Journal of Intelligent and Robotic Systems **30:** 399–414, 2001.

The Effect of Evolution in Artificial Life Learning Behavior

MAL-REY LEE
Department of Multimedia Information & System, School of Multimedia, Yosu National University, San 96-1, Dunduckdong, Yosu, JunNam, 550-749, Korea; e-mail: mrlee@yosu.ac.kr

HUINAM RHEE
School of Mechanical and Automotive Engineering, Sunchon National University, Maegok-Dong, Sunchon City, Chonnam 540-742, Korea; e-mail: hnrhee@sunchon.ac.kr

(Received 19 October 1999; in final form: 19 July 2000)

Abstract. In this paper, we add learning behavior to artificial evolution simulation and evaluate the effect of learning behavior. Each individual establishes its own neural network with its genetic information. Also, we propose a reward function to take reinforcement learning in a complicated and dynamically-determined environment. When the individual-level learning behavior was introduced, evolution of each simulation model got faster and the effectiveness of evolution was significantly improved. But the direction of evolution did not depend on learning and it was possible to affect the forms of evolution through reinforcement learning. This provides the mechanism that can apply the artificial life technique to various fields.

Key words: genetic algorithm, artificial life, reinforcement learning.

1. Introduction

Artificial evolution is usually to study the artificial life. Learning by artificial evolution is a gene-level instinctive behavior and is inherited by the next generation. An instinctive behavior is modified by means of selection computation in the artificial evolution. An instinctive behavior is determined when the individual is formed and not changed within a generation.

An instinctive behavior cannot adapt instantly to the changes in the environments. It cannot reflect the previous experiences to decide the next behavior. Therefore, in order to adapt effectively to the changes of the environments and to utilize the previous experience, we need to introduce the concept of learning in artificial evolution. With the introduction of the learning behavior, the artificial organism can utilize its experiences to decide how to behave and also can actively change its compatibility rate. Nolfi introduced the individual-level learning behavior to study the artificial evolution using the learning technique. Nolfi [13] showed an experimental result using the artificial life simulation. The artificial life simulation was performed in a 2-dimensional lattice structure. The artificial organism, which looks for foods, was represented by a multi-layered neural network including hidden layers. The inputs into the neural network were the direction and distance to the nearest food. The output from the neural network is the behavior of the artificial organism. The individual level learning behavior employed the back propagation algorithm using the error between the output and object patterns. However, Nolfi's work is limited to a simple environment case where the exact object pattern can be calculated for the behavior of artificial organism.

Gruau [6] worked on the cross evolution through competitions among groups of artificial organisms. The artificial organisms established their own neural network with their genetic information and they competed with one another by a simple rule. Through the cross evolution among the groups of artificial life, various competition strategies appeared. However, Gruau's work did not include the learning technique.

In this paper, we add learning behavior to artificial evolution simulation and study the evolution characteristics of artificial organisms having the function of learning behavior in a complicated and dynamically-determined environment where the exact object pattern cannot be obtained. In order to do this, we introduce the learning behavior in the experiment of competition among organism groups having a complicated environment and various behaviors.

2. Artificial Organism Learning Algorithm

In this Section, we propose a compensation model which generates reinforcement information for learning in a complicated and dynamically-determined environment.

2.1. REWARD MODEL

In reward model 1, the artificial organism considers all others except itself as environments. Therefore, if the individual successfully attacks the environment, it receives a (−) reinforcement signal. The purpose of reward model 1 is to activate the artificial organism's movement and to induce its offensive spirit, therefore, to allow for more competition strategies to appear. Reward model 1 computes the reinforcement signal after considering the loss and gain at the individual level. The default value of reward model 1 is (+1). The behavior is reinforced when the artificial organism does not interact with the environment. Behavior reinforcement learning is performed in case of the artificial organism's simple movement or when it is attacked by the environment, the strength of behavior constraint learning is varied according to the number of attacks. When the attack is successful, a higher reinforcement signal of (+2) (default value + behavior judgement value) is provided for balancing between the behavior reinforcement and constraint learning. For an unsuccessful attack behavior, (−) value is added to the default value, therefore, no learning occurs. The movement and attack cannot take place simulta-

```
Step 1: Initialize the reinforcement signal to the default value
Step 2: If (the attack is successful), then the reinforcement signal =
        reinforcement signal +1
Step 3: If (the attack is unsuccessful), then the reinforcement signal =
        reinforcement signal −1
Step 4: If (being attacked), then reinforcement signal = reinforcement
        signal = reinforcement signal −(number of being attacked)
```

Figure 1. Computation algorithm of reward model 1.

```
Step 1: Initialize the reinforcement signal to the default value
Step 2: If (enemy's attack is successful), then reinforcement signal =
        reinforcement signal +3
Step 3: If (friend's attack is successful), then reinforcement signal =
        reinforcement signal −1
Step 4: If (failed attack), then reinforcement signal = reinforcement
        signal −1
Step 5: If (failed movement), then reinforcement signal = reinforcement
        signal −1
Step 6: If (being attacked), then reinforcement signal = reinforcement
        signal −(number of being attacked)
```

Figure 2. Computation algorithm of reward model 2.

neously. For example, if the artificial organism moves ahead once and is attacked twice by the enermy, as a result, the reinforcement signal is (-1) $(= 1 + 0 - 2)$.

In reward model 2, the artificial organism divides the environment into the friend and the enemy one. When the attack on the environment is successful, the reinforcement signal is $(+3)$ in case of the enemy, and $(-)$ in case of the friend. This is to balance the reinforcement signal of the enemy's successful attack behavior and the constraint signal of being attacked behavior. In reward model 2, the default value of reinforcement signal, which was the cause of unnecessary learning, is changed to 0. The constraint signal for behaviors of failed movement or attack is a (-1) reinforcement signal. Also, for behaviors that induce the attack of the environment, a constraint signal of (− number of being attacked) is generated to control the strength of the constraint effect. In reward model 2, constraint learning for behaviors of the failed movement and failed attack is performed. This induces the organism to behave in a more exact and reasonable way. It also induces the artificial organism's attack behavior by eliminating unnecessary reinforcement learning for a simple movement or stop behavior.

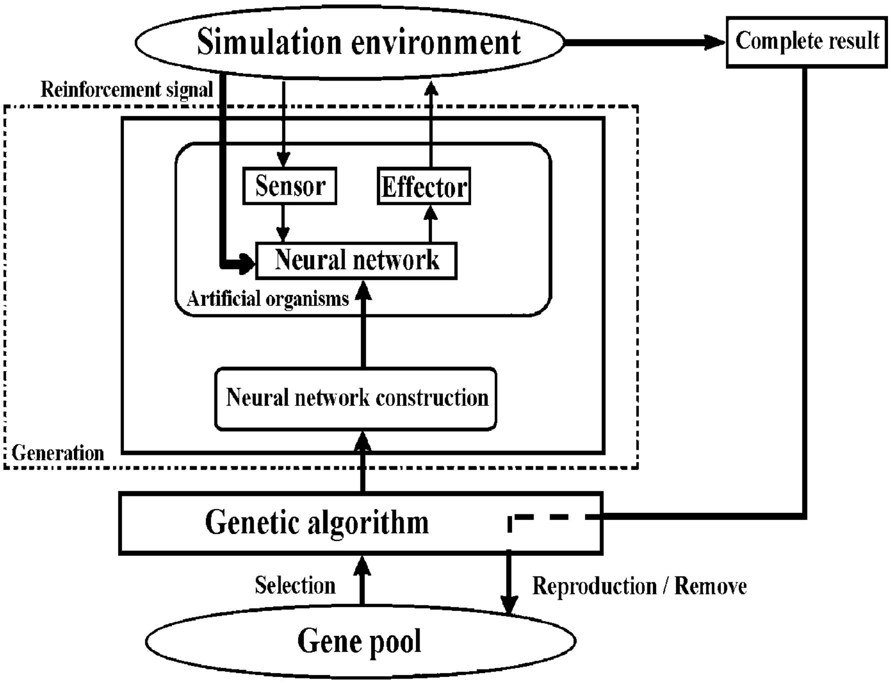


Figure 3. General structure of the simulator.

3. Artificial Life Learning

In this section, we discuss the system structure of artificial life simulation and the simulation model, as well as the learning method.

3.1. ARTIFICIAL LIFE SIMULATION SYSTEM

3.1.1. *Structure of the System*

The structure of the simulation is shown in Figure 3. While the artificial organism interacts with the environment, the artificial organism receives a reinforcement signal from the environment for reinforcement learning of the neural network.

3.1.2. *Environment*

The competition environment has a 12 × 5 lattice structure and is continuous without any boundary. Initially two competing groups are located arbitrarily on the left- or right-hand side in the competition environment. If the movement of artificial organisms has absolute directions, such as east-west-north-south, in the competition environment, the results would be sensitive to the initial positions of the groups. Therefore, each artificial organism has directions of forward, backward,

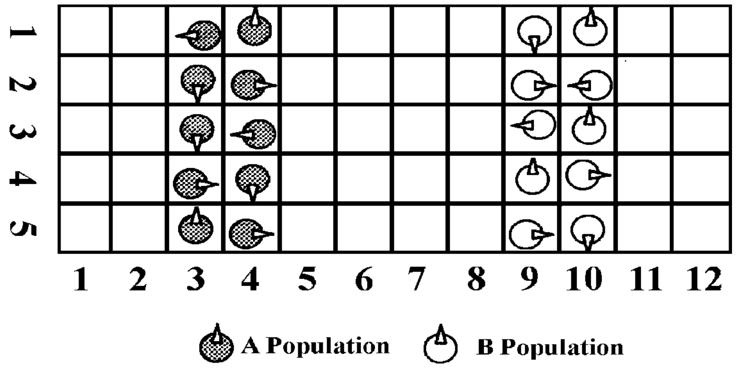


Figure 4. Initial locations of the two populations.

left and right, based on its head direction. In the competition environment there exist only two populations selected for competition, and no obstacle or food. Initial positions of the two competitive populations are sequentially located in the predetermined lattice. As shown in Figure 4, A and B populations are located at 3/4 and 9/10 columns, respectively. To avoid a static behavior pattern due to the initial positioning of the populations, the initial head directions are arbitrarily determined.

3.1.3. *Competition Rule*

Artificial organisms may move forward, backward, in the left- and right-direction in the competition environment. They attack and compete with artificial organisms of the opposite population. The artificial organisms of the two populations move alternatively one at a time. Initially, a population starts to move first. One step is defined as “every artificial organism of the two populations moves once”. After a few steps, the winner is determined by computing the compatibilty of each population. The artificial organism does not die when it is attacked one time. Rather the artificial organism has the initial life power. Whenever it is attacked, the life power decreases by a specific amount and it dies when the life power becomes zero. This method allows for a more realistic and reasonable competition mechanism excluding any accident.

A detailed competition rule is presented in Table I.

3.2. REPRESENTATION OF THE ARTIFICIAL ORGANISM

The neural network of the artificial organism has 9 output nodes. Each output node means a primitive behavior of the artificial organism listed in Table II.

3.2.1. *Genetic Information*

For the evolution of neural network, the neural network structure needs to be represented in the form of genetic information. In this simulation the neural network

Table I. Competition rule

Initialization	1. Initial life power of organisms is set to 50. 2. Each colony is composed of 10 organisms. 3. Organisms are positioned at specific locations and head directions are arbitrary
Progress	1. Total 50 steps are progressed. 2. One at a time from two populations alternatively. 3. When all organisms have moved one time, 1 step is finished. 4. It can move or attack at each movement. 5. Before determining the behavior in the present step, it learns the reinforcement signal given to the behavior in the previous step.
Movement	1. It can move 1 lattice forward, backward, left or right relative to the head direction. 2. When it moves, the head direction is the present moving direction. 3. It cannot move to the location occupied by the opposite or selfish and only the head direction can be forward to the location.
Attack	1. It can attack the adjacent 4 lattices forward, backward, left or right relative to the head direction. 2. Whenever it is attacked, the life power decreases by 10 and it dies when the life power reaches zero. 3. It can attack the selfish and decrease the life power.
Evaluation	1. The population which has more live individuals wins. 2. If the number of live individuals is the same, the total life power is evaluated. 3. If the total life power is the same the winner is arbitrarily selected.

Table II. Artificial organism's behavior

Move	Move forward	Attack	Attack forward
	Move backward		Attack backward
	Move left		Attack left
	Move right		Attack right
	Stop		

Table III. Construction components and magnitudes of the connection descriptor

	From node	To node	Weight
Size	7 bits	7 bits	8 bits
Representation scale	0–127	0–127	−128–(+127)

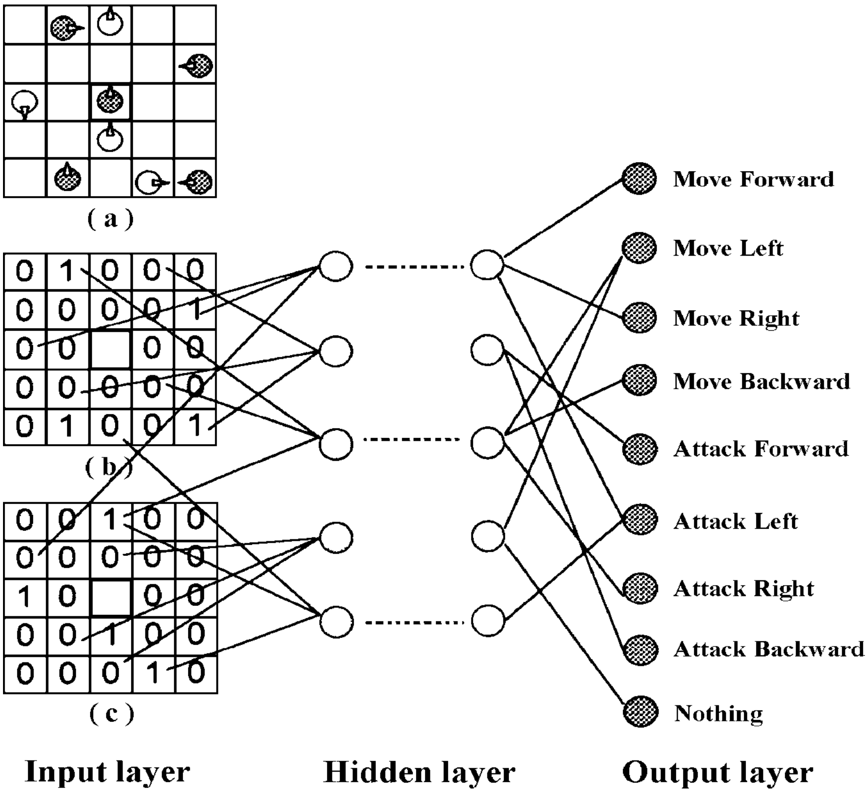


Figure 5. Neural network structure of artificial organism.

is constructed by a binary code using a connection descriptor [4, 5, 7, 9]. The neural network structure and weights are represented as connection descriptors in Table III.

3.3. ARTIFICIAL LIFE LEARNING

Competition proceeds in two tournaments, the winner and defeated team determination [1–3]. In each tournament, the organism group participating in the competition is arbitrarily selected from the gene pool, and the tournament is determined. Then the descendents of the winner replaces. In this simulation a steady-state genetic algorithm (SSGA) is used for evolution. SSGA does not retain the compatibility for the entire individual group. SSGA utilizes the actual competition results as the compatibility. In a dynamic competition environment, the compatibility cannot be represented as a certain numerical form. But only the actual competition results can be a basis for judgement. The win or defeat in the competition of the artificial organism group is determined by the number of live organisms after the competition. If A population has more live organisms than B population, A wins.

Table IV. Simulation model

Experiment model	Learning function	Reward model	Learning rate
A	no	–	–
B	yes	1	2
C	yes	1	4
D	yes	2	2
E	yes	2	4

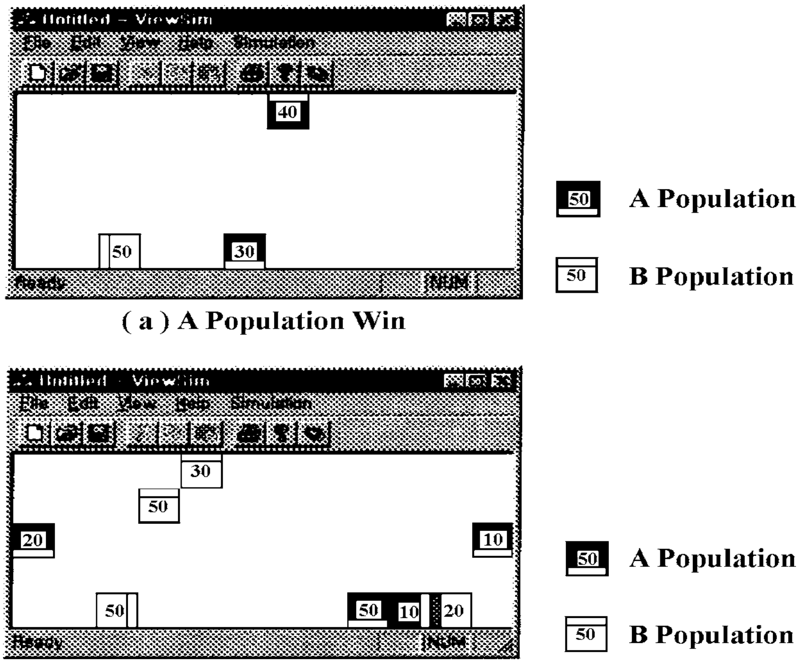
In case the two populations have the same number of live organisms, the total sum of organisms life power in each population determines the winner.

4. Artificial Life Simulation

The artificial life simulation in this paper has 1000 individuals in each population and the mutation rate is fixed as 3%. The simulations were performed using 5 experimental models as shown in Table IV. The models have a different combination of learning capability, compensation model and learning rate.

Model A is obtained by repeating the simulation used in [6], and is comparable to the simulation method proposed in this paper. Reward model 1 is the method for computing the reinforcement signal for neural network learning of artificial organisms. Reward model 2 has the purpose of constraining the movement and attack failure behaviors, and the reinforcement of a more exact behavior. The learning rate controls learning strength of the reinforcement signal given by reward model 1 for neural network learning. Learning rate 4 allows learning of the reinforcement signal at twice faster rate than learning rate 2. Therefore, the artificial organism's neural network is more affected by the kind and magnitude of the reinforcement signal. By applying various the learning rates [2, 3, 5, 6, 13] to the experimental environment of the same condition, more effective learning rates [3, 6] are selected and then used.

During the simulation, the behavior data is stored in every 100 generations and then recorded as a file. The behavior data is the number of behaviors for 6 items such as movement success, movement failure, total attack, opposite attack, selfish attack, and attack failure. The data provides analysis data of the evolution for 100 generations. Also, the behavior data allows the estimation of characteristics of the entire organism population. By comparison and analysis of the behavior data from each experimental model, the directions and speeds of evolution of the organism population of each model are compared. Then, the effect of an individual-level learning function on the evolution of entire population is evaluated. By the comparison and analysis of behavior data recorded in every 100 generations, it is not possible to make sure that the learning behavior of an



(a) A Population Win

(b) B Population Win

Figure 6. Winner determination.

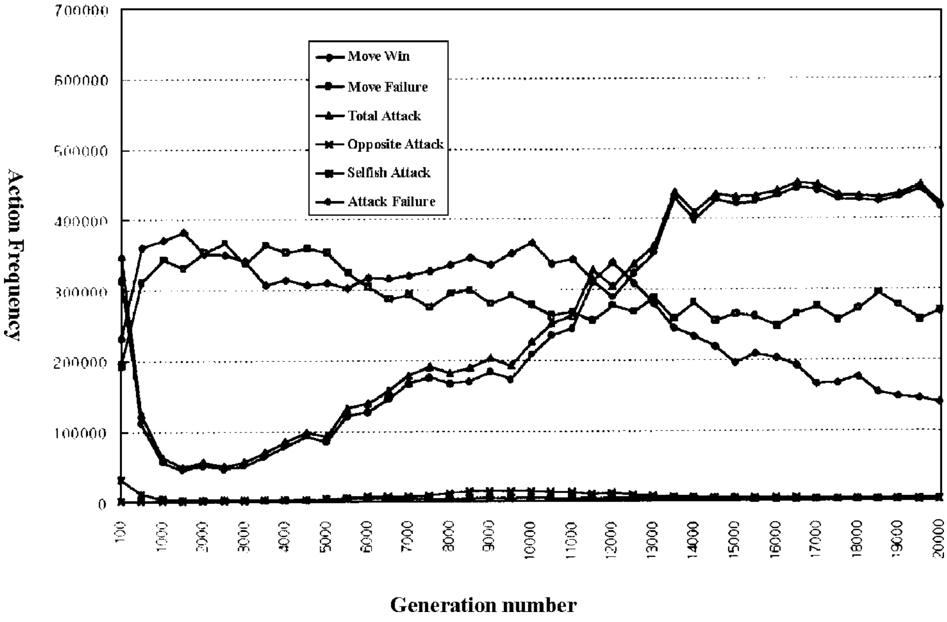


Figure 7. No learning case (model A).

artificial organism actually promotes the competition strategy. Therefore, the advantages/disadvantages of each methodology are judged by the actual competition between organism populations of a specific generation.

4.1. ANALYSIS AND EVALUATION OF THE RESULTS

4.1.1. *Behavior Analysis*

All experimental models globally showed the same direction of evolution. As the evolution progressed, global movabilities of populations decreased and the number of attacks against empty spaces increased. Therefore, the defensive strategy globally became prevailing. Also, every experimental model commonly showed a fast learning speed for the constraining function of selfish attack behavior. At the beginning of evolution, movements rapidly increased in all populations. This is because of the fact that at the beginning of evolution the population which can reduce the number of selfish attacks by moving has more advantages. Experimental models A, B, and C basically showed consistent graphs. On the other hand, experimental models D and E showed a little different graphs compared with experimental models A, B, and C.

Experimental model A did not use the individual-level learning function for the evolution of artificial organism. This result was obtained by repeating Gruau's simulation and is used to compare with the other experimental models. At the beginning of evolution, a rapid movability increases and attack characteristics decrease. As the evolution progresses, from around 3000 generations, the total attack number begins to increase and it continues to increase until 14,000 generations. It takes about 8000 generations for the total attack number to pass the 100,000–400,000 generation interval.

During the entire evolution, the number of attack failures, which mean attack behavior against empty spaces where no organism exists, makes up the main portion of the total attack number. This means that the global direction of evolution is a defensive strategy rather than an offensive strategy.

For the experimental model B, reward model 1 and learning rate 2 are used. Like the no learning case, the opposite attack is active only near a specific generation and the improvement of attack win behavior was not outstanding (Figure 8). At the beginning of evolution, the movability rapidly increased and,vice versa, the attack characteristic remarkably decreased. The attack behavior increased after 5000 generations and it reached the action frequency of 100,000–400,000 at the 3000 generations of total attacks. Comparing this result with model A, for which it took about 8000 generations, the speed of evolution was improved by approximately 267%. It means that dominant characters rapidly propagate inside the population. Generation is a concept of logical time used in simulation and it is assumed that the same generation has the same computing time. The attack characteristics increased at around the 5000 generation. The reason that it is slower than the 3000 generation for model A is through to be a delayed evolution due to unnecessary learnings. At

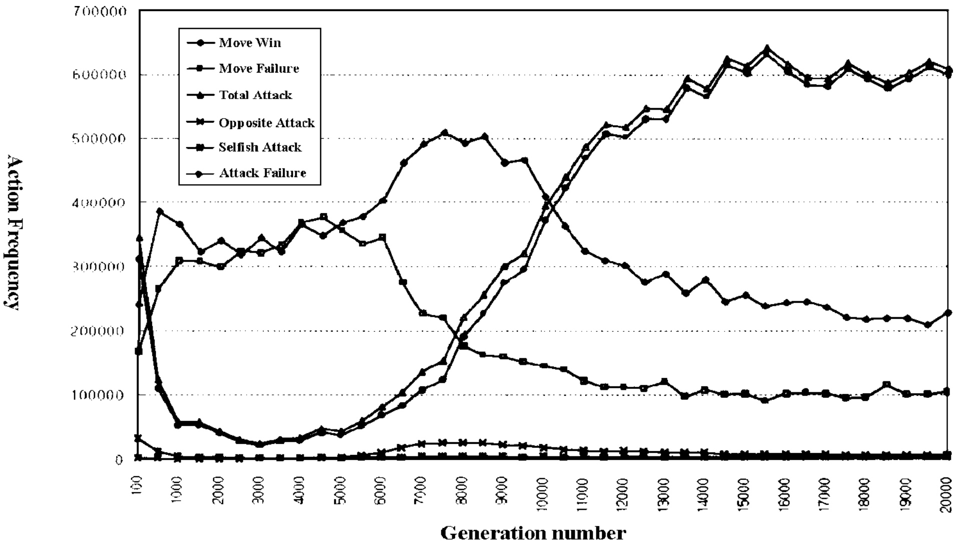


Figure 8. Reward model 2, learning rate 2 case (model B).

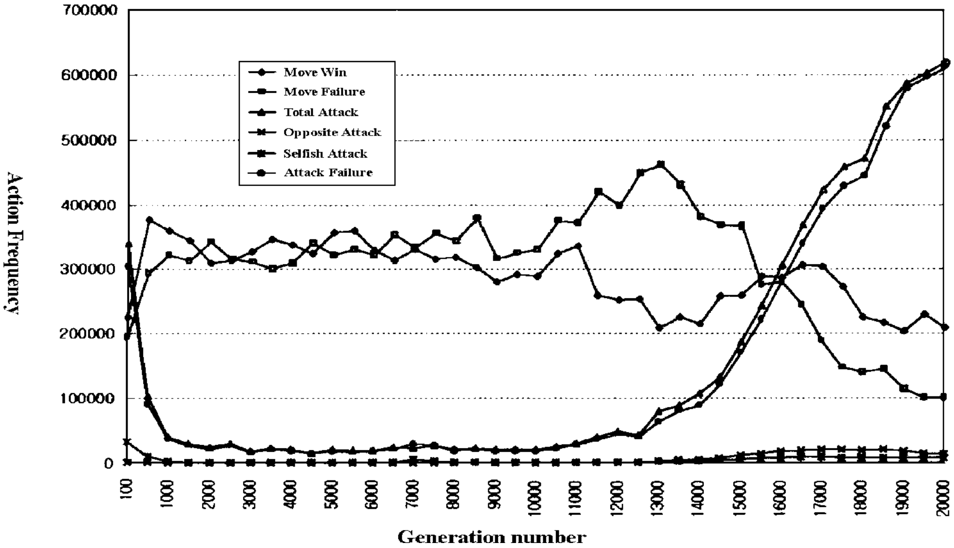


Figure 9. Reward model 1, learning rate 4 case (model C).

the last stage of evolution, a rapidly-increasing attack failure behavior remained constant and the organism's movability decreased.

For the experimental model C, reward model 1 and learning rate 4 are used. The characteristics of model C, compared to model A and B, are that the attack characteristics appeared very late at around the 12,000 generation. This is because (+) learning is unnecessarily performed for simple movement behaviors as addressed before. Like the other models, for model C, at the beginning of evolution

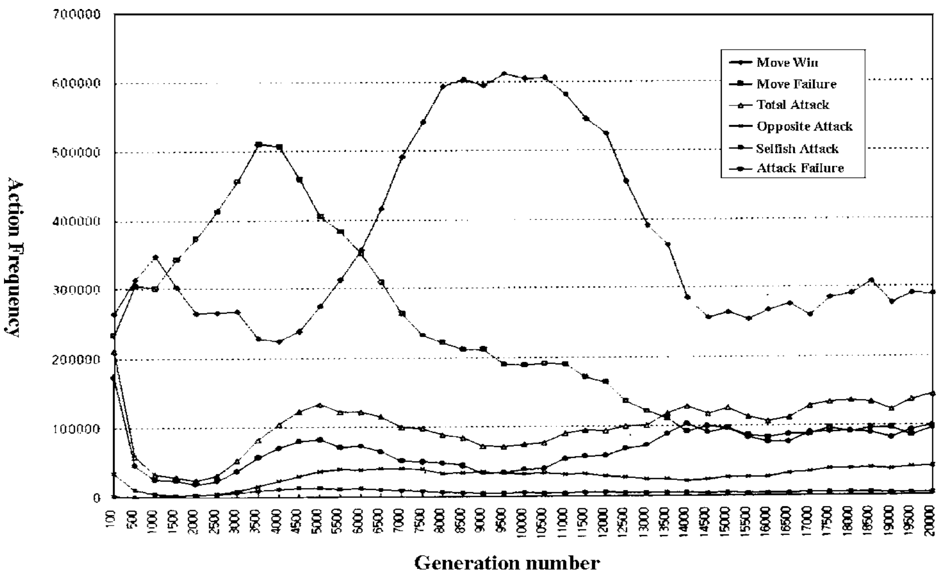


Figure 10. Compensation model 1, learning rate 2 case (model D).

movability increased and attack characteristic decreased. This is because selfish attack decreases and search of the neighboring area increases. As the evolution progressed, selfish attack number rapidly decreased, and then remained low-level. For the experimental model C, as model C, it took about 3000 generations to pass 100,000–400,000 intervals of the total attack number.

The experimental model D showed a different behavior graph compared to the previous models A, B, and C. Model D used compensation model 2. Compensation model 2 performs (−) learning for move and attack failures and no learning for a simple move behavior. Also, it can distinguish a selfish attack from the opposite attack. As a result, move and attack failure behaviors were remarkably constrained compared to the previous models and the opposite attack number constantly remained high. As the attack behavior is constrained, through the last phase of evolution, the frequency of move behavior is relatively high. However, regarding the global direction of evolution, like the other previous models, the movability gradually decreased and attack against empty spaces slowly increased, therefore, the strategy is defensive. The move failure number decreased after 4000 generations.

The experimental model E uses compensation model 2 and learning rate 4. Like model D, it showed a different behavior graph compared to models A, B, and C. At the beginning of evolution, the movability rapidly increased similarly to other models. However, it started to do learning for the move failure as well as, attack behavior. Like model D, the opposite attack number remained constant at a certain level. This moves that compensation model 2 had the same effect on the form of evolution. For model E at the last phase of evolution the movability gradually

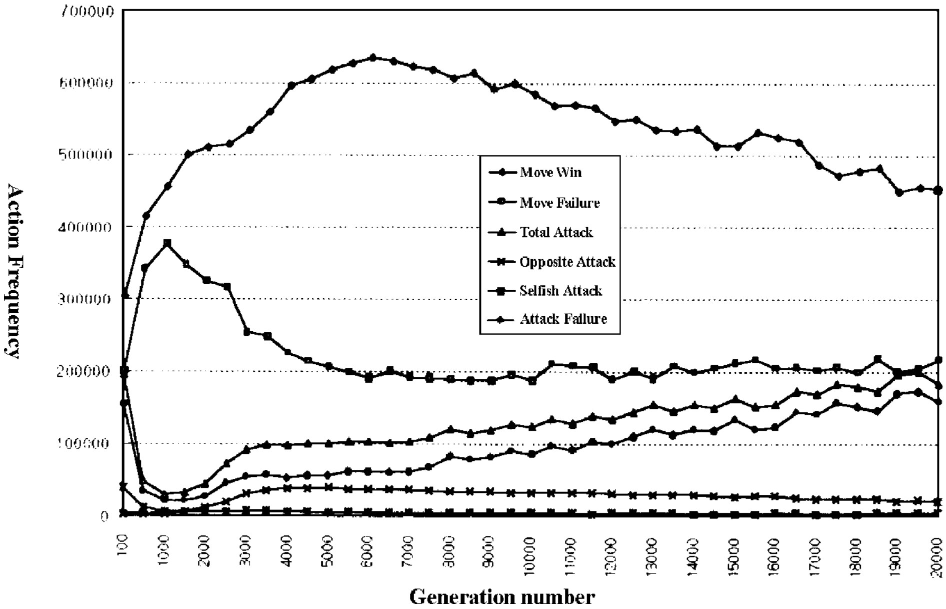


Figure 11. Compensation model 1, learning rate 2 case (model D).

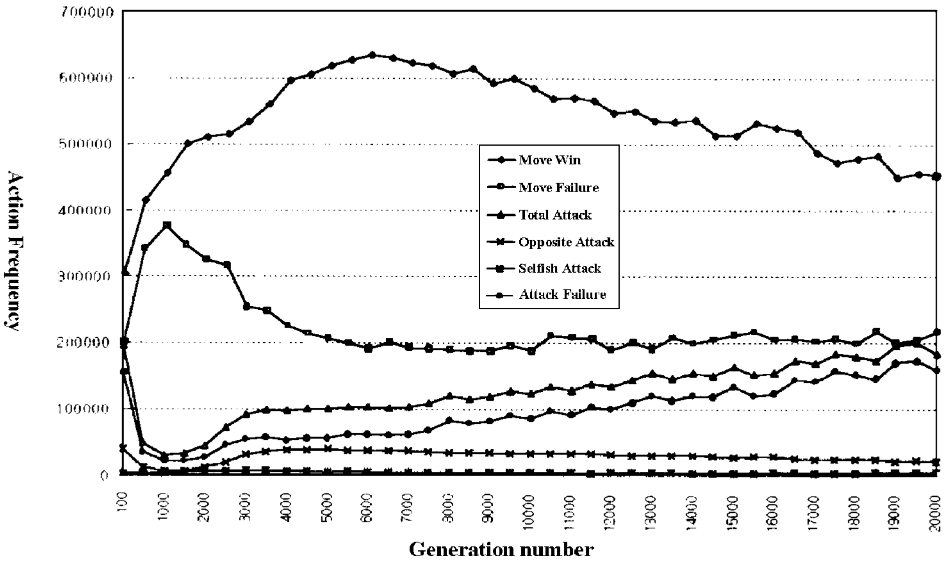


Figure 12. Compensation model 2, learning rate 4 case (model E).

decreased while attack against empty spaces gradually increased. The following problems were observed in experimental models B and C which compensation model 1 was applied to.

First, for move and stop behaviors, it always receives reinforcement signal 1 and, therefore, the move behavior is unnecessarily reinforced. Due to the activation of move behavior, learning for the attack behavior started late as compared to models A, D, and E. Second, the compensation for constraining the move or attack failure was not made. Therefore, constraints for move and attack failure behaviors were not effective even though evolutions took place. For models B and C, changes in the entire population appeared fast. It took 8000 generations for model A to pass the 100,000–400,000 interval of the total attack number. However, it was improved to 3000 generations for models B and C. The effect of constraining attack failure was remarkable for models D and E which compensation model 2 was applied to. Also, the learning for move and opposite attack win behaviors was made, therefore, a high-level performance was maintained compared to models A, B, and C. This suggests that the direction of evolution of the entire population can be controlled by the individual-level compensation function. By introducing the individual-level learning behavior concept in to the evolution of artificial organism population, the speed of change in the population globally showed a trend of being fast, but the direction of evolution was identical for all experimental models. The global direction of evolution was determined by the competition environment, and with the introduction of individual-level learning behavior, a faster search was possible.

4.2. EVALUATION

The evolution was strongly affected in time and form when the learning behavior was applied. Especially, the effect of the reward model used to simulate the learning behavior of an artificial organism was significant. For experimental models B and C which used the same reward model, the total attack number increased up to about 700,000, which is significantly higher than 450,000 of model A that did not use the learning behavior, and the move falure number decreased. It means that in case the reward model 1 was applied, a more effective learning for population competition environment was performed. Also, the comparison of superiority/inferiority shows on average 71.24% of the win rate against model A. It means that in case reward model 1 is applied, the artificial organism population searched for a more prevailing strategy during the same time interval. For experimental models D and E which reward model 2 is applied to, the constraining effect on the move failure and attack failure behaviors, which is the design purpose of reward model, was remarkable. And the opposite attack win number remained high-level during the evolution compared to other experimental models. The comparison of superiority/inferiority, models D and E, show average 78.91% of the win rate. It means that they searched for a more prevailing strategy during the same time period like models B and C. Although there are some variations in the results, depending on reward models, resultantly, in case the learning behavior is applied on the artificial organism-level, the artificial organism population performs more effective learning to the environment, and basing on this, there were many improvements in the form and time in the evolution.

5. Result

In this paper, the artificial organism-level learning behavior is introduced into the simulation of cross evolution among populations and the effect is evaluated. The reinforcement learning is applied. Also, the improved reward model 1 is designed for reinforcement learning application under a complicated and dynamically-determined environment. When the artificial organism's learning behavior is introduced, the speed of change in each experimental model population became fast and the global performance of evolution is improved. Also, when the learning behavior is introduced, during a certain time period, a better competition strategy can be found. The global direction of evolution is affected by the environment rather than learning and the form of evolution can be controlled by the artificial organism-level learning behavior. Application of artificial organism's learning behavior allows for the evolution of artificial organism population having specific purposes, and it provides a mechanism which makes it possible to apply the artificial life technique an various fields. For example, by a faster speed of change in population, the adaptability of a system to environment can be enhanced. By controling the evolution form, in unmanned tank simulation, the competition strategy having strong attack characteristics can be found, or an artificial organism having specific purposes such as strongly defensive unmanned tank behavior can be evolved. Also, in case several robots have to work together the behavior strategy can be found.

The simulation result in this paper was much influenced by reward models which produced reinforcement signals. To establish a more adaptive system, the reward model designed by the heuristic method as an initial step for the application of learning behavior, should find out appropriate values to environment for itself, and a more profound study of this topic needs to be performed in the future. Also, this paper has studied organism's ontogenesis which changes the behavior by the interaction between organism and environment.

References

1. Baldwin, J. M.: A new factor in evolution, in: R. K. Belew and M. Mitchell (eds), *Adaptive Individuals in Evolving Populations: Models and Algorithms*, Addison-Wesley, Reading, MA, 1996.
2. Collins, R. J.: Studies in artificial evolution, PhD Thesis (Philosophy in Computer Science), University of California, Los Angeles, 1992.
3. Collins, R. J. and Jefferson, D. R.: AntFarm: Towards simulated evolution, in: J. D. Farmer, C. Langton, S. Rasmussen, and C. Taylor (eds), *Artificial Life II*, Addison-Wesley, Reading, MA, 1991.
4. Filho, J. R., Alippi, C., and Treleaven, P.: Genetic algorithm programming environment, *IEEE Computer Journal* (1991).
5. Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
6. Gruau, F. and Whitley, D.: Adding learning to the cellular development of neural networks: Evolution and the Baldwin effect, in: *Evolutionary Computation I*, Vol. 3, 1993, pp. 213–233.

7. Holland, J. H.: *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press, reprinted by MIT Press, 1992.
8. Kaelbling, L. P., Littman, M. L., and Moore, A. W.: Reinforcement learning: A survey, *J. Artificial Intelligence Res.* **4**, AI Access Foundation and Morgan Kaufmann Publishers (1996).
9. Koza, J. R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, 1992.
10. Langton, C.: Studying artificial life with cellular automata, *Phys. D* **22** (1986), 120–149.
11. Langton, C.: in: C. Langton (ed.), *Artificial Life II*, Addison-Wesley, Reading, MA, 1989, pp. 1–47.
12. Lin, C. T. and Lee, G.: *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent System*, Prentice- Hall, Englewood Cliffs, NJ, 1996.
13. Nolfi, S., Elman, J. L., and Parisi, D.: Learning and evolution in neural networks. CRL Technical Report 9019, University of California, San Diego, 1990.