Implementing Dejong Function by Random Initialization

Noopur Tyagi[1] Rakesh Kumar[2]
[1]M.Tech. Student [2]Assistant Professor
[1,2]Department of Computer Science & Engineering
[1,2]Modern Institute of Engineering and Technology, Mohri, Haryana, India

Abstract— Genetic Algorithm (GAs) are search procedures based on principles derived from the dynamics of natural population genetics. Performance of genetic algorithms mainly depends on type of genetic operators – Initialisation, Selection, Crossover, Mutation and Replacement used in it. Success of Genetic Algorithm mainly depends upon the individuals selected in the initial population and the size of population. If the individuals chosen in the initial population are poor, it will result in weaker solutions and premature convergence towards optima. There are different methods to initialize Genetic algorithms but most of the time random initialization is used. In this paper minimum value of dejong's function is observed at different generations.

Key words: Dejong Function, Initialization, Genetic algorithm (GA)

I. Introduction

Genetic algorithms [1] are search and optimization algorithms based on the principles of natural evolution, which were first introduced by john Holland in 1970First process is selection of individual for the production of next generation and second process is manipulation of the selected individual to form the next generation by crossover and mutation techniques [2]. Good initial population facilitates a GA's convergence to good solutions while poor initial population can hinder Genetic Algorithms (GA) convergence [3]. Two chromosomes are physically aligned and break over the one or more location so as to exchange their fragments. In genetic algorithms, chromosomes represented as linear strings of symbols [4]. Using a local search method within the framework of genetic algorithm can improve the exploiting ability of search without limiting its exploring ability [5]. A local search method within the genetic operator can introduce new genes than can overcome the problem of genetic drift and accelerate the search towards global optima [6]. Maaranen et al. introduced quasi-random population initialization for genetic algorithms [7]. In this paper we have done random initialization of dejong function.

A. *How do Genetic Algorithms work*

1) *Initialization:*

Genetic Algorithm is an iterative process. Generally started with initialization based on knowledge, some research has been conducted into using special techniques to produce a higher quality initial population.

Selection Mechanism: Parents can be selected by applying selection mechanism; result may vary from one calculation to another. Parent's selection is based on some fitness based selection, rank based selection and tournament based selection.

2) *Crossover:*

Crossover is a process yielding recombination of bit strings via an exchange of segments between chromosomes. One point crossover, two point crossover uniform crossover are used to yield some new strings.

3) *Mutation:*

Mutation has the effect of ensuring that all possible chromosomes are reachable. This is useful since crossover and inversion may not be able to produce new alleles if they do not appear in the initial generation.

4) *Termination Condition*

Notice that GA is stochastic and mostly there are no guarantees to reach an optimum.

Commonly used conditions for terminations are the following:

1) The maximally allowed CPU times elapses
2) The total number of fitness evaluations reaches a given limit
3) The fitness improvement remains under a threshold value

II. Related work

Rahnamayan, Shahryar et. al. have introduced a new population method for accelerating evolutionary algorithm. This research proposes a new initialization approach which employs an opposition based learning [8] to generate initial population. Sivaraj et al [9] discussed about a novel approach to improve the performance of genetic algorithm by using initialization through tournament selection, which aims at supplying more fit individuals in the beginning. Louis, sushil J. & Johnson, Judy proposed a research paper on robustness of case-initialized genetic algorithms [10]. They investigate the robustness of case initialized genetic algorithm (CIGAR) system with respect to problem indexing. This approach borrows ideas from case –based reasoning (CBR) in which old problem and solution information, stores as cases in a case –base, help solve a new problem. Indexing is a major issue for case retrieval, especially in "poorly-understood" problems. Ramsey, C.L. suggest case based Initialization approach by including strategies in initial population of genetic algorithm [11]. The success of all approach relies on expertise of user in finding best individual in search space in 1993.

III. Proposed work

In the proposed algorithm the initial population is generated by applying random selection to each individual generated randomly which controls the flow of the chromosomes in the genetic algorithm process. The aim of applying more fit individuals in the beginning phase is to increase the chance of obtaining better optimal solution. To get the best optimal solution one must initialize the chromosomes in best possible way. Random generation can find out different solutions.

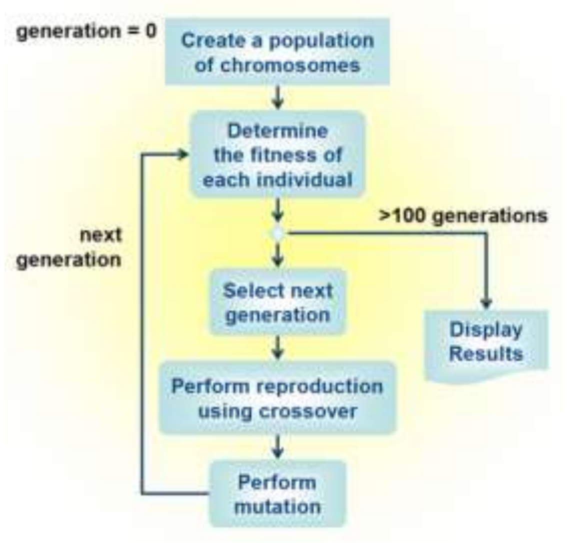
generation = 0
Create a population of chromosomes
Determine the fitness of each individual
>100 generations
next generation
Select next generation
Display Results
Perform reproduction using crossover
Perform mutation

Fig. 1:

IV. IMPLEMENTATION & OBSERVATIONS

In this section of paper Matlab code has been developed for dejong's function. The problem which the author discusses is dejong function which is one of NP hard problems often used as a benchmark for optimization. The code has been developed for dejong function like Rastrigin function. In this Implementation author show the result by graphs and table.

A. *Dejong's Function:*

Rastrigin's function is based on function 1 with the addition of cosine modulation to produce many local minima. Thus, the test function is highly multimodal. However, the location of the minima is regularly distributed.

$$f_6(x) = 10 \cdot n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i) \right) \qquad -5.12 \leq x_i \leq 5.12$$

F6(x) =10.n+sum(x (i)^2-10.cos(2.pi.x(i))), i=1:n; -5.12≤x (i) ≤ 5.12

Min value of dejong function is computed for 50 generation. This section contains the result of code runs.

The optimization problem was run for 3 different cases of generation, 50 generation, 100 generation and 150 generation. Graphs of minimum values are plotted.

1) *Parameters Used For Implementation Are-*

- size of individual
- No. of generation
- Number of individuals

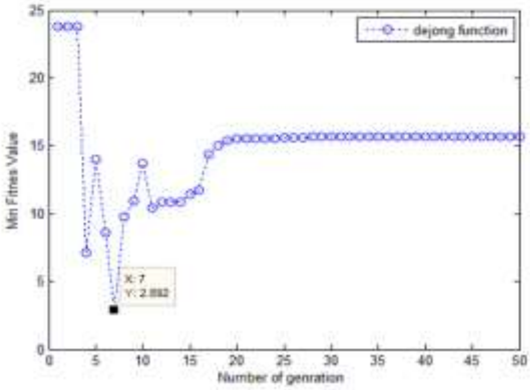

Fig. 2: Graph of 50 generation has minimum value of dejong function is at 7 generation

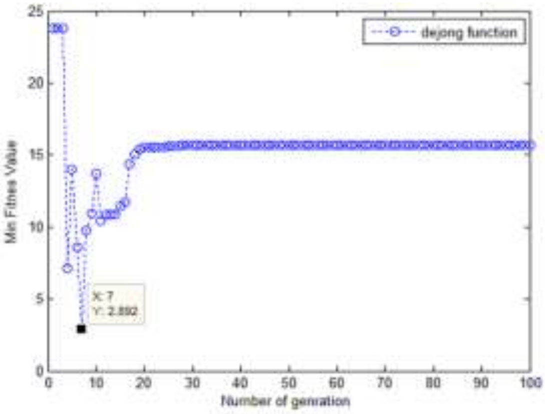

Fig. 3: Graph of 100 generation has minimum value of dejong function is at 7 generation

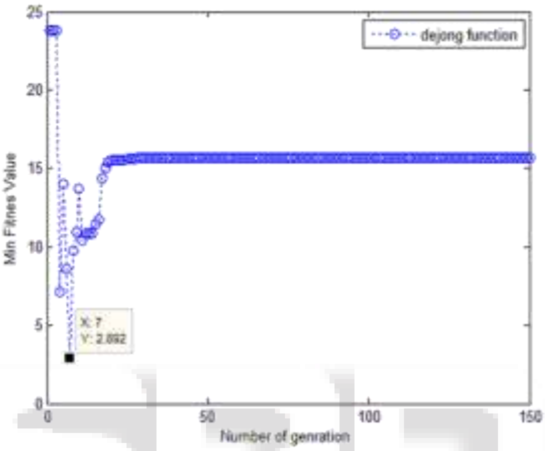

Fig. 4: Graph of 150 generation has minimum value of dejong function is at 7 generation

From these graphs we can conclude that dejong function has its minimum value at 7 generation in all these graphs.

V. CONCLUSION & FUTURE SCOPE

GA is good in exploring the search space but slow in convergence while local search techniques have been found good in exploiting the knowledge & resulting into fast convergence but sometime trapped into foot hills. But having a mix of these two, performance can be improved. It has been observed that hybridizing hill climbing into initialization step of genetic algorithm, performance improvement is there. It is further opine that other local search techniques may also be mixed in different steps of GA & can be tested on a number of different benchmark problems. In future knowledge based initialization can be done to get the result according to the problem. By knowing the approximate result in advance can reduce the complexity.

REFERENCES

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

[2] Noraini Mohd Razali, John Geraghty "A genetic algorithm performance with different selection strategies", Proceedings of the World Congress on Engineering Vol II, 2011.

[3] Raymond R. Hill (1999), "A Monte Carlo Study of genetic algorithm initial population Generation method", In Proceeding of the 1999 Winter simulation Conference, on Genetic Algorithms, Lawrence Erlbaum Associates, pp 1-8

[4] C. Ryan, Automatic re-engineering of software using genetic programming, Genetic Programming Series. Kluwer Academic Publishers, ISBN 0-7923-8653-1, 2000.

[5] W. E. Hart, "Adaptive global optimization with local search", Doctoral diss., San Diego, University of California, 1994.

[6] W. E. Hart, "Adaptive global optimization with local search", Doctoral Dissertation San Diego, University of California, 1994

[7] H. Maaranen, K. Miettinen, M.M. Makel ¨ a, A Quasi-Random Initial Population for Genetic Algorithms, in: Computers and Mathematics with ¨ Applications, vol. 47, 2004, pp. 1885–1895.

[8] Shahryar Rahnamayan , Hamid R. Tizhoosh, Magdy M.A. Salama,"A Novel Population initialization method for accelerating evolutionary algorithms", an international journal computer and mathematics with application.

[9] R. Sivaraj, T. Ravichandran and R. Devipriya " Boosting performance of genetic algorithm through selective Initialization.", European Journal Of Scientific Research, ISSN 1450-216X , vol. 68, No 1, pp 93-100, 2012.

[10] Sushil J.Louis,Judy Johnson, "Robustness of Case-Intialized Genetic algorithms".

[11] Ramsey and C.L, "Case-Based Initialization of Genetic Algorithms", Proceeding of 5th international Conference on genetic algorithm, pg-84-91, 1993.