Concept of "Genetic Algorithm" An Evolutionary Optimization Technique

Smt. Manjulata Soni[1], Dr. G. K. Agrawal [2], Dr. (Miss) S. L. Sinha[3]

[1]Ph.D. Scholar Deptt of Mechanical Engineering , G. E. C. Bilaspur, India
[2]Department of Mechanical Engineering, Government Engineering College Bilaspur, India
[3]Department of Mechanical Engineering, National Institute of Technology Raipur, India

ABSTRACT

Genetic Algorithm (GA) is a global optimization algorithm derived from evolution and natural selection. and is a powerful tool for solving complex problems .Evolutionary computation (EC) techniques abstract these evolutionary principles into algorithms that may be used to search for optimal solutions to a problem. In a Traditional search algorithm, a number of possible solutions to a problem are available and the task is to find the best solution possible in a fixed amount of time. For a search space with only a small number of possible solutions, all the solutions can be examined in a reasonable amount of time and the optimal one found. This exhaustive search, however, quickly becomes impractical as the search space grows in size.

Keywords: Population, Encoding, Decoding, Cross Over, Mutation, Selection

INTRODUCTION

Many graphical and analytical techniques were initially used for dimensional synthesis. The use of graphical methods is limited to simple path generation problems. Analytical methods require algebraic expressions and can become cumbersome for complex problems. Some of the short comings of graphical and analytical method are overcome by numerical optimization methods which may either be direct search or gradient based. In direct search method a small increment is given to and initial feasible design and different increments are tried until a better solution is obtained. Gradient based optimization methods require gradient information which is generally obtained using finite difference approximation techniques. These and other similar numerical optimization methods have been found to give reasonably accurate results for simple problems. In the case of highly non-linear optimization problems the number the number of iteration required to find the optimum step size and the descent direction can be quite large thus consuming significant computation time. In particular applications of these method are often characterized by excessive function evaluation, premature termination of the algorithm and inaccurate solutions.

The exact gradient method solves these problems by determining the exact gradient of the objective function and constrained functions instead of using finite difference approximation. However it is still a gradient based search method and as such requires derivative knowledge. Moreover all the optimization techniques described so far aim for and result in a single optimal point though there may exist other similar of better design in the feasible design domain.

An alterative approach for dimensional synthesis is to use probabilistic methods as opposed to the deterministic methods discussed above. These methods are called hill climbing" methods as they accept a relatively bad result with a given probability. Thus these algorithms are capable of breaking out of the local minima and finally converging to the absolute optimum point. Simulated annealing and genetic algorithms are some examples of hill climbing algorithms. Simulated annealing. Like the numerical optimization methods, gives only a single optimum design point. Instead of a simple design, if a set of non-inferior design is presented, the designer has the option of selecting the one that best suits the problem. Genetic algorithm search from a population and offer a set of equally good solutions to the designer. Unlace the gradient methods, no gradient information is necessary which makes the implementation of genetic algorithm even more appealing from the point of view of user effort.

A major drawback in applying genetic algorithm in its "as is" form to complex non linear constrained optimization problems such as optimal solution is the computational burden involved. Conventionally genetic algorithm operate on binary string. Whose recessing is time consuming and computationally expensive. If the genetic algorithm can be designed to operate on real numbers instead of binary strings, it will be more efficient in terms of time and computation.

Genetic algorithms (popularly known as GAs) have now gained immense popularity in real-world engineering search and optimization problems all over the world. Due to globalization of our economy, Indian industries are now facing design challenges not only from their national counterparts but also from the international market. To survive in the steep competition, they can no longer afford to adopt just any feasible solution obtained usually by trial-and-error means, they have to choose an *optimal* solution which is best (from cost, performance, safety, etc.) compared to any other solution.

Concept of Genetic Algorithm:

Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. The genetic algorithm is a probabilistic technique that uses a population of design rather that a single design at a time. It is analogous to natural selection in the evolution of living organisms in that the fittest members in the population have a better chance to survive, reproduce and thus transfer their genetic material to the successive generations. The initial population is formed by a set of randomly generated members.

Each generation consists of members whose constituents are the individual design variables that characterize a design and these are embedded in a binary string. Each member is evaluated using the objective function and is assigned a fitness value, which is an indication of the performance of the other members in the population. A biased selection depending on the fitness value decides which members are to be used for producing the next generation. The selected strings (a new set of artificial creatures) are the parents for the next generation which evolves from the use of two genetic operators namely crossover and mutation.

These operators give a random displacement to the parent population and generate a new population of designs. The cross over operator takes two parents strings, splits them at a random location, and swaps the sub-string so formed. A probability of mutation operator invents a bit in the string depending on the probability of mutation . The new strings formed are evaluated and the iteration continues until a maximum number of generations has been reached or until a user defined termination criteria has been met.

Figure shows the sequence of steps in a basic genetic algorithm. The control parameters that have to be initially specified are the population size the crossover and the mutation probabilities the maximum number of generations and the termination criteria. There are many alterations that may have to be introduced into the basic genetic algorithm described above, depending upon the problem. For example the whole population can be used for reproduction (generational replacement), only a part of the population can under go reproduction (steady state replacement) or the best member in the population can be passed on to the next generation without any changes (elitist selection). The crossover operator can occur at a single point or at more then one point (multi point crossover).

The fitness function may be based on the objective function, value or on the position of the member in the population (linear normalization). The control parameters of the genetic algorithm any be fixed at a particular value of can be made to vary as the genetic algorithm progresses. Real numbers may be used for representing the design variable or they may be mapped to a binary string. There is no single variation that out performs the other for all types of problems and therefore the designer has to decide as to which variation to implement.

Genetic Algorithm Differs From Conventional Optimization Techniques In Following Ways:

1. GAs operate with coded versions of the problem parameters rather than parameters themselves i.e., GA works with the coding of solution set and not with the solution itself.
2. Almost all conventional optimization techniques search from a single point but GAs always operate on a whole population of points(strings) i.e., GA uses population of solutions rather than a single solution for searching. This plays a major role to the robustness of genetic algorithms. It improves the chance of reaching the global optimum and also helps in avoiding local stationary point.
3. GA uses fitness function for evaluation rather than derivatives. As a result, theycan be applied to any kind of continuous or discrete optimization problem. The key point to be performed here is to identify and specify a meaningful decoding function.
4. GAs use probabilistic transition operates while conventional methods for continuous optimization apply deterministic transition

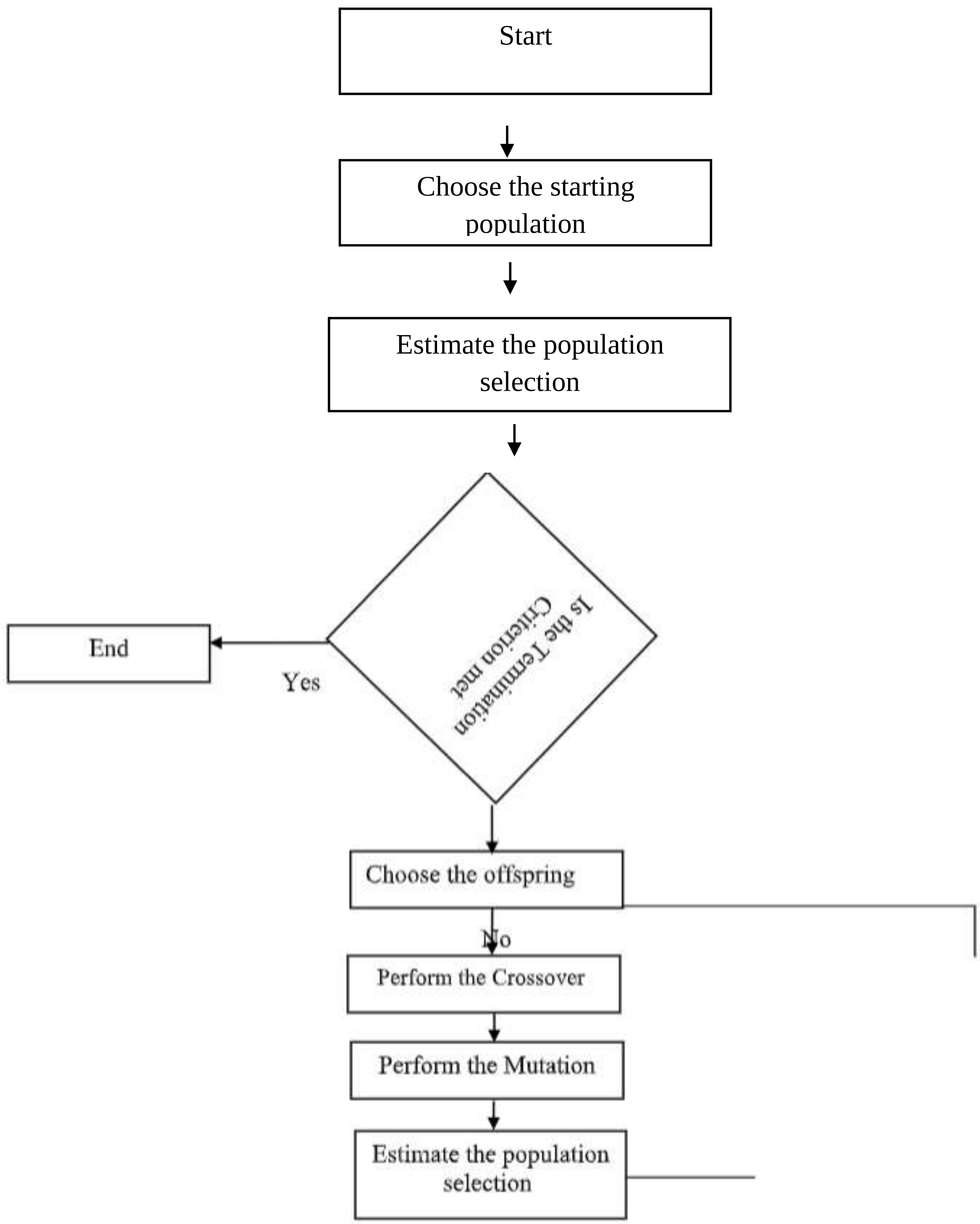

Figure 1

Application Of Genetic Algorithm To Optimization Problem:
The genetic algorithm procedure described can not be applied directly to solving equations, which is a highly non-linear constrained optimization problem. Certain modifications are necessary in the basic genetic algorithm for the treatment of constraints and to avoid premature convergence of the solution.

The initial population for the genetic algorithm is randomly generated such that all constraints are satisfied. The design variables associated with each of the design can be represented either as binary strings or as real numbers. The advantage of the binary representation is that conventional crossover and mutation operations become very simple. The disadvantage is that the binary numbers have to be converted to real number when the design is to be evaluated. Moreover, binary operation consume a lot of computer memory as well as computation time. On the other hand if crossover and mutation operators can be formulated to act on real numbers, the binary representation can be eliminated.

Each of the randomly generated designs is evaluate by the objective function and the constraints are checked separately. Since objective problems are minimization problems, the objective function value can not be used as a measure of fitness of the design. These have to be transformed so that a better design will have a higher fitness value.

In constrained optimization the design space gets restricted and therefore, genetic algorithms cannot be directly applied. Davis, 1991 has outlined the process of formations of genetic algorithms in each of the cases.

The selection of members for the next generation is done by the Roulette wheel selection method (Goldberg) wherein members with high fitness value have a higher probability of getting replicated in the next generation. The selected members are then operated on by one of the two genetic operators, crossover or mutation, both of which have a user defined probability of occurrence. The design resulting from the crossover or mutation operations are evaluated and subjected to the selection procedure. This iteration process continues. Until maximum number of generation has been reached.

CONCLUSION

Genetic algorithms are computerized search and optimization methods that work very similar to the principles of natural evolution. Based on Darwin's survival-of-the fittest principles, GA's intelligent search procedure finds the best and fittest design solutions, which are otherwise difficult to find using other techniques. GAs are attractive in engineering design and applications because they are easy to use and they are likely to find the globally best design or solution, which is superior to any other design or solution. Some of the GA applications include mechanical component design, structure design, process design, planning, job shop scheduling, VLSI design, control systems, electrical power systems, pattern recognition, classification problems, protein folding, neural network design, operations research, and machine learning. GAs are also suitable for multi-objective optimal design problems, involving multiple objectives.

REFERENCES

[1]. Arum kunjur, Sunder Krishnamurthy (A.P.) Dept. of Mech. Engg. University of Massachusetts "Genetic Algorithms in Mechanism Synthesis.
[2]. Davis L. 1991, "Handbook of Genetic Algorithms, "Van No Strand Reinhold, New York.
[3]. Goldberg D.F. 1989, "Genetic Algorithms in Search Optimization Machine Learning" Addison-Wesley Publications.
[4]. F. Ferudensrein, Approoximate synthesis of four-bar linkage', Trans. A.S.M.E. 77853-861 (1968)
[5]. Fang E.W. 1994 "Simultaneous Type and Dimensional Synthesis of Mechanisms by Genetic Algorithms, "Mechanism Synthesis and Analysis-DE-Vol.70.
[6]. H. chem.-Yen,'A general method for optimum design mechanisms', J. Mech. 1,301-313(1966).
[7]. J. Denavit and R.S. Hartenberg, 'Approximate synthesis of spatial linkages', J. Appl. Mech. Trans. A.S.M.E. ser. B, 27 (I), 201-206 (1960).
[8]. J. Toomas', The synthesis of mechanisms as a nonlinear programming problem', J. Mech. 3,119-130 (1968).
[9]. R.L. Fox and k.D. Wilmert,' Optimum design of curve generation linkages with inequality constrains', J.Eng. Ind, Trans A.S.M.E., ser. B,89(I) 144-152 (1967)
[10]. Rao Singiresu S. "Engineering Optimization Theory and Practice" Third Enlarged Edition, New Age International (p) Limited, New Delhi.
[11]. Rosen, J.B. 1960 "The Gradient Projection Method for Nonlinear Programming Part I, Linear Constraints, "SIAM journal of Applied Mathematics Vol.8, PP 181-217.
[12]. Srinvas M and Patnaik L.M. 1994 , "Genetic Algorithms: A Survey, "in Computer June 1994.