Cmsc 191 Exercise: Solving a Quadratic Function using the Genetic Algorithm

Gimel David F. Velasco
Department of Mathematics and Computer Science
University of the Philippines Baguio
gfvelasco@up.edu.ph

ABSTRACT
This paper demonstrates a simple implementation of the Genetic Algorithm on solving a quadratic function using binary arrays to represent integers.

1. INTRODUCTION
The Genetic Algorithm is an algorithm made by Holland on 1975. The Algorithm is inspired by the concept of Genetics and Natural Selection. It approaches a particular problem by looking for the fittest individual and considers it as the solution to the problem.

2. METHODOLOGY
The program will seek the solution of the following quadratic equation

$$f(x) = x^2 \tag{1}$$

in the interval [-100,100]. Equation 1 is also the set fitness function of the Genetic Algorithm. Furthermore, there will only be a maximum of 10 generations of which the fittest of this generation will be considered as the solution to the quadratic equation. If ever a generation reaches a certain average, the cycle will stop and the fittest of the said generation will be picked as the solution. Every generation will only hold a maximum of 40 chromosomes. Each chromosome consists of a binary array of 8-bits.

The program followed the flowchart of the Genetic Algorithm over several generations [1]. Except that the mutation process is not applied on the chromosomes (binary arrays) in the algorithm. The different processes and how they are implemented in the code is further explained in the succeeding subsections.

2.1 Initialization
In the initialization step, a zero 3D array of dimensions 10x8x40 was initialized. This 3D array will hold the 10 generations of 8-bit binary population of 40 chromosomes per generation. Found in the 8-bit chromosome is 1 sign bit and 7 value bits. The code is designed so that the value bits won't exceed 100 since 7-bits could reach up to the integer value of 127. If the program detects an unqualified chromosome, it will replace it with another one until it is qualified to be part of the population. The Initialization process ends when there are 40 chromosomes of which their integer value is within the interval [-100,100].

2.2 Selection
On selecting the parents of the next generation, the program will first rank the entire generation based on the fitness function. The chromosomes will be sorted in ascending order using the bubblesort. After this is done, the algorithm will pick the first 15 chromosomes and have them crossed over with the second 15 chromosomes. The fittest will then be found in the very first index of the population.

2.3 Crossover
The algorithm implements a single point crossover at the middle of each chromosome. The first half of the first chromosome will be swapped with the second half of the second chromosome. For example we'll take two chromosomes about to be crossed over: 1000|0001 and 0110|0011. This then will become 0011|0001 and 0110|1000. These crossovers will then be done for the 15 pairs of chromosomes in the generation.

2.4 Insertion
After the 15 crossovers, the resulting 15 offsprings will then be replacing the last most unfit 15 chromosomes of the generation. After inserting the 15 offsprings, the generation will be ranked again to be checked for the stopping criteria. All in all, the Elitist Strategy is implemented in the algorithm which results in the survival of the few fittest and the replacement of the unfit chromosomes.

2.5 Stopping Criteria
Found in this algorithm is two stopping criterion. One is if the generations have reached 10 (Number of Generations Completed) and the other is if the immediate generation has reached a favorable fitness average of less than 30 (Population Mean Deviation). If the algorithm has stopped, the program will take the fittest of the latest generation and consider it as the solution to the quadratic equation 1.

3. RESULTS AND DISCUSSION

For several runs and testings, the algorithm ends before the generations would reach 10. Averagely, the algorithm ends at te 6th generation of average fitness less than 30. The solution found by the algorithm is 0. Which is exactly the solution to the quadratic equation. The said events happen if there was a 0 in the first initialized generation with a total run time of averaging 0.7 second. However, if the initial generation did not have any 0 in it, the algorithm still stops by population mean deviation and stops at the 8th generation with a runtime of averaging 1.0 second. But with all those said, in all the runs and testings, all the solution that was found is 0.

4. CONCLUSION

From finding the solution to the quadratic equation 1, the way the genetic algorithm is implemented in this code is capable of solving for the exact solution. Furthermore, it gets the exact solution consistently in all the test runs performed.

5. REFERENCES

[1] Goldberg, D.E.,"Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley, 1989