Q-learning based on hierarchical evolutionary mechanism

MASAYA YOSHIKAWA[(1)], TAKESHI KIHIRA[(2)], HIDEKAZU TERAI[(2)]

[(1)] Department of Information Engineering
Meijo University
1-501, Tenpaku, Nagoya, Aichi, 468-8502,
JAPAN

[(2)]Department of VLSI System Design
Ritsumeikan University
1-1-1, Nojihigashi, Kusatsu, Shiga, 525-8577
JAPAN

Abstract: - Reinforcement learning is applied to various fields such as robotics and mechatronics control. The reinforcement learning is an efficient method to control in unknown environment. This paper discusses a new reinforcement learning algorithm which is based on Genetic Algorithm and has a hierarchical evolutionary mechanism. The proposed learning algorithm introduces new adaptive action value tables and it enables sharing knowledge among agents effectively. Regarding sharing knowledge among agents, the knowledge is inherited not only across the generations, but also in one generation. As a result, the proposed algorithm achieves effective learning, and realizes robustness learning. Computational simulations using the pong simulator which executes table tennis prove the effectiveness of the proposed algorithm.

Key-Words: - Q-Learning, Genetic Algorithm, Sharing knowledge among agents, Robustness of learning, Hierarchical evolutionary mechanism

1 Introduction

Recently, reinforcement learning is applied to various fields such as robotics and mechatronics control [1]-[11]. The reinforcement learning is an efficient algorithm to acquire adaptive behavior of the agent without a priori knowledge of the environment. Q-learning which was proposed by Watkins *et al.* [1], is the most basic learning algorithm in reinforcement learning.

Conventional Q-learning algorithm deals with states and action as discrete data. However, input data as environment information is usually continuous data. As a result, discretization of the input data is important. Thus, the unit of discretization influences learning efficiency. Three parameters (learning rate, discount rate, and searching rate) required by Q-learning also affect the learning efficiency.

In this paper, we propose a novel learning algorithm to optimize these parameters automatically. The proposed learning algorithm adopts Q-learning as base algorithm, and achieves discretization of the information from environment using Genetic Algorithm (GA)[12]-[21]. Moreover, we also propose a new technique to merge action value tables of agents for sharing knowledge. Regarding sharing knowledge among agents, the knowledge is inherited not only across the generations, but also in one generation. As a result, the proposed algorithm achieves not only effective learning but also robustness learning. The target model is Pong simulator which plays table tennis and the objective of learning is to obtain the behavior of rally of the table tennis in this paper. Experiments prove the effectiveness of the proposed learning algorithm.

2 Preliminaries

2.1 Q-learning

Q-learning is most famous learning algorithm and is defined by

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + a[\gamma_{t+1} + \gamma \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (1)$$

Where, α is the learning rate, γ is the discount rate, s_t is the current state, s_{t+1} is the next state, α_t is the action in state s_t, t+1 is the received reward to execute α_t in state s_t.

2.2 Related work

Many works for reinforcement have done using Q-learning. Examples of Q-learning are found Y.Maeda[9], C.F.Juang[10], K.S.Hwang *et al.*[11], and so on. Y.Maeda proposed an adaptive Q learning method tuned learning parameters by fuzzy rules and showed several results of artificial ants simulation. C.F.Juang proposed combination of online clustering and Q-value based genetic algorithm learning scheme for fuzzy system design with reinforcement and the feasibility of the proposed algorithm was demonstrated through simulations in cart-pole balancing, magnetic levitation, and chaotic system control problems with only binary reinforcement signals. K.S.Hwang *et al.* proposed a self-learning cooperate strategy for robot soccer systems. The strategy enables robots to cooperate and coordinate with each other to achieve the objectives of offense and defense.

However, no previous works have, to our knowledge, introduced the hierarchical evolutionary mechanism for sharing knowledge among agents.

3 Genetic Learning Algorithm

3.1 Agent model

Fig.1 shows the composition of the agent in this paper. The agent receives the environment information data, which are continuous data, as input data. The agent executes discretization of the received information according to pre-defined unit. The agent recognizes environment using the discrete data, that is, the agent creates a state for input to learning module. And then, the agent has an action value table as shown in Table.1. It consists of pair of state and action.

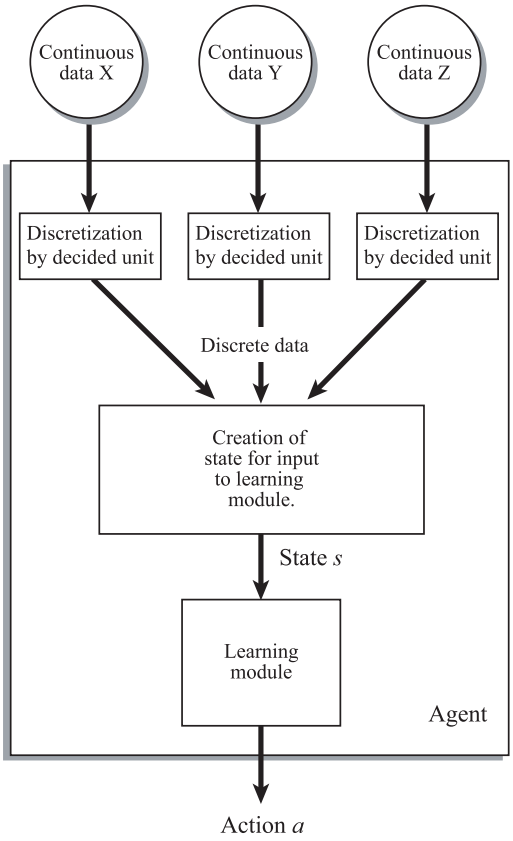


Fig.1 Example of agent model

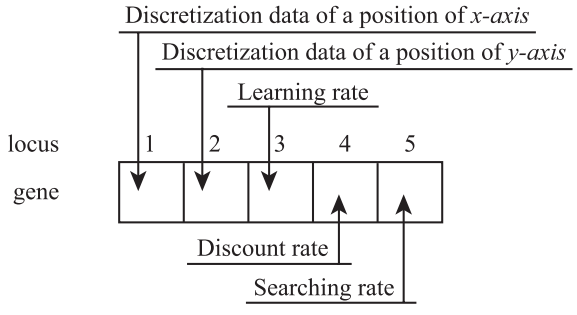
Table.1 Example of action value table

State s \ Action a	a_0	a_1	a_2	…	a_i
s_0	$Q(s_0, a_0)$	$Q(s_0, a_1)$	$Q(s_0, a_2)$		$Q(s_0, a_i)$
s_1	$Q(s_1, a_0)$	$Q(s_1, a_1)$	$Q(s_1, a_2)$	…	$Q(s_1, a_i)$
s_2	$Q(s_2, a_0)$	$Q(s_2, a_1)$	$Q(s_2, a_2)$		$Q(s_2, a_i)$
…	…	…	…		…
s_i	$Q(s_i, a_0)$	$Q(s_i, a_1)$	$Q(s_i, a_2)$		$Q(s_i, a_i)$

3.2 Learning procedure

The proposed learning algorithm optimizes the parameters on Q-learning by Genetic Algorithm. It is a powerful optimization algorithm, which is based on the mechanism of biological evolution. It needs to model for adopting GA at optimization problem.

The proposed algorithm creates a state *s* for input to learning module using two kind of information. One is discretization data of a position of *x*-axis, the other is that of *y*-axis. Thus, an individual composes of discretization data of a position of *x*-axis, that of *y*-axis, learning rate, discount rate, and searching rate. Fig.2 shows an example of coding. The proposed learning procedure will be explained concretely using Fig.3 as an example.



Fig,2 Example of coding

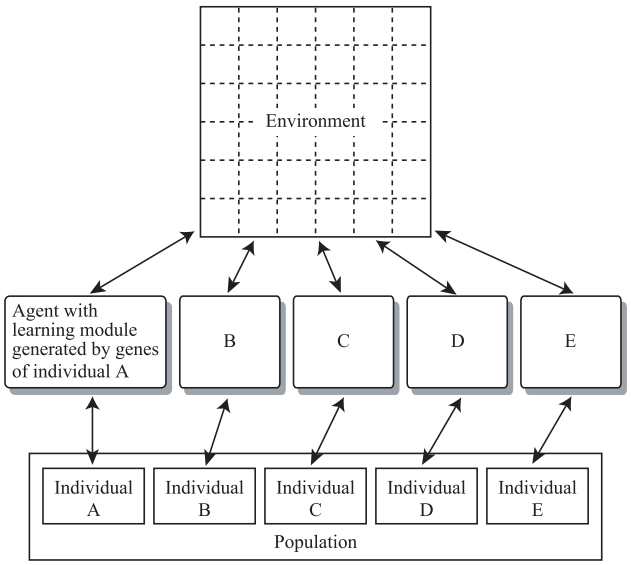


Fig.3 Example of learning procedure in one generation

The example represents one generation. As first step of learning, the learning module is created by using of gene's information of one individual.

Next, an agent with the learning module learns for pre-defined period. And then, the reward, which is obtained by the learning, is assigned to fitness of the individual. These operations are executed to all individuals.

Regarding genetic operation, it consists of selection operation and generating operation. The generating operation functions as crossover and mutation operator of ordinary GA and means creating new individuals. Here, there is a case of which fitness value has minus value, because the proposed algorithm adopts reward obtained by learning as fitness. As a result, the roulette wheel selection operator can not be used as the selection operator. The proposed algorithm adopts elitism and random replacement as selection operator of ordinary GA. Fig.4 shows an example of selection operator in the proposed algorithm.

The proposed algorithm introduces two techniques as generating operation. One is copy processing of genes at random, and the other is processing of merging with action value tables. The copy processing is executed for individuals of random replacement. Fig.5 shows the copy processing. It enables to copy genes of individuals with high fitness value, because the proposed algorithm adopts elitisum.

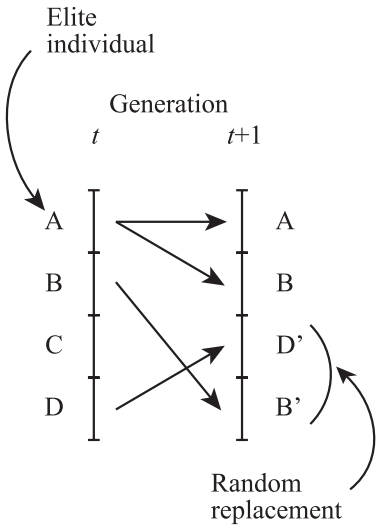


Fig.4 Example of selection procedure

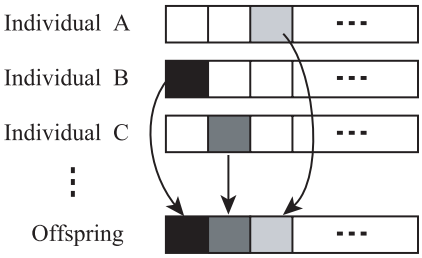


Fig.5 Example of copy processing for generating offspring

3.3 Adaptive action value table

From the viewpoint of the efficiency improvement of learning, each agent should share the knowledge obtained by the learning. However, it is necessary to adjust the size of the table, because the size of each action value table is difference.

Therefore, the proposed algorithm introduces a new technique to merge the tables. The value corresponding to the frequency of which each agent took the action of a in s is assigned to $Q(s,a)$. Next, the tables are merged using weighted average. A concrete procedure is as follows.

$$Q(s,a) \text{ of } Agent1 \leftarrow agent1Q \times \frac{expNumAgent1}{expNumAgent1 + expNumAgent2} + agent2Q \times \frac{expNumAgent2}{expNumAgent1 + expNumAgent2} \quad (2)$$

$$Q(s,a) \text{ of } Agent2 \leftarrow agent1Q \times \frac{expNumAgent1}{expNumAgent1 + expNumAgent2} + agent2Q \times \frac{expNumAgent2}{expNumAgent1 + expNumAgent2} \quad (3)$$

Here, expNumAgent1 is a frequency of which Agent1 took the action of a in s. Similarly, expNumAgent2 is a frequency of which Agent2 took the action of a in s. The agent1Q indicates $Q(s,a)$ of Agent1, and agent2Q indicates that of Agent2.

3.4 Modification of table size

Each agent recognizes his environment, which is a state s for input to the learning machine, based on information of discretization data.

The proposed learning algorithm deals with a state as three dimensions, because input information has three kinds. That is, information x, y, and z correspond to x-axis, y-axis, and z-axis respectively as shown in Fig.6. Thus, the state s for input to the learning machine is represented by coordinates (x,y,z). State s is concretely as follows.

State s of input to learning module
= Continuous data X
+ (Continuous data Y
× # partitions of continuous dataX)
+ (Continuous data Z
× # partitions of continuous data X
× # partitions of continuous data Y) (4)

Here, #partition indicates the maximum value of discretization data. For instance, #partition is 4, if the range of information x has [0,100] and the unit size is 30.

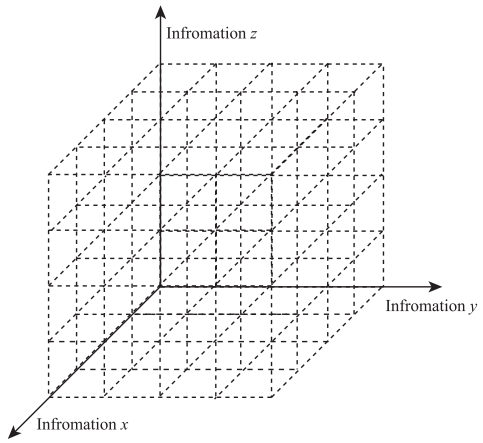


Fig.6 Example of a state as three dimensions

Regarding modification of action value tables, it is important to correspond between discretization data obtained by new unit and that by previous unit. The modification procedure is as follows.

$$temp \leftarrow \frac{\#\text{ partitions of continuous data in new unit}}{\#\text{ partitions of continuous data in add unit}} \quad (5)$$

$$\text{Continuous data in add unit} \leftarrow \begin{cases} temp < 1.0 & \dfrac{\text{Continuous data in new unit}}{temp} + 1 \\ temp \geqq 1.0 & \dfrac{\text{Continuous data in new unit}}{temp} \end{cases} \quad (6)$$

As a result, state s_{new} obtained by new unit is similar to state s_{old} obtained by previous unit.

3.5 Hierarchical learning

Regarding sharing method of knowledge, the proposed algorithm introduces a hierarchical evolutionary mechanism. That is, the knowledge is inherited in one generation. Fig.7 shows the hierarchical evolutionary mechanism.

The result of learning of individual i is used as an initial value of action value table of individual i+1. That means knowledge is propagated in one generation in the order of studying the individual. It indicates the knowledge's propagation in one generation according to the order of learning. The knowledge's propagation achieves effective sharing knowledge.

4 Experiments

4.1 Experimental conditions

In order to evaluate the proposed algorithm, it is implemented on a pong simulator.

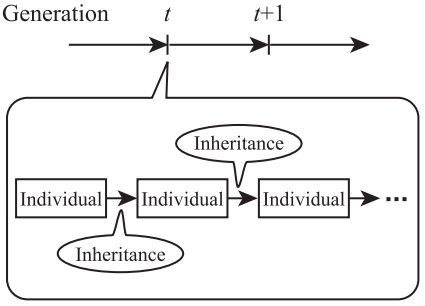


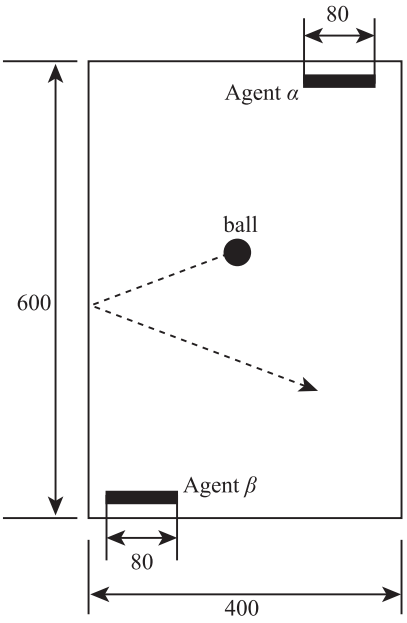
Fig.7 Hierarchical evolutionary mechanism

The specification of the pong simulator is as follows.

1. There is no concept of height.
2. The ball is placed with the speed in the direction of *x* and that of *y* at the center of the field as an initial state. The range of speed in the direction of *x* and *y* have (-50, 50) respectively.
3. The ball returned by the agent accelerates or decelerates in the direction of *x* and that of *y* at random.
4. The play finishes when the ball touches the wall which is behind the agent.
5. The action of agent consists of "move to right", "move to left", and "don't move".
6. It is called one episode from the initial state to the play's finish.

Fig.8 shows the environment of pong simulator. The size of field is 400 x 600, and that of racket is 80. Fig.9 shows relationship between agent and pong simulator. The agent receives information from the pong simulator, and returns the pong simulator the result of learning. The pong simulator changes the state of the environment, after receiving information on behavior selection from the agent. The simulator executes the transition by repeating these operations. Thus, pong simulator consists of two units which are simulation unit and control unit for agents.

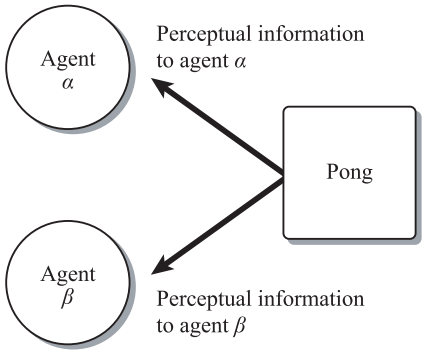
All experiments run on the same platform. The specification of platform is shown in Table.2.



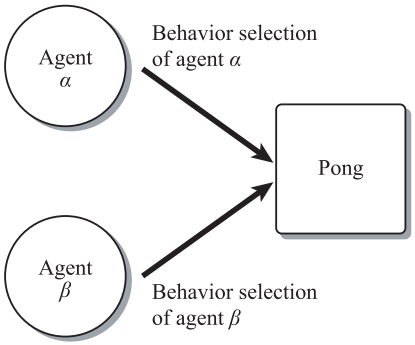
Fig,8 Example of Pong simulator

4.2 Simulation results: Evaluation of learning model

First, we experiments to evaluate the learning model. The parameters are decided form view points of accuracy of solutions. And, the proposed learning algorithm adopts three kinds of reward. Table.3 shows the reward of learning. Table.4 shows parameters using in the experiments.



(1) Perceptual information to agents



(2) Behavior selection of agents

Fig.9 Relationship between agent and pong simulator

Table.2 Specification of simulation platform

Processor	2GHz Intel Core2 Duo
OS	Mac OS X Version 10.4.11
Compiler	i686-apple-darwin8-g++-4.0.1(GCC)

Table.3 Reward of learning

Behavior	Reward
Agent returns a ball.	100
The ball touches the wall which is behind the agent.	-100
Agent moves outside of the range.	-1.0

In the proposed learning algorithm, these parameters are acquired by GA automatically. Fig.10 shows experimental results. A horizontal axis represents the number of episodes and a vertical axis represents the number of rally in Fig.10. The learning completed within 500 episodes. We can see the effectiveness of the proposed learning model from Fig.10.

4.3 Simulation results: Evaluation of characteristic of GA

Next, we experiments to evaluate the characteristic of GA. Parameters of GA are shown in Table.5. Fig.11 and Fig.12 show the simulation results.

Table.4 Parameters for learning

Unit for discretization of coordinates x of agent	80
Unit for discretization of coordinates x of ball	80
Unit for discretization of coordinates y of ball	80
Previous unit for discretization of coordinates x of ball	80
Previous unit for discretization of coordinates y of ball	80
Learning rate	0.1
Discount rate	0.9
Searching rate	0.0001

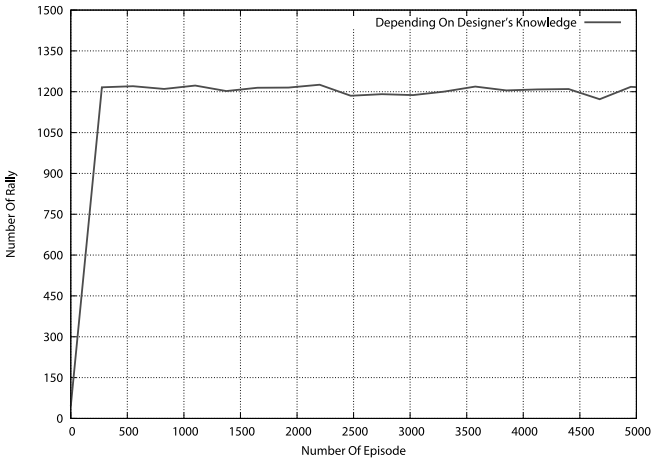


Fig.10 Simulation result of relationship between evaluation vale (fitness) and generations

A horizontal axis represents generations and a vertical axis represents evaluation value in Fig.11. A horizontal axis represents generations and a vertical axis represents the number of kinds of individuals in Fig.12.The proposed algorithm achieves effective learning as shown in Fig.11. We can see the reduction the number of kinds of individuals in the tenth generation from Fig.12.

Moreover, Fig.13 shows the ratio of which individuals have the same genes at each generation. Fig.13(1) shows the state of each individual of initial generation, Fig.13(2) shows that of 11th generation, and Fig.13(3) shows that of 41st generation. The transition of state in population indicates the changing from global search to local search in GA.

When Fig.13(2) is compared with Fig.13(3), population has a few kind of individual in 41st generation. It indicates the completion of learning at the generation. It indicates the completion of learning at the generation.

Next, we experiments to evaluate the robustness of the proposed algorithm. The environment is changed every the 50th generation. Fig.14 shows the experimental result. A horizontal axis represents generations and a vertical axis represents evaluation value in Fig.14. And then, Fig.14 shows the state of diversity of the population in changing environment..

Table.5 Parameters for GA

Solution space of unit for discretization of coordinates *x* of agent	[30, 200]
Solution space of unit for discretization of coordinates *x* of ball	[30, 200]
Solution space of unit for discretization of coordinates *y* of ball	[30, 200]
Solution space of previous unit for discretization of coordinates *x* of ball	[30, 200]
Solution space of previous unit for discretization of coordinates *y* of ball	[30, 200]
Solution space of learning rate	[0.1, 0.5]
Solution space of discount rate	[0.5, 1.0]
Solution space of searching rate	[0.01,0.000001]

In Fig.15, a horizontal axis represents generations and a vertical axis represents evaluation value as well as in Fig.12. The proposed algorithm enables to follow to the change of environment, and has achieved enough robustness.

4.4 Simulation results: Evaluation of sharing knowledge technique

Lastly, we experiments to evaluate the proposed sharing knowledge technique. Fig.16 shows the experimental result. It represents the total number of the rally between 100 episodes. Moreover, each transition of parameters, which are target of learning, is presented by from Fig.17 to Fig.24.

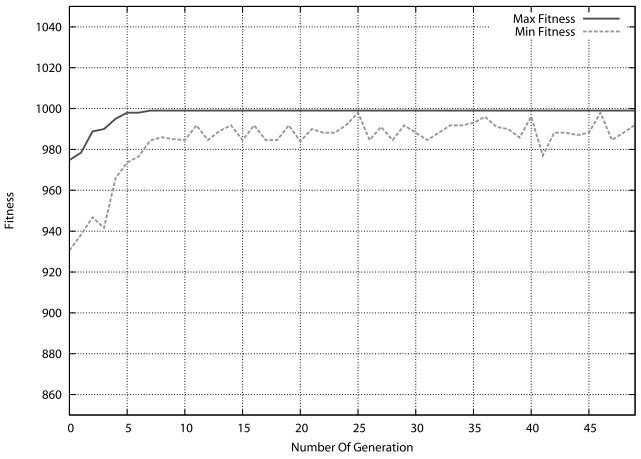


Fig.11 Simulation result of relationship between evaluation vale (fitness) and generation

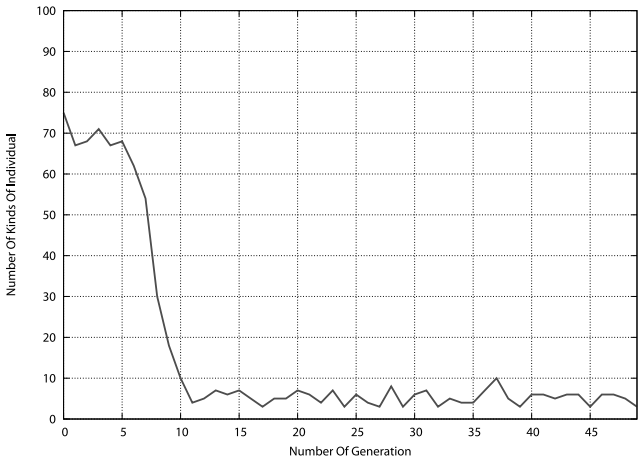
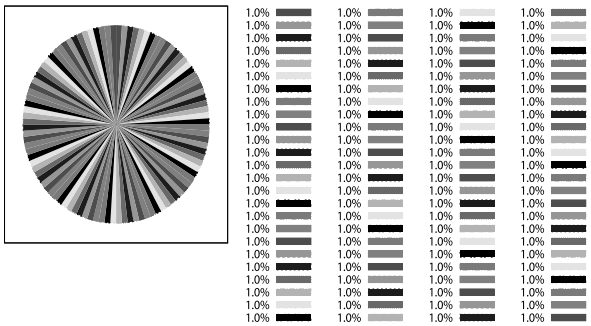
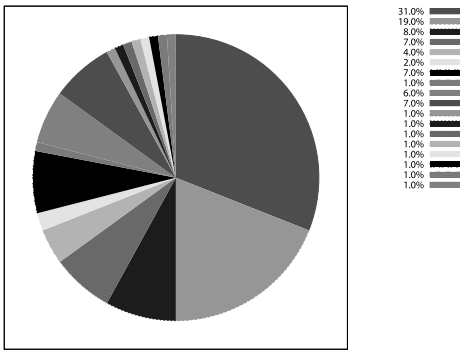


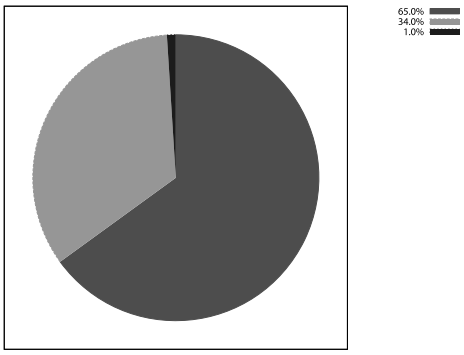
Fig.12 Simulation result of relationship between population versatility and generation



(1) State of population in initial generation



(2) State of population in 11th generation



(3) State of population in 41st generation

Fig.13 Comparison between generations

We can see the proposed learning algorithm improves the performance of learning from Fig.13. However, the learning efficiency of the proposed algorithm is worse than that of Fig.10. We guess the specific case of modification of table size causes the decrease in the learning efficiency. The specific case is the case of which table size is modified from big size to small size, because learning information is lost.

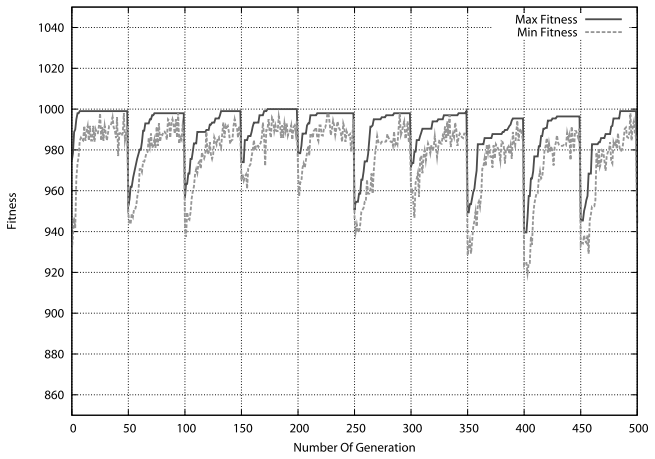


Fig.14 Result of simulation for robustness in changing environment

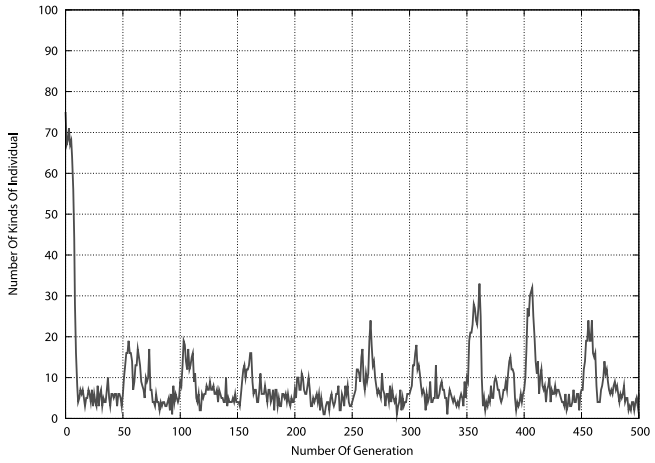


Fig.15 State of diversity of population in changing environment

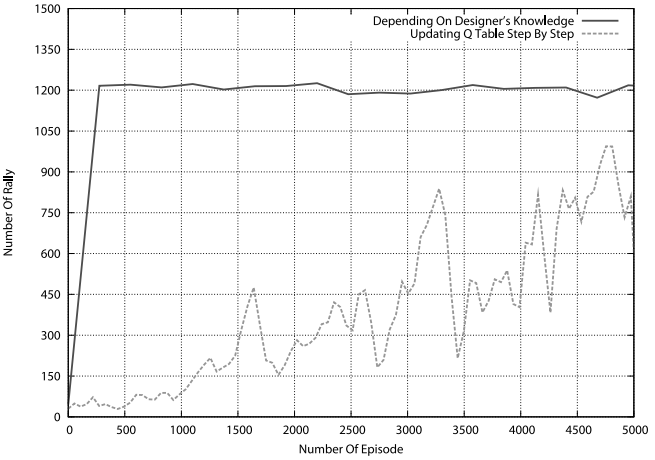


Fig.16 Result of simulation for sharing knowledge

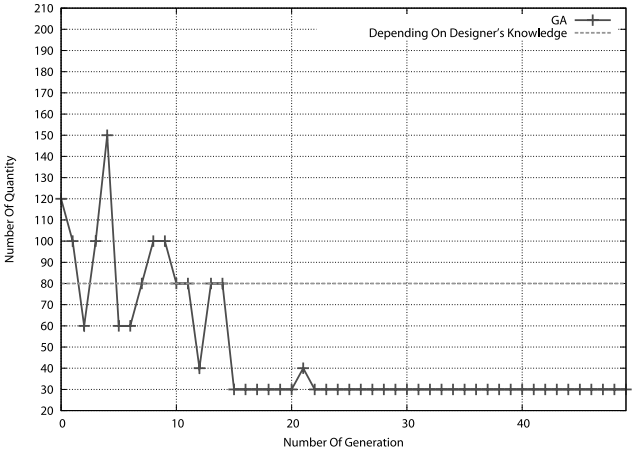


Fig.17 Transition of unit for discretization of coordinates *x* of agent

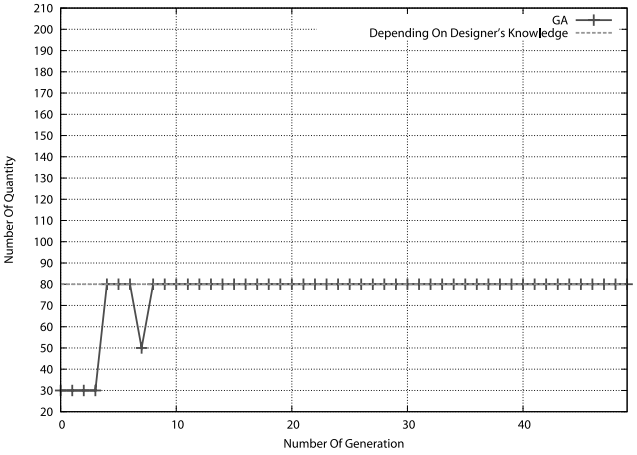


Fig.18 Transition of unit for discretization of coordinates *x* of ball

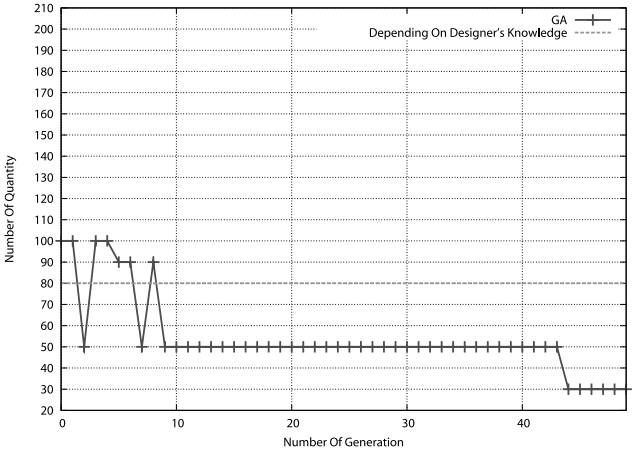


Fig.19 Transition of unit for discretization of coordinates *y* of ball

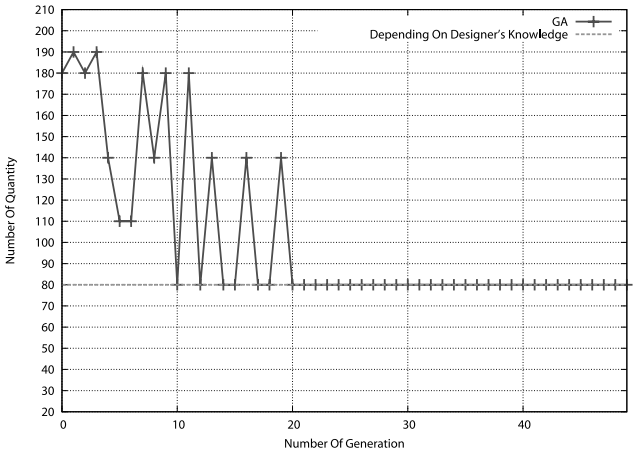


Fig.20 Transition of previous unit for discretization of coordinates *x* of ball

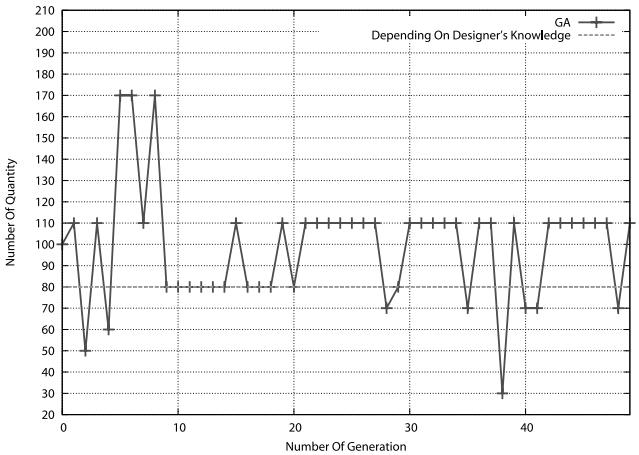


Fig.21 Transition of previous unit for discretization of coordinates *y* of ball

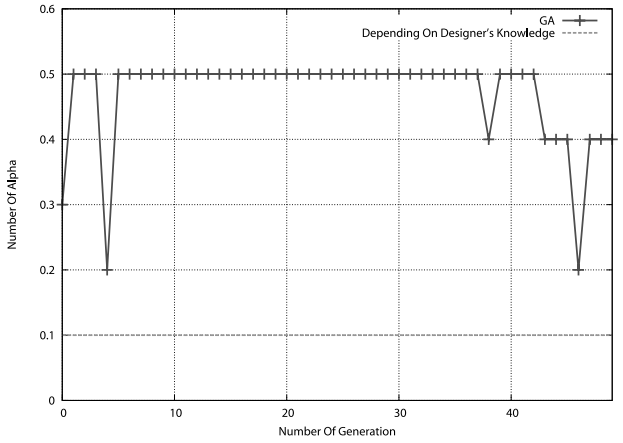


Fig.22 Transition of learning rate of the proposed learning module

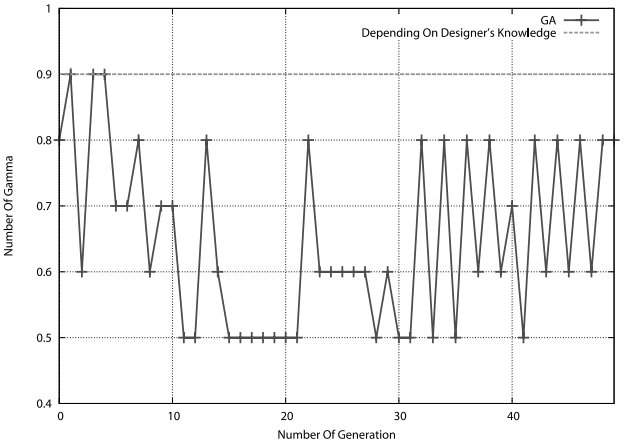


Fig.23 Transition of discount rate of the proposed learning module

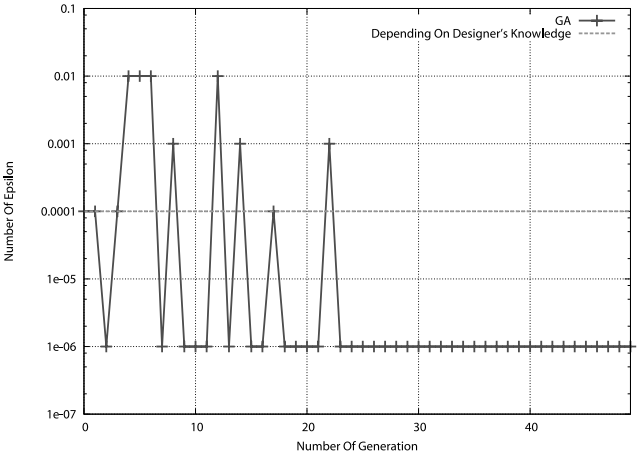


Fig.24 Transition of searching rate of the proposed learning module

5 Conclusion

In this paper, we proposed the new genetic learning algorithm based on hierarchical evolutionary mechanism. The proposed learning algorithm introduced new adaptive action value tables and it enabled sharing knowledge among agents effectively. Regarding sharing knowledge among agents, the knowledge was inherited not only across the generations, but also in one generation. As a result, the proposed algorithm achieved not only effective learning but also robustness learning. Experiments using pong simulator proved the effectiveness of the proposed algorithm.

In relation to future works, simulation in complex environment is the most important priority. We will

also apply the proposed learning algorithm to actual robot control.

Acknowledgments

This study was supported in part by the telecommunication advancement foundation. The authors would like to thank their supports.

References:

[1] C.J.C.H.Watkins, Learning from Delayed Rewards, Phd thesis, University of Cambridege, 1989.

[2] R.S.Sutton, Integrated Architecture for learning, Planning, and Reacting Based on Approximating Dynamic Programming, *Proc. of 7th International Conference on Machine Learning*, pp.216-224, 1990.

[3] S.P.Singh, Transfer of Learning by Composing Solutions of Elemental Sequential Task, *Machine Learning*, pp.323-339, 1992.

[4] R.A.McCallum, Using Transitional Proximity for Faster Reinforcement Learning, *Proc. of 9th International Conference on Machine Learning*, pp.316-321, 1992.

[5] D.Issicaba, *et al.*, Optimal capacitor placement in radial distribution systems by reinforcement learning approach, *WSEAS Transactions on Power Systems*, Issue 8, No.1, pp.1389-1395, 2006.

[6] O.Karaduman, *et al.*, Integrated treatment planning using an intelligent learning for clinical decision support by reinforcement feedback, *WSEAS Transactions on Computers*, Issue 7, No.5, pp.1541-1548, 2006.

[7] A.Abbaspour, *et al*., Emotional reinforcement learning for portfolio selection, *WSEAS Transactions on Systems*, Issue 1, No.3, pp.306-309, 2004.

[8] X.Zhuang, *et al.*, The strategy entropy of reinforcement learning in discrete state space, *WSEAS Transactions on Systems*, Issue 9, No.3, pp.2813-2820, 2004.

[9] Yoichi Maeda, Fuzzy Adaptive Q-learning Method with Dynamic Learning Parameters, *Proc. of Joint 9th IFSA World Congress and 20th NAFIPS International conference*, pp.2235-2240, 2001.

[10] Chia-Feng Juang, Combination of Online Clustering and Q-Value Based GA for Reinforcement Fuzzy System Design, *IEEE Trans. on Fuzzy Systems*, Vol.13, No.3, pp.289-302, 2005.

[11] K.S.Hang, et al., Cooperative Strategy Based on Adaptive Q-Larning for Robot Soccer Systems, *IEEE Transactions on Fuzzy Systems*, Vol.12, No.4, pp.569-576, 2004.

[12] Holland, *Adaptation in Natural Artificial Systems*, the University of Michigan Press (Second edition ; MIT Press)(1992).

[13] Goldberg,D.E, *Genetic algorithms in search optimization*, and machine learning; Addison Wesley,(1989)

[14] M.Yoshikawa, and H.Terai, Genetic Algorithm Engine for Scheduling Problems, *WSEAS Transactions. on Circuits and System*, Issue3, Vol.5, pp.397-407, 2006.

[15] M.Yoshikawa, H.Terai, Dedicated Hardware for scheduling problems using Genetic Algorithm, *Proc. of The 5th WSEAS International Conference on Applications of Electrical Engineering*, pp.208-212, 2006.

[16] M.Yoshikawa, H.Terai, Bus-Oriented Floor-planning Technique Using Genetic Algorithm, *WSEAS Transactions. on Circuits and* System, Issue2, Vol.6, pp.253-258, 2007.

[17] M.Yoshikawa, H.Terai, Constraint-Driven Floorplanning based on Genetic Algorithm, *Proc. of 2007 WSEAS International Conference on Computer Engineering and Applications*, pp.147-151, 2007.

[18] C.Shihchieh *et al.*, The development of a credit scoring system based on case-based reasoning with genetic algorithm applied, *WSEAS Transactions on Computers*, issue3, No.6, pp.394-399, 2007.

[19] N.Kovshov, *et al.*, About one approach for detecting logical dependencies in recognition by precedents based on the genetic algorithm, *WSEAS Transactions on Computers Research*, Issue 2, No.1, pp.152-155, 2006.

[20] S.Javad, Software development for optimum allocation of power system elements based on genetic algorithm, WSEAS Transactions on Power Systems, Issue 7, No.1, pp.1229-1234, 2006.

[21] L.Roseiro, *et al.*, Genetic algorithms and neural networks in optimal location of piezoelectric actuators and identification of mechanical properties, *WSEAS Transactions on Systems*, Issue 12, No.5, pp2911-2916, 2006.