Fitness Modification in Genetic Algorithms for Function Optimization Problems

Takanori Yoshida, Tomoharu Nakashima, and Hisao Ishibuchi
College of Engineering, Osaka Prefecture University
Gakuen-cho 1-1, Sakai, Osaka 599-8531, Japan
{dayoshi, nakashi, hisaoi}@ie.osakafu-u.ac.jp

Abstract

In this paper, we propose a fitness modification method to improve the performance of genetic algorithms. In the proposed method, the distance from each individual to the best one in the current population is used to modify its fitness value. First we examine the behavior of a genetic algorithm for various function optimization problems from the viewpoint of diversity of the population. Next we demonstrate that the performance of the genetic algorithm is improved by the proposed method for those problems that are difficult for the genetic algorithm to efficiently find optimal solutions. Finally, we examine the performance of our fitness modification method on a dynamically changing function optimization problem.

1 Introduction

The balance between exploration and exploitation is undoubtedly a key issue in the research of genetic algorithms. In genetic algorithms, this issue can be discussed in terms of the diversity and the selection pressure. That is, the diversity in the population corresponds to the exploration ability and the selection pressure corresponds to the exploitation ability of genetic algorithms. It is, however, difficult to find the best balance between the diversity and the selection pressure. For example, if one tries to keep a large diversity in the population, the search speed may slow down. On the other hand, if the selection pressure is high, the population is likely to converge to a local optimal solution. This issue has been discussed in various literature (for example, see [1]).

In this paper, we examine the effect of fitness modification on the performance of genetic algorithms from the viewpoint of diversity of the population. The aim of our fitness modification is to have high search ability with a large diversity in an entire population. We also examine the performance of the fitness modification method in a dynamic function optimization problem where the shape of the objective function changes over generation.

2 Performance of GAs

This section examines the performance of genetic algorithms on some function optimization problems. The performance is measured from the viewpoint of diversity as well as the search ability.

2.1 Function Optimization Problems

In the computer simulations in this paper, we use the following functions for function optimization. All of them are to be minimized.

$$\mathrm{F1}: f_1(x_i|1 \le i \le n) = \sum_{i=1}^{n} x_i^2, \quad x_i \in [-5.12, 5.11], \tag{1}$$

$$\mathrm{F2}: f_2(x_i|1 \le i \le n) = 10 \times n + \left[\sum_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i)\right], \quad x_i \in [-5.12, 5.11], \tag{2}$$

$$\mathrm{F3}: f_3(x_i|1 \le i \le n) = \sum_{i=1}^{n-1}\left[100\big(x_{i+1} - x_i^2\big)^2 + \big(1 - x_i\big)^2\right], \quad x_i \in [-2.048, 2.047], \tag{3}$$

$$\mathrm{F4}: f_4(x_i|1 \le i \le n) = \sum_{i=1}^{n-1}\left(100\big((i+1)y_{i+1} - (iy_i^2)\big)^2 + \big(1 - iy_i\big)^2\right), \quad y_i = x_i/i, \quad x_i \in [-2.048, 2.047]. \tag{4}$$

Function F1 is the first function of the DeJong's test suite, Function F2 is the Rastrigin function, Function F3 is the Rosenbrock function and Function F4 is the ill-scaled Rosenbrock function. The optimization problems of the above functions are static (that is, they do not change their shapes during the execution of optimization algorithms).

2.2 Entropy

In this paper, we use entropy as a measure of the diversity of the population [2]. The entropy measure of a population P is calculated as follows:

$$E(P) = -\sum_{l}^{L} \{p_l \log p_l + (1 - p_l) \log(1 - p_l)\}, \qquad (5)$$

where p_l is the proportion of the value 1 in the l-th bit in the population P, and L is the length of a bit string (i.e., an individual). We examine the entropy measure of the following subpopulations: $20 \times m$ neighboring individuals for best one in genotype space, where $m = 1, 2, 3, \cdots$. Figure 1 shows the subpopulations of which we examine the entropy. A low entropy measure means that a large number of individuals are the same bit string.

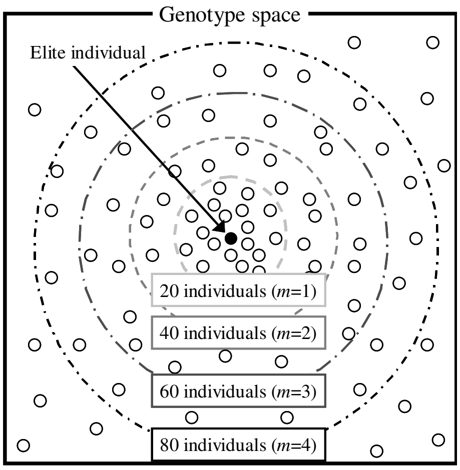

Figure 1: Examined subpopulations (m=1,2,3,4)

2.3 Experiments

We applied a genetic algorithm to each function optimization problem ten times. During the execution of the algorithm, we monitored the diversity of the subpopulation as well as the function values of elite individuals. Parameter specifications of the genetic algorithm in our experiments are shown in Table 1.

Table 2 shows the average function value of the obtained solutions over ten runs and the number of successful runs (R) where the optimal solution was found. We can see that the genetic algorithm could not find the optimal solution of F3 nor F4. In Figure 2 and Figure 3, we show the entropy measure of 20 neighboring individuals for the best individual and the entropy measure of the entire population, respectively. From these figures, we can see that during the execution for F3 and F4 the genetic algorithm keeps higher diversity maintained in 20 neighboring individuals and in the entire population than for F1 and F2. This is why the genetic algorithm could not find the optimal solution of F3 nor F4. We also show in Figure 4 the distribution of individuals in the final generation for F3. From this figure, we can see that the population does not converge around the optimal solution.

Table 1: Parameter settings

Population size	400
Crossover probability	1.0
Mutation probability	0.05
Stopping criterion (no. of generations)	500

Table 2: Simulation results for each problem

	Function value	R
F1	0.0	10
F2	0.0	10
F3	0.000219	0
F4	0.000155	0

3 Fitness Modification

In this section, we propose a fitness modification method and demonstrate that the performance of the genetic algorithm is improved by our fitness modification method.

3.1 Fitness Modification Method

In our fitness modification method, the best individual with the lowest function value in the population is selected from the current population. The selected

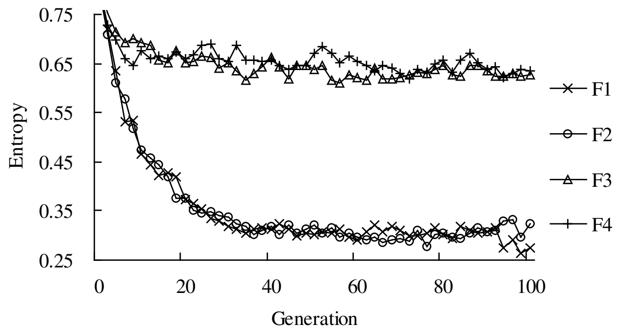

Figure 2: Entropy measure of 20 neighboring individuals for the best individual

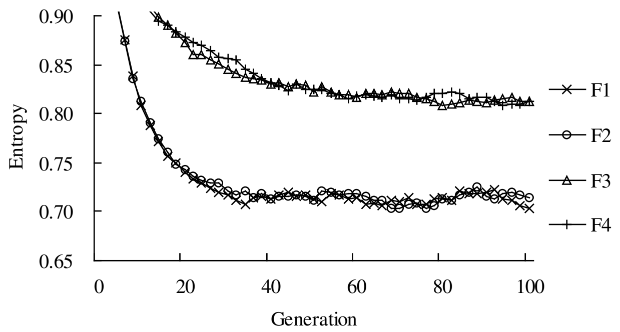

Figure 3: Entropy measure of the entire population

individual is called a reference individual. The fitness value is calculated for each individual in the population for a function minimization problem as follows:

$$fitness_i = F_i \times \left\{ \left(\frac{10d_i}{D} \right)^2 + 1 \right\}, \tag{6}$$

where F_i is the function value of the i-th individual, d_i is the phenotype distance from the i-th individual to the reference individual, and D is the length of the diagonal line in the phenotype search space.

The proposed fitness modification method in (6) means that the fitness value is increased if d_i is large (i.e., if the i-th individual is far from the reference individual.). On the other hands, if d_i is not large, the fitness value of an individual is almost the same as its objective function value.

3.2 Experiments

In this section, we use F3 and F4 in computer simulations. We applied the genetic algorithm with our fitness modification method to each function optimization problems ten times.

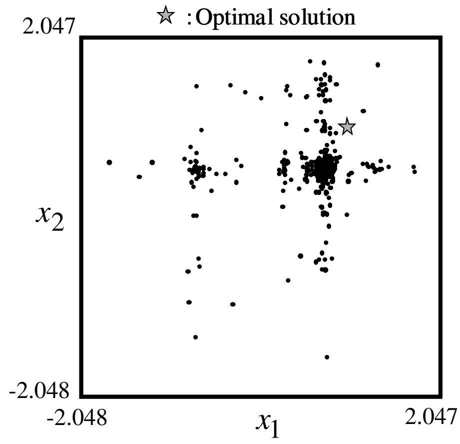

Figure 4: Distribution of individuals in the final generation for F3

Table 3 shows the average function value of obtained solutions over ten runs and the number of runs where the optimal solution was found. In this table, the genetic algorithm (GA) with our fitness modification method is denoted as GA+. From this table, we can see that the performance of genetic algorithms was improved by our fitness modification method. Figure 5 shows the entropy of various subpopulations in the final generation. From this figure, we can see that the entropy was decreased by our fitness modification method.

Table 3: Simulation results for F3 and F4

		Function value	R
F3	GA	0.000219	0
	GA+	0.0	10
F4	GA	0.000155	0
	GA+	0.0	10

4 Adaptability for Dynamic Environment

Dynamic environments form a difficult class of optimization problems for evolutionary algorithms. Branke [3] suggests that his memory-based genetic algorithm is able to efficiently adapt to changing environments. In this section, we examine the performance of our fitness modification method for a dynamically changing function.

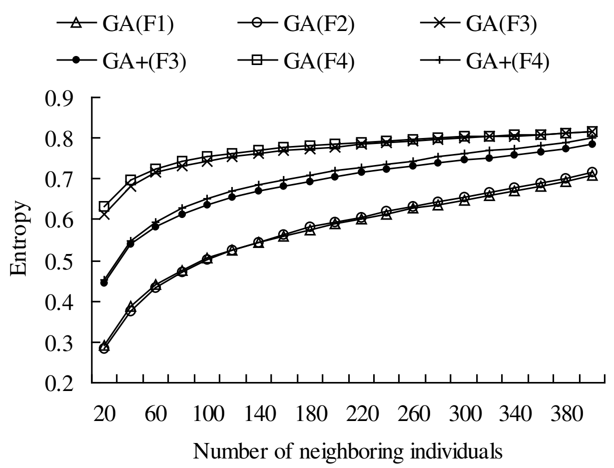

Figure 5: Entropy measure of subpopulations in the final generation

4.1 Function Optimization Problems

In this section, We examine the performance of our method on the following function:

$$\begin{aligned} \text{F5} : & f_5(x_i | 1 \le i \le n) \\ &= \sum_{i=1}^{n-1} \Big[100\big(y_{i+1} - y_i^2\big)^2 + \big(1 - y_i\big)^2 \Big], \\ y_i &= x_i - (\frac{t}{500} - 1), \quad x_i \in [-2.048, 2.047] \end{aligned} \tag{7}$$

where t is the index of the generation, which serves as a time index during the execution of genetic algorithms. The characteristic feature of the above function is that the optimal point gradually moves over generations while the optimal function value is always the same (i.e., 0.0).

4.2 Experiments

We examine the performance of the genetic algorithm and the genetic algorithm with our fitness modification method on the dynamically changing function optimization problem in (7). The genetic algorithms were executed ten times in the same manner as in the previous section. In the genetic algorithm, we examined three specifications of the tournament size (TS): TS=2, 3, 5. Figure 6 shows the best function value for F5 at each generation of each algorithm. From Figure 6, we can see that the modified version of the genetic algorithm (GA+) is more adaptable to the changing environment than the original version without the fitness modification method.

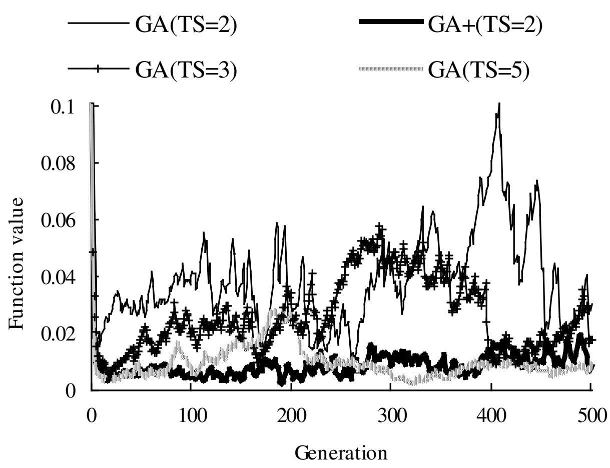

Figure 6: Best function value for F5

5 Summary

In this paper, first we examined the behavior of the genetic algorithm for several function optimization problems from the viewpoint of the diversity of the population. Next, we examined the effect of our fitness modification method for difficult problems where the genetic algorithm can not efficiently find their optimal solutions. In our computer simulations, we showed that the performance of the genetic algorithm was improved by our fitness modification method. Finally, we examined the performance of our fitness modification method on a dynamically changing function optimization problem. It was shown that the fitness modification method improved the adaptability of the genetic algorithm.

References

[1] D. Whitley, "The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best", *Proc. of the third International Conference on Genetic Algorithms*, pp. 116–121, 1989.

[2] E. K. Burke, S. Gustafson, G. Kendall, "Diversity in Genetic Programming: An Analysis of Measures and Correlation with Fitness", *IEEE Trans. on Evolutionary Computation*, Vol. 8, No. 1, pp. 47-62, 2004.

[3] J. Branke, "Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems", *Proc. of Congress on Evolutionary Computation*, pp. 1875-1882, 1999.