Comparison of Evolutionary Methods for Smoother Evolution

Tomofumi Hikage, Hitoshi Hemmi, and Katsunori Shimohara

NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho
Soraku-gun, Kyoto 619-0237, JAPAN
E-mail:{hikage, hemmi, katsu}@cslab.kecl.ntt.co.jp

Abstract. Hardware evolution methodologies come into their own in the construction of real-time adaptive systems. The technological requirements for such systems are not only high-speed evolution, but also steady and smooth evolution. This paper shows that the Progressive Evolution Model (PEM) and Diploid chromosomes contribute toward satisfying these requirements in the hardware evolutionary system AdAM (Adaptive Architecture Methodology). Simulations of an artificial ant problem using four combinations of two wets of variables — PEM vs. non-PEM, and Diploid AdAM vs. Haploid AdAM — show that the Diploid-PEM combination overwhelms the others.

1 Introduction

The ultimate goal of evolutionary system research is to create a system able to provide the flexibility, adaptability, and robustness of living organisms with the speed necessary for engineering application. Evolutionary methods, like Genetic Algorithms and Genetic Programming, have already been applied in various fields like optimization and pattern recognition. These methods, however, require a large amount of computations. Solving this problem will provide a large number of new applications for evolutionary systems.

Simulated evolution using electronic circuits, or in general hardware evolution, represents a significant breakthrough in speeding up the computations. However, simply speeding up individual operations will not necessarily solve the computation problem.

Yao et al. [6] discussed "the danger of relying too much on hardware speed". They described the possibility that the time required in evolution is exponential in terms of the target circuit scale. However, if we could navigate such evolution aptly, it would perhaps be possible to substantially reduce the problem space, i.e., the space in which the evolutionary process actually performs exploration, and to avoid, or at least to reduce, the difficulty. In this paper, we show that the Progressive Evolution Model (PEM) along with Diploid (DL) chromosomes can be employed for this purpose.

In our earlier work on PEM [3], evolution took place in a stepwise manner to match the stepwise environmental changes. This is an efficient way to lower the problem "hurdle". An essential part of PEM involves environmental changes, which often destroy well-adapted individuals. Individuals specialized to a previous environment may be unable to cope with environmental changes. Although this is a necessary cost for PEM, techniques for easing the damage are strongly desired.

On the other hand, our diploid-chromosomes-based model broadens the population diversity [2], and this makes it easier for the population to find the optimal solution. Combining these two techniques provides an efficient method for smoother evolution, as will be shown.

2 Diploid AdAM and Progressive Evolution Model

2.1 Hardware Evolutionary System: AdAM

In experiments, we used our HDL (Hardware Description Language) - based hardware evolutionary system named AdAM [4].

An initial chromosome population is first created at random according to a set of production rules, which are of the Backus-Naur form of the HDL named SFL (Structured Function description Language). Each chromosome then generates an individual SFL program through interpretation. Next, the individual SFL programs, i.e., individual hardware behaviors, are simulated and evaluated by a certain fitness criterion. Finally, the selection and a set of genetic operations are performed on the chromosome population to produce the next generation.

To expresses SFL efficiently, the chromosomes of this system are in a tree structure (see Fig. 2). The tree is a parse tree. By assuming the above production rules to be rewriting rules, information on these rewriting rules is stored in each node of the tree structure chromosomes. Two or more rewriting rules exist in one non-terminal symbol. The stored information on rewriting rules include rewritten non-terminal symbols and information of which rewriting rule is to be used, or terminal symbols.

2.2 Biological Background of Diploid-chromosomes-based Model

Most multi-cellular organisms are diploid, having a set of diploid chromosomes, and exhibit dominant and recessive heredity; a haploid, by comparison, has only one set of chromosomes. Since a model in which one individual has one "chromosome" is usually employed, we postulate that a diploid has a pair of chromosomes and a haploid has a single chromosome to simplify the following discussion.

In a haploid, a change in its chromosome immediately appears as a change in the pheno-type experiencing selection pressure. In general, most changes in a chromosome are harmful in the sense that the probability of destroying a gene that generates an effective function has been predicted to be higher than that of gaining a new gene with a superior function. For a haploid to acquire a new function effectively, a gene should be duplicated before it is mutated. This would allow multiple copies of a gene for an existing function, so the chromosome could acquire a new gene by mutation while maintaining the original gene. That is, gene duplication must precede mutation; as a result, haploid evolution would generally take a long time [1].

A diploid has a recessive gene, and a change in one of the chromosomes does not appear in the pheno-type if the change occurs in the recessive gene. Therefore, it is very likely that harmful changes in a chromosome would survive and be stored up in a selected lineage until a new surpassing combination is eventually constructed. From this viewpoint, the genetic diversity of a diploid is larger than that of a haploid.

The process of meiosis, i.e., a cell division process into a germ cell, is as follows. First, each chromosome in a diploid cell is replicated. Next, mother chromosomes and father chromosomes become involved with each other. From this, crossover occurs in this stage. Then, at division I of meiosis, the involved four chromosomes are divided into two pairs of chromosomes. Finally, at division II of meiosis, each pair of chromosomes is divided into two germ cells which have only one chromosome each.

When a germ cell meets a germ cell of another individual, the two cells unite into one cell which has two chromosomes (i.e., fertilization, Fig. 1).

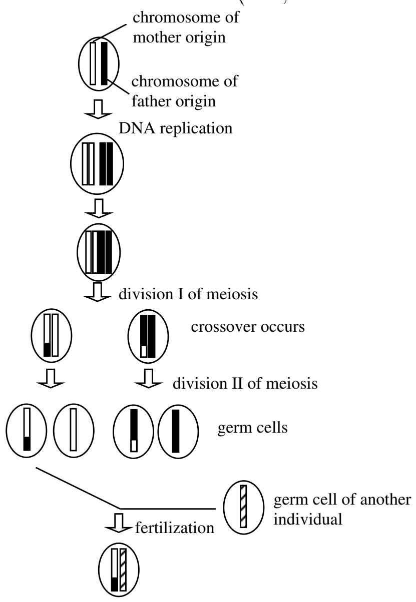


Fig. 1. Simplified illustration of the biological process of meiosis and fertilization

2.3 Chromosome of Diploid AdAM

A diploidy with a bit string chromosome is treated in [5]. Here, we introduce a tree structured chromosome with dominant-recessive heredity (see Fig. 2). We simplify the model of dominant-recessive heredity by assigning two sub-trees to one node corresponding to dominant-recessive alleles, which are initially defined randomly as dominant or recessive. A node having dominant-recessive alleles is initially selected at random.

2.4 Genetic Operations

Along with introducing dominant-recessive heredity into a chromosome, we employ the following genetic operations.

- Mutation: This changes the rewriting rule in a node, or changes the terminal symbol (constant, variable name, etc.) in a node.
- Duplication/Deletion: These duplicate or delete a node.
- HL (Haploid)-type crossover: This exchanges the rewriting rules of two nodes, each of which belongs to a different chromosome.
- DL (Diploid)-type crossover: This newly introduced operation exchanges the rewriting rules of two nodes, each of which belongs to either of two sub-trees corresponding to alleles in the same chromosome (see Fig. 3).

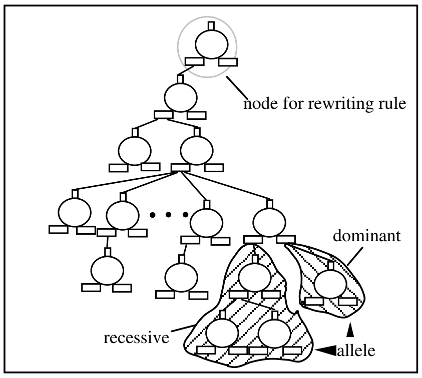


Fig. 2. A chromosome with dominant-recessive heredity

- Meiosis/Fertilization: These newly introduced operations are the operations most characteristic of dominant-recessive heredity. They exchange the sub-trees of two nodes that have the same non-terminal symbol, but each of which belongs to a different chromosome.

Fertilization fuses two "germ cells", i.e., the germ cells of parent A and parent B are fused to yield child C and child D. The dominant or recessive tags of the parent sub-trees are already determined before the fertilization. In Fig. 3, if we assume that sub-trees 'm' and 'o' are dominant over sub-tree 'n', then sub-tree 'n' can not appear in the pheno-type in the generation of the parents. However, in the generation of the children, child D has two copies of the same recessive sub-tree in its two sub-trees, that is, both sub-trees are recessive and a recessive sub-tree appears in the pheno-type.

2.5 Progressive Evolution Model

Basic idea Organisms evolve while acquiring new functions to match environmental changes. We therefore believe that environmental changes drive evolution, so our idea is to use environmental changes actively to accelerate evolution.

In the progressive evolution model, evolution occurs in environments that change in a stepwise manner toward the final target environment. The purpose of the model is to divide a large "hurdle" into a series of small steps that the evolutionary process can easily handle.

Because this model depends on the applied problem and is implemented, this model is concretely explained after the problem is described in the following section.

3 Experiments

Our target is to evolve a hardware system to control the autonomous behaviors of a robot. In order to test our evolutionary method, a software simulation of the ant problem explained above is used.

3.1 The Ant Problem

In our artificial ant problem, artificial ants adapt to a certain environment so that they can gather food faster and more effectively; the best artificial ant is an ant that collects the most food in the fewest number of steps. The environment is usually an arrangement of food in a toroidal space of m $\times$ n cells. There are q pieces of food, indicated by black squares. In our case, each artificial ant takes five inputs showing the existence of food and performs one of three actions (Fig. 4).

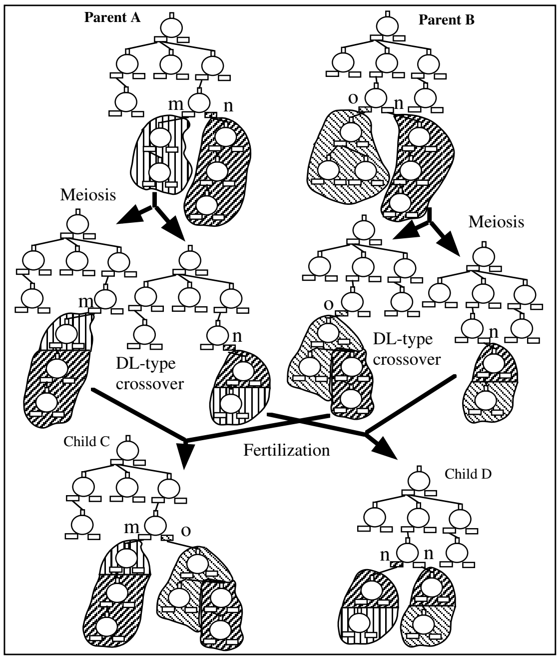


Fig. 3. Example of Meiosis/Fertilization and DL-type crossover with dominant and recessive heredity.

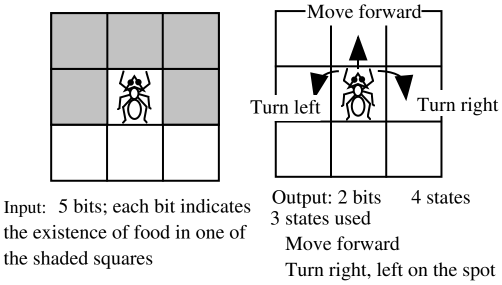


Fig. 4. Artificial Ant

3.2 Measuring the Environmental Complexity

In PEM, a measure of the environmental complexity is needed to divide the environment into a series of easier environments. As shown in Fig. 4, each ant has five inputs and two outputs. In the ant's inner mechanism, each input needs to be taken into account, and the arbitration among them is indispensable; an ant may encounter two different food items.

It is thought that, from the artificial ant's standpoint, the count of input-action pairs defines the environmental complexity. In this case, there are 96 ($= 2^5 \times 3$) combinations of inputs and actions. We call each combination an action primitive. These action primitives express the difficulty of the environment; the number of action primitives an ant needs to successfully collect all food from the environment is the complexity factor of the environment.

We write action primitive as follows. {Input form left, left front, front, right front, right: ACTION}, where ACTION means "move forward", "turn left", and "turn right". For instance, {0 0 1 0 0: "move forward"}, {0 0 1 0 1: "turn right"} means a complexity factor of 2.

3.3 Progressively Changing Environments

We designed a series of environments for artificial ants. Figure 5(e) shows the final target environment, and Figs. 5(a – d) show intermediate environments leading to the target environment. These environments are designed so that action primitives are obtained incrementally.

The first intermediate environment [Fig. 5(a)] requires four action primitives, {0 0 1 0 0: "move forward"}, {0 0 0 1 0: "move forward"}, {0 0 0 0 0: "move forward"}, {0 0 0 0 1: "turn right"}, so the complexity is 4. The second intermediate environment [Fig. 5(b)] requires three action primitives, {0 1 0 0 0: "move forward"}, {0 1 1 0 0: "move forward"}, {1 0 0 0 0: "turn left"}, in addition to those of the first, so the complexity of this environment is 7. In the same way, the third intermediate environment, [Fig. 5(c)] requires two more action primitives, {0 0 1 1 0: "move forward"}, {0 1 0 0 1: "turn right"}, so the complexity is 9. The fourth environment [Fig. 5(d)] adds three more action primitives, {1 1 1 0 0: "move forward"}, {0 0 1 1 1: "turn right"}, {1 1 0 0 0: "turn left"}, for a complexity of 12. Finally, the target environment requires two more action primitives in addition to the previous ones, {0 1 1 1 0: "move forward"}, {1 0 0 0 1: "turn right"}, and therefore the complexity measure is 14. The timing to exchange the environments is set at five generations after the best individual appears in each environment. In addition, due to the elite strategy, the first successful individuals spread through the population and most of the circuits adapt to the environment during the five generations.

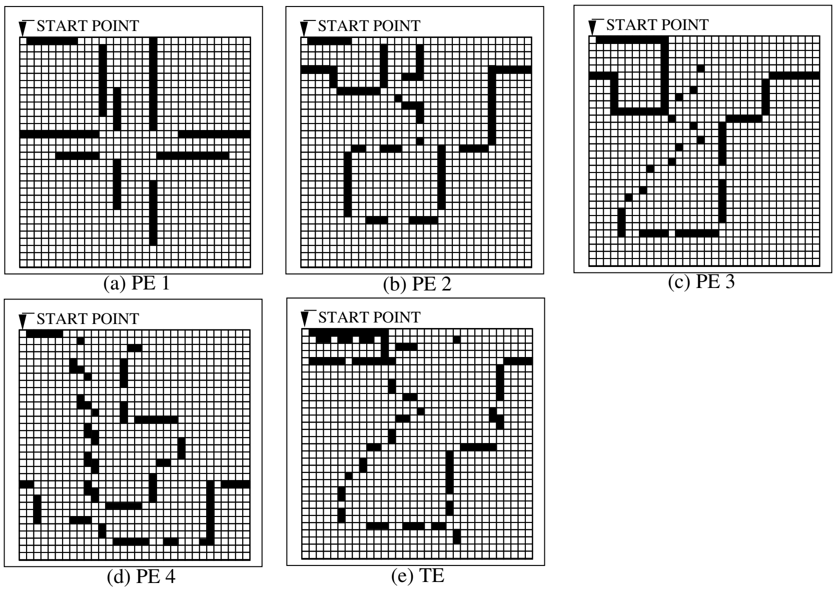


Fig. 5. Progressive Environments and Target Environment (TE)

3.4 Simulation Details

We carried out an experiment in which the controller circuit of an artificial ant was generated evolutionarily. We compared PEM to nonPEM by examining their performance in terms of the number of generations until convergence over all environments used. We also compared Diploid AdAM to Haploid AdAM. There were four types of simulations. The experimental conditions were as follows.

– Food: 89 types of food/environment
– Population size: 256 individuals
– Selection Methods: roulette model and elite strategy
– Fitness function
  - Fitness = Score + Limit - Step + 1

    Score: units of food collected; Limit: Maximum step number allowed in this system (350 steps); Step: move from one square to another

4 Results

The results for Diploid AdAM with PEM and Haploid AdAM with nonPEM are shown in Figs. 6(a – c) and 6(d), respectively. The experimental results obtained in early generations with PEM are shown in Fig. 6(a). The best individuals in the first, second, and third intermediate environments appear in the 1st, 11th, and 54th generation, respectively. Figure 6(b) shows the result for the last stage of PEM. From these figures, one can see that evolution takes place in a stepwise manner to match the environmental changes. Figure 6(c) shows the fitness transition for all generations. It takes 3,331 generations to produce the best individual in the final target environment.

Figure 6(d) shows the result obtained for evolution in the final target environment with an ordinary evolution model. The best individual appears in 18,427 generations, indicating that a progressive model is more efficient. Table 1 shows results for all types. Only Diploid AdAM with PEM could obtain the best ant within 10,000 generations. Accordingly, PEM is effective for Diploid AdAM, which is probably because of the ability of Diploid AdAM to treat dominant and recessive heredity works well for environmental changes. Conversely, it seems that in a stable environment, there is hardly any difference between Haploid AdAM and Diploid AdAM, as indicated by Haploid AdAM with nonPEM and Diploid AdAM with nonPEM. We think that the effect of PEM is lost for Haploid AdAM with PEM , in which the environment is changed in Haploid AdAM, because Haploid AdAM is unable to follow the changing environment.

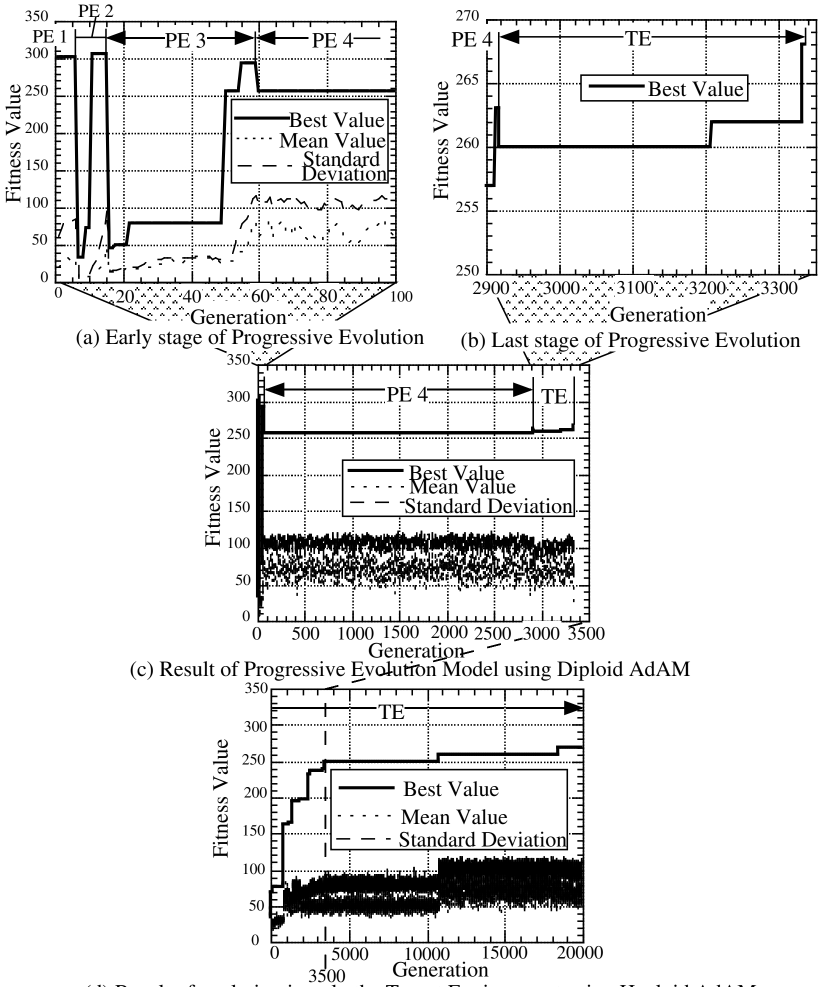
Table 1. Number of generations to produce the best individual in the final target environment

	Haploid AdAM	Diploid AdAM
PEM	13532	3331
nonPEM	18427	21698

5 Conclusion

Smooth evolution is a must for real-time evolutionary systems. The combination of the progressive evolution model and diploid-chromosome-based model was proposed to achieve smooth evolution. The effect of using this method was shown by comparing four types of evolutionary methods for an ant problem.

Although the experiments were limited to one application, the comparison clearly shows that the combination of the two methods is effective. It remains to be seen whether this is the case with other applications as well. The result, however, are so promising that we plan to establish the generality of the method in a wide range of other potential applications like job scheduling. The measurement of the environmental complexity requires further research, especially with respect to how this measurement can be generalized.



(a) Early stage of Progressive Evolution (b) Last stage of Progressive Evolution

(c) Result of Progressive Evolution Model using Diploid AdAM

(d) Result of evolution in only the Target Environment using Haploid AdAM

Fig. 6. Results of simulations

References

1. Bruce Albers et al. *MOLECULAR BIOLOGY OF THE CELL SECOND EDITION*, chapter 15. Garland Publishing, Inc., 1989.
2. Tomofumi Hikage, Hitoshi Hemmi, and Katsunori Shimohara. Hardware evolution system introducing dominant and recessive heredity. In *Evolvable Systems: From Biology to Hardware (LNCS 1259)*, pages 423–436. Springer, 1996.
3. Tomofumi Hikage, Hitoshi Hemmi, and Katsunori Shimohara. Progressive evolution model using a hardware evolution system. In *Artificial Life and Robotics*, pages 18–21, 1997.
4. Jun'ichi Mizoguchi, Hitoshi Hemmi, and Katsunori Shimohara. Production genetic algorithms for automated hardware design through an evolutionary process. In *IEEE Conference on Evolutionary Computation*, 1994.
5. R. E. Smith and D. E. Goldberg. Diploidy and dominance in artificial genetic search. *Complex Systems*, 6(3):251–285, 1992.
6. Xin Yao and Tetsuya Higuchi. Promises and challenges of evolvable hardware. In *Evolvable Systems: From Biology to Hardware (LNCS 1259)*, pages 55–78. Springer, 1996.