Effect of Mutation and Effective Use of Mutation in Genetic Algorithm

Palitha Dassanayake
Department of Mechanical Engineering,
University of Moratuwa, Katubedda, Sri Lanka
palitha@mech.mrt.ac.lk

Abstract

Author tries to analyze the effect of genetic algorithm when mutation rate is selected randomly compared to fixed mutation rates or/and adaptive mutation rates. The results shows that though it is not always possible to get comparable results using randomly selected mutation rates, it is possible to obtain the required range with less number of trials than using fixed mutation rates for this nature applications.

Key Terms:

Genetic algorithm, mutation, fitness function.

1. INTRODUCTION

Genetic algorithm (GA) has been used for optimization problem for various applications [1] [2]. The best or the optimum result is not always guaranteed in using GA though in many cases, it can be expected to get good results but may not be the best. The reason for not getting the best solution could be due to the poor formulation of fitness function or the poor selection of the genetic parameters such as crossover method and the rate, mutation rate, parent selection method or/and next generation selection method. Various studies have been conducted to verify the effect of mutation in GA [4]-[6]. In this paper, author tries to investigate the effect of mutation in GA for randomly selected mutation rates.

The lower static mutation rates limit the ability to explore in the solution space and higher static mutation rates can unnecessary corrupt the good genes. Therefore, it is necessary to find the correct optimum mutation rate. Although many researchers have proposed different static mutation rates to obtain the optimum result, it is very difficult, though not impossible, to find an appropriate parameter setting for mutation rate or probabilistic mutation for the optimal performance [4].

To overcome the problem of selecting the right mutation rate, ImtiazKorejo and el [4] had conducted a comprehensive study of adaptive mutation operators for GA. The study was carried out for different types of adaptive mutation for different mathematical optimization problems. ImtiazKorejo and el [1] concludes that the performance of different adaptive mutation operators varies on different functions.

In using static mutation rates, it needs time and experience to find out the good mutation rate to yield the results. When adaptive mutation rate is used, it is not straight forward to find the right type of adaptive mutation since it is problem dependent.In adaptive mutation too, the change of the mutation rate is not purely random. Hence, in this paper purely random mutation rate is considered for a problem where the optimum solution is known. The author tries to explore the power of randomness in selecting the mutation rate.

2. THE MATHEMATICAL OPTIMIZATION PROBLEM

A linear programming mathematical problem is considered as it is possible to find out the optimum solution using simplex method. When the optimum solution is known, it is possible to identify whether the genetic algorithm yields the best solution or not. This example is only taken as a case study to check the effect of mutation.
The considered problem is as follows:
Maximize Z where,

$$Z = x_1 + 4x_2 + x_3 + 2x_4 + x_5 \tag{1}$$

Subjected to the following constraints,$x_i \geq 0, i = 1, 2, 3, 4, 5$

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 200 \tag{2}$$

$$2x_1 + x_2 + 3x_3 + x_4 \leq 250 \tag{3}$$

$$x_2 + 5x_3 + x_5 \leq 400 \tag{4}$$

$$x_1 + 2x_2 \geq 50 \tag{5}$$

$$x_1 + 2x_3 + x_5 \leq 100 \tag{6}$$

Using simplex method it can be found, that the maximum $Z = 800$ that is used as the bench mark for this analysis.

3. EXPERIMENTAL STUDY

Genetic algorithm gives the solution by evolving each generation. The first generation is randomly selected out of a total population. The total population has to be selected such that the optimum solution should be in the total population space. Also, the total population space should be very large so that it should not be possible to evaluate every solution in the total space computationally. If it is not large then, it is not worth using genetic algorithm but evaluate each solution separately and have the best solution found.In this work there are five variables having 8 bits for each variable. In that contest, the total population is selected so that code length is 40 (8×5) so that total population space is 2^{40} that is too large to evaluate for each possible combination.

Since genetic algorithm tries to obtain the best solution by evolving the initial generation randomly selected, the final result depends on the initially generated population. Hence, it is not correct to mention that one genetic parameter is better than the other parameter by merely comparing two solutions that could be a result of a pure chance. Therefore in this paper for each case, 20 trials are conducted starting with different initial generations randomly selected. For each individual evaluation, all the genetic parameters are as given in Table 1 and the same except for the mutation rate.

Parent selection is done by tournament selection from the randomly selected 3 individuals. One point crossover is done. Crossover point is selected randomly but only at the variable

Table 1: GENETIC PARAMETERS AND VALUES USED

Parameter	value
Number of trails with different initial generations	20
Number of generations for a trial	2000
Number of individuals per generation	60
Crossover rate	0.9
Bit size for each variable	8
Code length for five variables (5×8)	40
Number of elitist for next generation	8

boundaries since the concept of crossover is to only pass the gene information [3]. If any other point other than boundary is selected, the gene information of that particular variable gets corrupted in both springs resulting unexpected mutation. Since this study is to analyze the effect of mutation such unexpected gene corruption (can consider as unexpected mutation) should be avoided. Phenotype to genotype conversion is simply the same as the genotype so that minimum value is 0 and the maximum value is 255. Elitist strategy is used in next generation selection so that best 8 individuals are kept and taken for next generation and only 52 offspring are produced totaling of 60 (52+8) individuals for the next generation. Each trial is stopped after the maximum generation of 2000.

4. FITNESS FUNCTION AND DIFFERENT MUTATIONS USED

The objective function itself is considered as the fitness function if all constraints are satisfied when the solution is in the feasible region. If one or more constraints are not satisfied penalty is given and total penalty is given as the fitness function. Therefore if the fitness value is positive, the individual is in the feasible region and if it is negative one or more constraints are violated. Fitness function can be formulated from the eq. 7 to eq. 13 where f refers to fitness function.

$$f = \begin{cases} Z, & \text{if } P = 0 \\ P, & \text{if } P \neq 0 \end{cases} \tag{7}$$

$$P_i = \begin{cases} C_i, & \text{if } C_i < 0 \\ 0, & \text{if } C_i \geq 0 \end{cases} i = 1, 2, .., 5. \tag{8}$$

$$C_1 = 200 - x_1 + x_2 + x_3 + x_4 + x_5 \tag{9}$$

$$C_2 = 250 - (2x_1 + x_2 + 3x_3 + x_4) \tag{10}$$

$$C_3 = 400 - x_2 + 5x_3 + x_5 \tag{11}$$

$$C_4 = x_1 + 2x_2 - 50 \tag{12}$$

$$C_5 = x_1 + 2x_3 + x_5 - 100 \tag{13}$$

When the fitness function is constructed this manner, any solution outside the feasible region is negative and within the feasibility region is positive. With that the sign of the fitness function reflect whether a solution is in the feasible region or not. But this will have the disadvantage since there is sudden step variation of the fitness function at the boundaries of the feasible region.

To see the variation of results, different static mutation rates have been used. This is done to analyze the results of variation of mutation rate and to find out the mutation rates that yields better solutions. It may be possible to obtain much better solution or the best by changing other parameters or fitness function. But the objective of this exercise is to obtain the effect of mutation and hence other parameters are unchanged. In addition, one adaptation mutation is used so that mutation rate is high at early generations and it reduces with high generation. In the selected adaption mutation rate is a function of the number of generation so that it is obtained by eq.14 where m refers to mutation rate or the mutation probability and n refers to the number of generation.

$$m = 0.5/n \tag{14}$$

To introduce the random mutation rate for each individual, mutation rate is decided with a uniform random distribution between a maximum value and a minimum value that is given by Eq. 15. RAND function gives a random value between the maximum and minimum value and the probability of selection is uniform.

$$m = RAND(Maximum, Minimum) \tag{15}$$

This is conducted for 3 different sets of maximum and minimum values. Visual C++ is used as a programming language in the dot.net environment. The object oriented programming is used with user defined classes created by the author.

5. RESULTS

Table 2 presents the mutation rate considered, the maximum, mean and the standard deviation of the best fit values after 2000 generations of 20 trials for each mutation rate.

Table 2: BEST FIT RESULTS OF 20 TRIALS

Mutation Rate	Maximum	Mean	Std. deviation
0.01	652	758.95	61.85
0.02	780	656.40	32.22
0.04	782	741.00	58.05
0.05	790	780.00	10.65
0.06	798	782.10	4.51
0.07	785	776.00	3.30
0.1	780	758.95	14.27
0.25	759	690.15	30.68
RAND(0.02-0.1)	791	781.10	2.75
RAND(0.03-0.08)	794	782.95	3.67
RAND(0.01-0.25)	774	746.05	16.89
adaptive	744	637.20	59.89

The results from Table 2 shows that for the given fitness function and the genetic parameters for the given problem, constant mutation rate such as 0.05 and 0.06 gives better results. None of the trials could yield the best solution of 800. When the mutation rate is too law or too high, it gives poor performance. The given adaptive mutation is not comparable for this application. When the random mutation range is large (between 0.01 and 0.25), the result is not comparable. However when the random mutation range is little narrowed, the result is quite similar or better than the best fit values obtained at constant mutation 0.05 or 0.06. In all these four cases (constant rate 0.05, constant rate 0.06, RAND (0.02-0.1), and RAND (0.03-0.08) standard deviation is also law. That means getting a good solution for any trial is more.

Fig. 1 to Fig. 5 shows the results of each trial the final generation (after 2000 generations) for a selected mutation rates given in Table 2. Fig. 1 shows a low constant mutation rate 0.02, Fig. 2 shows better solutions at a medium constant mutation rate 0.06, Fig. 3 shows a high mutation rate 0.1, Fig 4 shows the adaptive mutation and Fig. 5 shows the random mutation between the range 0.02 and 0.1.

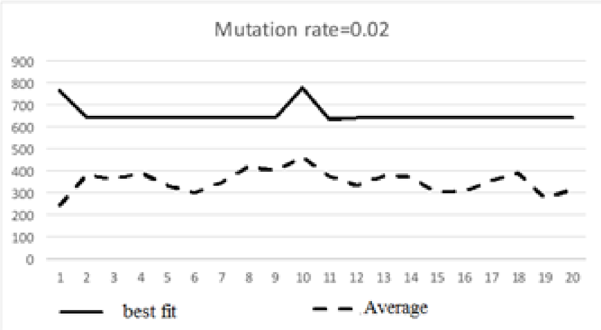

Figure 1: Final generation of 20 trials at mutation rate 0.2

From Fig. 1 to Fig. 3, it is clear that with the increase of mutation rate average value of the final generation decreases. This means that even after 2000 generations there are many individuals within the generation that is not even inside the feasible region because the average value is negative or near zero though the best fit has a higher value. This fact can be easily justified since higher the mutation, the probability that child carries muted genes is high. The result of the simulation obtained via selecting the mutation rate randomly is quite similar to the result in one of the optimal mutation rates. This can be obtained by the result given in Table 2 as well as the comparing the average and the best fit values in Fig. 2 and Fig. 4. The result obtained for the adaptive mutation is quite different from others when we compare the average value. Average value is quite near the best fit value and in certain trials it is the same. The reason could be due to the fact that the mutation is a very low value at the final generation (0.00125 = 0.5/2000 but a high value in first generations. It is necessary to get each generation best fit and the average value to see the effect of mutation rate. Fig. 6, Fig. 7 and Fig 8 shows the all generation result of one trial at mutation rate =0.02 (low rate) , mutation rate=0.1 (high rate) and adaption mutation where it is low at the initial generations and gradually decreases to a low value at higher generations.

At low rate of mutation average value is high (Refer Fig.6) and at high rate of mutation average value is low or negative (Refer Fig. 7). For both instants the variation of the average value is quite high (Refer Fig 6 and Fig. 7). But in adaptive

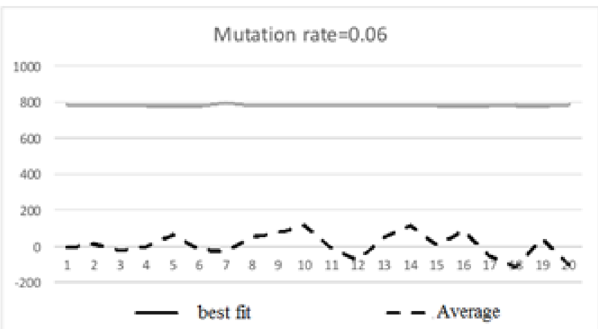

Figure 2: Final generation of 20 trials at mutation rate 0.06

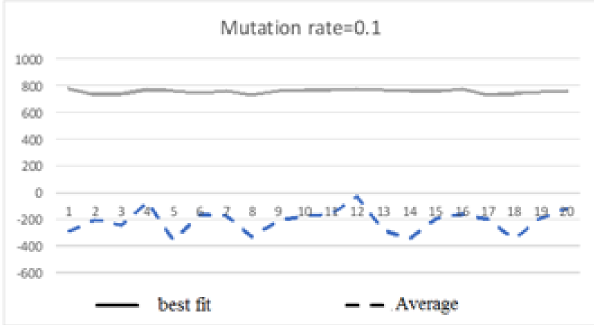

Figure 3: Final generation of 20 trials at mutation rate 0.1

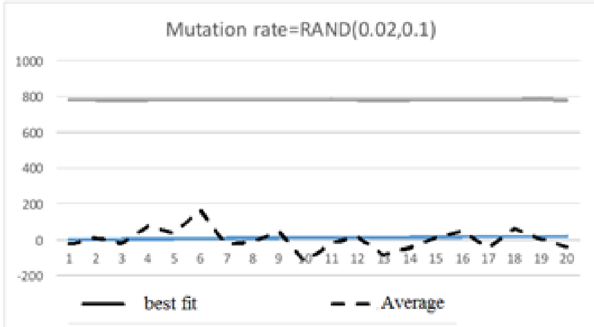

Figure 4: Final generation of 20 trials at random mutation

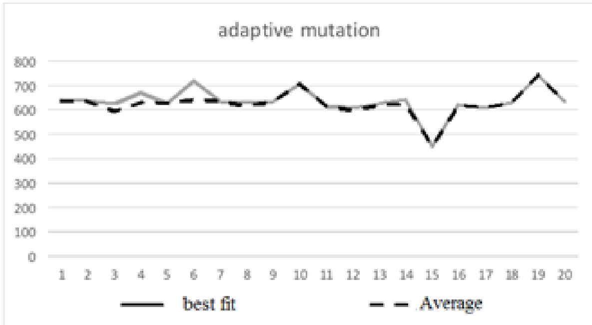

Figure 5: Final generation of 20 trials at adaptive mutation

mutation after a few number of generation (< 100), the average value is near the best fit value and the variation is low or negligible (Refer Fig. 8)

6. CONCLUSIONS AND FUTURE WORK

The results show that at a particular constant mutation rate GA can give a better solution. But it requires a lot of trials and experience to find out required mutation rate. There are different types of adaption mutation rates. ImtiazKorejo and el [4] has proved that it is applicant dependent and hence only adaptation type is used in this work. It has not given a comparable solution. Selection of the mutation rate randomly gives similar result to optimum mutation rate solution provided that the selection range is not too large. Use of random mutation rates does not require many trials to obtain a comparable solution since it is always easier to find a range than a specific value. Hence, it can be concluded that use of random mutation rate is better than using a constant mutation rates and/or trying different types of adaption mutations.

This is conducted only for one type of application. It is expected to verify this results by conducting the experiments

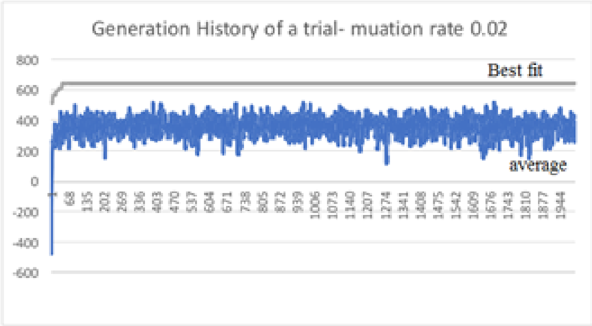

Figure 6: Best fit and average of a trial at mutation rate 0.02

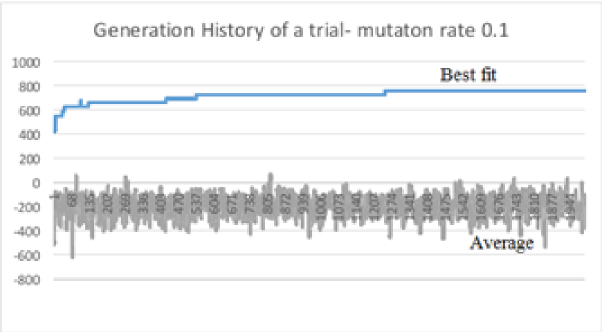

Figure 7: Best fit and average of a trial at mutation rate 0.1

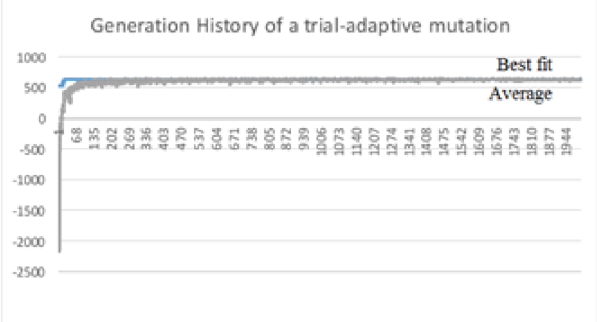

Figure 8: Best fit and average of a trial of adaptive mutation

for different types of applications. The fitness function used in this application may not be the optimum since using optimum rates has not yield the best solution. Therefore different fitness functions has to be verified for this application and variety of application to see how best to use fitness functions to get the best solution.

References

[1] P Dassanayake, K Watanabe, and K Izumi, Fuzzy Behavior-Based Control for a Task of Three-Link Manipulator with Obstacle Avoidance, J. of Robotics and Mechatronics, vol. 11, no. 6, pp. 502-509, 1999

[2] P Dassanayake, K Watanabe, K Kiguchi and K Izumi, Robot Manipulator Task Control with Obstacle Avoidance Using Fuzzy Behavior Based Strategy, J. of Intelligent and Fuzzy Systems, 2002

[3] K.F. Man, K. S. Tang, S. Kwong and W. A. Halang, Genetic Algorithm for Control and Signal Processing, London, UK: Springer-Verlag, 1997, pp. 27-28.

[4] ImtiazKorejo, Shengxiang Yang, and ChangheLi, A Comparative Study of Adaptive Mutation Operators for Genetic Algorithms, Metaheuristics International Conference, 2009, pp id1-id10.

[5] S. MarsillLibelli, and P. Alba, Adaptive mutation with Genetic Algorithm, Soft Computing 4(2000), pp. 76-80

[6] Matthias Khn, Thomas Severin and Horst Salzwedel, Variable Mutation Rate at Genetic Algorithms: Introduction of Chromosome Fitness in Connection with Multi-Chromosome Representation, International Journal of Computer Applications , Volume 72 No.17, June 2013.