Volume 4, Issue 4, April 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Implementation of Dejong Function (Sphere model) by using Genetic Algorithm

Shaifali Aggarwal*
GGITC, Ambala
India

Dr. Puneet Goswami
Professor, GGGI, Ambala
India

Abstract— :- Genetic algorithms are searching and optimization techniques inspired by biology. Genetic algorithm can be applied in many problems like mathematical optimization and simulation parameterization. The main use of genetic algorithm is in optimization problems like travelling salesman problem(TSP), dejong function etc. In this paper, the problem of finding the optimized value of dejong function by genetic algorithm has been discussed. The researcher found the lot of results which is shown by graphs and table. All the results which are shown in this paper are of dejong function1's result. This paper shows the minimum values of the genetic algorithm by implementing the dejong function.

Keywords— Crossover, Dejong Function, Genetic algorithm, Mutation, Selection.

I. INTRODUCTION

Genetic algorithm[1] was first invented by john Holland in 1970's . Genetic algorithm are basically the search methods based on the principle of natural selection and genetics[2]. Genetic algorithms give the step by step procedure to solve the problems but the algorithm is based on the genetic models. Each solution of the problems in genetic algorithm is represented by the chromosomes[3]. Chromosomes are the set of genes or the string which contains the numbers of genes. Each chromosome is evaluated by their fitness value which is computed by the objective function of the problem. Basic flow chart of genetic algorithm is:

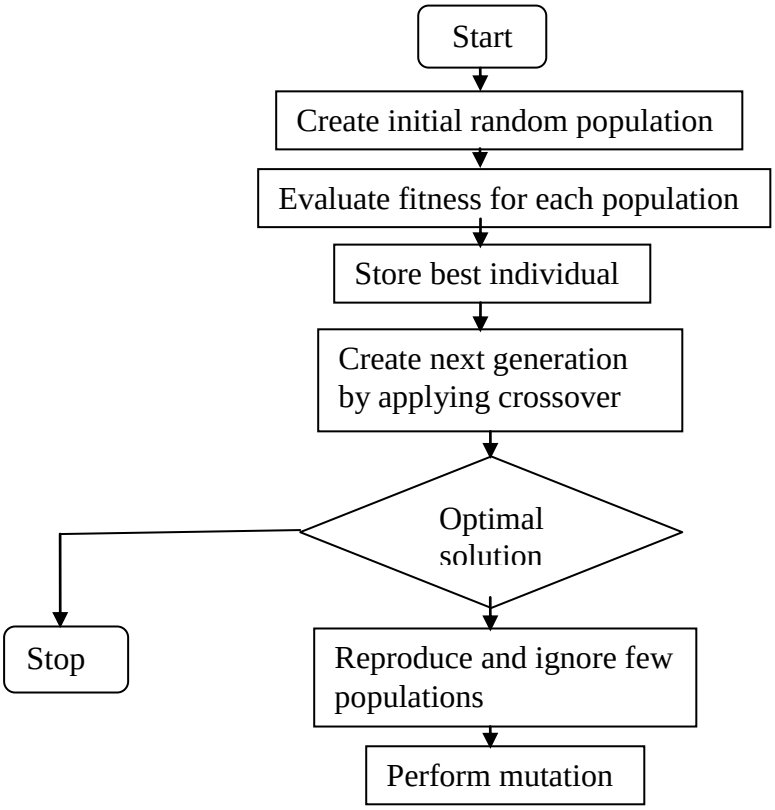

Fig: 1 Flow chart of genetic algorithm

In the genetic algorithm there are some steps which are used to solve the problems and to find the optimal solution of the problems. These steps are as:

1. Initialization
2. Selection
3. Crossover
4. Mutation
5. Replacement

1. **Initialization:** Initialization is the first step of genetic algorithm. In this step individual solutions are randomly generated to form an initial population. The size of the population depends on the nature of the problem but it randomly contains several hundreds or thousands of possible solutions.
2. **Selection:** Selection is the next genetic operator in the reproductive phase of genetic algorithm. Selection is the process in which the chromosomes are randomly selected from the population according to their evaluation function. Selection of an individual is directly proportional to their fitness function. Selection of an individuals in the population is fitness dependent [4] .Selection chooses more fit individuals in analogy to Darwin's theory of evolution that is the survival of fittest [5].
3. **Crossover:** The crossover operator is a genetic operator that combines the two chromosomes (parents) to produce a new chromosome (offspring). This scheme is used to produce a better offspring after the selection methods. Crossover operator is applied to the mating pool with the hope that it creates a better offspring. This crossover operator proceeds in three steps as:
   i. The operator selects the random pair of the two individual strings for the mating.
   ii. A cross site or the points is selected at random along the string length.
   iii. Finally the position values are swapped between the two strings.
4. **Mutation:** Mutation performs after the crossover. Mutation changes one or more gene values in a chromosome from its initial state. Mutation gives the better solution to the genetic algorithm. Mutation helps to prevent the algorithm to be trapped in a local minima[3]. Mutation occurs during evaluation according to a user defined mutation probability. Mutation probability should be low if the probability is very high then the search will turn in to the random search.
5. **Replacement:** Replacement is the last step. In this two parents are drawn from a fixed size of population , they breed two children but not all four can return to the population so two of them must be replaced that is once an offspring are produced, a method must be determine which of the current members of the population, is replaced by the new solution. There are two methods of maintaining the population that describe that which of the individual is replaced and which is not these are:

   i. **Generational updates**: In this genetic algorithm creates the new offspring from the individuals of an old population by using some genetic operators and then places these individuals in a new population.
   ii. **Steady state updates**: This is different to the generational update in that there is typically one single new member inserted in to the population at any one time. Replacement strategy is that which describes that which member of the population is replaced with the offspring. The algorithm of the steady state is as[6]:
      1. Select two parents from the population.
      2. Create an offspring using crossover and mutation.
      3. Evaluate the offspring with the fitness function.
      4. Select an individual in the population, which may bereplaced by the offspring.
      5. Decide if this individual will be replaced.

II. RELATED WORK

A paper by Ebrahim Bagheri and Hossein Deldari[7] describes that genetic algorithm is the search method which is used for solving the optimization problems. Parallel versions of the genetic algorithms are easily implemented and increases the performance of the algorithm[8]. In this paper authors propose a new method for fuzzy parallel genetic algorithms, in which a parallel client server single population fuzzy genetic algorithm is configured to optimize the performance of the first three Dejong functions in order to reach a global solution in the least possible iterations[8]. Rakesh et al[9] describes that genetic algorithm can be used for the number of NP hard problems such as dejong function, travelling salesman problem(TSP), etc. This paper describes that the performance of the genetic algorithm can be improved by adding the local seach in any phases of the algorithm. In this paper performance of the genetic algorithm is observed. Zhang fang, Li na and Li jinhui[10] focuses design and realization of the test paper composition model that is established and chromosome encoding method of test paper composition, adaptability function and genetic operator. This paper basically describes the test composition of the algorithm. K.F.Man, K.S.Tang and S.Kwong[11] describe that "why" and "when" the genetic algorithm is used as an optimization tool. This paper also describe the capability of genetic algorithm for solving the complex and conflicting problems. The basic purpose of this paper is to introduce the emerging technology to engineers who may have less or no knowledge of genetic algorithm. Meera kapoor and vaishali vadhwa[12] introduced to maximize or optimize de jong's function 1 in genetic algorithm using roulette wheel selection, rank selection, tournament selection and random selection. In this paper authors applied for the maximization value is in real encoding and for that they uses the different types of crossover methods for producing the better offspring's. in this paper mutation is used for the implementation of the algorithm. After the experiment and the implementation results came out and that result describes that the best selection method is the roulette wheel selection. This selection technique

is the probability based selection technique. Jason G. digalakis and Konstantinos G. margaritis[13] discusses the benchmarking functions used for performance control genetic algorithm. In this paper behavior of the generational replacement model(GRM) and the steady state replacement model(SSRM) is evaluated. In this experiments were performed to search for the optimal genetic algorithms for a given set of numerical optimization problems.

III. IMPLEMENTATION & OBSERVATION

In this section of paper Mat lab code has been developed for genetic algorithm. The test problem which is taken is the dejong function which is one of the important NP hard problems often used as a benchmark for the optimization. This function is popular in genetic algorithm literature. It is also known as "sphere model". It is continuous, convex and unimodal. The code has been developed for dejong functions i.e. sphere model function, axis parallel hyper ellipsoid function. The code uses the same crossover operator and mutation operator. The code for both also uses the same crossover probability and mutation probability. The test functions which are used for this research are given below:
This function the simplest test function in all dejong functions [4].
Function definition:

$$F1(x)=\sum_{i=1}^{n} x_i^2 \quad -5.12 \le x_i \le 5.12$$

F1(x)=sum(X(i)^2),
i=1: n, -5.12<=x(i)<=5.12.
Global minimum: F1(x)=0, x(i)=0, i=1 to n.

This optimization problem was run for two different cases of generations as 100 generation and 200 generation. The minimum values of these 100 generation and 200 generations are described in the graph below. In the dejong function we have some parameters which are used for the implementation are as:

i. **Encoding**: Real value encoding is used in the dejong function.
ii. **Selection**: The selection technique which is used in the dejong function is the roulette wheel selection.
iii. **Crossover**: The crossover operator is the arithmetic crossover operator.
iv. **Mutation**: Mutation operator is the uniform mutation.
v. **Replacement**: In this step generational update is used.
vi. **Crossover probability (Pc):** The crossover probability for the dejong function 1 is 0.7.
vii. **Mutation probability (Mc):** The mutation probability for the dejong function 1 is 0.01.
viii. **Termination criteria**: Maximum number of generations.

Table 1:

Algorithm		Dejong Function 1	
		100 Generation	200 Generation
GA	MIN	1.363	1.363

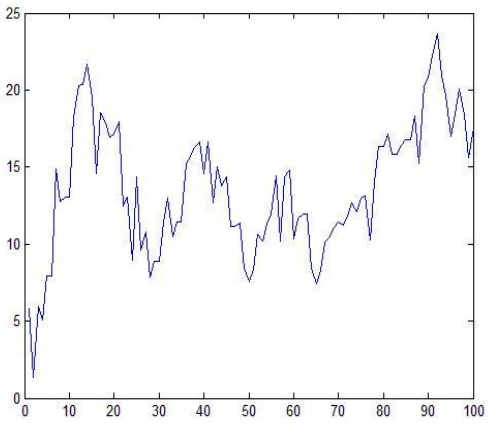

Fig 2: Dejong function 1(100 generation)

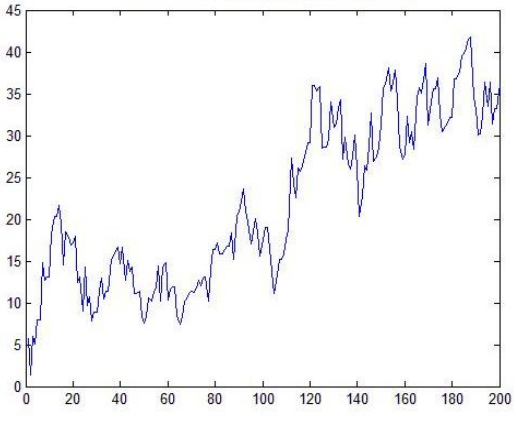

Fig 3: Dejong function 1(200 generation)

Figure 2 and figure 3 shows the minimum results of genetic algorithm in matlab. These results are of dejong function1's result.

IV. Conclusion:

This paper presents the importance of genetic algorithm which is used for optimization of dejong fuction1(sphere model). Genetic algorithm is one better function optimization method used by authors in this paper. Dejong function1 is one of the various benchmark functions that are described in genetic algorithm. Results of sphere model are shown by graphs and table which is described by using roulette wheel selection and real value encoding of genetic algorithm. Graphs are plotted for the 100 generation and the 200 generation. Minimum values of sphere model are considered by the authors.

REFERENCES

[1] J. Holland, Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, 1975.

[2] D.E. Goldberg, Genetic algorithms in search, optimization, and machine learning. Addison Wesley Longman, Inc., ISBN 0-201-15767-5, 1989.

[3] S.N. Sivanandam and S.N.Deepa, Introduction to genetic algorithm , 1st edition, 2008 Springer-Verlag Berlin Heidelberg.

[4] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms", Foundations of Genetic Algorithms, San Mateo, CA, Morgan Kaufmann, pp 69-93, 1991.

[5] D. Fogel, Evolutionary Computation, IEEE Press, 1995

[6] D. Whitley, The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best. Proc. Of the Third Int. Conf. on Genetic Algorithms, J. David Schaffer (Ed.), (Morgan Kaufmann Publishers, San Mateo, 1989) 116-121.

[7] Ebrahim Bagheri, Hossein Deldari "Dejong Function Optimization by means of a Parallel Approach to Fuzzified Genetic Algorithm" Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC'06) 0-7695-2588-1/06.

[8] Goldberg D. E.," Genetic and evolutionary algorithms come of age". Communications of the ACM, vol.37,1994.

[9] Rakesh Kumar, Sudhir Narula, Rajesh Kumar " A Population Initialization Method by Memetic Algorithm" International Journal of Advanced Research in Computer Science and Software Engineering, volume 3, issue 4, april 2013

[10] Zhang fang, Li na and Li jinhui "application research of the genetic algorithm on the intelligent test paper composition of examination database" J.Zhang(Ed.): ICAIC 2011, Part 4, CCIS 227, pp. 443-448, 2011. © Springer-verlag Berlin Heidelberg 2011

[11] .K.F.Man, K.S.Tang and S.Kwong "Genetic algorithm: concepts and application" IEEE transactions on industrial electronics, VOL-43, NO. 5, OCTOBER 1996

[12] Meera kapoor, vaishali wadhwa "Optimization of DE jong's function using genetic algorithm approach" International journal of advanced research in electronics and communication engineering(IJARECE), volume 1, issue 1, july 2012.

[13] Jason G. digalakis and Konstantinos G. margaritis "An experimental study of benchmarking functions for genetic algorithm" Intern. J. computer math. , volume 79(4), pp. 403-416, ISSN: 0020-7160 ©2002 Taylor and francis ltd. DOI: 10.1080/00207160290017942.

