Evolution of Learning Robot Team via Local Mating Strategy*

Tatsuo UNEMI and Masahiro NAGAYOSHI†
Department of Information Systems Science, Soka University
1-236 Tangi-cho, Hachioji, Tokyo 192, JAPAN
{unemi,masahiro}@iss.soka.ac.jp
http://www.intlab.soka.ac.jp/˜unemi/

Abstract

This paper describes an experimental result on evolutionary processes of learning agents in a multi-agent environment, under our objective to propose an orientation toward a feasible design of a group of autonomous mobile robots that can evolve in the software level. After the work of L steps, each agent gathers the degree of task achievement and the genetic information from the nearest $N-1$ robots to revise its own genome, here named *Local Mating Strategy*. It starts learning again after this genetic operation. The experiment of ash sweeping by fifty autonomous mobile vacuum cleaners on the computer simulation is presented to show how this method is effective. It focuses on the effects of mating group size N and life span length L. According to the results, proposed method works well even if $N = 3$.

1 Introduction

Evolutionary robotics is one of the challenging fields related to Artificial Life, but only several practical results have been reported, such as optimizing control strategy of autonomous mobile robot by evolutionary adaptation of neural network [Floreano 93, Harvey 96, Hoshino 94], by genetic programming [Koza 92], and by classifier system [Dorigo 95]. As [Nolfi 94] stated, there are several possible approaches, but combination of simulation and real environments is one of the feasible methods described in [Miglino 96] because it is difficult to execute a huge number of iteration of testing individual performance using the real hardware robot, however there still remain the issues for building an appropriately precise simulation program including noise of sensor signal, fluctuation of actuator's motion, interaction between physical environment, and so on.

The primary purpose of our research partially described here is to propose an orientation toward a feasible design of evolvable robot team. Here we focus on a design of life cycle that can realize a type of evolution in a multi-agent environment with relatively small number of generations and small amount of communication cost. This paper proposes *Local Mating Strategy* to satisfy this requirement, and presents some experimental results to show the effectiveness of the method.

Learning is also a strategy to adapt to the environment for autonomous agent. In a multi-agent situation, it also works enough with communication among agents as described in [Weiß 93, Tan 93, Unemi 93]. There are several researches on relation between evolution and learning, such as Baldwin effect [Baldwin 1896, French 94], optimizing learning parameters by a genetic algorithm [Unemi 94], optimization by combination of artificial neural network and genetic algorithm [Belew 92], and so on. These papers has reported that the combination tends to provide earlier achievement to the superer performance. By this reason, we primarily examined evolution of reinforcement learning robots.

The rest sections describe our proposed method, experimental task, and experimental results of our computer simulation.

2 Local Mating Strategy

Evolution is an adaptive process by a population of organisms, which seems suitable for a framework to design an adaptive multi-agent system if an agent can spawn its offspring. However, it is difficult for artificial robot system to reproduce its physical body in the current technology. It seems feasible so far to let it evolve not in the hardware level but in the software level. One elegant idea to apply an evolutionary computation to a real robot is proposed in [Floreano 93] though it is only for a single agent. The following mechanism we propose here is for multi-agent evolution.

Proposed method named Local Mating Strategy is to communicate with only a small number of nearest robots for mating, to avoid global communication. Ordinary genetic algorithm needs global communication to select superer individuals from whole of population as parents to reproduce offsprings. This type of global comparison is useful to obtain the optimal solution, but wastes a long time to gather fitness values of all of robots, because it requires a *many to one* communication process. In the proposed method, each robot gathers the information on fitness and genetic code of the nearest $N-1$ others around it, and then embeds the genetic information of superer one if exists. This is in a style of

*Presented at the Fourth European Conference on Artificial Life, Poster Session in July 28–31 1997 in Brighton UK.

†Now associated with IBM Japan.

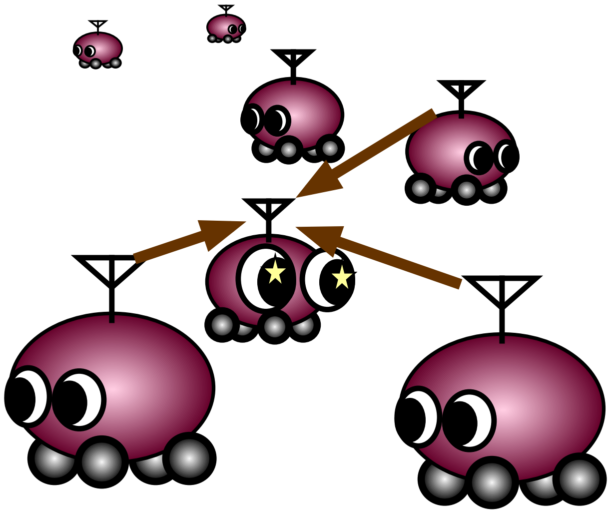

Figure 1: Local mating strategy, where each robot communicates with the nearest few others.

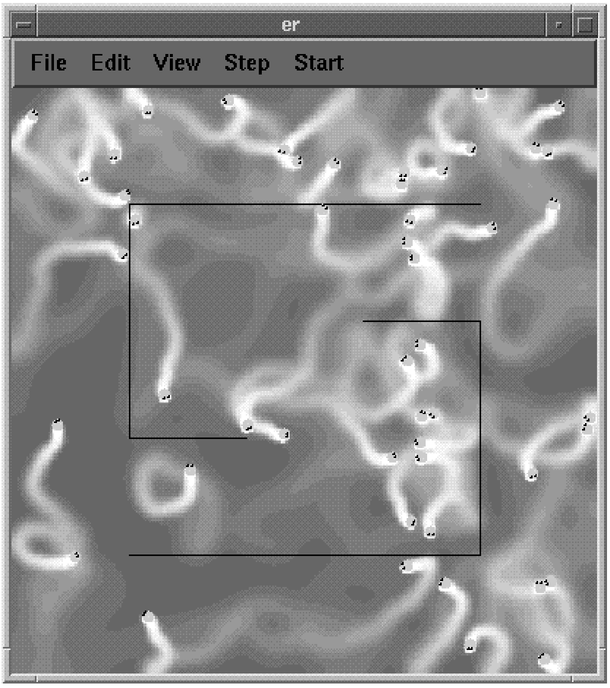

Figure 2: Example view of sweeping ashes task.

communication by observation as Figure 1 illustrates.

Fitness is measured as the score of task achievement. Selection is done among local sub-group. For each robot, the following genetic operation is applied.

1. After L steps passed, that is, it reaches the end point of life, it gathers information on fitness and genetic code of the nearest $N-1$ others around it to make the list of information of N individuals including itself, and sorts them by fitness.
2. It makes no change if it is in the upper third of N.
3. It embeds a part of genetic code of randomly selected one from the upper third if it is in the middle third. This operation corresponds to the crossover operation in genetic algorithms.
4. It replaces its genetic code by a mutant of randomly selected one from the upper third if it is in the lower third. This operation corresponds to the mutation in genetic algorithms.

After the above genetic operation, each robot resets the intrinsic parameters according to the new genetic information, then starts its work again. The reason why we denote the number of steps between genetic operations by *life span* is that this restart process is similar to the birth of baby in the software level.

This paper focuses on the effect of the mating group size and life span length. As described below, experiments by computer simulation were done on a variety of mating group size and life span length to investigate their effects.

There have been some researches on local mating strategies for *distributed (or parallel) genetic algorithms* [Tanese 89, Manderick 89, Schleuter 92, Belding 95] to speed up and to avoid premature convergence on a massively parallel machine [Spiessens 91] and a distributed computer network [Maruyama 93]. The experimental results presented in these papers showed that local mating strategies are effective enough compared with global mating of a canonical genetic algorithm. We can expect the similar effects in our algorithm for an autonomous mobile robot team.

3 Experimental task and robot

This section describes the specification of the experimental task, physical features of robot assumed, learning mechanism of robot, and some details related on genetics.

3.1 Task

The task we designed to examine the evolutionary process is to sweep ashes on the floor by a group of autonomous mobile vacuum cleaners. Each cleaner is an autonomous robot viewed as an individual in the evolutionary process. Each robot cleans up all amount of ashes just under its body in each time step, but ashes spread and increase gradually. The working space is a room of square shape surrounded by walls and there are some other walls as obstacles. Figure 2 shows an example display of the simulator.

The above settings are designed to satisfy the following conditions.

1. The work is by a group but it doesn't require sophisticated

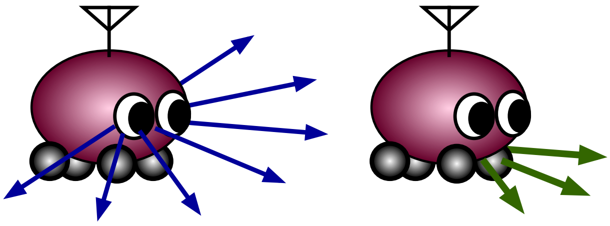

Figure 3: Vision (left side) and action (right side) of the robot.

cooperation among members.

2. Each member can measure the degree of individual task achievement.

3. Information on behavior strategy of a robot is reusable for another robot.

Total amount of ashes the robot absorbed in its life span is used as the fitness value.

3.2 Robot

Each robot has seven distance sensors to detect any object at its front in a limited area. The robot detects a wall and another robot by sensors but it has no explicit signal to recognize which type of object it is seeing. Each sensor outputs a real number between 1.0 for the touched object and 0.0 for nothing in the sensing area. The robot is also able to detect the amount of ashes on the floor at just its front. The robot moves forward by constant length and can turn by constant degree in each time step. It stops but can turn when it collides against any object. Figure 3 illustrates sensing and action of the robot.

The main strategy the robot should acquire to achieve the task is collision avoidance, such as rules if you detected any object in the left side then turn right and if you detected any object in the right side then turn left. Some proportion of random action possibility is also needed to avoid any deadlock situation.

3.3 Learning

Each robot has a potential ability of reinforcement learning based on simplified neural Q-learning as shown in Figure 4.

In each time step, the robot selects its action under the following probability.

$$P(a) \propto \exp\left(\frac{\mathbf{w}_a^T \mathbf{x}}{\tau}\right) \qquad (1)$$

where a indicates the type of action, turn left, go straight or turn right, $\mathbf{w}_a$ is the vector of connection weights corresponding to the action a, $\mathbf{x}$ is the input vector, and τ is the exploration rate given as a part of genetic information. The value of τ should gradually decreases during learning process since a robot would be better to aggressively explore to find better strategy in the early phase and would be better to exploit its strategy after enough times of experience. Because of this reason, the genome includes the value of τ at both the initial step and the final step of robot's life.

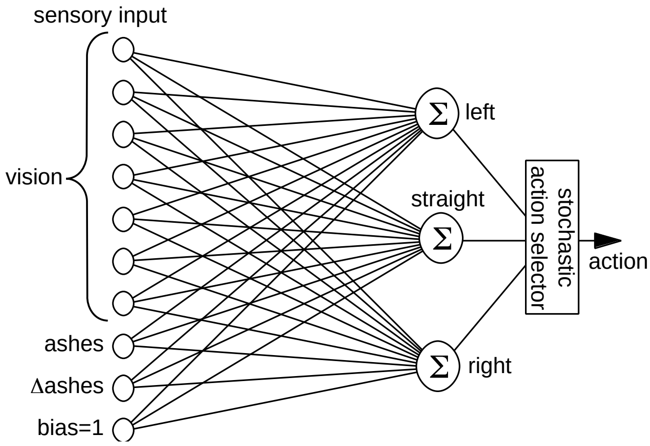

Figure 4: Neural network to decide robot's behavior.

The learning rule is based on the one step Q-learning [Lin 92] as follows.

$$\Delta\mathbf{w}_a = \beta \cdot (r_t + \gamma \max_b \mathbf{w}_b^T \mathbf{x}_t - \mathbf{w}_a^T \mathbf{x}_{t-1}) \cdot \frac{\mathbf{x}_{t-1}}{|\mathbf{x}_{t-1}|^2} \qquad (2)$$

where r_t is the reward acquired at time t, β is learning rate ($0 \le \beta \le 1$), and γ is discount rate ($0 \le \gamma \le 1$). The values of learning rate and discount rate are encoded in the genome.

In the experiment, the input vector consists of seven visual information, amount of front ashes, its time difference, and constant value one as a bias, totally ten elements. The value of reward is the amount of ashes the robot absorbed.

3.4 Genetics

Genetic information is represented on two chromosomes in this experimental model, one includes the initial values of connection weights w_i $(i = 1, 2, \ldots, 30)$ of the neural network and the other one includes some learning parameters described above. Each real value is encoded in eight bits integer on the gene. The integer values on chromosome corresponding to connection weights are signed, that is from -128 to 127, and corresponding to learning parameters are unsigned, that is from 0 to 255. At the start point of each generation, each robot starts learning after resetting the values of learning parameters and connection weights of the neural network according to the new genetic information. The correspondence between the integer values on genotype and the real values on phenotype is shown in Table 1. Learned characteristics do never inherit directly to the successors, that is, it is not Lamarckian but Darwinian evolution. Crossover operation is done as uniform crossover on each chromosome. Mutation is done as one point mutation by adding/subtracting a uniformly

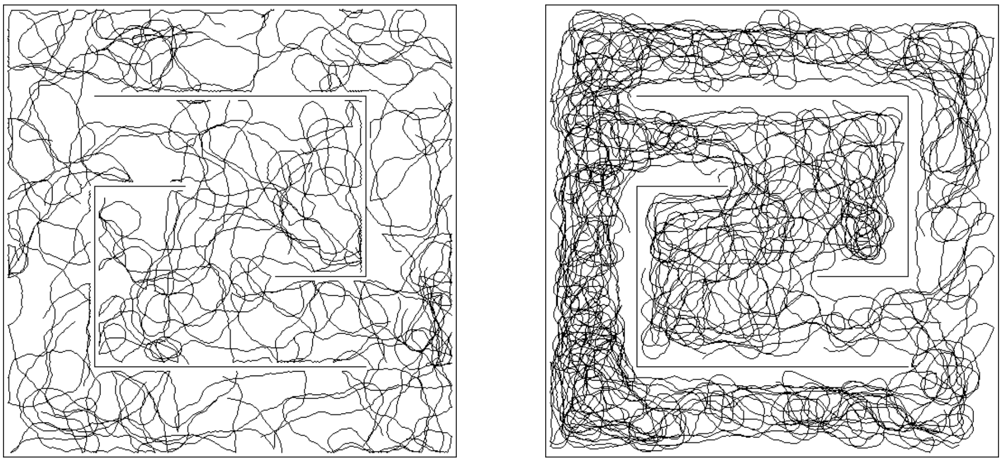

Figure 5: Traces of robots in one generation where life span is 500 steps.

Table 1: Correspondence between the values on genotype and phenotype.

	genotype	divider	phenotype
w_i	-128–127	64.0	-2.0–1.98
τ	0–255	127.5	0.0–2.0
β	0–255	511.0	0.0–0.5
γ	0–255	255.0	0.0–1.0

distributed random integer in $[-13, 13]$ to/from randomly selected locus. The result value of mutation is limited in the range of eight bits signed or unsigned integer.

It is difficult for some robots in the initial population to carry out the task because learning parameters of random values often makes it worse to do it. Guided by the fitness function, intrinsic characteristics are adjusted through the evolutionary process.

4 Experimental results

In this section, we show two kinds of experiments by computer simulation. The first is on the effect of the size of mating group N, and the second is on the effect of the length of life span L. In both cases, robots acquired collision avoidance strategy that enables efficient ash sweeping, as shown in Figure 5. To clarify the effect of combination of learning and evolution, we added experimental results on the cases of only learning and only evolution.

4.1 Effects of mating group size

We examined evolutionary process in the environment shown in Figure 2 that includes 50 robots with a variety of the size

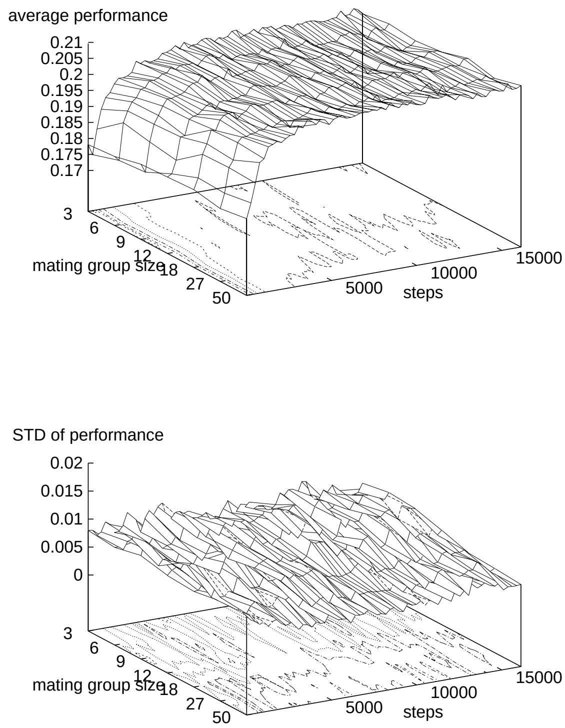

Figure 6: Evolutionary process of average performance of individual on a variety of mating group size over thirty trials, and the standard deviation over the trails, where life span $L = 128$ steps.

Table 2: Average performance of 128th generations and the standard deviation over thirty trials, for various size of mating group (N).

N	average	STD
3	0.203754	0.00776261
6	0.205030	0.00859252
9	0.204091	0.00821120
12	0.204938	0.00683015
18	0.207323	0.00536445
27	0.205464	0.00503067
50	0.207607	0.00464391

of mating group N, where $N = 3, 6, 9, 12, 18, 27, 50$. We did the simulation of 128 generations on thirty distinct random number sequences where the life span is 128 steps for each N. Figure 6 shows evolutionary processes of the average performance of individual and the standard deviation over thirty trials on each N.

The best case is on $N = 50$, but it is possible to achieve enough performance even if $N = 3$ in this task as show in Table 2. This indicates that local mating strategy is effective enough comparing with global selection. The difference of average performance between the case of $N = 3$ and 50 is 0.003853, that is less than the value of standard deviation for $N = 3$.

4.2 Effects of life span length

In the same settings as the previous experiment, we examined a variety of life span length L, where $L = 16, 32, 64, 128, 256, 512$ and $N = 3, 6$. We did the simulation of 16,384 steps on thirty distinct random number sequences for each combination of L and N. Figure 7 shows evolutionary processes of the average performance of individual and the standard deviation over thirty trials on each L.

It is difficult to evolve under too short life span, because the performance depends on the robot's position rather than the control strategy and the fitness values are unstable for the same genetic information. On the other hand, long life span guarantees a stable improvement process but it takes many steps to achieve enough performance because of slow generation exchanges. As shown in Figure 7, there is the optimal length of life span in terms of fast convergence, possibly depending on the mating group size N and the type of application domain.

4.3 Effects of combination of learning and evolution

Figure 8 shows the performance improvement on the case of learning without evolution by average and standard deviation among thirty trials of distinct random number sequences. Learning parameters are set that $\beta = 0.3, \gamma = 0.9$ and $\tau = 0.9 - (0.8/16383)\ t$ where t indexes the number

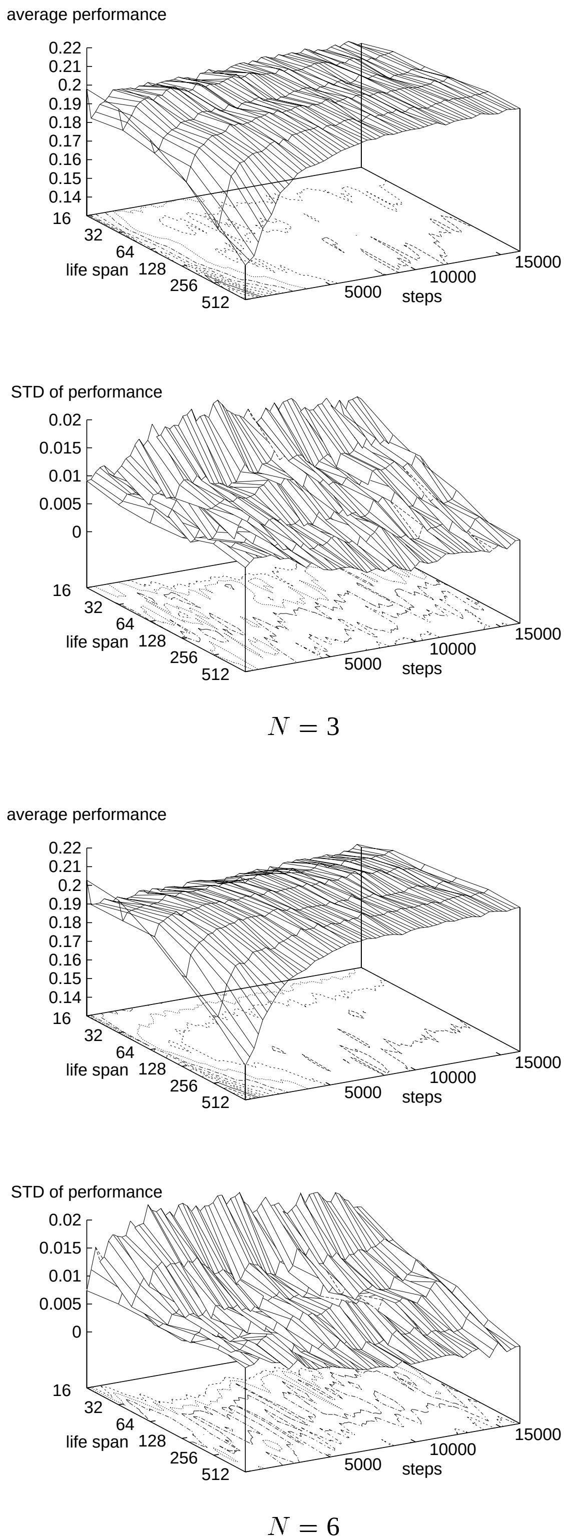

Figure 7: Evolutionary process of average performance of individual on a variety of life span length over thirty trials, and the standard deviation over the trails, where mating group size $N = 3$ and 6.

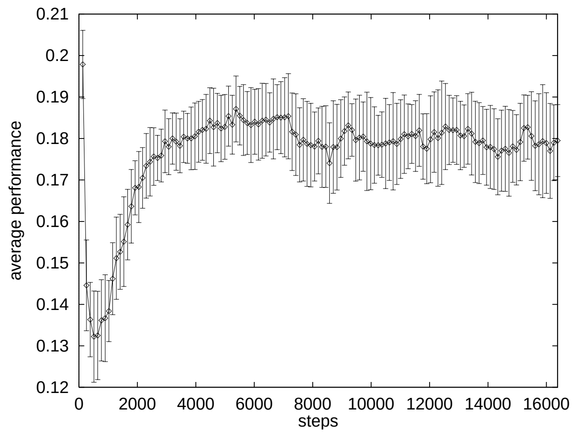

Figure 8: Learning process without evolution.

of steps ($t = 0, 1, 2, \ldots, 16383$). These values were carefully selected through some times of preliminary experiments to produce the best performance. We initialize the value of connection weight by random numbers.

As the figure clearly indicates, it is difficult to realize enough performance only by independent learning. The average performance at 16,384th step is less than 0.18 though it is more than 0.2 in the case shown above.

Figure 9 shows an evolutionary process without learning. It looks better than the case of combination. It is true when the life span length is either short or long, but, as Figure 10 shows, learning is effective when $L = 32$ and $N = 3$, and $L = 64$ and $N = 6$.

This phenomenon might be by Baldwin effect [Baldwin 1896, French 94] in which plasticity of phenotype guides evolution to jump up to the next stage. Figure 11 shows a typical evolutionary process of the distribution of gene values corresponding to learning rate β. The gene of the value about 130 suddenly disappeared at about 50th generation, then the value less than 64 got a majority. The lower learning rate leads less modification of weights, that is, learning is suppressed. It suggests that there occurred replacement from plastic learners to intrinsic winners. However, we have never found any evidence that learners guide the evolutionary process yet, at least in our experiments.

5 Conclusion

We proposed an orientation toward a feasible method to apply an evolutionary computing scheme to real robot team. Through the experiments described above, we certified on the framework of local mating strategy that

1. it works well enough even if the size of mating group is three,

2. the optimal length of life span exists for fast adaptation, and

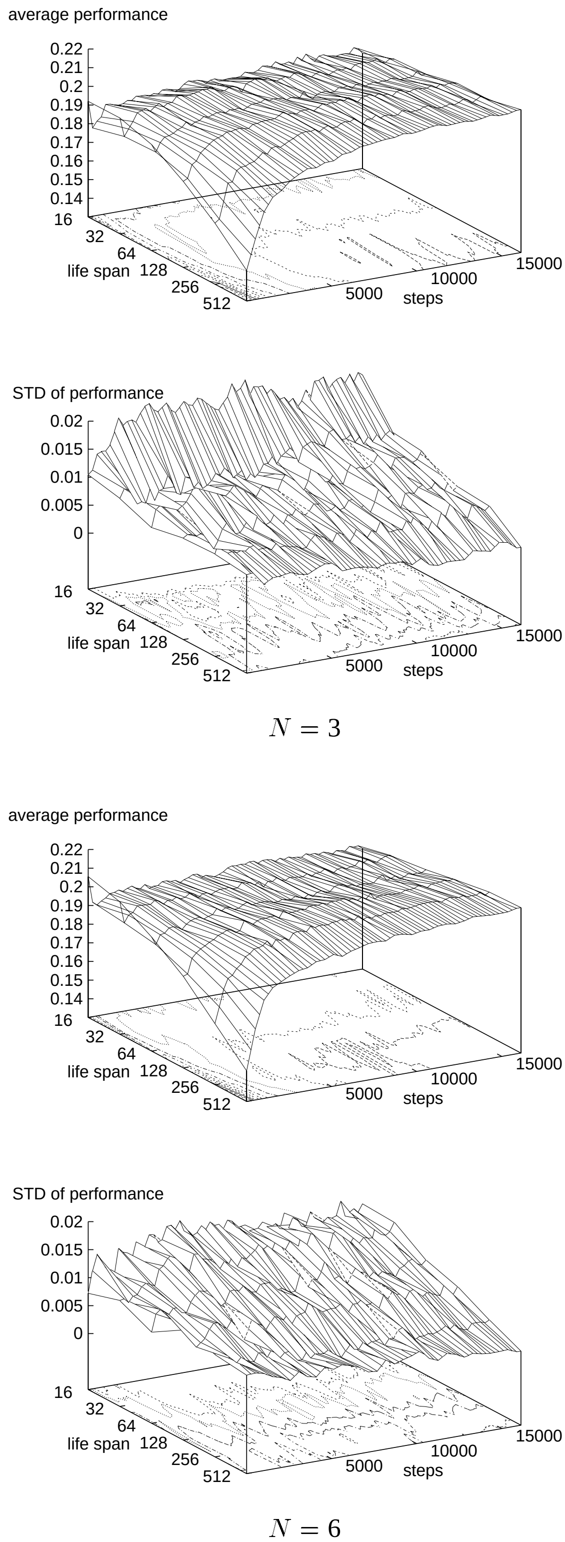

Figure 9: Evolutionary process without learning.

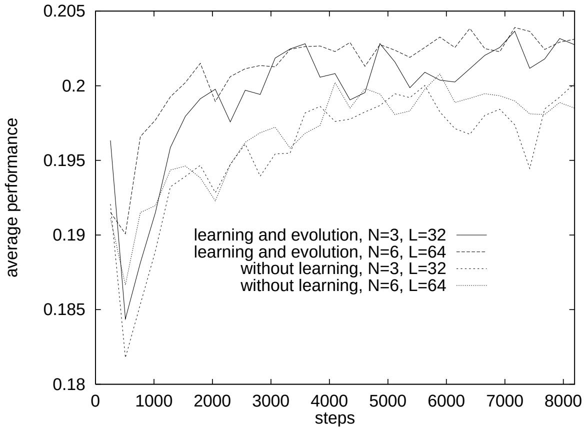

Figure 10: Comparison between evolutionary process with learning and without learning in the case of $N = 3$ and $L = 32$, and $N = 6$ and $L = 64$.

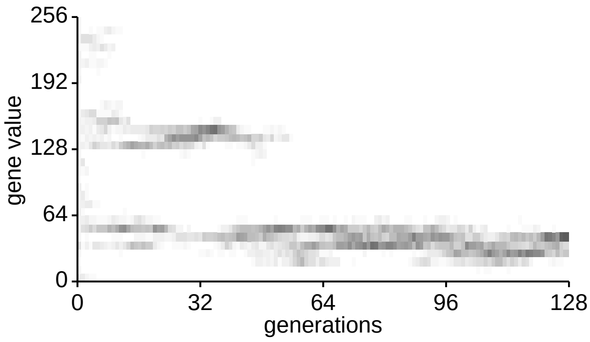

Figure 11: Typical evolutionary process of the distribution of gene values corresponding to learning rate β in the case of $L = 64$ and $N = 6$.

3. learning helps or disturbs evolutionary improvement depending on the life span length.

These statements might stand at least in the application domain we examined, but we can expect that the proposed method is effective in some similar type of other domains.

Adding to applying to the other types of tasks, our future work will include adaptive length of life span and mating group size. The method to find partner to exchange genetic information may have to be reconsidered in order to make it more feasible for hardware realization. There are also many interesting theme to challenge such as emergence of cooperation, species differentiation, global versus selfish goal, and so on.

The local mating strategy proposed here has a potential possibility to produce species differentiation in local area because the genetic operation is only possible among neighbor robots. An experiment on the more complex environment might be required to see this type of phenomena.

References

[Baldwin 1896] Baldwin, J. M. 1896. A new factor in evolution. *American Naturalist* 30: 441–451.

[Belding 95] Belding, T. C. 1995. The Distributed Genetic Algorithm Revisited. In Proceedings of the Sixth International Conference on Genetic Algorithms, 114–121. Morgan Kaufmann Publishers, Inc.

[Belew 92] Belew, R. K.; McInerney, J.; Schraudolph, N. N. 1992. Evolving networks: using the genetic algorithm with connectionist learning. In Langton, C. G. *et al* eds. Artificial Life II, 511–547. Addison Wesley.

[Dorigo 95] Dorigo, M. 1995. ALECSYS and the AutonoMouse: Learning to Control a Real Robot by Distributed Classifier Systems. *Machine Learning* 19: 3, 209–240.

[Floreano 93] Floreano, D.; Mondada, F. 1993. Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot. In Proceedings of the Third International Workshop on Simulation of Adaptive Behavior, 421–430. MIT Press.

[French 94] French, R. M.; Messinger, A. 1994. Genes, Phenes and the Baldwin Effect: Learning and Evolution in a Simulated Population. In Proceedings of the Forth International Workshop on the Synthesis and Simulation of Living Systems, 277–282. Mass.: MIT Press.

[Harvey 96] Harvey, I.; Husbands, P; Cliff, D.; Thompson, A.; Jakobi, N. 1996. Evolutionary Robotics: the Sussex Approach, *Robotics and Autonomous Systems*, to be published.

[Hoshino 94] Hoshino, T. 1994. *Dreams and Worries of Artificial Life (Jinkou-Seimei no Yume to Nayami)*. Shouka-Bo. (in Japanese)

[Koza 92] Koza, J. R. 1992. Evolution of Emergent Behavior. in *Genetic Programming: on The Programming of Computers by Means of Natural Selection*, 329–355. MIT Press.

[Lin 92] Lin, L.-J. 1992. Reinforcement Learning, Planning and Teaching, *Machine Learning* 8: 293–321.

[Manderick 89] Manderick, B.; Spiessens, P. 1989. Fine-Grained Parallel Genetic Algorithms. In Proceedings of the Third International Conference on Genetic Algorithms, 428–433. Morgan Kaufmann Publishers, Inc.

[Maruyama 93] Maruyama, T.; Hirose, T.; Konagaya, A. 1993. A Fine-Grained Parallel Genetic Algorithm for Distributed Parallel Systems. In Proceedings of the Fifth

International Conference on Genetic Algorithms, 184–190. Morgan Kaufmann Publishers, Inc.

[Miglino 96] Miglino, O.; Lund, H. H.; Nolfi, S. 1996. Evolving Mobile Robots in Simulated and Real Environments. *Artificial Life* 2: 417–434.

[Nolfi 94] Nolfi, S.; Floreano, D.; Miglino, O.; Mondada, F. 1994. How to Evolve Autonomous Robots: Different Approaches in Evolutionary Robotics. In Proceedings of the Forth International Workshop on the Synthesis and Simulation of Living Systems, 0–197. Mass.: MIT Press.

[Schleuter 92] Schleuter, M. G. 1992. Comparison of Local Meting Strategies in Massively Parallel Genetic Algorithms. In Männer, R.; Manderick, B. (eds). *Parallel Problem Solving from Nature, 2*, 553–562. Elsevier, Amsterdam.

[Spiessens 91] Spiessens, P.; Manderick, B. 1991. A Massively Parallel Genetic Algorithm – Implementation and First Analysis. In Proceedings of the Forth International Conference on Genetic Algorithms, 279–286. Morgan Kaufmann Publishers, Inc.

[Tan 93] Tan, M. 1993. Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents. In Proceedings of the Tenth International Conference on Machine Learning, 330–337. San Mateo, CA.: Morgan Kaufmann Publishers, Inc.

[Tanese 89] Tanese, R. 1989. Distributed Genetic Algorithm. In Proceedings of the Third International Conference on Genetic Algorithms, 434–439. Morgan Kaufmann Publishers, Inc.

[Unemi 93] Unemi, T. 1993. Collective Behavior of Reinforcement Learning Agents. In Proceedings of 93 IEEE/Nagoya University *WWW* On Learning and Adaptive System, 92–97. Japan: Nagoya University.

[Unemi 94] Unemi, T.; Nagayoshi, M.; Hirayama, N.; Yano, K.; Nade, T; Masujima, Y. 1994. Evolutionary Differentiation of Learning Abilities – a case study on optimizing parameter values in Q-learning by a genetic algorithm, In Proceedings of the Forth International Workshop on the Synthesis and Simulation of Living Systems, 331–336. Mass.: MIT Press.

[Weiß 93] Weiß, G. 1993. Learning to Coordinate Actions in Multi-Agent Systems. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, 331–316. International Joint Conference on Artificial Intelligence, Inc.