Evolution of Reinforcement Learning Agents – toward a feasible design of evolvable robot team

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

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 a constant of steps, each agent gathers the degree of task achievement and the genetic information from the nearest $N-1$ agents to revise its own genome. It starts learning again after this genetic operation. The experiment described here focuses on the effects of mating group size N and life span length.

Introduction

Learning is a strategy to adapt to the environment for autonomous agent. In a multi-agent situation, it also works enough with communication among agent as described in (Weiß 1993; Min 1993; Unemi 1993). On the other hand, evolution is also another strategy to adapt to the environment for a population of agents, though it seems hard to apply its framework directly to real robot team. The primary purpose of our research partially described here is to propose an orientation toward a feasible design of evolvable robot team.

There are several researches on relation between evolution and learning, such as Baldwin effect (Baldwin 1896; French & Messinger 1994), optimizing learning parameters (Unemi *et al* 1994), combination of artificial neural network and genetic algorithm (Belew, McInerney & Schraudolph 1992), and so on. Evolutionary robotics is also a challenging field but only few practical results have been reported, such as evolutionary adaptation of neural network to control a mobile robot (Floreano & Mondada 1993). Here we focus on the design of life cycle that can realize a type of evolution in a multi-agent environment.

Each agent proposed here has a mechanism of simple reinforcement learning, but learning parameters depend on individual genetic information. It is difficult for almost all of agents in the initial population to carry on the task because learning parameters of random values usually makes it worse to do it. Guided by the fitness function, intrinsic characteristics are adjusted through the evolutionary process.

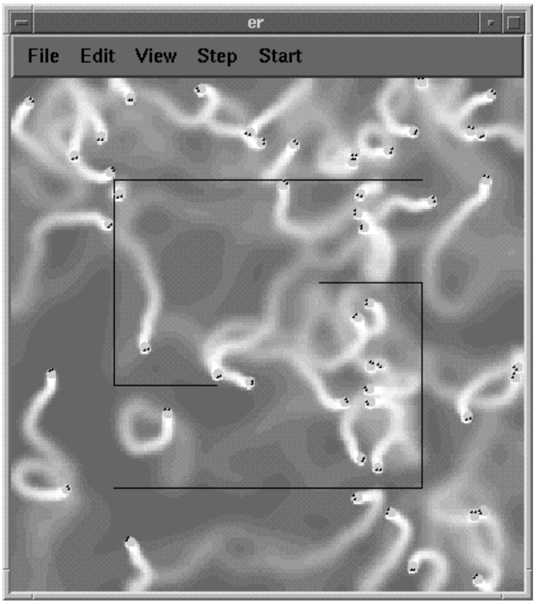


Figure 1: Example view of sweeping ashes task.

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 thier effects.

Experimantal task and agent

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 agent viewed as an individual in the evolutionary process. Each agent cleans up all amount of ashes just under itself in each time step, but ashes spreads and increases gradually. Figure 1 shows an example of display of the simulator.

Each agent has seven distance sensors to detect any

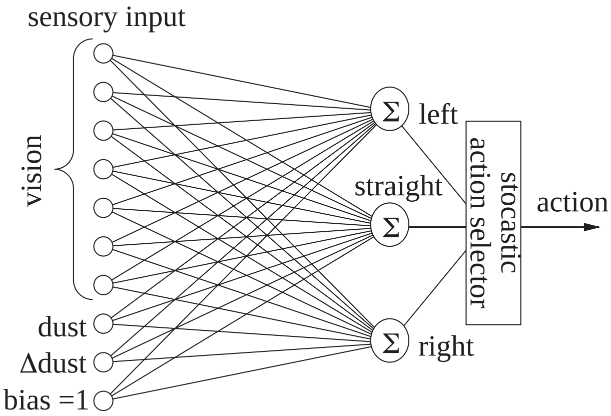


Figure 2: Neural network to decide agent's behavior.

object at its front in a limited area. The working space is a room of square shape surrounded by walls and there are some other walls as obstacles. The agent detects a wall and another agent by sensors but it has no explicit signal to divide which type of object it is seeing. The agent is also able to detect the amount of ashes on the floor at just its front.

The agent 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.

The above settings are designed to satisfy the following condition.

1. The work is by a group but it doesn't require sophisticated cooperation among members.
2. Each member can measure the degree of individual task achievement.
3. Information on behavior strategy of an agent is reusable for another agent.

The main strategy the agent should acquired 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.

Learning

Each agent has a potential ability of reinforcement learning based on simplified neuro Q-learning as shown in Figure 2.

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

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

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 learning rule is as follows, based on the one step Q-learning (Lin 1992).

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

where r_t is the reward acquired at time t, β is learning rate ($0 < \beta < 1$), and γ is discount rate ($0 \leq \gamma \leq 1$). 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 agent absorbed.

Evolution

Evolution is an adaptive process by a population of organisms, which seams suitable for a framework to design a adaptive multi-agent system if an agent can spawn its offspring. However, it is difficult for artificial robot system to reproduce itself in the current technology. One elegant idea to apply an evolutionary computation to a real robot is proposed in (Floreano & Mondada 1993) though it is only for a single agent. The following mechanism we propose here is for multi-agent evolution.

To avoid difficulty to apply evolutionary computing scheme to a real robot system, each member gathers 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. Fitness is measured as the score of task achievement, that is equal to total amount of ashes it absorbed in its life span.

Genetic information is represented on two chromosomes in this experimental model, one includes the initial value of connection weights 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.

We employ a local mating strategy in which selection is done among local sub-group. For each agent, 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 is done as one point crossover on each chromosome.
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 is done as one point mutation by adding/subtracting a random number to/from randomly selected locus.

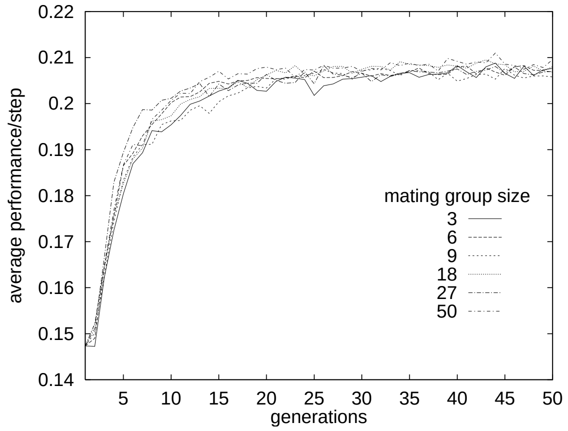


Figure 4: Evolutionary process of average performance of individual on a variety of mating group size.

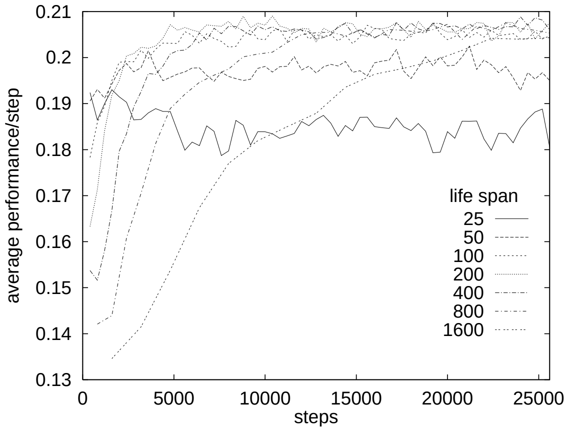


Figure 5: Evolutionary process of average performance of individual on a variety of life span length.

After the above genetic operation, each agent starts learning again resetting the weight values of neural network connection according to the new genetic information. 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.

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, agents acquired collision avoidance strategy, as shown in Figure 3. To clarify the effect of combination of learning and evolution, we added experimental results on the cases of only learning and only evolution.

Effect of mating group size

We examined evolutionary process in the environment shown in Figure 1 that includes 50 agents with a variety of the size of mating group N, where $N = 3, 6, 9, 18, 27, 50$. We did the simulation of 50 generations on ten distinct random number sequences where the life span is 500 steps for each N. Figure 4 shows evolutionary processes of the average performance of individual 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.

Effect of life span length

In the same settings as the previous experiment, we examined a variety of life span length L, where $L = 25, 50, 100, 200, 400, 800, 1600$ and $N = 6$ fixed. We did the simulation of 25,600 steps on ten distinct random number sequences for each L. Figure 5 shows evolutionary processes of the average performance of individual on each L.

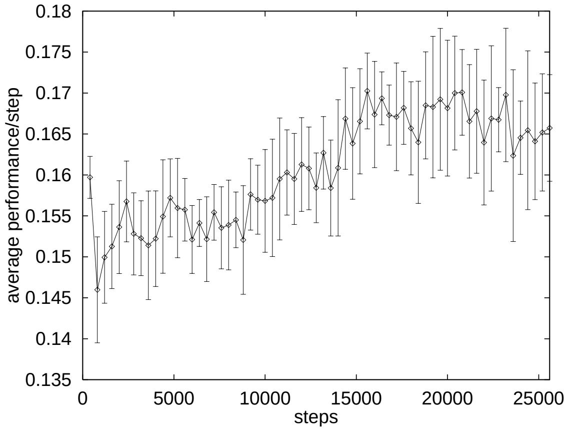


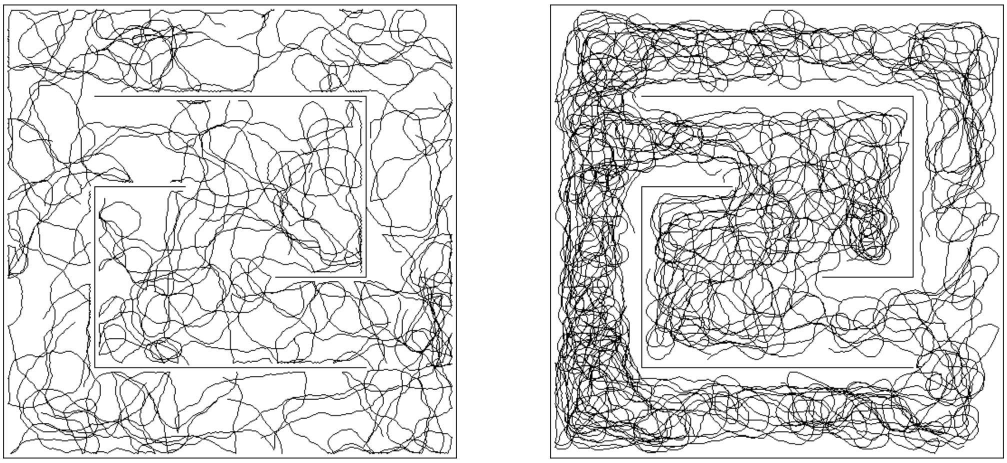
Figure 6: Learning process without evolution.

It cannot evolve under too short life span because it makes the fitness values unstable. On the other hand, long life span guarantees a stable process but it takes many steps to achieve enough performance. As shown in Figure 5, there is the optimal length of life span possibly depending on the type of application domain.

Effect of combination of learning and evolution

Figure 6 shows the performance improvement on the case of learning without evolution by average and standard deviation among ten trials of distinct random number sequences. Learning parameters are set as $\beta = 0.5, \gamma = 0.8, \tau = 1 - (0.9/25600) \cdot t$ where t indexes the number of steps ($t = 0, 1, 2, \ldots, 25599$). 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



First generation

50th generation

Figure 3: Traces of agents in one generation where life span is 500 steps.

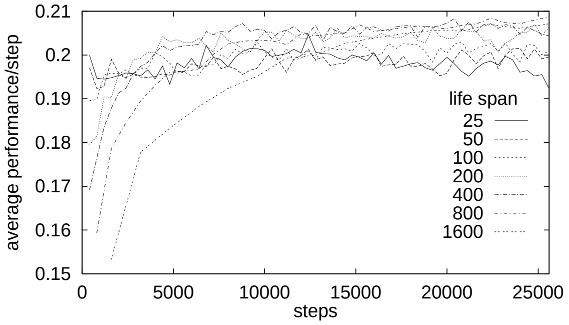


Figure 7: Evolutionary process without learning.

enough performance only by independent learning.

Figure 7 shows an evolutionary process without learning. It looks better than the case of combination. It is true when the life span length is eather short or long, but, as Figure 8 shows, learning is effective when $L = 100$. This phenomenon should be by Baldwin effect (Baldwin 1896; French & Messinger 1994) in which plasticity of phenotype guides evolution to jump up to the next stage, though we have not certified it by precise analysis on the genetic trace of evolutionary process.

Conclusion

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

1. local mating strategy is good enough even if the size of mating group is three,
2. the optimal length of life span exists for fast adaptation, and

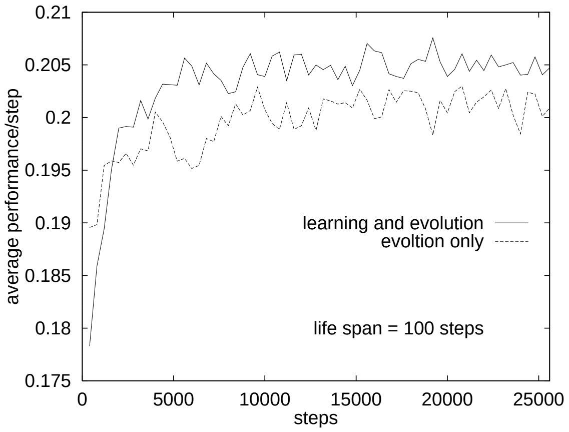


Figure 8: Comparison between evolutionary process with learning and without learning in the case of $L = 100$.

3. learning helps evolutionary improvement with a suitable life span length,

at least in the application domain we examined.

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 near agents. An experiment on the more complex environment might be required to see this type of phenomena. This is also interesting from a view point of Artificial Life.

References

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

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.

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, 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.

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

Min, T. 1993. Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents. In Proceedings of the Tenth International Conference on Machine Learning, 330–337. San Mateo, Calif.: Morgan Kaufmann Publishers, Inc.

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

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, 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.