数理モデル化と問題解決　35−2
（２００１．６．２６）

適応度評価を導入した拡張二分決定グラフの進化手法

武藤 敦子　　中村 剛士　　伊藤 英則
名古屋工業大学

遺伝的アルゴリズム (GA) は組合せ最適化問題を解くための有効な手法の１つである。我々はこれまで遺伝的操作が効果的に作用する表現方法として n-BDD（多出力二分決定グラフ）を提案し、それが人工生命の振る舞いを表現するのに適した方法であることを示してきた。GA を本表現方法に適用する際の課題として、GA における一般的な課題である実行時間の短縮が挙げられる。本稿では、高適応度個体を得るまでの実行時間の短縮を目的とし，新たにニューラルネットを用いた適応度評価を導入することにより，無駄な遺伝的操作を省略する手法を提案した。また、これを簡単な競合問題に適用し提案手法の有効性を確認した。

An Evolutionary Method of an Extended Binary Decision Diagram using Fitness Prediction

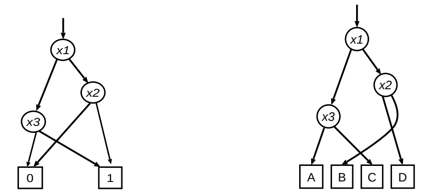
Atsuko Mutoh　　Tsuyoshi Nakamura　　Hidenori Itoh
Nagoya Institute of Technology

Genetic algorithm (GA) is an effective method to solve combinatorial optimization problems. We had proposed and developed a gene expression n-BDDs (n-output Binary Decision Diagrams), which lets genetic operations work efficiently. This gene expression is more effective in some of combinatorial optimization problems using GA, however it required a lot of execution time to acquire optimum solutions. Generally speaking it is a common issue of most of problems using GA. This paper proposes a novel approach to acquire the high-fitness individuals as fast as possible by using fitness prediction. We had adopted artificial neural networks to predict fitness, and attempted to verify its effect in a simple competition problem. In the experiments the proposed method had higher fitness than our conventional method.

1 Introduction

Binary Decision Diagrams (BDDs), which were originally proposed by Akers in 1978[1], are graph representations of Boolean functions. BDDs representation has been applied to various engineering fields. A gene expression system using n-BDDs, which are extended BDDs, is proposed in [2] to represent multi-value functions. Though the n-BDDs gene expression system represents behavior of agents and lets genetic operations work more efficiently in a simple competition problem, the simulation of agents takes a long time. Genetic algorithm (GA) is well known to be one of the most useful methods for solving combinatorial optimization problems, on the other hand in some problems have difficulty to solve in realistic time since calculating time is huge. Many studies have been made to find the more suitable individuals as fast as possible[3]. The purpose of this paper is to propose a method to acquire the high-fitness individuals as fast as possible for problems using n-BDDs. One of the most important genetic operations is crossover and it is expected an efficient evolution, however it does not always give good children. Thus it is the important matter of whether crossover is performed or not. We incorporate artificial neural networks (ANN)[4] to our method. ANN has been applied widely in many application domains, and it is capable of making predictions on process output based on the learned knowledge. Though there exists many combinations of GA and ANN in literature, many of them is that GA is used learning parameters selection and weight initialization for ANN[5]. On the other hand, ANN have been applied as fitness evaluators in genetic and evolutive applications, to a varying degree of success[6]. Our proposed method is similar approach to the fitness evaluators, and fitness prediction realizes to acquire the individuals which have high fitness in comparison with our conventional method. In the proposed method the ANN predicts children's fitness. Only when the predicted fitness is higher than a threshold value, the crossover is performed.

Section 2 revisits n-BDDs and a model of a simple competition problem is introduced in Section 3. We propose an evolutionary



(a) a BDD (b) an n-BDD

Figure 1: Graphical notation.

Table 1: Bit Assignments.

$X0$	hungry
$X1$	repletion
$X2$	enemy is visible far
$X3$	enemy is visible near
$X4$	plant is visible far
$X5$	plant is visible near

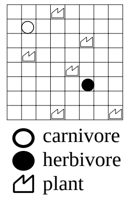


Figure 2: A field.

Table 2: Actions.

walk(W)	Animal moves to 8 neighborhoods at random.
runaway(R)	Animal moves in the opposite direction to the carnivore.
eat(E)	Animal moves in the direction of the food, and eat it if reached.
nothing(N)	Animal does not move.

method predicting fitness using ANN in Section 4. Section 5 gives experiments which compare the proposed method with the conventional method. Section 6 discusses the results of experiments and the future work using this method.

2 A gene expression with n-BDDs

A BDD is the graphical notation of a logical function (see Figure 1(a)). It has terminal nodes labeled by only two labels, true or false, while an n-BDD can have more than two labels and gives a value from some set of values which the labels denote (see Figure 1(b)). A circle denotes an input bit and is called a decision node, as shown in Figure 1. Each decision node has 3 directional edges, one of which comes from the outside and two of which go to the outside. An output value is denoted by a square which is called a terminal node. An output value is calculated from input bits as follows: First, look at the top decision node $X1$ and take either a left edge if $X1$ is 0 or a right edge if $X1$ is 1. The left edge and the right edge are called a 0-edge and a 1-edge respectively. Iterate this for the decision node indexed by the 0-edge or 1-edge until the edge indexes a terminal node. If the edge indexes a terminal node, the value in the node is the output. Genetic operations, such as crossover and mutation are defined to operate n-BDDs as gene[2].

3 A simple competition problem

A simple competition problem using n-BDDs is introduced in [2] and is a good test bed for evaluation. In an environment two artificial animals act. One of them, carnivore has a fixed action strategy and tries to capture another one, herbivore, which is an individual for GA and evolved using genetic operations. Both animal agents act according to their own n-BDDs. Mearnings of input bits are shown in Table 1. The order of the variables also obeys this table. A gene outputs a value out of the four actions given in Table 2. The carnivore genes do not output runaway.

The animal agents act in a two-dimensional array field shown in Figure 2 and proper number of plants are distributed in the field randomly. When a plant is eaten by a herbivore, another plant appears in a random position. A stage in our simple competition simulation starts with one herbivore, one carnivore and n plants. The number of steps for which the herbivore survived is taken as its fitness, which is used to select the next generation of herbivores. This model uses elitist strategies for selection.

Many experiments need to be done to evaluate the fitness considering the different initial conditions. Iterating many experiments, the individuals which have the same or similar chromosome appear often all throughout the evolutionary process. Our conventional method does crossover, mutation and calculates the fitness of every individual, and is considered to do unnecessary calculation as a result.

4 Fitness prediction using ANN

Figure 3 outlines the procedure used to treat one generation of a population. First of all, the same selection process as the conventional model is performed, and in the proposed method a temporary partner is selected randomly from the population. Then fitness

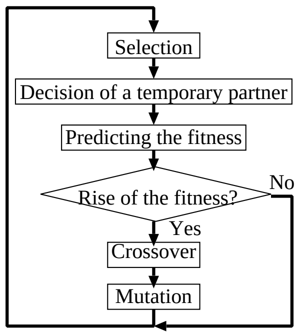


Figure 3: A flow chart of one generation.

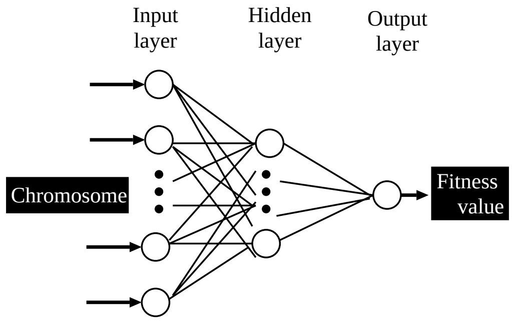


Figure 4: ANN used to predict fitness.

prediction is performed under the assumption that an individual is crossed with the temporary partner. As a result, if the child's fitness predicted is larger than the fitness of the individual, actual crossover, mutation and calculation of the fitness are carried out. If not, the individual joins to make the next generation without actual crossover, mutation and calculating fitness. Notice that the execution of mutation obeys user-specified probability and the mutation is not always performed in a processing cycle.

Fitness prediction is performed using ANN. The type of ANN adopted for this model is the multi-layered ANN shown in Figure 4. The input to the ANN is the chromosome of the child under the assumption that the individual is crossed with the temporary partner. In order to apply the ANN to a gene expression of n-BDDs, the two-dimensional graph expression of n-BDDs is converted into the one-dimensional chromosome. Figure 5 is an example of the conversion. N-BDDs have one output for one input, and the graph expression of n-BDDs could be rearranged to the truth table. The input data to the ANN, a chromosome, can now be represented by an action-output vector. Regarding action-outputs, there are 4 kinds in this model (see Table 2), and they can be arranged to be 4-bit binary expression. Thus the input data to ANN is a chromosome which consists of $2^6 \times 4$ long, the output is a child's predicted fitness.

The pre-experiments have been done for several populations in advance. ANN uses several individuals' chromosomes and their fitness as training data. Our proposed method uses the

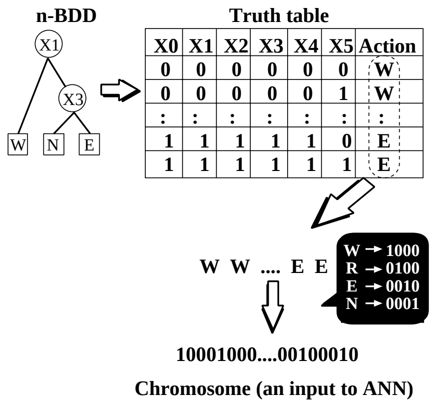


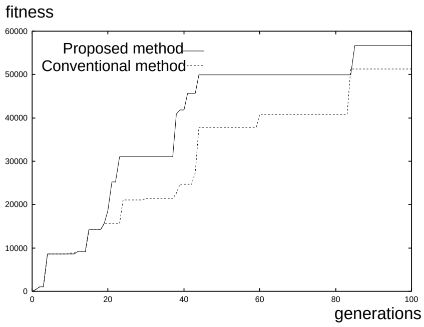
Figure 5: Conversion of n-BDD to chromosome

constructed ANN and does predictions. The structure of the ANN is set to 256:100:1 and is trained using standard back propagation.

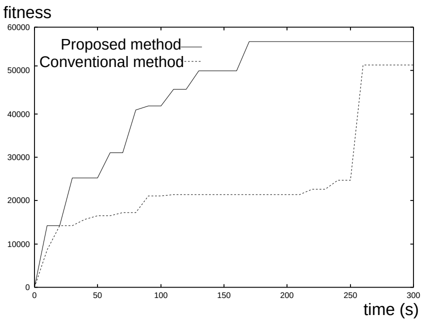
This model defines a fitness as the number of steps for which the herbivore survived. The simulation actually has to be done in order to calculate the fitness. Thus the higher fitness agents are generated, the more calculating time is need for the simulations. Comparing the simulation time with ANN's processing time, it would be better that the simulation is done if the former has a shorter time than the latter and vice versa. In the following experiments if the fitness of the individual become larger than 1,000, ANN processing starts and predicts the children's fitness.

5 Experimental results

This simple competition experiment has been done under the following conditions. The field size has 20×20. There is a carni-



(a) maximum fitness at each generation.



(b) maximum fitness at each execution time.

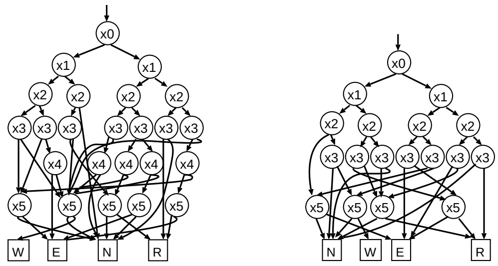
Figure 6: Experimental results.

vore, a herbivore and 30 plants. There are 100 identical populations and each of them contain 30 herbivore agents. The experiment has been performed on each population. Pre-experiments were performed using another 5 populations before the experiment was started. ANN was trained on the data generated from the pre-simulations before starting the 100 experiments. Figure 6(a) shows maximum fitnesses at each generation, and Figure 6(b) shows the one at each execution time.

6 Discussions

Our purposed method is to find a high-quality individuals as fast as possible as shown in Figure 6. Considering that, the fitness prediction of ANN has worked efficiently and the purpose has been achieved.

Figure 7 shows the n-BDD trees acquired by using each method. Each of Figure 7 (a) and (b) is shows the best individual's n-BDD at the 100th generation. It is observed that the proposed method has fewer decision nodes compared with the conventional method. One



(a) the conventional. (b) the proposed.

Figure 7: The best individual's n-BDDs.

of the major factors may be that the proposed method does not perform very many unnecessary crossover operations. This is something we must analyze and discuss more closely in a future project.

Generally speaking ANNs need a lot of time to learn training data. There is also the issue of how to choose the proper training data. This is an important issue in the use of ANN. These issues must also be considered and solved in the future. Solving this would improve our method's performance. In addition to this, we have future plans to apply our proposed methods to other problems using GA.

References

[1] S. B. Akers, 1978, *Binary Decision Diagrams*, IEEE Transaction on Computers, Vol. C-27, No. 6, pp. 509-516.

[2] A. Mutoh, S. Oono, K. Moriwaki, T. Nakamura, N. Inuzuka and H. Itoh, 1999, *An Evolutionary Method Using Crossover in a Food Chain Simulation*, Proc. 5th European Conference on Artificial Life, LNAI 1674, pp. 89-93, Springer.

[3] H. Mühlenbein, 1991, *Evolution in time and space-The parallel genetic algorithm*, Foundations of Genetic Algorithms, pp. 316-337, Morgan Kaufmann Publishers.

[4] D. E. Rumelhart, J. L. McClelland and the PDP Research Group, 1986, *Parallel Distributed Processing*, Vols. 1 and 2, Cambridge, MA, MIT Press.

[5] X. Yao, 1993, *Evolutionary artificial neural networks*, International Journal of Neural Systems, vol. 4, no. 3, pp. 203-222.

[6] A. R. Burton and T. Vladimirova, 1997, *Genetic Algorithm Utilising Neural Network Fitness Evaluation for Musical Composition*, Proc. International Conference on Artificial Neural Networks and Genetic Algorithms, pp. 220-224, Springer-Verlag.