

Instruction Knowledge Acquisition for Reinforcement Learning Scheme by PSO Algorithm

Toru Sawa
Osaka Electro-Communication University
Neyagawa, Osaka, Japan
m10709@oecu.jp

Toshihiko Watanabe
Osaka Electro-Communication University
Neyagawa, Osaka, Japan
t-wata@isc.osakac.ac.jp

Abstract— In order to realize intelligent agents such as autonomous mobile robots, Reinforcement Learning is one of the necessary techniques in control systems. It is desirable in terms of knowledge or skill acquisition of agents that reinforcement learning is based only upon rewards instead of teaching signals. However, there exist many problems to apply reinforcement learning to real-world tasks. The most severe problem is a huge number of iterations in the learning phase. In order to deal with the problem, the instruction approach for reinforcement learning agents based on sub-rewards and forgetting mechanisms were proposed and shown to be effective. However, the relationship between the instruction and the learning performance of reinforcement learning has not been adequately clarified. In this study, in order to clarify the instruction performance in the reinforcement learning, we propose an instruction knowledge acquisition method for the reinforcement learning scheme by the particle swarm optimization (PSO) algorithm. Through numerical experiments of the mountain car task and the Acrobat task, we show the validity of the proposed approach in terms of learning speed and accuracy.

Keywords: *Reinforcement Learning, Q-Learning, Fuzzy Q-Learning, Instruction, PSO, Machine Learning*

I. INTRODUCTION

The Reinforcement Learning[1-4] will be one of the necessary techniques to realize intelligent agents such as autonomous mobile robots. The reinforcement learning is a desirable idea that can acquire the skill and knowledge without using the teaching signal based upon only the rewards from environments. However, the huge iterations of the trial for accomplishment of the reinforcement learning become a problem in actual applications in which the appropriate learning of the compound task or complicated task is needed. This huge trial frequency is one of the issues of the reinforcement learning. In order to deal with the complicated task problem, several effective approaches are studied. For relaxation of task complexity, several types of hierarchical reinforcement learning have been proposed to apply to actual applications[5,6]. The hierarchical reinforcement learning has been widely recognized as an effective scheme for complicated tasks. However, the approach is based on the assumption that the compound task is decomposed hierarchically into subtasks as internal models of the agent in advance. The assumption is sometimes not suitable for learning of agents in an unknown environment, i.e. basic concepts of the reinforcement learning. When we consider

realizing the automatic task decomposition function by means of the reinforcement learning, improvement of learning performance such as iterations and precision of acquisition is indispensable for the reinforcement learning technique as well as a concrete segmentation technique of the learning results. The authors proposed an instruction based reinforcement learning applying sub-rewards and forgetting mechanisms[7,8] in order to improve the learning performance of reinforcement learning agents. In the study, we aimed at constructing a new learning scheme that can utilize the instruction, i.e. not strictly teaching, in the reward based reinforcement learning scheme and showed effectiveness in terms of the necessary learning speed in the reinforcement learning process through numerical experiments of not so complicated tasks. However, the relationship between the instruction and the learning performance of the reinforcement learning has not been adequately clarified. In this study, in order to clarify the instruction performance further in the reinforcement learning, we propose an instruction knowledge acquisition method for reinforcement learning scheme by the particle swarm optimization (PSO) algorithm[9]. The efficacy is shown by the numerical experiment of the mountain car task and the Acrobat task.

The paper is organized as follows. In section 2, we discuss the issue of instructions for task acquisition. In section 3, we introduce a learning method based on the instruction. An instruction knowledge acquisition algorithm is proposed in section.4. Then the results of numerical experiments are shown in section 5. Finally, conclusions are drawn in section 6.

II. INSTRUCTIONS FOR TASK ACQUISITION

The skillful task performed by human experts is indispensable even in the modern society. Even in a highly automated factory, skillful experts usually take part in important tasks. Moreover, tasks in traditional artistic crafts such as ceramic art or woodworking are also highly skillful. On the other hand, techniques in sports can also be said to be skillful for not only the athlete but also general people in our daily life. Since it is obvious that the sports are good for health, learning the techniques(task acquisition) in the sports is necessary process to all the people. Even injured people under treatment, struggle to restore their movement functions through rehabilitation physical trainings which are difficult tasks for themselves as though skillful techniques of the sports.

There exist many skillful tasks to acquire in many areas.

Though the skillful task is highly complicated at a glance through an observation, many tasks consist of the combination of simple sub-tasks. As seen from the potential ability of human recognition, the simple task is clearly performed from the selected inputs among huge information, such as vision, sound, and measured data. The comprised simple task is not difficult itself to be developed into an automated machine owing to the modern technology progress. However the task decomposition without knowledge is not easy to develop. In the soft computing area, the automatic task decomposition by the computer system has been studied. Modular neural network[10,11] is proposed to realize the modularization of learning results. This is equivalent to the task decomposition represented as “gating network.” Modular fuzzy model[12] is also proposed to deal with the problem. The task decomposition without knowledge is more difficult without explicit teaching.

This problem corresponds substantially to the reinforcement learning concept. The model for the reinforcement learning is usually constructed as a memory type, neural network, fuzzy model, locally weighted regression, CMAC and so on. However, non structured model acquired is essentially a mimic system that is not robust for changes of environment, condition, and so on. Moreover, the segmentation of native acquired model is an important problem to realize the task decomposition. Recently, study of the task decomposition is advanced by learning by the intrinsic motivation[13,14]. The methods are based on internal rewards to formulate behavioral rules abstracted from the results of the reinforcement learning expressed as action rules. In the abstraction process, the segmentation of learning is an indispensable and essential technique. Although they are promising techniques in terms of task decomposition or task acquisition of complicated tasks of the agent, the substantial difficulty of the intrinsic reward concept is still remained in actual applications.

On the other hand, in terms of realization of applicable task decomposition or task acquisition, the authors proposed an “Instruction based reinforcement learning”[7,8] applying environmental sub-rewards and forgetting mechanism in the agent and showed promising results through not so complicated tasks. We consider that there exist some situations that the environmental sub-reward is more suitable than the intrinsic concept in the task acquisition. “Instruction” plays an important role in acquiring the task of human works. For example, when we teach someone the skill we already have, we often give only an instruction as a knack or a point that is necessary to perform the skill, instead of teaching every detail including trivial matters in the task procedure. Our motivation of “Instruction based reinforcement learning” is to realize such instructions in the reinforcement learning scheme. The basic concept is to give “instruction” that is not strict and fragmental compared with the teaching data used in the teaching based learning such as the multi-layered perceptron,

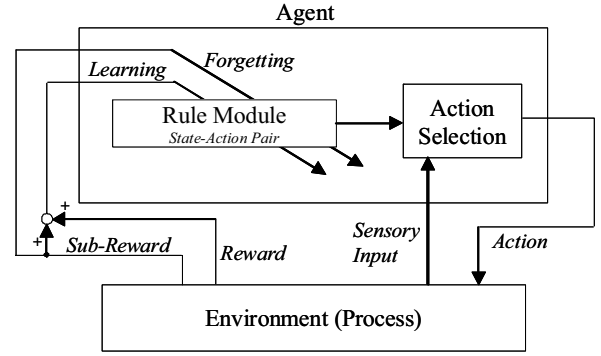


Figure 1. Conceptual Diagram of Instruction Based RL Architecture

to the reinforcement learning agent. The instruction concept can be realized as sub-rewards in addition to the main reward as well as forgetting mechanism to improve the learning performance in the reinforcement learning process.

However, there exist a lot of tasks difficult to give the instructions even by the skillful human experts. Moreover, the instruction concept is essentially subjective and is not clear its optimality or validness in terms of the reinforcement learning. Our primary motivation in this study is to investigate the relationship between instructions and learning performance of the reinforcement learning. Since the instruction can be considered as additive information to accelerate the learning process of the task, valid instruction is necessary to the task acquisition. Furthermore the instruction acquisition might lead to significant tools to analyze the skillful task and to hand down the skillful task to the next generation by interacting human and agent(robot) models. In this study, in order to clarify the instruction performance further in the reinforcement learning, we propose an instruction knowledge acquisition method for the reinforcement learning scheme

III. LEARNING METHOD BASED ON INSTRUCTION

In order to commission sub-rewards mechanism appropriately, the issue of consistence maintenance between sub-rewards and the main reward should be dealt with. Then we proposed a learning algorithm based on sub-rewards with forgetting mechanism and eligibility trace[7, 8]. The sub-rewards are defined corresponding to necessary state-action pairs. Moreover, when the agent obtains the sub-reward, forgetting mechanism is applied at the same time. Fig.1 shows the conceptual diagram of proposed learning architecture.

A. Q-learning algorithm

In Q-learning reinforcement learning algorithm, an optimal policy is formulated by maximizing rewards received over time. Q-function is defined as values for each pair of state and action:

$$Q(x_t, a_t) = Q(x_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_{\eta} Q(x_{t+1}, \eta) - Q(x_t, a_t) \right) \quad (1)$$

where x_t is the state vector of t -th input, a_t is the t -th action,

r_{t+1} is a reward, γ is a discount rate, and α is a learning rate. The reward based on the sub-reward is defined as:

$$r_{t+1} = hr_{t+1} + R_{t+1} \quad (2)$$

where hr_{t+1} denotes a sub-reward, and R_{t+1} is a main reward. At learning, neither a sub-reward nor a main reward distinguishes the kind. Therefore, the possibility of inconsistency of the expectation reward in the reinforcement learning is not negligible. Then, forgetting is executed also at giving the instruction to maintain consistency of the expectation reward. Forgetting mechanism is defined as:

$$Q(x, b) = \Gamma \cdot Q(x, b), \quad b \neq a_t, 0 \leq \Gamma \leq 1 \quad (3)$$

where Γ is a forgetting parameter and a_t is the current action.

B. Fuzzy Q-learning algorithm

In Fuzzy Q-Learning[15], Q-function is approximated by the fuzzy model. The fuzzy model is defined by FCS (Fuzzy Classifier System) form[16] as:

$$\text{if } x_t \text{ is } A_j \text{ then } a_t \text{ is } c_j^k \text{ with } q_j^k \quad (4)$$

where c_j^k is a k -th concrete action, A_j denotes the fuzzy set defined in the input states, and q_j^k is the Q-value of the j -th rule. The output Q-value is calculated as:

$$Q(x_t, c^k) = \frac{\sum_{s=1}^m h_s \cdot q_s^k}{\sum_{s=1}^m h_s} \quad (5)$$

where h_s denotes the membership degree of the rule and m is the number of the fuzzy rules. From (1), the TD error that should be corrected is defined as:

$$\Delta Q(x_t, a_t) = \left(r_{t+1} + \gamma \max_{\eta} Q(x_{t+1}, \eta) - Q(x_t, a_t) \right) \quad (6)$$

The correction of the q-value in the rule is defined as follows by the request of the gradient of (5) and (6).

$$\Delta q_j^k = \Delta Q(x_t, a_t) \cdot \frac{\partial Q(x_t, a_t)}{\partial q_j^k} = \Delta Q(x_t, a_t) \cdot \frac{h_j}{\sum_{s=1}^m h_s} \quad (7)$$

where a_t is the action c^k in this equation. The maximum Q value at the right of (6) is obtained from (5).

Moreover, when the sub-reward and forgetting are applied to the fuzzy Q-learning, (2) and (4) are applied based on the next equation that modifies the Q value.

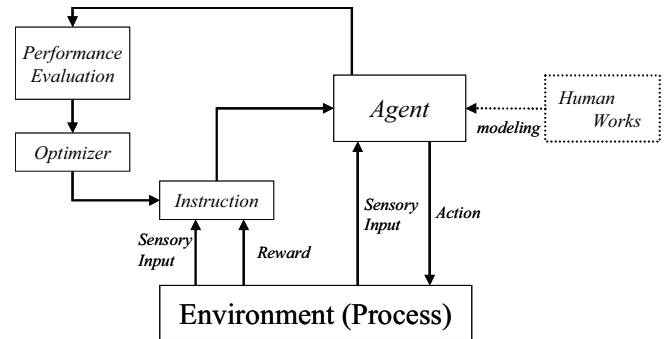


Figure 2. Conceptual Diagram of Instruction Knowledge Acquisition

$$q_j^b = \Gamma \cdot q_j^b$$

$$\Gamma = 1 - (1 - \beta)(1 - \mu(d, b)) \cdot h_j, \quad 0 \leq \beta \leq 1 \quad (8)$$

where μ is a membership function that shows the degree of agreement of selected action d and the other actions b , and β is a forgetting parameter. The degree of forgetting becomes high when we set a small value as the parameter.

The eligibility trace can also be successfully applied at the same time of the instruction[8].

IV. INSTRUCTION KNOWLEDGE ACQUISITION BY PSO ALGORITHM

Our objective in this study is to acquire effective instructions through reinforcement learning simulations. Fig.2 shows the conceptual diagram of instruction knowledge acquisition. Let us assume that the human works such as actions or movement are modeled as the agent or the robot. By evaluating the learning performance using episode lengths, appropriate instructions are decided through optimization procedures. Empirically, there could exist many alternatives of good instructions. In order to deal with such nature, we adopt PSO algorithm to acquire the instructions.

A. Particle swarm optimization algorithm

Particle swarm optimization[9] is a population-based iterative search procedure, which simulates the social behavior such as flocks of birds or schools of fish. N individuals, referred to as particles, constitute a swarm. Each particle is a candidate solution to the D dimensional problem. The position of the i -th particle at t -th iteration is represented by:

$$X_i^{(t)} = (x_{i1}, x_{i2}, \dots, x_{iD}) \quad (9)$$

During the search process, the particle successively adjusts its position toward the global optimum according to two factors. The best position encountered by itself (pbest) is denoted as:

$$P_i = (p_{i1}, p_{i2}, \dots, p_{iD}) \quad (10)$$

The best position encountered by the whole swarm (gbest) is denoted as:

$$P_g = (p_{g1}, p_{g2}, \dots, p_{gD}) \quad (11)$$

Its velocity at t iteration is represented by:

$$V_i^{(t)} = (v_{i1}, v_{i2}, \dots, v_{iD}) \quad (12)$$

The position at the next iteration is calculated according to the following equations:

$$V_i^{(t)} = \lambda \left(w * V_i^{(t-1)} + c_1 * rand() * (P_i - X_i^{(t-1)}) + c_2 * rand() * (P_g - X_i^{(t-1)}) \right) \quad (13)$$

$$X_i^{(t)} = X_i^{(t-1)} + V_i^{(t)} \quad (14)$$

where c_1 and c_2 are two positive constants, called cognitive learning rate and social learning rate respectively; $rand()$ is a random function in the range $[0;1]$; w is inertia factor; and λ is a constriction factor. In addition, the velocities of the particles are confined within $[Vmin, Vmax]^D$. If an element of velocities exceeds the threshold $Vmin$ or $Vmax$; it is set equal to the corresponding threshold. From equation (13) and (14), it can be seen that the velocity determines search direction and step-size simultaneously for the particle according to the social learning and the cognitive learning.

The point of the instruction for the reinforcement learning is defined as a particle (position X) for PSO algorithm in this study. The particles(instruction points) search towards the point in which the trial episode lengths of learning are minimized.

V. NUMERICAL EXPERIMENTS

In order to evaluate the performance of the proposed approach, the numerical experiments are conducted for the mountain car task and the Acrobot task. The search of the instruction points is performed by using PSO algorithm.

Each parameter of PSO is as follows. c_1 and c_2 are 2.0, respectively. The number of particles in the swarm is 10. Constriction factor λ is set as 0.1. Inertia factor w is set as 1.0. Let us assume that the positions of particles are also confined within the range $[Xmin, Xmax]^D$. In this study, $Xmax$ is set as 100, and $Xmin$ is set as -100. The state variables including the action for the instruction are normalized to $[Xmin, Xmax]^D$ according to the task. In the mountain car task, the dimension of the state is three(position, speed, and acceleration operation of the car agent). In acrobot task, the dimension of the state is five (angle and angular velocity of the first link, angle and angular velocity of the second link, and torque operation of the Acrobot).

Initial positions of particles are decided at random within the range of $Xmin$ and $Xmax$. The number of generations of PSO is set as 100 in this study.

A. Mountain Car Task(Crisp)

Fig.3 shows the settings of the mountain car task. The purpose of this task is to climb a right-hand mountain. The power of the car agent is restrained as lower value than the gravity. Therefore, the acquisition of the action knowledge that shuttles several times, enlivens, and reaches a goal becomes indispensable. The state of the agent (car) is two continuous values of speed v_t and position p_t . The action can be selected from $\{-1, 0, 1\}$. The action is each full throttle retreat (-1), zero throttle (0), and full throttle advancement (+1). When the car finishes climbing the mountain, the reward is given. Agent's initial state was assumed to be $p_0 = -0.5$ and $v_0 = 0$. The state is quantized by the crisp partition. The number of partitions is 15, the learning rate α is 0.10, and the discount rate γ is 0.95. The simulations are conducted 100 times. The averaging value is used to evaluate. The agent tries the mountaineering 5 times by one simulation. When the following condition is satisfied, the instruction is given to the agent. When the state and the action of the agent falls into the range of the instruction point(particle) as shown in Fig.4, the instruction is applied to the agent. In the figure, position p and

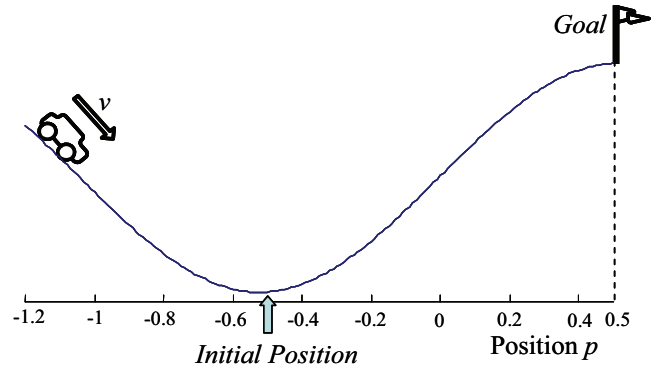


Figure 3. Mountain Car Task

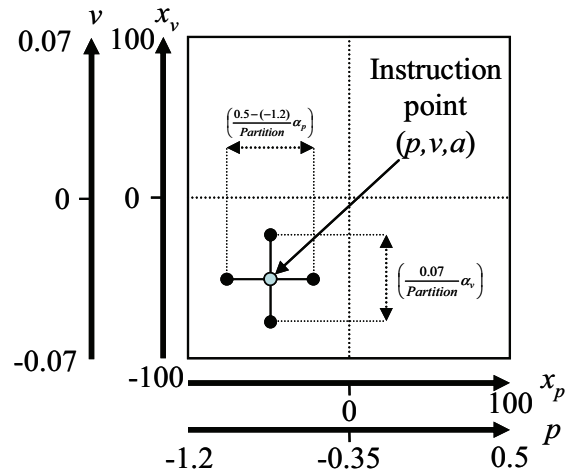


Figure 4. Normalization of Mountain Car Task

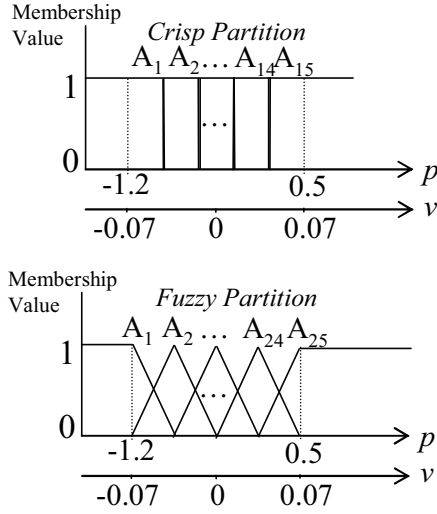


Figure 5. Membership Function

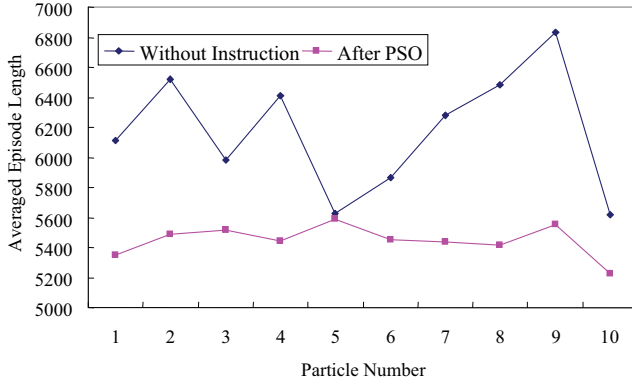


Figure 6. Results of Crisp Mountain Car Task

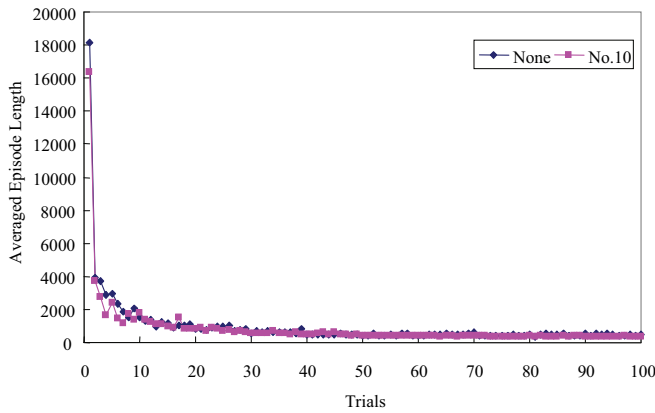


Figure 7. Results of Crisp Mountain Car Task(No.10)

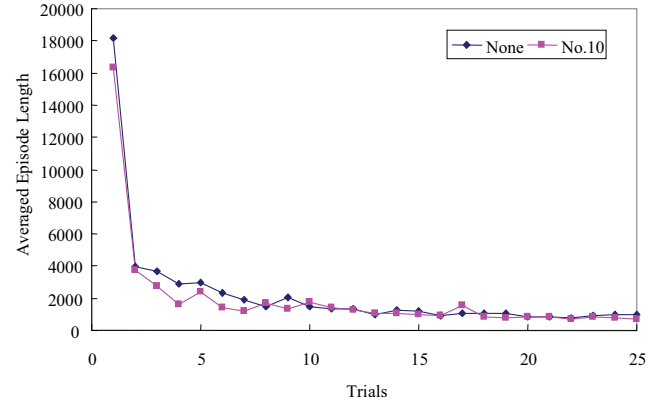


Figure 8. Results of Crisp Mountain Car Task(No.10)

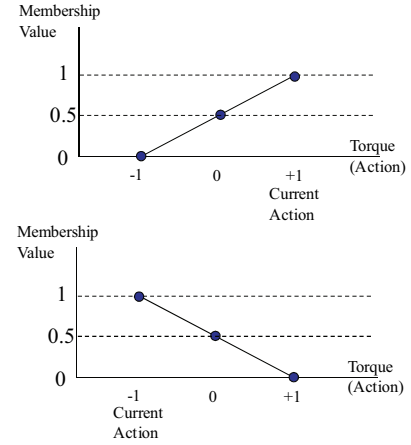


Figure 9. Membership Function for Forgetting

speed v of the instruction are normalized from position x_p and x_v of the particle in PSO. Moreover, the action a is defined as +1 (full throttle advancement) when x_a is positive, and action a is defined as -1 (full throttle retreat) when x_a is negative. α_v and α_p are predefined parameters to decide the range of the instruction, and $0 < \{\alpha_v, \alpha_p\} \leq 1.0$. The range of the instruction becomes wider if they are set near 1.0. α_v and α_p are set as 0.1, in this study.

The membership function for the state quantization for the agent is shown in Fig.5. The averaged episode lengths of the initial states of the particles and the averaged episode lengths after the 100 generations in PSO are shown in Fig.6. About 700 or more of the averaged episode length is decreased by the proposed method. As for the #9, we can see the averaged episode length becomes 5551 from initial value of 6831, i.e. decrease over 1200 episodes. Fig.7 shows the initial 100 trial process of the #10, i.e. the best instruction, after the optimization as well as the results without the instructions. Fig.8 also shows the initial 25 trial process. It is confirmed that the episode lengths are decreased by the proposed algorithm.

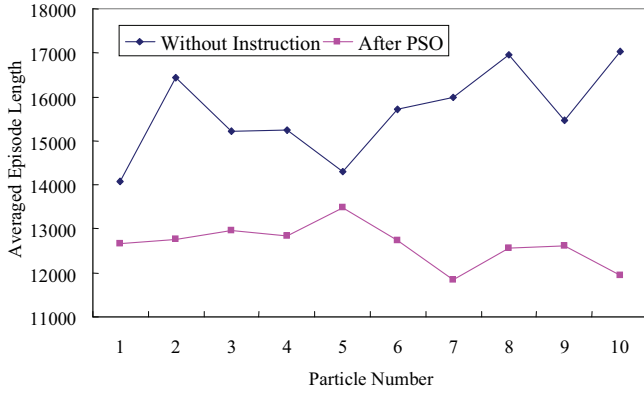


Figure 10. Results of Fuzzy Mountain Car Task

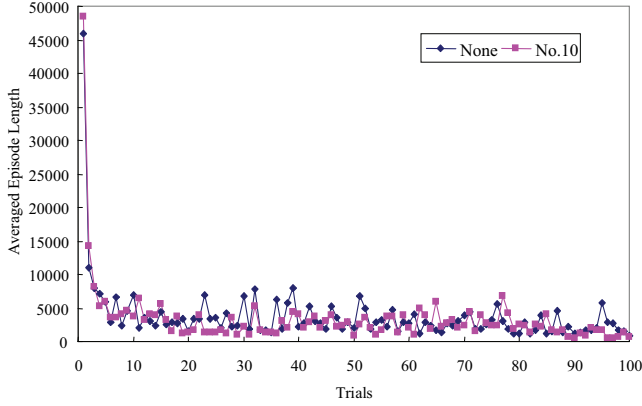


Figure 11. Results of Fuzzy Mountain Car Task(No.10)

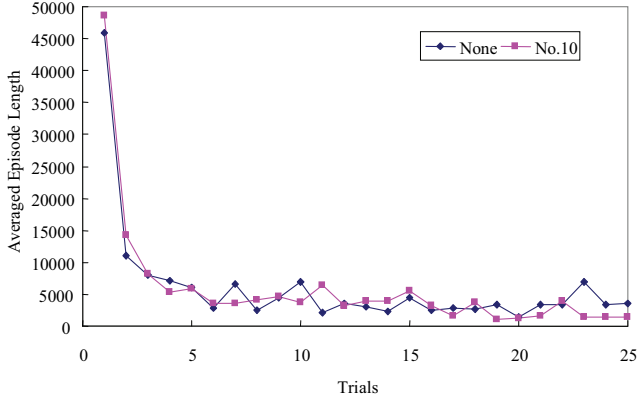


Figure 12. Result of Fuzzy Mountain Car Task(No.10)

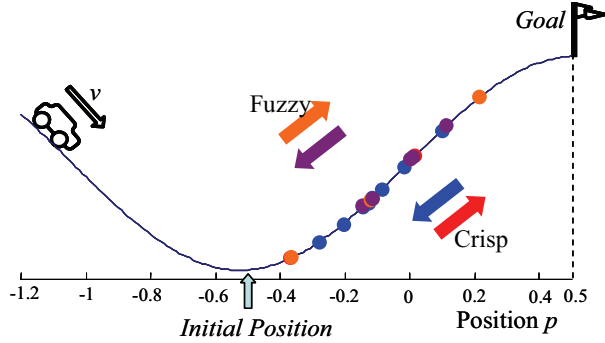


Figure 13. Crisp and Fuzzy Instruction Points

B. Mountain Car Task(Fuzzy)

Fuzzy Q-Learning agent is evaluated applying the instruction acquisition. The state is partitioned according to the membership functions of 25 fuzzy sets as shown in Fig.5. The degree of agreement in (8) that employs forgetting is assumed to be formulated as the membership functions shown in Fig.9. The averaged episode lengths of the initial states of the particles and the averaged episode lengths after the 100 generations in PSO are shown in Fig.10. About 3000 or more of the averaged episode length is decreased by the proposed method. As for the #10, we can see the averaged episode length becomes 11931 from initial value of 17036, i.e. decrease over 5000 episodes. Fig.11 shows the initial 100 trial process of the #10 after the optimization along with the results without the instructions. Fig.12 also shows the initial 25 trial process. It is confirmed that the total episode lengths are decreased by the proposed algorithm (from 367906 to 324872).

Examples of typical instruction points are shown in Fig.13. The action of a full throttle advances is decided around the initial position and the goal as the instruction. The full throttle retreat is decided in the other parts. These results are similar to the results by the Crisp partitioned agents.

C. Acrobot task (Crisp)

Acrobot is a mimic system of the “horizontal bar” of human gymnastics. The objective of the task is to swing the tip of the robot up to the predefined height as shown in Fig.14. A unique characteristic of the task is that the robot adjusts only the second joint torque being affected by its dynamics to achieve the goal. The dynamical equations of the Acrobot are as follows:

$$\begin{aligned}
 d_{11}\ddot{\theta}_1 + d_{12}\ddot{\theta}_2 + h_1 + \phi_1 &= 0 \\
 d_{21}\ddot{\theta}_1 + d_{22}\ddot{\theta}_2 + h_2 + \phi_2 &= T \\
 d_{11} &= m_1 r_1^2 + m_2 l_1^2 + m_2 r_2^2 + 2m_2 l_1 r_2 \cos \theta_2 + I_1 + I_2 \\
 d_{12} &= m_2 r_2^2 + m_2 l_1 r_2 \cos \theta_2 + I_2 \\
 d_{21} &= m_2 r_2^2 + m_2 l_1 r_2 \cos \theta_2 + I_2 \\
 d_{22} &= m_2 r_2^2 + I_2 \\
 h_1 &= -m_2 l_1 r_2 (2\dot{\theta}_1 + \dot{\theta}_2) \dot{\theta}_2 \sin \theta_2 \\
 h_2 &= m_2 l_1 r_2 \dot{\theta}_1^2 \sin \theta_2 \\
 \phi_1 &= (m_1 r_1 + m_2 l_1) g \sin \theta_1 + m_2 r_2 g \sin(\theta_1 + \theta_2) \\
 \phi_2 &= m_2 r_2 g \sin(\theta_1 + \theta_2)
 \end{aligned}$$

where θ_1 , θ_2 are the angles of joint1 and joint2, $\dot{\theta}_1$, $\dot{\theta}_2$ are the angular velocities of joint1 and joint2, and $\ddot{\theta}_1$, $\ddot{\theta}_2$ are the angular accelerations of joint1 and joint2, respectively. T is the torque applied to the joint2. m_i is mass, l_i is length, r_i is length to center of mass, and I_i is moment of inertia of link i , respectively. Actual setting of constant parameters is as $m_1=1$, $l_1=1$, $r_1=1$, $I_1=1$ ($i=1,2$), and $g=9.8$ in this study. The angular velocities are bounded by $\dot{\theta}_1 \in [-4\pi, 4\pi]$ and $\dot{\theta}_2 \in [-9\pi, 9\pi]$. The target height H is set as 0.5. The equations of motion are numerically solved by means of Runge-Kutta method in 0.05sec interval. The sampling period is 0.2 sec. The agent chooses the torque T from $\{-1, 0, 1\}$ as action at every sampling

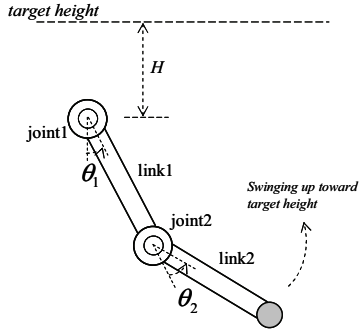


Figure 14. Acrobot Task

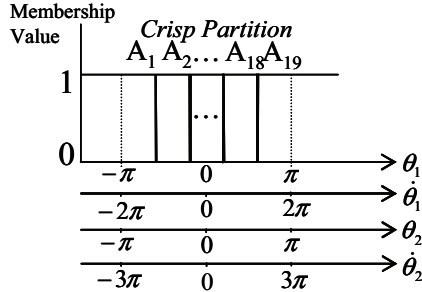


Figure 15. Membership Function

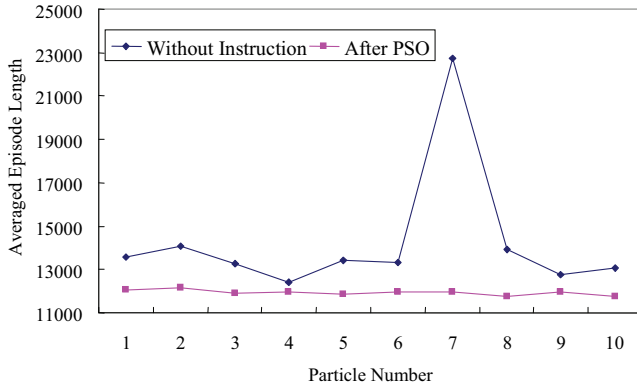


Figure 16. Results of Crisp Acrobot Task

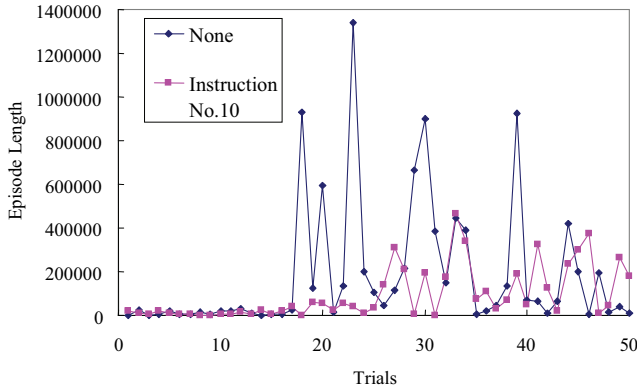


Figure 17. Results of Crisp Acrobot Task(No.10)

TABLE I. ACROBOT INSTRUCTION POINTS

Particle number	Acrobot Instruction Points				
	θ_1	$\dot{\theta}_1$	θ_2	$\dot{\theta}_2$	act
1	0.49	3.44	-0.53	1.79	1
2	0.83	3.85	-0.60	2.83	1
3	0.64	6.81	0.27	-3.25	1
4	1.04	0.64	-1.11	4.83	-1
5	0.57	3.55	-0.42	2.73	1
6	0.70	3.77	-0.31	2.81	1
7	1.44	7.04	-1.41	0.93	1
8	-0.21	2.34	-0.29	0.89	-1
9	-0.70	-6.56	2.52	-18.1	1
10	0.19	2.59	-1.37	-2.98	1

time(step). The number of partitions is 19 as shown in Fig.15, the learning rate α is 0.20, and the discount rate γ is 0.95. The agent tries the mountaineering 5 times in one simulation and the simulation is conducted 100 times. The averaged episode length is used to evaluate the learning performance.

When the following condition is satisfied, the instruction is given to the agent. The condition of the angle is defined in the same way of the mountain car task experiments as follows:

$$\frac{|\theta_{\max}| \cdot x_{\theta}}{|X_{\max}|} - \frac{|\theta_{\max}|}{\text{Partition size}} \leq \theta \leq \frac{|\theta_{\max}| \cdot x_{\theta}}{|X_{\max}|} + \frac{|\theta_{\max}|}{\text{Partition size}} \quad (15)$$

where θ denotes the state value of the angle, x_{θ} is the normalized particle state corresponding to the angle value, and $|X_{\max}|$ is the maximum range value of the particle state. $|\theta_{\max}|$ is the maximum range value of the angle. *Partition size* is a parameter of granularity of PSO search. The other conditions of states such as angular velocities are defined in the same manner.

The averaged episode lengths of the initial states of the particles and the averaged episode length after the 100 generations in PSO are shown in Fig.16. About 2000 or more of the averaged episode length is decreased by the proposed method. As for the #7, we can see the averaged episode length becomes 11951 from initial value of 22724, i.e. decrease over 10000 episodes. Fig.17 shows the initial 50 trial process of the #10 after the optimization along with the results without the instructions. It is confirmed that the total episode lengths are decreased drastically by the proposed algorithm (from 9169856 to 4721048 episodes). The acquired instructions by PSO algorithm are shown in TABLE.I. Fig.18 schematically depicts the examples of the acquired instruction points of Acrobot by the proposed algorithm. Although the Acrobot task is high dimensional and complicated task that the appropriate instruction is not clear intuitively, the proposed algorithm performs successfully the acquisition of the instruction in the reinforcement learning. Moreover, the episode length, i.e. the learning performance, can be drastically decreased by giving

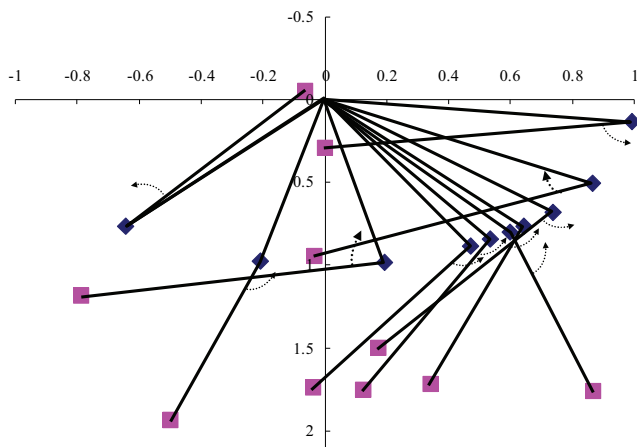


Figure 18. Acquired Instructions of Crisp Acrobot Task

even only one instruction point acquired by the proposed algorithm. It can be considered that this is because the necessary knack existing in the task is successfully acquired by the proposed algorithm.

VI. CONCLUSION

In this study, in order to clarify the instruction performance in the reinforcement learning, we proposed an instruction knowledge acquisition method for the reinforcement learning scheme by the particle swarm optimization (PSO) algorithm. Through numerical experiments of the mountain car task and the acrobat task, we showed the validness of the proposed approach in terms of learning speed and accuracy. Moreover, the effective instruction point for the reinforcement learning process was successfully acquired even in the Acrobot task that was the complicated task. In this study, the number of the instruction was assumed as 1 for simplicity. Our future plan includes acquisition of appropriate combination of some instructions for high dimensional and complicated tasks.

REFERENCES

- [1] R. S. Sutton and A. G. Barto, "Reinforcement Learning", MIT Press, 1998.
- [2] C. J. H. Watkins and P. Dayan, "Technical Note: Q-Learning", Machine Learning, Vol.8, pp.58-68, 1992.
- [3] K. Miyazaki, H. Kimura, and S. Kobayashi, "Theory and Application of Reinforcement Learning Based on Profit Sharing," J. of JSAI, Vol.14, No.5, pp. 800-807, 1999.
- [4] K. Miyazaki, S. Arai, and S. Kobayashi, "A Theory of Profit Sharing in Multi-agent Reinforcement Learning," J. of JSAI, Vol. 14, No.6, pp.1156-1164, 1999.
- [5] Y. Takahashi and M. Asada, "Behavior Acquisition by Multi-Layered Reinforcement Learning," Proc. of the 1999 IEEE Int. Conf. on Syst., Man, and Cybern., pp.716-721, 1999.
- [6] J. Morimoto and K. Doya, "Acquisition of Stand-up Behavior by a Real Robot using Hierarchical Reinforcement Learning," Proc. of International Conference on Machine Learning, pp. 623-630, 2000.
- [7] T. Watanabe and T. Sawa, "Instruction for Reinforcement Learning Agent Based on Sub-Rewards and Forgetting", Proc. of the IEEE World Congress on Computational Intelligence, FUZZ-IEEE 2010, pp.1984-1990, 2010.
- [8] T. Sawa and T. Watanabe, "A Study on Evaluation of Eligibility Trace in Instruction Method for Fuzzy Q-learning Agent," Proc. of Joint 5th International Conference on Soft Computing and Intelligent Systems and 11th International Symposium on Advanced Intelligent Systems, pp.310-315, 2010.
- [9] Yu Liu, Zheng Qin, and Zhewen Shi, "Hybrid Particle Swarm Optimizer with Line Search", Proceedings of the 2004 IEEE International Conference on Systems, Man & Cybernetics, pp. 3751-3755
- [10] R. A. Jacobs, M. I. Jordan, and A. G. Barto, "Task Decomposition through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks," Neural Computation, Vol.3, pp.79-87, 1991.
- [11] G. Auda and M. Kamel, "Modular Neural Networks: A Survey," International Journal of Neural Systems, Vol.9, No.2, pp.129-151, 1999.
- [12] T. Watanabe and Y. Takahashi, "Hierarchical Reinforcement Learning Using A Modular Fuzzy Model for Multi-Agent Problem," Proc. of the 2007 IEEE International Conference on Systems, Man, and Cybernetics, pp.1681-1686, 2007.
- [13] A. Stout, G. D. Konidaris, and A. G. Barto: "Intrinsically motivated reinforcement learning: A promising framework for developmental robot learning," Proc. of the AAAI spring symposium workshop on developmental robotics, 2005.
- [14] Y. Tanaka and M. Ishikawa: "Reinforcement Learning with Internal Rewards Based on Error in a Grid-based Map," IEICE Technical Report, NC2008-151, pp.285-290, 2009.
- [15] P. Y. Glorennec, "Reinforcement Learning: Overview," Proc. of ESIT, pp. 17- 35, 2000.
- [16] D. Gu and H. Hu, "Fuzzy Multi-Agent Cooperative Q-Learning," Proc. of the 2005 IEEE International Conference on Information Acquisition, pp.193-197, 2005.