Appendix VIII

Genetic Algorithm

The optimization problems reveal the fact that the formulation of engineering design problems could differ from problem to problem. Certain problems involve linear terms for constraints and objective function but certain other problems involve nonlinear terms for one or both of them. In some problems, the terms are not explicit functions of the design variables. Unfortunately, a single optimization algorithm does not exist which will work in all optimization problems equally efficiently. Some algorithms perform better on one problem, but may perform poorly on other problems. That is why, the optimization literature contains a large number of algorithms, each suitable to solve a particular type of problem. The choice of a suitable algorithm for an optimization problem is, to a large extent, dependent on the user's experience in solving similar problems. Since the optimization algorithms involve repetitive application of algorithmic steps, they need to be used with the help of a computer. The optimization algorithms are classified into a number of groups, which are now briefly discussed.

- ***Single-variable optimization algorithms***

These algorithms provide a good understanding of the properties of the minimum and maximum points in a function and how optimization algorithms work iteratively to find the optimum point in a problem. The algorithms are classified into two categories—direct methods and gradient-based methods. Direct methods do not use any derivative information of the objective function; only objective function values are used to guide the search process. However, gradient-based methods use derivative information (first and/or second-order) to guide the search process. Although engineering optimization problems usually contain more than one design variable, single-variable optimization algorithms are mainly used as unidirectional search methods in multivariable optimization algorithms.

- ***Multi-variable optimization algorithms***

These algorithms demonstrate how the search for the optimum point progresses in multiple dimensions. Depending on whether the gradient information is used or not used, these algorithms are also classified into direct and gradient-based techniques.

- ***Constrained optimization algorithms***

These algorithms use the single-variable and multivariable optimization algorithms repeatedly and simultaneously maintain the search effort inside the feasible search region. These algorithms are mostly used in engineering optimization problems.

- ***Specialized optimization algorithms***

There exist a number of structured algorithms, which are ideal for only a certain class of optimization problems. Two of these algorithms integer programming and geometric programming are often used in engineering design problem. Integer programming methods can solve optimization problems with integer design variables. Geometric programming methods solve optimization problems with objective functions and constraints written in a special form.

There exist quite a few variations of each of the above algorithms. These algorithms are being used in engineering design problems since sixties. Because of their existence and use for quite some years, these algorithms are termed as traditional optimization algorithms.

- ***Nontraditional optimization algorithms***

There exist a number of other search and optimization algorithms which are comparatively new and are becoming popular in engineering design optimization problems in the recent years. Two such algorithms are genetic algorithms and simulated annealing.

Investigators have put together about 34 different optimization algorithms. Over the years, investigators and practitioners have modified these algorithms to suit their problems and to increase the efficiency of the algorithms. However, there exist a few other optimization algorithms like stochastic programming methods and dynamic programming method which are very different than the above algorithms.

Many engineering optimization problems contain multiple optimum solutions, among which one or more may be the absolute minimum or maximum solutions. These absolute optimum solutions are known as global optimal solutions and other optimum solutions are known as local optimum solutions. Ideally, we are interested in the global optimal solutions because they correspond to the absolute optimum objective function value. Unfortunately, none of the traditional algorithms are guaranteed to find the global optimal solution, but genetic algorithms and simulated annealing algorithm are found to have a better global perspective than the traditional methods. Moreover, when an optimal design problem contains multiple global solutions, designers are not only interested in finding just one global optimum solution, but as many as possible for three main reasons. Firstly, a design suitable in one situation may not be valid in another situation. Secondly, it is also not possible to include all aspects of the design in the optimization problem formulation. Thus, there always remains some uncertainty about the obtained optimal solution. Thirdly, designers may not be interested in finding the absolute global solution, instead may be interested in a solution which corresponds to a marginally inferior objective function value but is more amenable to fabrication and operation. Thus, it is always prudent to know about other equally good solutions for later use. However, if the traditional methods are used to find multiple optimal solutions, they need to be applied a number of times, each time starting from a different initial solution and hoping to achieve a different optimal solution each time. Genetic algorithms allow an easier way to find multiple optimal solutions simultaneously in a single simulation.

Another class of optimization problems deals with simultaneous optimization of multiple objective functions. In formulating an optimal design problem, designers are often faced with a number of objective functions. Multi objective optimization problems give rise to a set of optimal solutions known as *Pareto-optimal* solutions all of which are equally important as far as all objectives are concerned. Thus, the aim in these problems is to find as many Pareto-optimal solutions as possible. Because of the complexity involved in the multi objective optimization algorithms, designers usually choose to consider only one objective and formulate other objectives as constraints. Genetic algorithms handle multiple objectives and help find multiple Pareto-optimal solutions simultaneously.

In many engineering design problems, a good solution is usually known either from the previous studies or from experience. After formulating the optimal problem and applying the optimization algorithm if a better solution is obtained, the new solution becomes the current best solution. The optimality of the obtained solution is usually confirmed by applying the optimization algorithms a number of times from different initial solutions.

Two nontraditional search and optimization methods have gained popularity in engineering optimization problems not because they are new but because they are found to be potential search and optimization algorithms for complex engineering optimization problems. *Genetic algorithms* (GAs) mimic the principles of natural genetics and natural selection to constitute search and optimization procedures, *Simulated annealing* mimics the cooling phenomenon of molten metals to constitute a search procedure. Since both these algorithms are abstractions from a natural phenomenon, they are very different search methods than other procedures.

Genetic algorithms are computerized search and optimization algorithms based on the mechanics of natural genetics and natural selection. GAs is radically different from most of the traditional optimization methods. GAs works with a string-coding of variables instead of the variables itself. The advantage of working with coding of variables is that the coding discretizes the search space, even though the function may be continuous. On the other hand, since GAs require only function values at various discrete points, a discrete or discontinuous function can be handled with no extra cost. This allows GAs to be applied to a wide variety of problems. Another advantage is that the GA operators exploit the similarities in string-structures to make an effective search.

The most striking difference between GAs and many traditional optimization methods is that GAs work with a population of points instead of a single point. Because there are more than one string being processed simultaneously, it is very likely that the expected GA solution may be a global solution. Even though some traditional algorithms are population-based, like Box's evolutionary optimization and complex search methods, those methods do not use previously obtained information efficiently. In GAs, previously found good information is emphasized using reproduction operator and propagated adaptively through crossover and mutation operators. Another advantage with a population-based search algorithm is that multiple optimal solutions can be captured in the population easily, thereby reducing the effort to use the same algorithm many times.

GAs does not require any auxiliary information except the objective function values. Although the direct search methods used in traditional optimization methods do not explicitly require the gradient information, some of those methods use search directions that are similar in concept to the gradient of the function. Moreover, some direct search methods work under the assumption that the function to be optimized is unimodal and continuous. In GAs, no such assumption is necessary.

One other difference in the operation of GAs is the use of probabilities in their operators. None of the genetic operators work deterministically. In the reproduction operator, even though a string is expected to have fixed copies in the mating pool, a simulation of the roulette-wheel selection scheme is used to assign the true number of copies. In the crossover operator, even though good strings (obtained from the mating pool) are crossed, strings to be crossed are created at random and cross-sites are created at random. In the mutation operator, a random bit is suddenly altered. The action of these operators may appear to be naive, but careful studies provide some interesting insights about this type of search. The basic problem with most of the traditional methods is that they use fixed transition rules to move from one point to another. For instance, in the steepest descent method, the search direction is always calculated as the negative of the gradient at any point, because in that direction the reduction in the function value is maximum. In trying to solve a multimodal problem with many local optimum points, search procedures may easily get trapped in one of the local optimum points. Consider the bimodal function shown Figure VIII.1. The objective function has one local minimum and one global minimum. If the initial point is chosen to be a point in the local basin (point $x^{(t)}$ in the figure), the steepest descent algorithm will eventually find the local optimum point. Since the transition rules are rigid, there is no escape from these local optima. The only way to solve the above problem to global optimality is to have a starting point in the global basin.

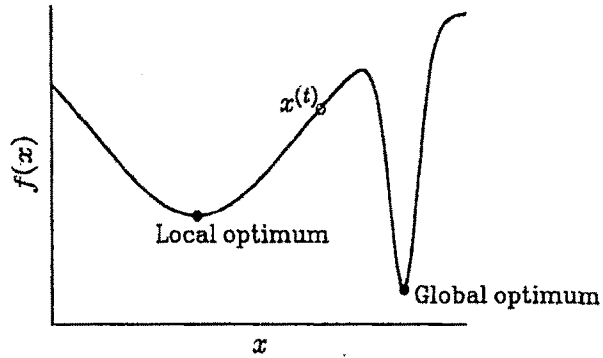

Figure VIII.1 An Objective Function with One Local Optimum and One Global Optimum

Since this information is usually not known in any problem, the steepest-descent method (and for that matter most traditional methods) fails to locate the global optimum. This phenomenon is termed as trapping in local optima. However, these traditional methods can be best applied to a special class of problems suitable for those methods.

For example, the gradient search methods will outperform almost any algorithm in solving continuous, unimodal problems, but they are not suitable for multimodal problem. Thus, in general, traditional methods are not robust. A robust algorithm can be designed in such a way that it uses the steepest descent direction most of the time, but also uses the steepest ascent direction (or any other direction) with some probability. Such a mixed strategy may require more number of function evaluations to solve continuous, unimodal problems, because of the extra computations involved in trying with non-descent directions. But this strategy may be able to solve complex, multimodal problems to global optimality. In the multimodal problem, shown in the above figure, the mixed strategy may take the point $x^{(t)}$ into the global basin (when tried with non-descent directions) and finally find the global optimum point. GAs use similar search strategies by using probability in all their operators. Since an initial random population is used, to start with, the search can proceed in any direction and no major decisions are made in the beginning. Later on, when the population begins to converge in some bit positions, the search direction narrows and a near-optimal solution is achieved. This nature of narrowing the search space as the search progresses is adaptive and is a unique characteristic of genetic algorithms.