Genetic Algorithm Overview and Modeling for Function Optimization Using Benchmark Function

***Parul Gupta**

***Abstract*— Genetic algorithms (GA) are a type of evolutionary algorithms based on the principle of natural evolution and heredity. Genetic algorithm works in the same way as nature does. In this paper, GA is described using De Jongs' benchmark Function example for function optimization with different selection techniques. The fitness value of function for different number of generations is compared keeping all other parameters constant, except the selection techniques i.e. roulette wheel and roulette wheel with elitism. The results show that roulette wheel with elitism gives better result than roulette wheel selection for function optimization.**

***Index Terms*— De Jongs' Function, Elitism, Evolutionary Algorithm, Genetic Algorithm (GA), Roulette Wheel.**

I. Introduction

Genetic algorithms (GA) are a type of evolutionary algorithms [1-4]. It is based on the principle of natural evolution and heredity. Genetic algorithm works in the same way as nature does. During evolution, each species faces the problem of searching for beneficial adaptations to complicated and changing environment. Genetic algorithm follows the structure of the evolutionary program and maintains a population of individuals where each individual represents a potential solution to the problem. Each individual fitness value is evaluated and new population is generated using probability distribution based on fitness value of individuals. Genetic operators are applied on the new population to form new individuals. Genetic algorithm works by maintaining a proper balance between exploration of search space and exploitation of best solution using genetic operators. After some generations the program is converged hoping best individual represents a near optimum solution of the problem.

II. Steps for implementation of GA

A genetic algorithm when applied for a particular problem follows following steps:

A. *Representation*

In GA, the individuals are represented as a string of length n like below:

$$x = x1\ x2\ \ldots\ldots xn$$

The individuals are also called as chromosomes. Chromosomes are made up of units called genes. The different values a gene can take are called alleles. The process of the representation of individuals is called coding. The coding depends on the problem that needs to be solved. The typical way of coding is the binary coding where each chromosome is represented as a binary string. The length of string used to represent chromosome depends on the user required precision.

B. *Initialization*

As GA starts with some initial population so it is generated in a totally random manner.

C. *Evaluation*

Each individual of the population is evaluated for some fitness value which is used to select an individual for reproduction of next generation. Evaluation function works like an environment rating solutions in terms of fitness value.

D. *Selection*

All the individuals in one population have the possibility of selection and an individual can be selected more than once as the selection is random. GA uses natures' principle of survival of the fittest i.e. Individuals with high fitness value are more likely to be reproduced as compared to those with low fitness value. There exist some schemes for the selection process like roulette wheel, ranking methods, tournament and elitism selection etc

E. *Genetic operators*

Genetic operators are used to create new solutions based on the existing solutions. Basically, there are two types of operators: crossover and mutation.

• *Crossover* takes two individuals and generates two new individuals by swapping corresponding segments. It combines the features of two individuals for information exchange. It works by exploitation of best solution. Every

Manuscript received Feb, 2016.

Parul Gupta, Asst. Professor, Computer Science, P.I.G.G.C.W. Jind (Haryana), India.

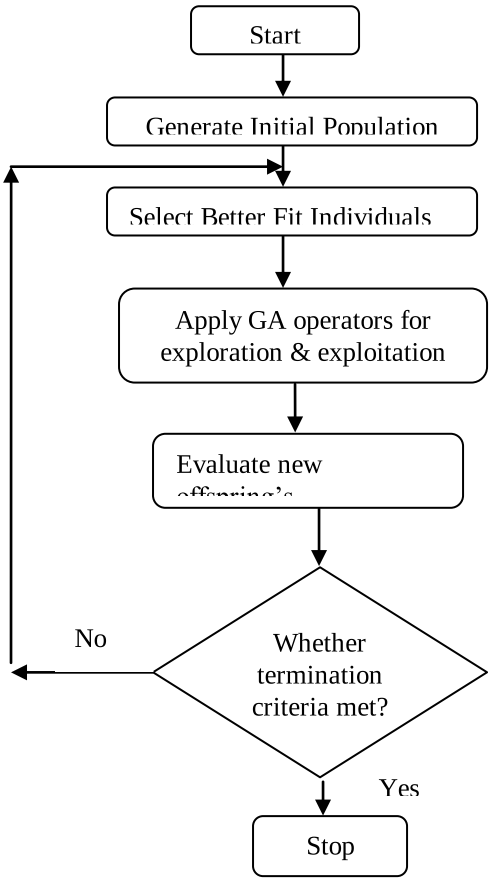
ISSN: 2278 – 1323

chromosome in the population has an equal chance on which crossover can be applied. The probability of crossover specifies the number of chromosomes on which crossover is to be applied.

- *Mutation* changes individual genes values to generate new solution. It is used for the introduction of some extra variability into the population. It works by exploration of search space. Every gene of all the chromosomes in the population has an equal chance on which mutation can be applied. The probability of mutation specifies the number of genes on which mutation is to be applied.

In GA algorithm, proper balance between exploration and exploitation is necessary and maintained by probability of crossover and mutation. After selection, crossover and mutation the new population is ready and evaluated and same process of selection, crossover and mutation is repeated for fixed number of generations.

III. Flowchart of Genetic Algorithm



IV. Proposed Work and Methodology

In this paper, De Jong's F1 test function which is continuous, convex and unimodal, used to evaluate the performance of GA [3]. The function is

$$f1 = \sum_{i=1}^{n} x_i^2$$

where x_i varies from $-5.12 \leq x_i \geq 5.12$ and i=1 to n.
Global minimum is at: f (x)=0, x(i)=0, where, i =1:n.

GA is implemented for above function optimization with i=1 with the following parameters:

i. ***Size of chromosome***: Binary representation is used for representation of chromosomes and precision of 6 digits after the decimal point is taken. So, the chromosome string is of 24 genes.

ii. ***Size of population***: 10

iii. ***Initialization:*** Initialization of chromosomes for initial population is done randomly.

iv. ***Selection method***: Roulette wheel and Roulette Wheel with Elitism used for selection of individuals for next generation.

In roulette wheel selection, selection of new population is done on the basis of probability distribution based on fitness value and there may be chances of not getting the most promising blocks in next generation either due to non selection or due to the applicability of operators.

In roulette wheel with elitism selection, selection of new population is done on the basis of probability distribution based on fitness value but copies best two individuals to new population to remove the chance of not having best solution in final generation which occurs in between generations.

v. ***Crossover***: single point crossover with a probability of 0.6

vi. *Mutation*: for each bit with a probability of 0.01

vii. ***Number of generations***: varies as 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100.

V. Results

In this section, comparison of two famous techniques i.e. roulettes wheel selection and roulette wheel with elitism is shown. In table I, fitness value of both techniques for different number of generations is shown.

Table I: Fitness values with different selection techniques

Number of generations	Fitness Value	
	Roulette Wheel Selection	**Roulette Wheel Selection with Elitism**
Initial Population	0.122625	0.152011
10	10.7743	0.0048405
20	10.799	0.00184996
30	3.36279	0.00161593
40	3.36294	1.23811e-007

ISSN: 2278 – 1323

50	3.36308	6.15129e-009
60	0.401803	1.54981e-009
70	0.188245	1.54981e-009
80	0.167184	2.32831e-012
90	0.00982977	2.32831e-012
100	0.00982977	9.31323e-014

From table I, it is clear that roulette wheel selection with elitism gives better result as compared to the roulette wheel selection without elitism. In roulette wheel selection, best results are not derived for each successive generation as most promising blocks of individuals undergoes changes and transformed to less fit individuals.

VI. Conclusion

In this paper, GA is described using example for function optimization with different selection techniques. The fitness value of function for different number of generations is compared. All other parameters are kept constant, except the selection techniques i.e. roulette wheel and roulette wheel with elitism. From the implementation results, it is found that roulette wheel with elitism gives better results than roulette wheel selection without elitism.

REFERENCES

[1] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs (New York : Springer-Verlag, 1994).

[2] D. E. Goldberg, The Design of Innovation: Lessons from and for Competent Genetic Algorithms.Vol. 7, Springer Science & Business Media, 2013.

[3]F. Della Croce, R. Tadei, and G. Volta, “A genetic algorithm for the job shop problem,” Computers & Operations Research, vol. 22, no. 1, pp. 15–24, 1995.

[4]D. E. Goldberg and H. John, “Holland. Genetic algorithms and machine learning,” Machine learning, vol. 3, no. 2–3, pp. 95–99, 1988.

[5] J. G. Digalakis and K. G. Margaritis, “An experimental study of benchmarking functions for genetic algorithms,” International Journal of Computer Mathematics, vol. 79, no. 4, pp. 403–416, 2002.

ISSN: 2278 – 1323