Neighboring Crossover to Improve GA-Based Q-Learning Method for Multi-Legged Robot Control

Tadahiko Murata
Faculty of Informatics,
Kansai University
2-1-1 Ryozenji, Takatsuki, Osaka 569-1095, Japan
+81-72-690-2429

murata@res.kutc.kansai-u.ac.jp

Masatoshi Yamaguchi
Computer Science Major
Kansai University Graduate School
2-1-1 Ryozenji, Takatsuki, Osaka 569-1095, Japan
+81-72-690-2429

fb5m113@edu.kutc.kansai-u.ac.jp

ABSTRACT

In this paper, we propose a crossover method to improve a GA-based Q-learning method for controlling multi-legged robots. As a GA-based Q-learning method, we employ a method called "Q-learning with Dynamic Structuring of Exploration Space Based on Genetic Algorithm (QDSEGA)". We propose a crossover for QDSEGA, and a method to reward a robot in Q-learning in order to follow a moving target. Simulation results clearly show the effectiveness of the proposed methods.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning – *Knowledge acquisition*

General Terms: Algorithms, Performance.

Keywords

Q-learning, Q-table restructuring by genetic algorithm, multi-legged robot control.

1. INTRODUCTION

Recently reinforcement learning has received much attention as a learning method [2,3]. The Q-learning [3], known as one of effective reinforcement learning, has difficulty in accomplishing learning tasks when the size of action-state space is large. To overcome this problem, Ito and Matsuno [1] proposed a GA-based Q-learning method called QDSEGA. In their algorithm, a genetic algorithm is employed to reconstruct an action-state space which is learned by Q-learning. That is, the size of the action-state space is reduced by the genetic algorithm in order to apply Q-learning to the learning process of that space. Since the way to apply a genetic algorithm seems so straightforward, we propose a crossover operation for QDSEGA. Furthermore we try to modify the learning algorithm of QDSEGA to follow the moving target.

2. PROPOSED METHODS FOR QDSEGA

We propose a crossover operation for the multi-legged robot control problem (MRC problem) employed in [1]. Q-table represents a subset of the action-state space to control a 12-legged robot. In QDSEGA, individuals are used for the both of actions and states. Fig. 1 shows an example of a Q-table created by n

individuals. In this problem, each individual is used twice in the

Q-table. "Current state" indicates the current positions of 12 legs. "Action" indicates a next position of 12 legs after a certain action. For example, if S1 is the current position of 12 legs and A2 is selected as an action, the 12-legged robot moves its 12 legs to become the next position as S2.

2.1 Neighboring Crossover

The crossover employed in [1] causes drastic change of legs' position because randomly selected bits are changed between two individuals. We propose a crossover between similar parent individuals to avoid a drastic change of legs. We define the similarity by the number of the same genes in the same locus of a chromosome. The similarity is defined by k_{sim}. Thus, the crossover is applied among similar individuals more than k_{sim}.

2.2 Modified Reward for Moving Target

In the previous study, the target of the MRC problem was fixed. That is, the target stays at the same place. We consider a moving target as a variant problem of MRC problems. In order to let the multi-legged robot follow a moving target, we specify the reward $r(s,a)$ for Q-learning as follows:

If $-45[\deg] \leq L_z < -15[\deg]$ and $d(t-1) - d(t) \geq 0$ then
$$r(t) = 50 \cdot (d(t-1) - d(t)), \tag{1}$$
If $-45[\deg] \leq L_z < -15[\deg]$ and $d(t-1) - d(t) < 0$ then
$$r(t) = 200 \cdot (d(t-1) - d(t)), \tag{2}$$
If $15[\deg] < L_z \leq 45[\deg]$ and $d(t-1) - d(t) \geq 0$ then
$$r(t) = 50 \cdot (d(t-1) - d(t)), \tag{3}$$
If $15[\deg] < L_z \leq 45[\deg]$ and $d(t-1) - d(t) < 0$ then
$$r(t) = 200 \cdot (d(t-1) - d(t)), \tag{4}$$
If $-15[\deg] \leq L_z \leq 15[\deg]$ then
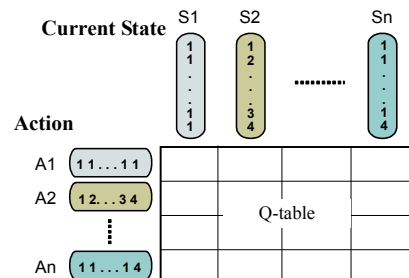$$r(t) = 100 \cdot (d(t-1) - d(t)), \tag{5}$$

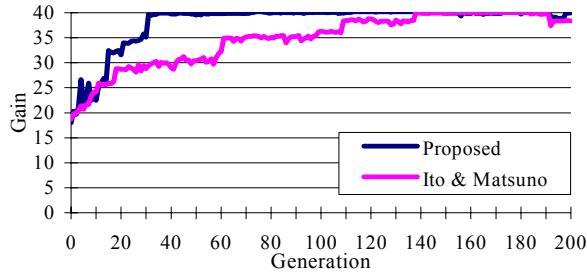**Figure 1. An example of Q-table created by *n* individuals.**

Figure 2. Average gain over the generation.

Table 1. Effect on the conventional method for moving target.

Conventional	Without adjustment		
Target position	Left	Ahead	Right
Ratio (%)	21.7	50.7	27.5

Conventional	With adjustment		
Target position	Left	Ahead	Right
Ratio (%)	17.8	63.4	17.8

Table 2. Effect on the proposed method for moving target.

Proposed	Without adjustment		
Target position	Left	Ahead	Right
Ratio (%)	21.0	50.6	28.2

Proposed	With adjustment		
Target position	Left	Ahead	Right
Ratio (%)	21.9	63.3	14.8

else ($L_z < -45[\deg]$ or $45[\deg] < L_z$ or stuck) then
$$r(t) = -100 . \qquad (6)$$

In the above equations, L_z is the direction of the target in degrees (not in radian), and $d(t)$ is the distance from the head of the robot to the target at the time t. L_z is zero [deg] if the target locates just ahead of the robot.

3. SIMULATION
3.1 Parameter Settings
We specified the parameter settings in the Q-learning, the genetic algorithm, and the multi-legged robot simulation model as follows:

[Q-learning] The number of trials for each Q-table: 3,000 trials, The number of steps in each trial: 100 steps, Learning rate: $\alpha = 0.7$, Discounting rate: $\gamma = 0.7$. Temperature of Bolzmann distribution: $T = 150 \times e^{(-0.002 \times trial)+1}$.
[Genetic Algorithm] The number of individuals: 50, The number of generations: 200, Crossover probability: $P_c = 0.5$, Similarity for the crossover: $k_{sim} = 4$, Mutation probability: $P_m = 0.2$,

Weigh value: $k_f = 200$, Ratio of reward to penalty: $p = 0.1$, Weight values: $w_1 = 0.6$, $w_i = 0$ $(i = 2,..., N_s -1)$, $w_{N_s} = 0.4$.

3.2 Effect of Neighboring Crossover
We applied the conventional method [1] and the proposed method with the neighboring crossover in Subsection 2.2. Fig. 2 shows the average "Gain" over the generation. We defined "Gain" as the gaining distance of the robot toward the target where the maximum gain of the problem was 40 during the allotted time. From Fig. 2, we can see that the proposed crossover clearly enhanced the performance of QDSEGA. We calculated the average number of generations when the robot gained more than 38 in average over 100 experiments. The conventional method required 108.5 generations, and the proposed method 56.2 generations to gain more than 38. We can see that the number of generations was drastically reduced by the proposed crossover in QDSEGA.

3.3 Effect of Modified Reward
Tables 1 and 2 show the results on the MRC problem with the moving target. We applied our modified reward to QDSEGA [1] and QDSEGA with the proposed neighboring crossover. In these tables, "Target position" indicates that the ratio of target positions from the robot. That is, when the target locates ahead of the robot within -15 [deg] to 15 [deg], the number of "Ahead" is incremented, and "Max gain" shows that the average maximum gain after the target starts to move. From these tables, we can see that the ratio of "Ahead" was increased by the introduction of the adjustment of rewards in Q-learning in both algorithms. Through the proposed modification of the reward we could improve the performance of the multi-legged robot to follow the moving target.

4. CONCLUSION
In this paper, we proposed the neighboring crossover that improves the performance of QDSEGA for the multi-legged robot control problems. By introducing the neighboring crossover, we could reduce the number of generations to attain the target gain in the problem. We also show the fine effect of the modified reward for the problems with the moving target. When the target is moving in the problem, the robot needs to learn how to adjust its position toward the target. That is, it should learn how to adjust its direction toward the target, and how to convey its body toward the target. By adjusting the reward, we could improve the control of the multi-legged robot.

5. ACKNOWLEDGMENTS

6. REFERENCES
[1] Ito, K. and Matsuno, F. A study of reinforcement learning for the robot with many degrees of freedom –Acquisition of locomotion patterns for multi legged robot-, In *Proc. of IEEE Int'l Conf. on Robotics and Automation*, pp. 392-397, 2002.

[2] Sutton, R.S. *Reinforcement Learning: An Introduction*. The MIT Press, 1988.

[3] Watkins, C.J.C.H. and Dayan, P. Technical note q-learning. *Machine Learning*, 8: pp. 279-292, 1992.