Swarm Reinforcement Learning Algorithm Based on Particle Swarm Optimization Whose Personal Bests Have Lifespans

Keywords: reinforcement learning, particle swarm optimization, swarm intelligence

Abstract

A swarm reinforcement learning algorithm based on particle swarm optimization was proposed in order to find optimal policies rapidly. In this algorithm, multiple agents are prepared, and they learn not only by individual learning but also by an update procedure of particle swarm optimization. In this update procedure, state-action values are updated based on the personal best and the global best which are found by the agents so far. It becomes a problem that an agent could possess an overvalued personal best, which brings inferior learning performance. In order not to update the state-action values based on the overvalued personal best, this paper presents a swarm reinforcement learning algorithm based on particle swarm optimization in which the personal best of each agent has a lifespan. In this algorithm, if the personal best is not replaced after experiencing a specified number of episodes which correspond to the lifespan, it comes the end of its life and is forced to be replaced.

1. Introduction

In ordinary reinforcement learning algorithms (Sutton & Barto, 1998), a single agent learns to achieve a goal through experiencing 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) (Kennedy & Eberhart, 2001) have been recognized that they are able to find rapidly global optimal solutions 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.

In order to find optimal policies rapidly, swarm reinforcement learning algorithms have been proposed (Iima & Kuroe, 2006; Iima & Kuroe, 2008). 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 algorithm such as Q-learning (Watkins & Dayan, 1992). In the latter strategy, the agents exchange their information among them and learn based on the exchanged information every after repeating the individual learning a certain number of times.

Learning methods called multi-agent reinforcement learning have been proposed (Busoniu et al., 2008). 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 other agents as a part of environments. The concept and objective of the swarm reinforcement learning algorithms are different from those of the multi-agent reinforcement learning methods. Basically, the swarm reinforcement learning algorithms 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 discuss swarm reinforcement learning algorithms for problems of single-agent tasks.

The performance of the swarm reinforcement learning algorithms highly depends on a method of exchanging the information, which should be appropriately designed. Iima and Kuroe (2008) propose the swarm reinforcement learning algorithm based on PSO (SRL-PSO) in which the update equations of PSO are used for exchanging the information. In this algorithm, Q-values (state-action values) of each agent are evaluated by a certain way every episode and the superior

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Q-values among the Q-values found by the agent so far are stored as the personal best. Moreover, the superior Q-values among the Q-values found by all the agents so for are stored as the global best. In learning through exchanging the information, the Q-values of each agent are updated based on its own personal best and the global best. In SRL-PSO, Q-values are not necessarily evaluated correctly and are sometimes overvalued. When such overvalued Q-values become the personal best, it is hard for the agent to find Q-values which are superior to the overvalued personal best. Since the Q-values of the agent continue to be updated based on the overvalued personal best until superior Q-values are found, SRL-PSO may not be able to find optimal policies.

In this paper, we propose SRL-PSO in which the personal best has a lifespan (SRL-PSO-L). In this algorithm, if the personal best of an agent is not replaced after experiencing a certain number of episodes corresponding to a specified lifespan, it comes the end of its life and is forced to be replaced with the Q-values of the agent. Even if the agent has overvalued Q-values, the overvalued personal best is replaced, which resolves this problem of SRL-PSO. SRL-PSO-L is applied to a shortest path problem, and its performance is examined through numerical experiments.

The rest of this paper is organized as follows. Section II outlines the swarm reinforcement learning method (Iima & Kuroe, 2006; Iima & Kuroe, 2008). Next, Section III proposes SRL-PSO-L, and Section IV shows its experimental results. Finally, Section V concludes the paper.

2. Swarm Reinforcement Learning Method

2.1. Basic Framework

The swarm reinforcement learning method (Iima & Kuroe, 2006; Iima & Kuroe, 2008) 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 algorithm. In the latter strategy, the agents exchange their information among them and learn based on the exchanged information every after repeating the individual learning a certain number of times.

In ordinary reinforcement learning algorithms with a single agent, it 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. Because each agent learns based on information exchanged with the agents who take the useful actions, it is expected that agents can acquire the optimal policy in a shorter learning time.

In the individual learning, Q-learning (Watkins & Dayan, 1992) which is a typical reinforcement learning algorithm is used in this paper. It is performed in the standard way as follows. When agent i takes action a in state s, the Q-value $Q_i(s,a)$ of the agent i for the state-action is updated by

$$Q_i(s,a) \leftarrow Q_i(s,a) + \alpha \{ r + \gamma \max_{a^* \in A(s_\mathrm{n})} Q_i(s_\mathrm{n}, a^*) - Q_i(s,a) \} \quad (1)$$

where
α : learning-rate parameter,
r : reward,
γ : discount-rate parameter,
s_n : next state,
$A(s)$: set of allowable actions in state s.

The ε-greedy method is used as an action selection method for all the agents. In this method, each agent takes an action randomly with probability ε, and the action whose value is maximum ($\max_{a^*} Q(s,a^*)$) with probability $1-\varepsilon$.

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 Q-learning is performed. In ordinary Q-learning with a single agent, Q-values are not necessary to be evaluated. By contrast, in the swarm reinforcement learning method, it is essential to introduce an appropriate criterion to evaluate the Q-values. By the evaluations, superior Q-values can be selected and exchanged among agents, and Q-values of each agent can be updated by using the superior Q-values.

The flow of the swarm reinforcement learning method is as follows.

Step 1 Each of multiple agents updates its own Q-values by performing in the individual Q-learning (1) for a specified number of episodes.
Step 2 The Q-values of each agent are evaluated by an appropriate method, which is described in Subsection 2.2.

Step 3 Based on the evaluation of Step 2, superior Q-values are selected and exchanged among agents. The Q-values of each agent are updated by using the superior Q-values. One method to realize this is an information exchange method based on PSO, which is described in Subsection 2.3.

Step 4 If the termination condition is satisfied, terminate this algorithm. Otherwise, return to Step 1.

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, it is not practical, because this calculation requires many simulations. Instead, the Q-values are evaluated in such a way that the evaluation results are close approximations of the returns.

Basically, in this 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, 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 same Q-values at the end of the previous episode, the same actions 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, the evaluated value E for the Q-values at the end of each episode is defined as

$$E = \sum_{k=1}^{N} d^{N-k} r_k \tag{2}$$

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 (2) could bring that the larger the evaluated value E for the Q-values of an agent is, the superior the Q-values are.

2.3. Information Exchange Method Based on Particle Swarm Optimization

This subsection describes an information exchange method based on PSO. PSO (Kennedy & Eberhart, 2001) is a population-based method which is often used for rapidly finding global optimal solutions in optimization. PSO originates in social behavior, and each agent updates its own candidate solution by utilizing its own personal best and the global best.

Here, PSO is outlined. Consider a problem of determining values of N decision variables $x = (x(1), x(2), \cdots, x(N))$ which maximize an objective function $f(x)$. In PSO, a swarm is prepared and all particles in the swarm are used for solving this problem. Let J be the number of the particles (the swarm size), and $x_j = (x_j(1), x_j(2), \cdots, x_j(N))$ be the decision variable vector (the candidate solution) of particle j $(j = 1, 2, \cdots, J)$. The candidate solution of j is updated as follows:

$$\begin{aligned} v_j(n) &\leftarrow wv_j(n) + c_1 r_1 \{p_j(n) - x_j(n)\} \\ &\quad + c_2 r_2 \{g(n) - x_j(n)\} \qquad (3) \\ x_j(n) &\leftarrow x_j(n) + v_j(n) \qquad (4) \end{aligned}$$

$$(j = 1, 2, \cdots, J, n = 1, 2, \cdots, N)$$

where

$v_j = (v_j(1), v_j(2), \cdots, v_j(N))$: velocity vector of particle j,

w : weight parameter called *inertia weight*,

c_1, c_2 : weight parameters called *acceleration coefficients*,

r_1, r_2 : uniform random numbers in the range from 0 to 1,

$p_j = (p_j(1), p_j(2), \cdots, p_j(N))$: best solution found by particle j so far, which is called *personal best*,

$g = (g(1), g(2), \cdots, g(N))$: best solution found by all particles so far, which is called *global best*.

The reinforcement learning problems can be considered to be a kind of optimization problems by regarding a candidate solution and an objective function value as Q-values and an evaluated value E, respectively. It is expected that optimal Q-values can be found rapidly by applying the procedure of updating the candidate solution in PSO to the update scheme of Q-value.

In the reinforcement learning, the personal best of each agent and the global best are determined by comparing evaluated values and are stored. Let the personal best P_i be the best Q-values found by the agent i so far, and let the global best G be the best Q-values found by all the agents so far. Each agent updates its Q-values by using these two kinds of best Q-values. Following the procedures (3) and (4), we give the update equations

as:

$$V_i(s,a) \leftarrow WV_i(s,a) + C_1R_1\{P_i(s,a) - Q_i(s,a)\} + C_2R_2\{G(s,a) - Q_i(s,a)\} \quad (5)$$

$$Q_i(s,a) \leftarrow Q_i(s,a) + V_i(s,a) \quad (6)$$

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.

3. The Proposed Swarm Reinforcement Learning Algorithm

Although SRL-PSO can find good policies in short learning time, these policies are not necessarily optimal. In this section, we propose an extension of SRL-PSO in order to enhance its performance.

SRL-PSO do not necessarily find optimal policies because overvalued Q-values sometimes become the personal best. Q-values of each agent are evaluated based on rewards obtained during the previous one episode, as mentioned in Subsection 2.2. Therefore, even if the Q-values are considerably different from the Q-values which bring the maximum return, they are evaluated to be much better in the case where the agent unexpectedly gains large rewards by actions selected randomly. If such overvalued Q-values become the personal best or the global best, then they are mostly better than Q-values updated by the individual Q-learning and they are unfortunately kept to be the personal best or the global best. Since the Q-values of agent continue to be updated by using the overvalued Q-values in learning through exchanging information, SRL-PSO may not be able to find optimal policies.

In the proposed method, if the personal best is not replaced after experiencing a specified number of episodes, it is forced to be replaced in order not to store overvalued Q-values over many episodes. For this purpose, the concept of age and lifespan is introduced into the personal best. After an agent performs the individual Q-learning, the age of its own personal best increases by one. If the personal best is not replaced after experiencing a certain number of episodes corresponding to a specified lifespan, it comes the end of its life and is replaced with the current Q-values of the agent. If it is replaced, its age is reset to zero. In addition, if the personal best which comes the end of its life is the global best, the global best is also replaced. In this case, the (replaced) personal bests of all the agents are compared, and the best among them is stored as the global best. By using this information exchange method, even if overvalued Q-values are the personal best, it is replaced after experiencing some episodes and it is expected that better policies are found.

Because the personal best and the global best of PSO for optimization problems are determined based on objective function values, to introduce the lifespan is not necessary. By contrast, since the personal best and the global best of SRL-PSO could be overvalued for the reason mentioned above, to introduce the lifespan is considered to be an important factor to find better policies.

The flow of SRL-PSO-L is as follows.

SRL-PSO-L

E_i : evaluated value for the Q-values $\{Q_i(s,a)\}$ of agent i.
E_i^{P} : evaluated value for the personal best $\{P_i(s,a)\}$ of agent i.
E^{G} : evaluated value for the global best $\{G(s,a)\}$.
i^{G} : agent number in which the Q-values are the origin of the global best.
age_i : age of agent i.
L : lifespan of the personal best.
Y : number of episodes for which the individual Q-learning is performed between the information exchange among the agents.

Step 1 Set the initial values of $Q_i(s,a)$ and $V_i(s,a)$ $(\forall i,s,a)$, and set $age_i \leftarrow 0$ $(\forall i)$, $E_i^{\mathrm{P}} \leftarrow -\infty$ $(\forall i)$ and $E^{\mathrm{G}} \leftarrow -\infty$.

Step 2 Perform the following procedures for each agent i.

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

Step 2-2 Update $Q_i(s,a)$ by performing the individual Q-learning (1) for one episode.

Step 2-3 Set $age_i \leftarrow age_i + 1$. If $age_i = L$, go to Step 2-7.

Step 2-4 Calculate the evaluated value E_i by (2). If $E_i > E_i^{\mathrm{P}}$, set $P_i(s,a) \leftarrow Q_i(s,a)$ $(\forall s,a)$, $E_i^{\mathrm{P}} \leftarrow E_i$ and $age_i \leftarrow 0$.

Step 2-5 If $E_i > E^{\mathrm{G}}$, set $G(s,a) \leftarrow Q_i(s,a)$ $(\forall s,a)$, $E^{\mathrm{G}} \leftarrow E_i$ and $i^{\mathrm{G}} \leftarrow i$.

Step 2-6 Go to Step 2-9.

Step 2-7 Set $P_i(s,a) \leftarrow Q_i(s,a)$ $(\forall s,a)$, $E_i^{\mathrm{P}} \leftarrow E_i$ and $age_i \leftarrow 0$.

Step 2-8 If $i = i^{\mathrm{G}}$, set $i^{\mathrm{G}} \leftarrow \arg\max_i E_i^{\mathrm{P}}$, $G(s,a) \leftarrow P_{i^{\mathrm{G}}}(s,a)$ $(\forall s,a)$ and $E^{\mathrm{G}} \leftarrow E_{i^{\mathrm{G}}}^{\mathrm{P}}$.

Step 2-9 If $y < Y$, set $y \leftarrow y + 1$ and return to Step 2-2.

Step 3 Update $V_i(s,a)$ and $Q_i(s,a)$ of each agent i by (5) and (6).

Step 4 If the termination condition is satisfied, terminate this algorithm. Otherwise, return to Step 2.

4. Numerical Experiments

The effectiveness of the proposed SRL-PSO-L is examined by applying it to a shortest path problem and by comparing its computational efficiency with that of other algorithms. A usual shortest path problem may easily be solved by ordinary Q-learning, 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

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 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.

Figure 1 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. Four cases 1, 2, 3 and 4 in which the number and positions of the walls are different among them are generated according to the above condition. Cases 1, 2, 3 and 4 are numbered in the ascending order of the number of walls, and there is no wall in Case 1.

4.2. Experimental Set up

In this paper we consider problems of single-agent tasks, and the shortest path problem explained in the previous subsection is a single-agent task. SRL-PSO-L, SRL-PSO and ordinary Q-learning with a single agent (called QL) are applied to this problem, and their experimental results are compared.

The following values are used for parameters of Q-learning:

learning-rate parameter : $\alpha = 0.1$,
discount-rate parameter : $\gamma = 0.999$,
probability of random action : $\varepsilon = 0.2$.

All the initial values of $Q_i(s, a)$ are set to be zero.

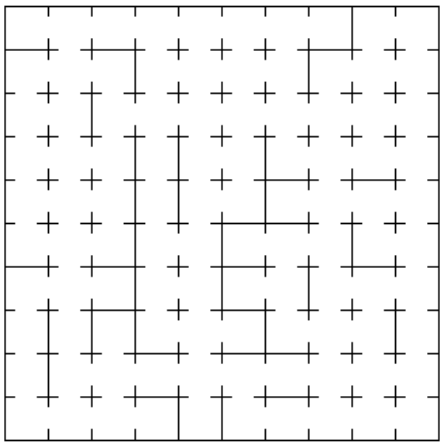

Figure 1. Example of grid world for n=10

The coordinates of the goal cell are changed randomly at every episode. When the agent reaches the goal cell, it gains the reward +100. Otherwise, it gains -1 whenever it takes an action. By using this rule of rewarding, the evaluated value E becomes larger when the number of actions to reach the goal cell is smaller.

For SRL-PSO-L and SRL-PSO, the following values are also used for their parameters:

number of agents : 4,
weights in (5) and (6) : $W = 0, C_1 = C_2 = 0.2$,
number of episodes for which the individual Q-learning is performed between the information exchange : $Y = 1$,
learning-rate parameter : $\alpha = 0.5$,
discount parameter in (2) : $d = 0.999$,
lifespan : $L = 50$ (for SRL-PSO-L).

The values of all the parameters used in our experiments are determined through the preliminary experiments in such a way that each algorithm works as good as possible.

4.3. Results and Discussion

Figures 2-5 show the variation of number of actions through the learning phase obtained by each algorithm. The x-axis in these figures represents the number of episodes. In SRL-PSO-L and SRL-PSO, the x-axis is not the number of episodes for each single agent, and is the sum of numbers of episodes for all the agents. The y-axis represents the minimum number of actions to reach the goal found so far. Each algorithm is performed thirty times with various random seeds, and their results are averaged.

It is observed from these figures that the numbers of actions in SRL-PSO-L and SRL-PSO are smaller than that of QL. The convergence speed of QL is gradu-

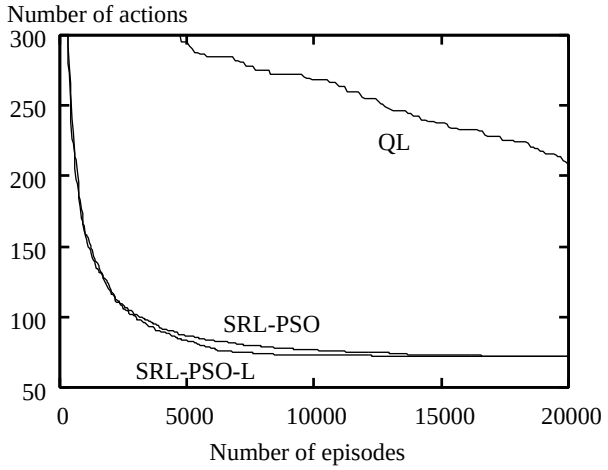

Figure 2. Variation of number of actions through the learning phase (Case 1)

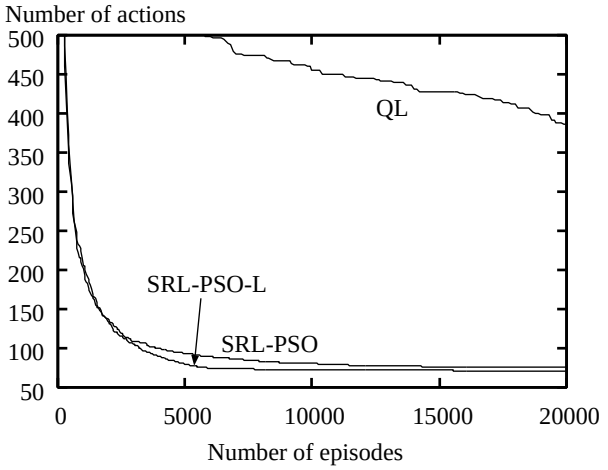

Figure 4. Variation of number of actions through the learning phase (Case 3)

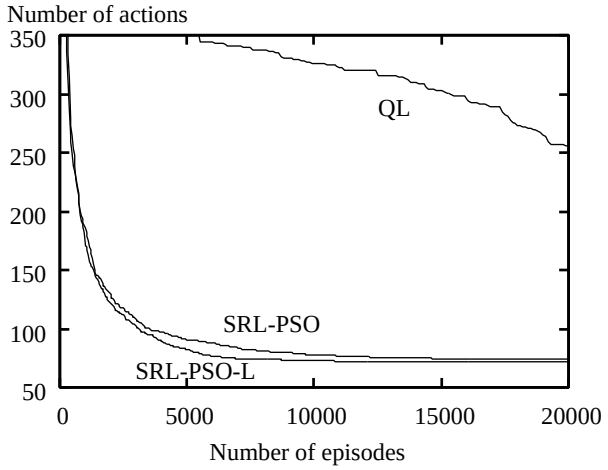

Figure 3. Variation of number of actions through the learning phase (Case 2)

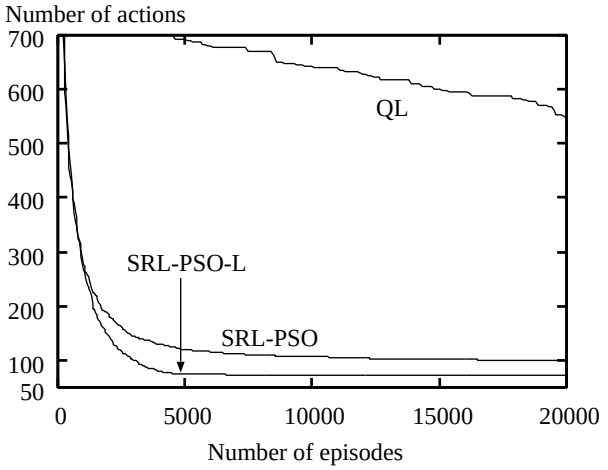

Figure 5. Variation of number of actions through the learning phase (Case 4)

ally slower when the number of walls increases. On the other hand, SRL-PSO-L and SRL-PSO can find rapidly good Q-values regardless of the number of walls and learning with the multiple agents works better.

Comparing SRL-PSO-L with SRL-PSO, SRL-PSO-L is better than SRL-PSO in Cases 1-3. In Case 4 in which many walls exist, SRL-PSO-L is much better than SRL-PSO. Therefore, better policies are obtained by introducing the lifetime into the personal best.

Next, simulations are performed by using the Q-values obtained through the previous learning phase with 20000 episodes in each algorithm, and each algorithm is evaluated by the number of actions to reach the goal cell obtained by the simulations. Since there exist 36 coordinates of the goal cell in this shortest path problem, the simulations are performed ten times for each of these goal coordinates and their results are averaged. The ε-greedy method is used for taking an action, and the probability of random action is 0.2, which is the same value as the learning phase.

Table 1 shows the mean number of actions to reach the goal cell obtained by the simulations. It is confirmed from this table that SRL-PSO-L outperforms SRL-PSO and QL.

In this paper, SRL-PSO-L is proposed under the assumption that overvalued personal bests exist for many episodes in SRL-PSO. In order to confirm that this assumption is acceptable, we re-investigate the performance of SRL-PSO from the following viewpoints:

(A) How many personal bests which are not replaced after many episodes are there?

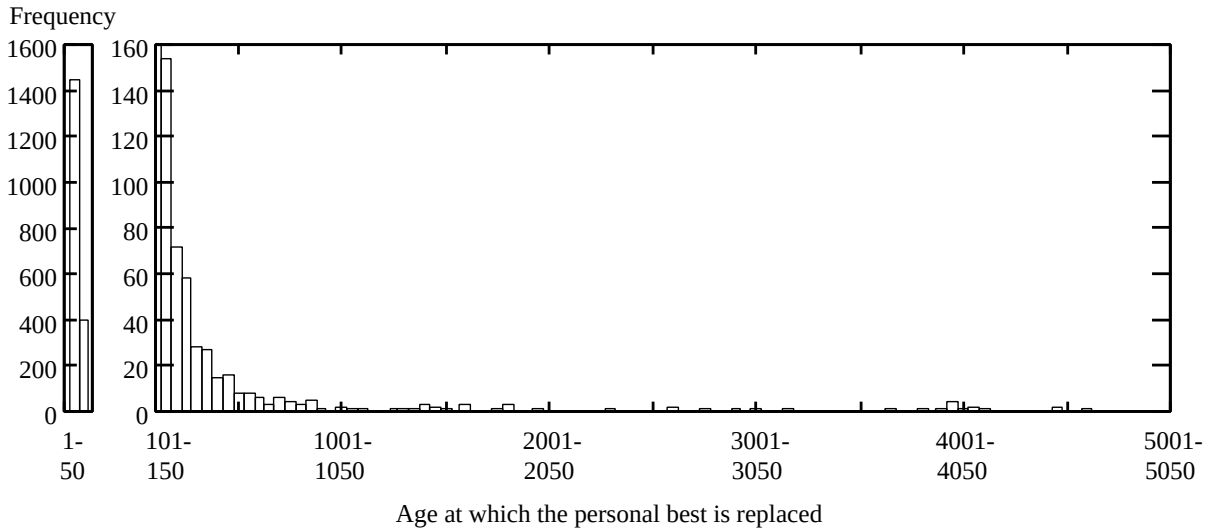

Figure 6. Histogram of ages at which the personal bests are replaced in the learning processes

Table 1. Comparison of number of actions in simulations performed by using the Q-values obtained

Case	SRL-PSO-L	SRL-PSO	QL
1	261	287	945
2	319	321	915
3	515	599	996
4	459	574	999

(B) Are most of the personal bests which are not replaced after the many episodes overvalued?

For this purpose, numerical experiments are carried out for Case 4 where the difference of performance between SRL-PSO and SRL-PSO-L is largest.

Figure 6 shows the histogram of ages at which the personal bests are replaced in the learning processes. In the x-axis of this figure, fifty consecutive ages are categorized into a single class. It is found from this figure that there exist more than 600 personal bests which are not replaced after more than 100 episodes. In addition, there are some long-lived personal bests which are not replaced after more than 4000 episodes. We call the personal best which is not replaced after more than 100 episodes the *long-lived* personal bests.

Figure 7 shows the histogram of episode numbers at which the long-lived personal bests appear in the learning processes. In the x-axis of this figure, one hundred consecutive episode numbers are categorized into a single class. From this figure, many long-lived personal bests appear after the 900-th episode while few long-lived personal bests appear before the episode.

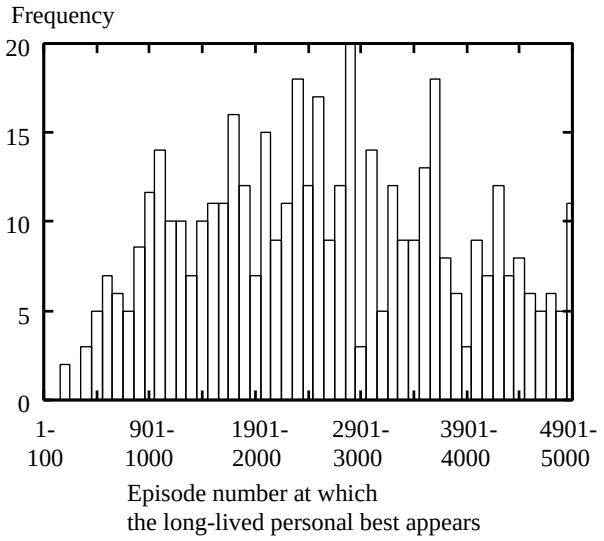

Figure 7. Histogram of episode numbers at which the long lived-personal bests appear in the learning processes

It is confirmed from Figs. 2-5 that the performance of SRL-PSO is inferior to that of SRL-PSO-L after about the 900-th episode at which the number of long-lived personal bests increases. Therefore, the long-lived personal bests are considered to bring the inferior performance of SRL-PSO.

In order to confirm whether long-lived personal bests are overvalued or not, the following experiment is carried out for each of the thirty learning processes:

Step 1 Pick up the long-lived personal best which first appears between the 900-th episode and

the 1100-th episode in the learning process. These episodes are selected for the reason that the performance of SRL-PSO is inferior after about the 900-th episode at which the number of long-lived personal bests increases.

Step 2 Re-calculate evaluated values E 300 times for the long-lived personal best, and determine the maximum $E_{\max}$ of these evaluated values.

Step 3 Calculate the difference $D(= E_{\text{org}} - E_{\max})$ between the original evaluated value E_{org} and $E_{\max}$ for the long-lived personal best. If E_{org} is much greater than $E_{\max}$, the personal best is determined to be overvalued.

Step 4 Select a personal best randomly from the personal bests at the 1000-th episode, and perform Steps 2-3 in a similar way.

For a learning process, a long-lived personal best between the 900-th episode and the 1100-th episode may not be found in Step 1. For this case, Steps 2-4 are not carried out.

Table 2 shows E_{org}, $E_{\max}$ and D for the long-lived personal bests and the randomly selected personal bests. It is confirmed from this table that D for the long-lived personal bests is much greater than that for the randomly selected personal bests. Hence, the long-lived personal bests are overvalued, which brings inferior learning performance of SRL-PSO.

Table 2. Comparison between the long-lived personal bests and the randomly selected personal bests

Learning process number	long-lived personal best			randomly selected personal best		
	E_{org}	$E_{\max}$	D	E_{org}	$E_{\max}$	D
2	−142	−476	334	−849	−566	−283
4	−130	−503	373	−435	−542	107
5	−121	−512	391	−536	−542	6
9	−161	−489	328	−548	−464	−84
11	−139	−524	385	−626	−473	−153
12	−122	−582	460	−302	−438	136
13	−92	−567	475	−787	−507	−280
15	−105	−476	371	−463	−532	69
17	−137	−568	431	−752	−400	−352
18	−128	−520	392	−718	−438	−280
19	−204	−554	350	−762	−470	−292
20	−200	−530	330	−759	−504	−255
21	−78	−626	548	−829	−622	−207
22	−161	−578	417	−888	−545	−343
23	−178	−510	332	−381	−565	184
25	−279	−573	294	−874	−640	−234
26	−181	−551	370	−743	−544	−199
27	−226	−532	306	−820	−520	−300
30	−104	−521	417	−312	−517	205
Average			384			−134

5. Conclusion

In SRL-PSO, multiple agents learn individually by using a usual reinforcement learning algorithm and also learn by using the update equations of particle swarm optimization. The performance of SRL-PSO is inferior because the overvalued personal bests appear.

This paper has proposed SRL-PSO in which a lifespan is introduced into the personal best. If the personal best is not replaced after experiencing episodes corresponding to the lifespan, it comes the end of its life and is forced to be replaced with the current Q-values. In addition, if the personal best which comes the end of its life is the global best, the global best is also replaced. It is confirmed from the experimental results that the proposed algorithm using the lifespan outperforms SRL-PSO and ordinary Q-learning using a single agent.

References

Busoniu, L., Babuska, R., & Schutter, B. D. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics (Part C)*, *38*, 156–172.

Iima, H., & Kuroe, Y. (2006). Reinforcement learning through interaction among multiple agents. *SICE-ICASE International Joint Conference 2006 CD-ROM* (pp. 2457–2462).

Iima, H., & Kuroe, Y. (2008). Swarm reinforcement learning algorithms based on particle swarm optimization. *Proceedings of 2008 International Conference on Systems, Man and Cybernetics* (pp. 1110–1115).

Kennedy, J., & Eberhart, R. (2001). *Swarm intelligence*. Morgan Kaufmann Publishers.

Sutton, R., & Barto, A. (1998). *Reinforcement learning*. MIT Press.

Watkins, C., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*, 279–292.