = 0.999$.

In this experiment, BEST is used for the learning through the information exchange. T is not the number of episodes for each agent, and is the sum of numbers of episodes for all the agents.

Each algorithm is performed thirty times with various random seeds, and their results are averaged.

4.3 Results and Discussion

Before we apply the four algorithms to the problem in which the goal cell is changing, we apply them to the simple problem in which the goal cell is fixed at (40,40) in order to show that Sarsa method is more effective than Q-learning method for problems with negative large rewards. Figure 3 shows the variation of number of actions through the learning phase obtained by each algorithm. The x-axis in this figure represents the number of episodes of all the agents. The y-axis represents the average number of actions to reach the goal where the value at the t-th episode is obtained by averaging the numbers of actions from $(t-99)$-th to t-th episodes. If an agent reaches a dangerous cell, the number of actions is counted as the maximum (10000). It is confirmed from Fig. 3 that 1agent-Sarsa is better than 1agent-Q. Since it is easy to solve this problem, 1agent-Sarsa can find a better policy rapidly.

Figure 4 shows the variation of number of actions obtained for the complicated problem in which the goal cell is changing. As seen from this figure, 1agent-Sarsa and 1agent-Q do not converge and can not find good policies. On the other hand, Swarm-Sarsa and Swarm-Q converge and can find better policies. Therefore, learning with the multiple agents works better. In particular, Swarm-Sarsa can find a good policy more rapidly than Swarm-Q. It is concluded that Swarm-Sarsa, the proposed algorithm is most effective for this problem with negative large rewards.

5. CONCLUSION

In swarm reinforcement learning algorithms, multiple agents learn individually by using a usual reinforcement

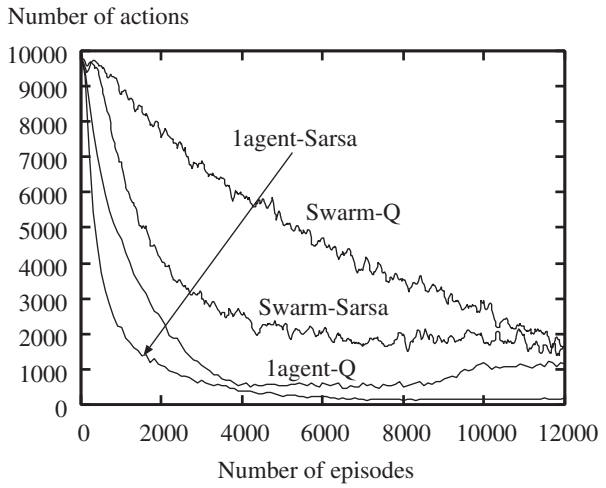


Fig. 3 Variation of number of actions for the simple problem in which the goal cell is fixed

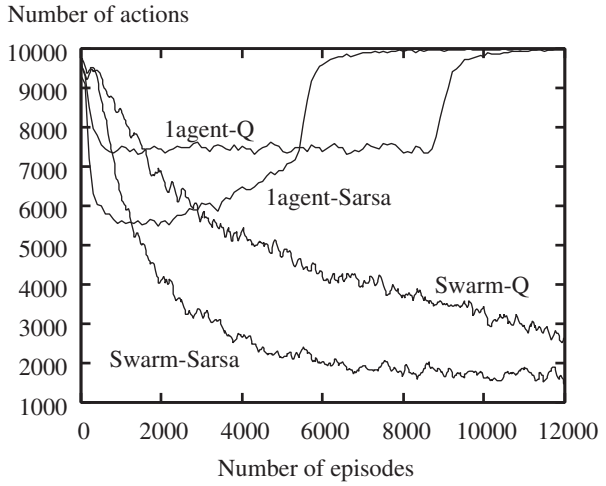


Fig. 4 Variation of number of actions for the complicated problem in which the goal cell is changing

learning method and also learn through the information exchange among them. The performance of the swarm reinforcement learning algorithms depends on that of the usual reinforcement learning method used in the individual learning.

This paper has proposed swarm reinforcement learning algorithms based on Sarsa method. It is known that Sarsa method is good at problems with negative large rewards compared with Q-learning method. It is confirmed from the experimental results that the proposed reinforcement learning algorithm using Sarsa method outperforms other reinforcement learning algorithms for problems with negative large rewards.

There have so far been proposed several usual reinforcement learning methods besides Sarsa method and Q-learning method, and each of them is good at some problems and is not at others. Given a problem, it is expected that more effective swarm reinforcement learning algorithms for it can be developed by introducing a usual reinforcement learning method which is good at it.

REFERENCES

[1] R.S. Sutton and A.G. Barto: *Reinforcement Learning*, MIT Press, 1998.

[2] J. Kennedy and R.C. Eberhart: *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.

[3] H. Iima and Y. Kuroe: "Reinforcement Learning through Interaction among Multiple Agent", *SICE-ICASE International Joint Conference 2006 CD-ROM*, pp.2457–2462, 2006.

[4] L. Busoniu, R. Babuska and B. De Schutter, "A Comprehensive Survey of Multiagent Reinforcement Learning", *IEEE Transactions on Systems, Man, and Cybernetics*, Part C, Vol.38, No.2, pp.156–172, 2008.