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

Review of Genetic Algorithm: An Optimization Technique

Gopesh Joshi

USICT, Guru Gobind Singh Indraprastha University Sector 16-C,
Dwarka, Delhi, India

Abstract—In this article we are presenting an optimization technique. Although there are several optimization techniques like ant colony, simulated annealing, greedy approach and more yet genetic algorithm is meta-heuristics search optimization technique largely focus on overall optimization. And then we review various methods under genetic algorithm for optimization.

Keywords—Optimization, Genetic Algorithm, Operators

I. Introduction

Solving the real life problems are always a great challenge for researchers. Although there are many real problems to solve but more or less they are focussed on optimizing the solution. Optimizing is of various types for instance code optimization, combinatorial optimization, mathematical optimization etc. In this paper we are trying to understand how genetic algorithm provides solution for optimization problem. Genetic algorithm is a part of soft computing a branch of computer science that deals with exploring the search space and select the best solution. Genetic algorithm is a subset of evolutionary computation that is a group of algorithms. The genetic algorithm works for global optimization. Hence this paper represents review of genetic algorithm for optimization.

II. Types of Optimization Technique

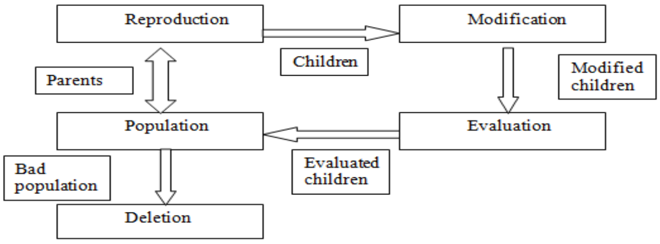
Optimization problems are set of problems that find the best result either by minimizing the effort or maximizing the productivity. There are wide ranges of application where optimization is needed for instance shortest path finding for graph for example travelling salesman problem, root finding for system of linear and nonlinear equation, design of pipeline, circuit and other. In general what we have to do is to reduce the cost or increase the profit.

Various types of Optimization techniques are:

1. Hill Climbing: Hill climbing technique is based on local search. In this method we first select a starting point and successively find next point closer to the solution. By this we will reach at goal point using greedy approach.
2. Ant Colony: Ant colony method simulates the behaviour of ants living in colony. Initially all ants start searching for food in arbitrary manner. Once a path is found by one ant that leads to food all other will follow that path.
3. Genetic Algorithm: Genetic algorithm explores search space for finding global optimization solution by iterative evolution.
4. Simulated Annealing: Simulated annealing comes from fact of lowering the energy of metal so that defects are minimum and size of crystal increase. In terms of computation, it finds the global minimum that consists of many local minima. Here objective function is depicted as internal energy of system, so we try to minimize it as low as possible.

III. Basic concept of Genetic Algorithm

Genetic Algorithm is related to biological background, in sense that it follows Darwin theory of natural evolution “the survival of the fittest”. In this theory Sir Charles Darwin implies that one individual who is fit among the population will survive and reproduce to next generation. The genetic algorithm is first proposed by John Holland in1975. In genetic algorithms first of all we will understand some terminologies to get insight of the process. Main terms are gene, chromosome, individual, population. Gene is smallest unit of information carrying capacity. Individual is a set of genes carrying information and further set of individuals is population. Here chromosome and individual are synonym and we interchange each other throughout this paper.



Process of genetic algorithm

Fig.1. Working of Genetic Algorithm

Continuous evaluation of population and deletion of bad individual is what we are doing in genetic algorithm. Basic flow of genetic algorithm is given in figure1. In figure 1 flow start from the generation of population. In next step parents are selected from the pool and reproduce to give children, this is known as crossover. Then these newly generated children are modified by the process known as mutation. They are evaluated against fitness criterion. Those children are selected that have high fitness function value and become a part of population others that do not pass the fitness test are bad population and are deleted.

IV. GENETIC ALGORITHM

In this section we are listing the steps involved in executing the genetic algorithm. There are 8 steps for the algorithm. Genetic Algorithm works as follows:

1. Encoding: First step towards solving problem using genetic algorithm is the encoding of solution. In this stage, phenotype is mapped to genotype. It means data is represented in genes.
2. Initialization: Take input parameter like population size, crossover probability, mutation probability, and number of generation for iteration.
3. Evaluation: Find the fitness value of each individual through fitness function.
4. Selection: Retainment of best fit individual and elimination of bad population is the task of selection.
5. Crossover: Recombination is another name for crossover of two selected parents from the pool of population.
6. Mutation: Mutation is adding new features form outside and permutation of gene within a chromosome.
7. Iterate steps 3 to 6 until terminate the loop.
8. Decoding: Decode the final solution back to phenotype.

Details of main operators are discussed in next section.

V. REVIEW OF GENETIC OPERATORS

Core functioning of any variants of genetic algorithm is consists of operators that will iterate to a number of times till best solution reach. The operators are discussed below:

A. Evaluation

This is the most important criterion that is used by selection operator. In population of size n there are n individuals. Each will have their own fitness value. The first task is the design of fitness function. One more function is objective function that directly takes data from genes and output the fitness value. For example in case of travelling salesman problem each gene represents a city and sum of genes is tour cost. This is a scenario of cost minimization so fitness function in this case is reciprocal of objective function. Similar in another scenario of maximizing the flow from one point to other like in traffic utilization fitness function will be identical to objective function. After this, next step is selection of best chromosome.

B. Selection

Selection is a process of retaining the best fit individual and eliminating rest one. There are two general methods for doing the selection. These are fitness proportionate and tournament selection method. These are discussed below:

1) Fitness proportionate selection

This technique is based on cumulative distribution of the individuals according their fitness value. One easiest way to select under this class is roulette wheel selection. In this method an imaginary wheel contains slots of varying size. Each slot is shows a individual cumulative statistics. One who gets maximum size will get selected.

2) Tournament selection

As the name suggest, population set is divided into tournament. A number of tournaments are played in this method. And the winner of each tournament get selection for the next generation and rest are eliminated.

C. Crossover

Crossover resembles the concept of reproduction of male and female parent. Same thing we do in crossover. We pick two randomly selected parents from the population and reproduce them to give birth to new children. Here one more point is the role of crossover rate. It is a probability that decides the child chromosome. This rate should be kept high usually in range of 0.6 to 0.8, so that replication of gene can be avoided. Main concern of the crossover operator is to put best features from both the parent into their children.

There are three main crossover techniques:

1) One-point crossover: It is simplest among all method. A random position is selected for crossover. In one point crossover, parents copy their features in their children (see figure 2).

PARENT 1	AAA | AAAA
PARENT 2	BBB | BBBB
CHILD 1	AAA | BBBB
CHILD2	BBB | AAAA

Fig.2. Single point crossover

2) Two-point crossover: In this method we select two random points for crossover such that child 1 adopt middle portion of parent 2 and rest from parent 1, similar for child 2 (see figure 3).

PARENT 1	AA | AAA | AA
PARENT 2	BB | BBB | BB
CHILD 1	AA | BBB | AA
CHILD2	BB | AAA | BB

Fig.3. Two point crossover

3) Uniform crossover: In this crossover is done at multiple sites but these are uniformly spread across the chromosome that is either even or odd ordering (see figure 4).

PARENT 1	A | A | A | A | A | A | A
PARENT 2	B | B | B | B | B | B | B
CHILD 1	A | B | A | B | A | B | A
CHILD 2	B | A | B | A | B | A | B

Fig.4. Uniform crossover

D. Mutation

Mutation operator modifies the solution obtained after doing the crossover. Chance of changes in the solution is very low. So the mutation probability should be kept as low as possible usually in range of 0.01 to 0.05. Without mutation same gene will inherited to all successive generation results in solution far from optimize one. There are several ways to implement mutation. These are as follows:

1) Insert Mutation: Insert mutation is one of the major types of mutation. In this method, random changes occur at gene. If the individual is at bit level, then it is called bit-level insertion. Advantage of this method is the new features and information coming from outside the population. Below figure 5 shows insert mutation.

BEFORE MUTATION	P Q R S T U
AFTER MUTATION	P Q **A** S T **B**

Fig.5. Insert mutation

2) Swap Mutation: Gene at random locations is replaced with each other so that a new combination will generate. Here the most of the ordering remain same expect swapped one (see figure 6).

BEFORE MUTATION	P Q R S T U
AFTER MUTATION	P Q **T** S **R** U

Fig.6. Swap mutation

3) Jumble Mutation: In this method, rearrangement are done in such a way that a new outcome will result. Benefit of this method is that same ordering will not inherit to next generation (see figure 7).

BEFORE MUTATION	P Q R S T U
AFTER MUTATION	**T S U Q R P**

Fig.7. Jumble mutation

4) Inverse Mutation: It is a new approach where allel value from left are moved to right. It is also similar to cyclic mutation where after each cycle allel values are shifted to right one position (see figure 8).

BEFORE MUTATION	P Q R S T U
AFTER MUTATION	**U T S R Q P**

Fig.8. Inverse mutation

VI. CONCLUSIONS & FUTURE WORK

We have successfully reviewed the genetic algorithm for the optimization problem. It is clearly understood from this paper how genetic algorithm originated, the basic concepts and the working of genetic algorithm. Genetic algorithm is one of the well known optimization technique. Other technique like hill climbing, simulated annealing and ant colony optimization are explained. In addition to this we have reviewed various ways to implement genetic operators. Also working of each operator is explained. In future we will design modified version of genetic algorithm and run, test it for various optimization problem. Also we will compare genetic algorithm with other optimization technique.

ACKNOWLEDGMENT

I would like to thanks to all the faculties of USICT, GGSIPU, Dwarka, Delhi for their support to write this paper. Also I would to thank the university resource center for providing valuable assistance and materials.

REFERENCES

[1] A.H. Wright, “Genetic Algorithms for real parameter optimization: Foundations of Genetic Algorithms” J. E. Rawlins, Ed. San Mateo, **CA:** Morgan Kaufmdnn, 1991, pp. 205-218.

[2] K.F. Mann, Member, IEEE, K.S. Tang, S. Kwong, Member, IEEE, “Genetic Algorithms: Concepts and Applications”, pp 519-534.

[3] C.A. Coello Coello, G.B. Lamont and D.A.V. Veldhuizen, “Evolutionary Algorithms for Solving Multi-objective Problems” 2nd ed. Springer.

[4] Battiti Roberto, Mauro Brunato, Franco Mascia (2008), “Reactive Search and Intelligent Optimization”, Springer Verlag. ISBN 978-0-387-09623-0.

[5] Zbigniew Michalewicz, “Genetic Algorithms+DataStructures=Evolution Programs” 3rd ed. Springer.

[6] Zbigniew Michalewicz, “Genetic Algorithms, Numerical Optimization and Constraitns”.

[7] R.V. Rao, V.J. Savsani, “Mechanical Design Optimization using Advanced Optimization Techniques” verlag. (2012)Springer.