Multimodal Optimization Based on Global and Local Mutation Operators

Yong-Gun Jo*, Hong-Gi Lee**, Kwee-Bo Sim***, and Hoon Kang****

*School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Korea
(Tel : +82-2-816-8234; E-mail: iceman@ms.cau.ac.kr)

**School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Korea
(Tel : +82-2-820-5317; E-mail: hglee@cau.ca.kr)

***School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Korea
(Tel : +82-2-820-5319; E-mail: kbsim@cau.ca.kr)

****School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Korea
(Tel : +82-2-820-5320; E-mail: hkang@cau.ca.kr)

Abstract: Multimodal optimization is one of the most interesting topics in evolutionary computational discipline. Simple genetic algorithm, a basic and good-performance genetic algorithm, shows bad performance on multimodal problems, taking long generation time to obtain the optimum, converging on the local extrema in early generation. In this paper, we propose a new genetic algorithm with two new genetic mutational operators, i.e. global and local mutation operators, and no genetic crossover. The proposed algorithm is similar to Simple GA and the two genetic operators are as simple as the conventional mutation. They just mutate the genes from left or right end of a chromosome till the randomly selected gene is replaced. In fact, two operators are identical with each other except for the direction where they are applied. Their roles of shaking the population (global searching) and fine tuning (local searching) make the diversity of the individuals being maintained through the entire generation. The proposed algorithm is, therefore, robust and powerful.

Keywords: multimodal optimization, genetic algorithm, mutation

1. INTRODUCTION

Genetic algorithm (GA) introduced by John Holland in 1975, has been making an outstanding performance in many fields through its characteristics of parallel searching mechanism and the individual fitness evaluation executed at every generation [1, 2]. It still shows good results in numerical optimization, especially in optimization problem with at least one global extreme. However, it is difficult for GA to solve multimodal optimization problem, one of the most interesting topics in evolutionary computational world. One of the reasons is because simple genetic algorithm has a problem in maintaining diversity of solution in multimodal optimization problems, which leads to poor performance like taking a long generation time to obtain the optimum and converging on local extrema in early generation.

In this paper, we propose a new genetic algorithm with two new genetic mutational operators we created, named global and local mutation operators, respectively, and no genetic crossover. The proposed algorithm is similar to simple GA and the two genetic operators are as simple as the conventional mutation operator. They just mutate the genes from left or right end of a chromosome till the randomly selected gene is replaced. In fact, two operators are identical with each other except for the direction where they are applied. Their roles of shaking the population (global searching) and fine tuning (local searching) make the diversity of the individuals being maintained through the entire generation. The proposed algorithm is, therefore, robust and powerful.

We test the proposed algorithm for a simple but somewhat difficult numerical optimization problem under the various conditions.

2. SIMPLE GA

Previously mentioned, GAs are used to solve optimization problems. GAs work on a set of possible solutions called population and a solution is called individual or chromosome and generally coded in a string. In general this string represents one possible solution of the optimization problem, and could be the optimum we are looking for. The quality of each individual is called fitness and computed by a fitness function which should be carefully created or chosen.

A new population is generated by three genetic operators: reproduction, crossover and mutation. The population breeds new population, called offspring, by these operators and offspring replaces its parents' position. By doing this, generation goes by. The exact procedure of a simple GA is as follows: [1-4]

```
begin
  initialization
  repeat
    reproduction
    crossover
    mutation
  until termination-condition = true
end.
```

With the use of these three operators, every generation generates a population with a higher average fitness than the previous population.

Simple GA has shown its capability in a variety of unimodal optimization problems. In a multimodal problem there are difficulties for simple GA, because after a certain generations pass by the greater part of population converges to a certain string pattern. This early convergence problem results from the lack of diversity of individual in solution space.

3. GLOBAL AND LOCAL MUTATION OPERATORS

In this paper, we suggest two new genetic operators based on the concept of mutation with which the two operators of simple GA, crossover and mutation, are to be replaced to resolve the early convergence problem. The new operators are very simple to be implemented, fast, and quite robust. Their very important role is that they provide the diversity of solution which is hard to be achieved by mutation and especially crossover of simple GA in multimodal situations.

Now we are to introduce our suggesting operators more in detail.

3.1 Global Mutation Operator

The most significant trouble of simple GA in multimodal problems is that it is difficult for it to maintain the diversity of solution. So, it fails to achieve the goal to find out the optimum with high possibility. One of our operators, global mutation operator (GM) is intended to resolve this issue.

Given a string selected from a population, a location in the string is selected randomly. After that GM mutates each bit, a cell of the string, from the most significant bit (MSB) of the string to the bit where the randomly generated location indicates regardless of the preoccupied value in the cell. In our GA process, GM is applied to the population depending on the fitness value of individuals to reproduce a half of the next population.

GM's mixing up the left part of a string makes new starting points on solution space to search for a possible global solution. This enables our simple GA to have more chances to escape from local extrema than the existing simple GA with crossover and mutation.

3.2 Local Mutation Operator

On the contrary, local mutation operator (LM) mutates each bit in the right part of a string. Bits in the right side of a string make up least significant bits, which means bit changes in this part do not affect the fitness value of the individual considerably.

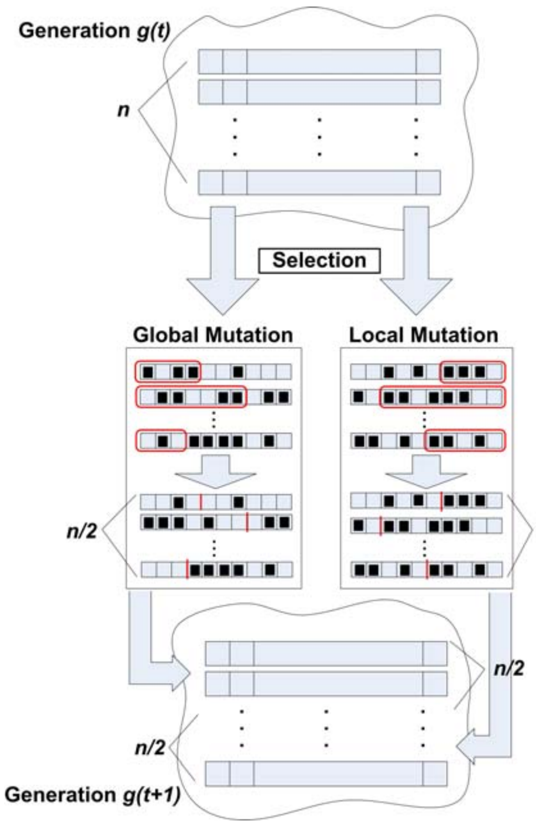

Fig. 1 This diagram describes how our new genetic operators, global mutation operator and local mutation operator, work. Each operator reproduces a half of a population.

A string is to be selected from a population usually by roulette wheel mechanism and a location in the string is selected randomly. By now the procedure is the same as GM follows. Then LM mutates each bit from the least significant bit (LSB) of the string to the bit where the randomly selected location regardless of the value in the cell. This is similar to that of GM except the part of a string where the operator works on. Next, LM reproduces a half of a population to complete the next generation's population.

The role of LM is fine-tuning solutions to move them close to the optimum. GM and LM cooperate with each other to implement a fast, reliable, and robust genetic algorithm for multimodal problems. GM makes individuals be jumping around the solution space to start searching new area for global minimum or maximum. Therefore, the diversity of solution is maintained through the whole generation. Meanwhile, LM lets chromosomes be fine-tuned and have much possibility to be reborn as better solutions than ever in generations. Fig. 1 shows the function diagram of the two new operators. The procedures depicted in Fig. 1 are repeated for every generation.

3.3 Our Simple GA, GLGA

We replace crossover and mutation genetic operator used in the conventional simple GA with our suggesting two genetic operators, global mutation and local mutation. We named this algorithm as Global & Local Genetic Algorithm (GLGA). All procedures of simple GA except for those for crossover and mutation remain untouched. All we have to do is to switch those genetic operators. The slightly modified procedure of a simple GA with the suggested genetic operators is as follows:

```
begin
  initialization
  repeat
    reproduction
    global mutation
    local mutation
    (elitism)
  until termination-condition = true
end.
```

As you see, there is another procedure for elitism. Elitism preserves the optimum found in the previous generation and passes it over to the next generation. It helps GA to avoid from shivering around the optimal value due to the effect of global mutation operator. [5]

4. SIMULATION

4.1 Test Function

In this paper, we have chosen a function for solving multimodal problem. This function equation is as follows:

$$f(x) = \begin{cases} a\sin\dfrac{\pi}{\alpha}x, & \text{if } 0 < x < \alpha \\ b\sin\dfrac{\pi}{\beta}(x-\alpha), & \text{if } \alpha < x < \beta \end{cases} \qquad (1)$$

where $\beta \ll \alpha$ and $a \ll b$

By making α very big rather than β, and letting b very big than a, the global maximum is hard to find because there is a trap, a local maximum, where GA must jump out of.

What the equation (1) look like can be confirmed in Fig. 2. In the conventional simple GA, once it has fallen into the local maxima it's very hard to jump out of that local extrema. We will show this in the result section.

4.2 Simulation Environment

First of all, coefficients of equation (1), a, b, α, and β should be determined. In our simulation, we choose coefficient values enough to make the problem sufficiently hard. We set each coefficient as follows: $a = 10,\ b = 1{,}000{,}000,\ \ \alpha = 1{,}000,$ and $\beta = 1$.

And the population size is 50. To get sufficient resolution in axis x ranging from 0 to $\alpha + \beta$, we decided that the length of a string is 17.

All simulation results are achieved through 50 times per test set.

4.3 Simulation Result I

First, we have a simulation result of comparing simple GA with the proposed genetic operators to the conventional simple GA with crossover and mutation with a certain probabilities. Fig. 3 shows the result. Crossover rate of simple GA is 80% and mutation rate is 1%. These values are just universal ones used in many applications. GLGA is a simple GA with our mutation operators. And the elitism is applied to both simple GAs. X axis means that 50 says one try leads us to the global solution within the 50th generation. Values on Y axis are the number of GA tries. For example, the first dot on the GLGA line indicates that for GLGA, 41 out of 50 tries found out the global value before 50^{th} generation is coming.

Global & local probabilities of GLGA are the probabilities on which GM and LM are applied to the selected chromosomes. So in this case, GM and LM are applied to all the selected individuals.

Fig. 3 shows that the suggested GA found the optimal value in very early generations but the conventional simple GA could not achieve the aim even after 500 generation. Its success rate is about 50%.

At least after 200 generation the modified simple GA using our genetic operators would find out the optimum with very high probability of almost 100%.

Fig. 4 shows the similar result as in fig. 3 and the elitism is not applied to GA. In this situation, our approach still shows better performance than the existing one.

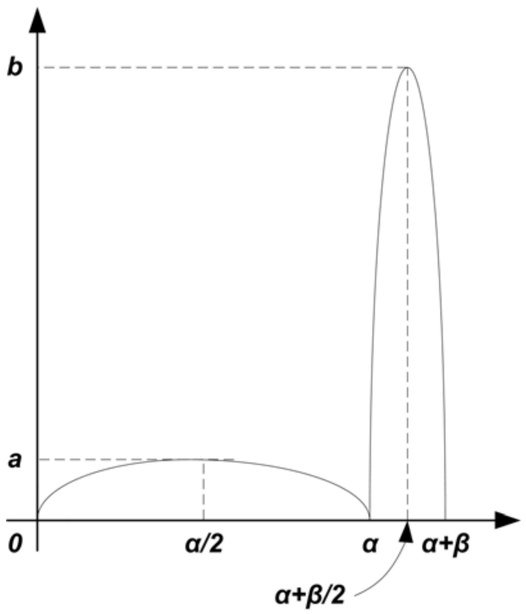

Fig. 2 The test function for GA to find out where the optimal value exists by using suggested genetic operators.

4.4 Simulation Result II

The aim of the second simulation is to examine the effect of probabilities that controls to apply global mutation and local mutation to the selected string. We set the probability 100%, 80%, and 50% and run simulations. The result is as we expected as shown in fig. 5. The higher the probability is the

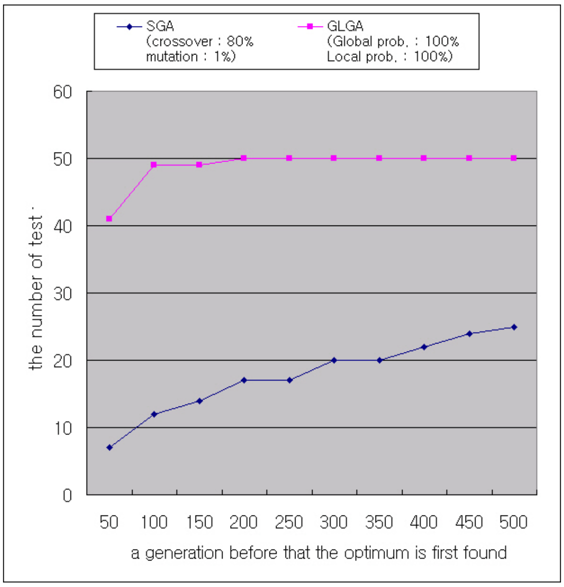

Fig. 3 This graph shows the comparison between the conventional simple GA and the suggested simple GA with the elitism. The above graph represents the result of the suggested simple GA. It has outstanding performance.

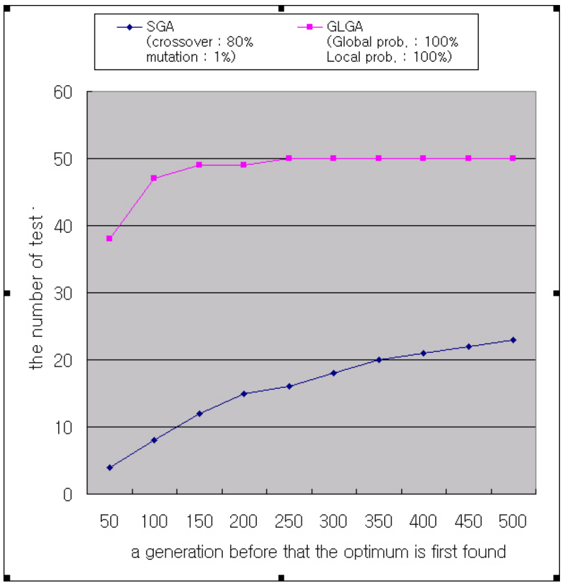

Fig. 4 This graph shows the similar result to that of fig. 3. In this case, the elitism is not used. The suggested GA still shows better performance.

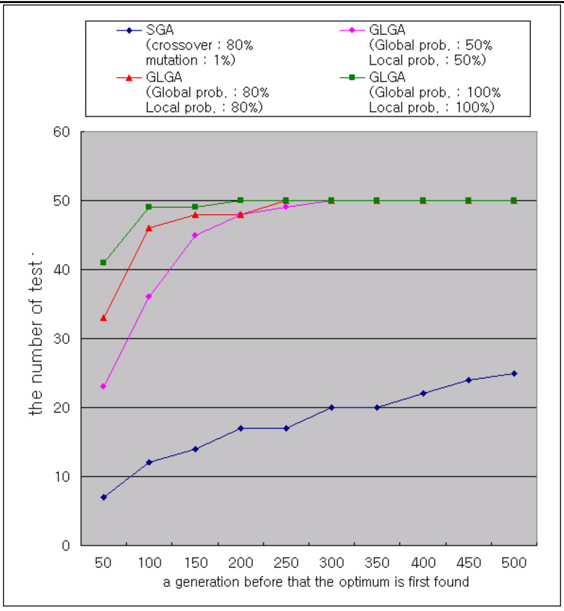

Fig. 5 The performance plots according to global mutation probability and local mutation probability. The lowest line represents simple GA.

more possible solutions produced.

In general, GLGAs makes very good performance over the conventional simple GA. Especially the higher the probability is the better the performance. Considering this result, there will be no need to introduce the probability of global and local mutation operators. Instead, we do two mutation operators on every selected string.

4.5 Simulation Result III

The last simulation result reveals the effect of elitism on the suggested operators and is shown in fig. 6. The result says that the elitism helps GA to find a global value quickly.

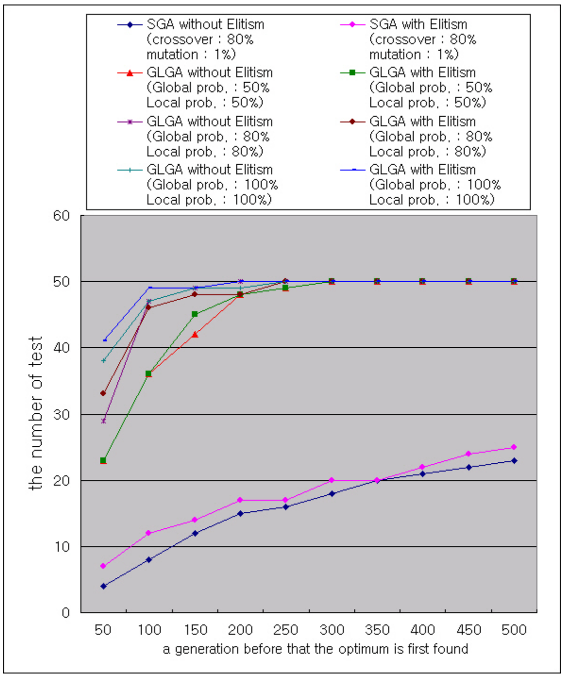

Fig. 6 The result plot graph shows that GA with elitism makes a good performance.

5. CONCLUSIONS

In spite of genetic algorithm's making an outstanding performance in many fields, especially in optimization problem with at least one global extreme, it is difficult for it to solve multimodal optimization problem, one of the most interesting topics in evolutionary computational world. One of the reasons is because simple genetic algorithm has a problem in maintaining diversity of solution in multimodal optimization problems.

In this paper, to overcome this GA's weakness we propose a new genetic algorithm armed with two new genetic mutational operators, global and local mutation operators. The proposed algorithm is similar to simple GA and the two genetic operators are as simple as the conventional mutation operator. Their roles of shaking the population (global searching) and fine tuning (local searching) make the diversity of the solution being maintained through the entire generation.

Our simulation results verify the suggested genetic algorithm is very fast, reliable, and robust in multimodal problem. Additionally, it is very effective to apply global and local mutation operators to all selected strings and it is wise to make use of the elitism.

REFERENCES

[1] J. H. Holland, "Adaptation in Natural and Artificial System," *The University of Michigan Press*, Michigan, 1975

[2] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," *Addison-Wesley Publishing Co. Inc.*, N.Y., 1989

[3] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs," *Springer-Verlag*, Berlin Heidelberg, 1996

[4] M. Mitchell, "An Introduction to Genetic Algorithms", MIT Press, 1996

[5] K. A. De Jong, "An Analysis of the Behavior of Class of Genetic Adaptive Systems," *Doctorial Dissertation, The University of Michigan*, Ann Arbor, Michigan. 1975.