Paper from Proceeding of the National Conference on Innovative Developments in Science, Technology & Management (NCIDSTM-2015) Organized by Ganga Technical Campus, Soldha, Bahadurgarh, Haryana (India) March 1st 2015 Published by International Journal of Engineering Sciences Paradigms and Researches (IJESPR) with ISSN (Online): 2319-6564, Impact Factor: 2.20 and Website: www.ijesonline.com

A Genetic Approach for Optimization

Sudesh Lathar[1], Savita Narang[2], Monika Chawla[3]

[1]Associate Profesor, Computer Science Department, Rohtak
[2]Assistant Lecturer, Computer Science Department, Rohtak
[3]Assistant Lecturer, Computer Science Department, Rohtak

Abstract

An optimization problem involves finding values of the variables that minimize or maximize the objective function while satisfying certain constraints. Genetic algorithms are the adaptive heuristic search algorithms based on evolutionary idea of natural selection and genetics. GA's are intelligent exploitation of random search used in optimization problems. GA although randomized, exploits historical information to direct the search into the region of better performance with in the search space.

In this paper genetic approach for solving the optimization problem will be presented. As Genetic algorithms are good to search a huge search spaces and navigating them looking for optimal solutions one might not otherwise find in a life time.

Keywords: *Optimization, search space, genetic algorithm*

1. Introduction

For three decades many mathematical programming methods have been developed to solve the optimization problems but until now there has not been a single totally efficient and robust method to cover all the optimization problems that we came across with in many engineering fields.

Optimization is a process that finds a better optimal solution for a problem. The optimization problems are centered on three factors

1. An objective function which is to be minimized or maximized for example: - In multiprocessor scheduling problems we need to minimize the total time needed to complete the task (make span).
2. A set of unknown variables that affect the objective function for example: - Time spent in the multiprocessor scheduling problems.
3. A set of constraints that allow us to take a certain value but exclude the other values for example:- In multiprocessor scheduling problem we are to complete all the tasks on a number of processors we are to select a schedule by which we are getting minimum time for completing all the tasks.

Optimization methods can be classified into three main categories; calculus based methods, numerical methods and random method. Calculus based methods are most popular and can be further classified as direct and indirect methods. In Direct methods the search is done to find a local maximum moving on a function over the relative local gradient direction. But in the indirect methods we try to find the local ends by solving the nonlinear equation, both methods are improved and studied but these methods are not considered as Robust methods because they have local focus as they try to find the maximum in the analyzed point neighborhoods, secondly they depends on the existence of their derivative.

But the real world has a large number of noisy spaces and discontinuities so the above methods are unsuitable for real world problems. A large number of schemes have been applied in many forms these are direct inside the finite search space or a discrete infinite search space where algorithm can locate the object function values in each space point one at a time.

This type of algorithm is possible to solve with above approaches when number of possibilities are very small but these approaches are inefficient not robust in case of big dimensional spaces where it become very difficult to find the optimal values because of limitations of the calculus based techniques and numerical method the random methods have become popular.

The random search methods are based on evolutionary algorithms then evolutionary techniques are parallel and globally robust optimization principle of natural selection of Darwin and the genetic theory of natural selection of R.A.Fischer . Genetic algorithm is the technique that successfully used in the optimization problems. The genetic algorithm is a search procedure that uses random selection for optimization of a function by means of the parameters space coding.

1.1. Historical Review

The term genetic algorithm was firstly given by John Holland (1) in his book "Adaptation in natural and artificial systems" (1975)

In 1960 Germany Ingo Rechenberg and Hans Paul Schwefel developed the idea of Evolutions strategic (also called Evolution Strategy). Bremermann Fogel and other in USA implemented the evolutionary programming. In 1960 they used the mutation and selection operators that were taken from Darwin theory of evolution.

In 1985 First conference on the subject was held and David Goldberg (1985) applied the genetic algorithm for gas pipeline optimization.

In 1989 the published a book "Genetic Algorithm in search optimization and Machine learning. After this the GA start growing rapidly. De Jon Firstly showed his intent in optimization.

All evolution algorithm till this time were based on mutation operator and they tried to develop the hill-climbing method but Holland's genetic algorithm use the idea of recombination same research at this time tried to solve the optimization problem using technique which provide good solutions. These techniques are known as heuristics and neighborhood search.

Lin (1965) found an excellent solution for the traveling salesman problem he investigate the problem by breaking any 3 links of a tour and then reconnecting. He found that 3 optimal solution got in this way were of excellent quality. They were very close to the global optimum

Lin's ideas were rediscovered a Multi parent recombination and convenience operation.

Robert and Flores (9) used a similar approach as Lin's for the TSP. While Nugent et.al. applied the same idea for quadratic alignment problem.
GA is very effective and efficient approach for searching problems for these reasons the genetic algorithm has proven successful for robust searches in complex spaces.

1.2. Why Genetic Algorithm

1. Concept of genetic algorithm is easy to understand.
2. It is better than conventional tools available. It is more robust.
3. Unlike older system for searching array set of data, the genetic algorithms do not break easily even if the inputs changed slightly, or in the presence of reasonable noise.
4. While performing search in large state space or multi modal state space or n-dimensional surface a genetic algorithm offers significant benefits over many other typical search optimization techniques like linear programming, heuristic, depth first, breadth first.
5. Genetic algorithm support multi objective optimization.
6. It always result in an answer which becomes better and better with time.
7. Can be easily run in parallel.
8. The fitness function can be changed from iteration to iteration, which allows incorporating new data in the model if it becomes available.

Genetic algorithms are good at taking large potentially huge search spaces and navigating them looking for optimal combination of things the solutions one might not otherwise find in a life time.

1.3. Traditional algorithm v/s Genetic algorithm

1. GA don't remain trapped in a sub optimal local maximum or minimum of the target function.
2. GAs search a population of points in parallel, not a single point.
3. GAs do not require derivative information or other auxiliary knowledge only the objective function and corresponding fitness levels influence the directions of search.
4. GAs use probabilistic transition rules, not deterministic ones.
5. GAs work on an encoding of the parameter set rather than the parameter set
itself (except in where real-valued individuals are used).

2. Genetic Algorithms

The GA is a stochastic global search method that mimics the metaphor of natural biological evolution. GAs operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a

new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

2.1 WORKING PRINCIPLE OF A SIMPLE GA

1. Begin
2. INITIALIZE population with random candidate solutions;
3. EVALUAE each candidate:
4. REPEAT UNTIL (TERMINATION CONDITION is satisfied)
5. DO
 a. SELECT parents;
 b.RECOMBINE pairs of parents;
 c. MUTATE the resulting offspring;
 d. EVALUATE new candidate;
 e. SELECT individuals for the next generation;
6. DO
7. END

2.2Genetic Operators

When GA proceeds, both the search direction to optimal solution and the search speed should be considered as an important factor, in order to keep a balance between exploration and exploitation in search space. In general, the exploitation of the accumulated information resulting from GA search is done by the selection mechanism, while the exploration to new regions of the search space is accounted for by genetic operators.
The genetic operators mimic the process of heredity of genes to create new offspring at each generation. There are three common genetic operators: crossover, mutation and selection.

2.2.1 Crossover

Crossover may be a main genetic operator. Crossover operator selects two chromosomes at a time and generates offspring by combining both individual's features. A simple way to achieve crossover is to choose a random cut-point and generate the offspring by combining the segment of one parent to the left of the cut-point with the segment of the other parent to the right of the cut-point.

Figure below shows an example of one-cut point crossover operation.

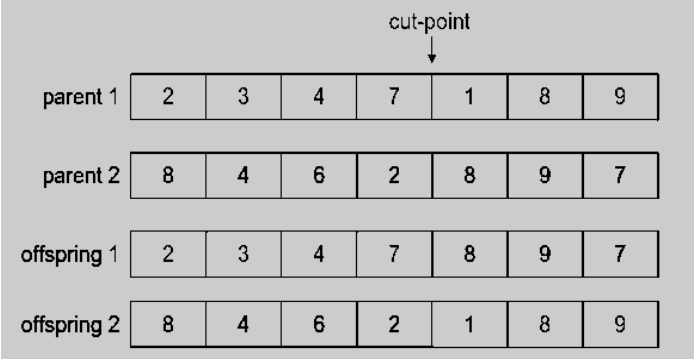

Fig1: one-cut point crossover operation

In Figure above the right side genes of the selecting point in the two individuals, are exchanged each other. After the crossover operation, the new individuals, are produced. Up to now, various types of crossover operators have been developed, which can be roughly put into four classes: conventional, arithmetical, direction-based and stochastic.

2.2.2) Mutation

Mutation is usually used as a background operator, which produces spontaneous random changes in various individuals. A simple way to achieve mutation would be to alter one or more genes. Figure below shows an example of mutation operation.

Fig2 :mutation operation

In the adjacent Fig the value of the gene at the selection point is changed to 0. In general, mutation operator serves the crucial role of either replacing the genes lost from the population during the selection process so that they can be tried in a new context or providing the genes that were not present in the initial population. The mutation operator arbitrarily alters one or more components of a selected structure, and this increases the variability of the population.

2.2.3) Selection

New offspring for next generation of GA are formed by selection method according to their fitness values. The selection process is known as evolution operator, directing a GA search toward promising region in the search space. There have been presented various types of selection strategies such as elitist selection strategy, stochastic tournament selection strategy, and so on, in order to improve population to more evolved population (with better fitness). These selection strategies provide the driving force in a GA and the selection pressure is critical in it. At one extreme, the search will terminate prematurely; while at the other extreme progress will be slower than necessary. It seems that there are two important issues in the evolution process of genetic search: population diversity and selective pressure. These factors are strongly related: an increase in the selective pressure decreases the diversity of the population, and vice versa.

2.3 Fitness Evaluation

Fitness evaluation is to check the solution value of the objective function subjected to constraints. In general, the objective function provides the mechanism evaluating each individual. However, its range of values varies from problem to problem. To maintain uniformity over various problem domains, the fitness function could be normalized to a range of 0 to 1. The normalized value of the objective function is the fitness of the individual, and the selection mechanism uses to evaluate the individuals of the population.

2.4 Termination of the GA

Because the GA is a stochastic search method, it is difficult to formally specify convergence criteria. As the fitness of a population may remain static for a number of generations before a superior individual is found, the application conventional termination criteria becomes problematic. A common practice is to terminate the GA after a prespecified number of generations and then test the quality of the best members of the population against the problem definition. If no acceptable solutions are found, the GA may be restarted or a fresh search initiated.

3. Conclusion

Genetic algorithms do not necessarily guarantee that the global optimum solution will be reached, although experience indicates that they will give near optimal solution after a reasonable number of evaluations.
These can be used to find the solutions when the search space is very large. It can also be used to solve the problems in which the solution can not be calculated by traditional algorithms in a long period of time.

References

[1] Allen B. Tucker (Jr.), The Computer Science and Engineering Handbook, CRC Press, USA, pp. 557-571, 1997.
[2] C. Darwin, The Origin of the Species, Cambridge, Ma., Harvard University Press, 1967.
[3] D.B. Fogel (1998) Evolutionary Computation: The Fossil Record, IEEE Press, Piscataway, NJ.
[4] D.E. Goldberg (1989) Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Massachusetts.
[5] D.E. Goldberg, Genetic Algorithms, in Search, Optimization & Machine Learning. Addison Wesley, 1997.
[6] H.P. Schwefel, Numerical Optimization of Computer Models, John Wiley & Sons, New York, 1981.
[7] J.H. Holland, Adaptive in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press, 1975.
[8] J.H. Holland (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan; re-issued by MIT Press (1992).
[9] J.J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithms". IEEE Trans on Systems, Man and Cybernetics. Vol. 16, No.1, pp 122-128, 1986.
[10] K.A. De Jong (1975) An analysis of the behavior of a class of genetic adaptive systems, Doctoral dissertation, University of Michigan, Ann Arbor, Michigan.
[11] R.A. Fisher, The Genetical Theory of Natural Selection. Clarendon press, Oxford 1930.
[12] R.L. Haupt, "An Introduction to Genetic Algorithms for Electromagnetics", IEEE Transactions on Antennas and Propagation Magazine, Vol. 37, No 2, pp. 7-11, 1995.
[13] R.L. Haupt "Thinned Arrays Using Genetics Algorithms", IEEE Transactions on Antennas and Propagation, Vol. 42, pp. 993-999, 1994.
[14] S. Lin (1965) Computer solutions of the traveling salesman problem. Bell Systems Tech. J., 44, 2245–2269.

[15] S.M. Roberts and B. Flores (1966) An engineering approach to the travelling salesman problem. Man. Sci.,13, 269–288.