NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]
ISSN: 2394-3696
VOLUME 1, ISSUE 2 DEC-2014

Review of Hooke and Jeeves Direct Search Solution Method Analysis Applicable To Mechanical Design Engineering

Mr. G.S KIRGAT
Department of Mechanical Engineering, Solapur University / W.I.Technology, Solapur, India
Prof. A.N Surde
Department of Mechanical Engineering, Solapur University / W.I.Technology, Solapur, India

Abstract
Role of optimization in engineering design is prominent one with the advent of computers. Optimization has become a part of computer aided design methodology. It is primarily being used in those design activities in which the goal is not only to achieve a feasible design, but also a design objective. The paper reviews the optimization in detail followed by the literature review and brief discussion of Hooks and Jeeves Method Analysis with an example.

Keywords: Optimization, Hooks and Jeeves, Design, Analysis, Operation Research.

Introduction:

Optimization:
In most engineering design activities, the common design objective could be simply to minimize the cost of production or to maximize the efficiency of the production. An optimization algorithm is a procedure which is executed iteratively by comparing various solutions till the optimum or a satisfactory solution is found. In many industrial design activities, optimization is achieved indirectly by comparing a few chosen design solutions and accepting the best solution. This simplistic approach never guarantees and optimization algorithms being with one or more design solutions supplied by the user and then iteratively check new design the true optimum solution [1]. There are two distinct types of optimization algorithms which are in use today. First there are algorithms which are deterministic, with specific rules for moving from one solution to the other secondly, there are algorithms which are stochastic transition rules. An important aspect of the optimal design process is the formulation of the design problem in a mathematical format which is acceptable to an optimization algorithm. Above mentioned theory (tasks) involve either minimization or maximization of an objectives. Mathematically programming techniques are useful in finding the minimum of a function of several variables under a prescribed set of constraints. Stochastic process techniques can be used to analyze problems described by a set of random variables having known probability distributions statistical methods enable one to analyze the experimental data and build empirical models to obtain the most accurate representation of the physical situation.

PRINCIPLES OF OPTIMAL DESIGN

History of Optimization Techniques
In an optimization problem one seeks to maximize or minimize a specific quantity called the objective, which depends on a finite number of (input) variables. These variables may be independent of one another or they may be related through one or more constraints. The existence of optimization methods can be traced to the days of Newton, Lagrange. and Cauchy, because of development of differential calculus, calculus of variations and other basic earlier methods. Despite these early contributions, very little progress was made till the middle of the 20th century. The advancement in optimization techniques resulted due to usage of high

speed digital computers in 1960(UK), 1947(Dantzig), 1957(Kuhn and Tucker), 1982(Von Neumann) [1] [4] [9].

What is optimization?

Optimization is the method of obtaining the best result under given circumstances. In engineering design activities (design, construction and maintenance), engineers have to take many technological and managerial decisions at several stages. The main and important goal of all such decisions is either to minimize the effort required or to maximize the desired benefit. For example: Mechanical engineers design mechanical components for the purpose of achieving either a minimum manufacturing cost or a maximum component life. Aeronautical engineers, in aerospace design activities, minimize the overall weight, since the minimization of the weight of aircraft components is of major concern to aerospace designers. Chemical engineers are interested in designing and operating a process plant for an optimum rate of production. Production engineers are interested in designing optimum schedules of various machining operations to minimize the idle time of machines and the overall job completion time. Civil engineers are involved in designing buildings bridges, dams and other structures in order to achieve a minimum overall cost or maximum safety or both. Electrical engineers are interested in designing communication networks so as to achieve minimum time for communication from one node to another. All the above mentioned tasks, such as effort required or the benefit desired involve either minimization or maximization (collectively known as optimization) of an objective. Also these tasks can be expressed as a function of certain (decision) variables, (optimization can be defined as the process of finding the conditions that give the maximum or minimum value of a function? [10] [11]

Methods of optimization (or operations research):

Because of growing complexity in engineering design activities, there is no single method for all optimization problems efficiently. Hence a number of optimization methods have been developed for solving different types of optimization problems. The optimum seeking methods are also known as mathematical programming techniques and they are generally studied as tools of operations research. Operations research is a branch of mathematics concerned with the application of scientific methods and techniques to decision making problems and with establishing the best or optimal solution. The following are the various mathematical programming techniques together with other well defined areas of operations research. Mathematical programming's Techniques are useful in finding the minimum of a function of several variables under a prescribed set of constraints:

1. Calculus methods
2. Calculus of variations
3. Nonlinear programming
4. Geometric programming
5. Quadratic programming
6. Linear programming
7. Dynamic programming
8. Integer programming
9. Stochastic programming
10. Separable programming
11. Multi objective programming
12. Network methods: CPM and PERT
13. Game theory
14. Simulated annealing
15. Genetic algorithms

16. Neural Networks.
17. Stochastic process techniques can be used to analyze problems described by a set of random variables having known probability distributions:
18. Statistical decision theory
19. Markov processes
20. Queuing theory
21. Renewal theory
22. Simulation methods and many others.

Choosing an efficient optimization technique

Because of the above spectrum of problems and methods of optimization, also due to the growing complexity in engineering optimization problems the engineer or scientist can no longer afford to rely on a particular method. Hence, an engineer specialized in a particular optimization problem is usually more informed about different factors governing that problem than anyone else, every engineer should know aspects to choose a proper optimization technique for the chosen optimal problem, i.e. the engineer must know the advantages and limitations of various methods, and about the working principles of different optimization methods, to choose the one that is more efficient to the problem at hand [12] [13].

Applications of Optimization in Engineering:

Optimization can be applied to solve any engineering optimization problem. To indicate the wide spectrum and scope of the optimization, some typical applications from different engineering domains are given below:

1. Design of air craft and aerospace structures for minimum weight
2. Finding the optimal trajectories of space vehicles
3. Optimum design of electrical machinery such as motors, generators and transformers.
4. Optimum design of electrical networks
5. Design of civil engineering structures such as frames, foundations, bridges, towers, chimneys and dams for minimum cost
6. Minimum weight design of structures for earth quake, wind and other types of random loading.
7. Design of water resources systems for maximum benefit.
8. Optimal plastic design of structures
9. Optimum design of control systems
10. Inventory control
11. Optimum design of linkages, cams, gears, machine tools, and other mechanical components
12. Selection of machining conditions in metal cutting processes for minimum production cost
13. Design of pumps, turbines and heat transfer equipment for maximum efficiency
14. Shortest route taken by a sales person visiting various cities during one tour
15. Allocation of resources or services among several activities to maximum the benefit.
16. Planning the best strategy to obtain maximum profit in the presence of a competitor
17. Optimal production planning, controlling and scheduling
18. Optimum design of chemical processing equipment and plants
19. Controlling the waiting and idle times and queuing in production lines to reduce the costs.
20. Selection of a site for an industry
21. Analysis of statistical data and building empirical models from experimental results to obtain the most accurate representation of the physical phenomenon
22. Design of optimum pipeline networks for process industries
23. Planning of maintenance and replacement of equipment to reduce operating costs.
24. Optimum design of electrical networks.
25. Solving for optimality in several mathematical economics and military problems [11].

Design Constraints:

The selected design variables have to satisfy certain specified functional and other requirements, known as constraints (restrictions). These constraints that must be satisfied to produce an acceptable design are collectively called design constraints.

Constraints that represent limitations on the behavior or performance of the system are termed behavior or functional constraints. Constraints that represent physical limitations on design variables such as availability, fabric ability and transportability are known as geometric or side constraints.

Mathematically, there are usually two types of constraints: Equality or Inequality constraints. Inequality constraints state that the relationships among design variables are either greater than, smaller than or equal to a resource value.

Equality constraints state that the relationships should exactly match a resource value. Equality constraints are usually more difficult to handle and therefore, need to be avoided wherever possible. Thus, the second thumb rule in the formulation of optimization problem is that the number of complex equality constraints should be kept as low as possible. (Optimal problem) optimization problem formulation: The purpose of the formulation procedure is to create a mathematical model of the optimal problem, which then can be solved using an optimization technique. The following figure shows an outline of the steps usually involved in an optimal problem formulation process [1] [12].

Optimal design Procedure:

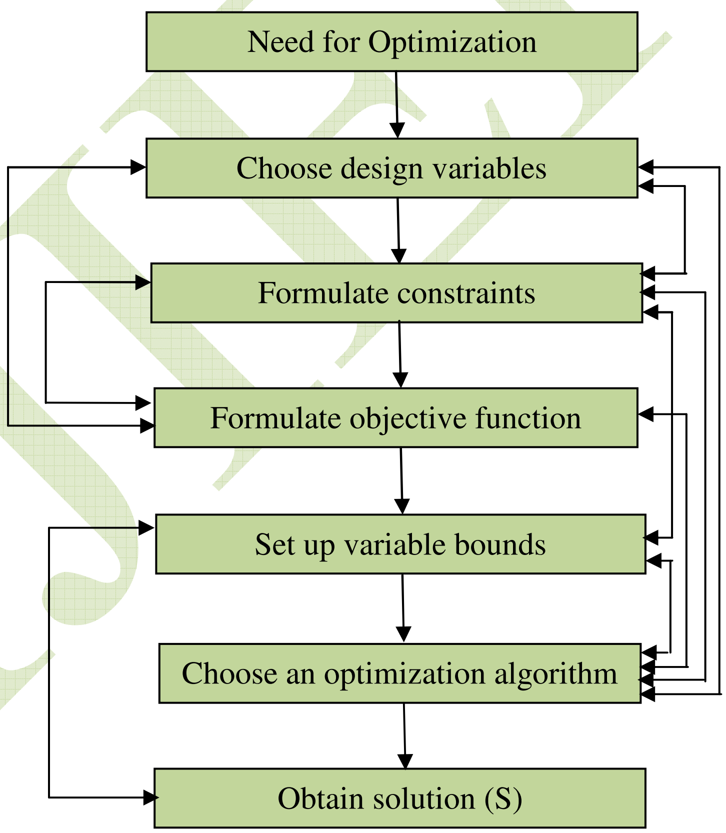

Fig 1 a flow chart of the Optimal design Procedure

Literature review

Many engineers and researchers in industries and academics face difficulty in understanding the role of optimization in engineering design. Too many of them, optimization is an esoteric technique used only in mathematics and operations research related activities. With the advent of computers, optimization has become a part of computer-aided design activities. It is primarily being used in those design activities in which the goal is not only to achieve just a feasible design, but also a design objective. In most engineering design activities, the design objective could be simply to minimize the cost of production or to maximize the efficiency of production. An optimization algorithm is a procedure which is executed iteratively by comparing various solutions till the optimum or a satisfactory solution is found. In many industrial design activities, optimization is achieved indirectly by comparing a few chosen design solutions and accepting the best solution. This simplistic approach never guarantees an optimal solution. On the contrary, optimization algorithms begin with one or more design solutions supplied by the user and then iteratively check new design solutions in order to achieve the true optimum solution [1] [4]. There are two distinct types of optimization algorithms which are in use today. First, there are algorithms which are deterministic, with specific rules for moving from one solution to the other. These algorithms have been in use for quite some time and have been successfully applied to many engineering design problems. Secondly, there are algorithms which are stochastic in nature, with probabilistic transition rules. These algorithms are comparatively new and are gaining popularity due to certain properties which the deterministic algorithms do not have. An important aspect of the optimal design process is the formulation of the design problem in a mathematical format which is acceptable to an optimization algorithm. However, there is no unique way of formulating every engineering design problem. To illustrate the variations encountered in the formulation process [9] [10].

Optimization Algorithms:

The above 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 them. In some problems, the terms are not explicit functions of the design variables. Unfortunately, there does not exist a single optimization algorithm 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. This book provides a number of optimization algorithms used in engineering design activities.
Since the optimization algorithms involve repetitive application of certain procedures, they need to be used with the help of a computer. That is why the algorithms are presented in a step-by-step format so that they can be easily coded. To demonstrate the ease of conversion of the given algorithms into computer codes, most chapters contain a representative working computer code. Further, in order to give a clear understanding of the working of the algorithms, they are hand-simulated on numerical exercise problems. Simulations are performed for two to three iterations following the steps outlined in the algorithm sequentially. Thus, for example, when the algorithm suggests moving from Step 5 to Step 2 in order to carry out a conditional statement, the exercise problem demonstrates this by performing Step 2 after Step 5. For the sake of clarity, the optimization algorithms are classified into a number of groups, which are now briefly discussed [1] [4] [10].

- **Single-variable optimization algorithms -** Because of their simplicity, single-variable optimization techniques are discussed first. 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 -** A number of algorithms for unconstrained, multivariable optimization problems are discussed next. 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 -** Constrained optimization algorithms are described next. These algorithms use the single-variable and multivariable optimization algorithms repeatedly and simultaneously maintain the search effort inside the feasible search region. Since these algorithms are mostly used in engineering optimization problems, the discussion of these algorithms covers most of the material of this book.
- **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 problems and are discussed. 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 tho above algorithms. These algorithms are being used in engineering design problems since sixties. Because of their existence and use for quite some years, we call these algorithms 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 past. Two such algorithms--genetic algorithms and simulated annealing are discussed [10] [11].

We have put together about 34 different optimization algorithms. Over the years, researchers 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 stochastic programming methods and dynamic programming method which are very different than the above algorithms. Because of the space limitation and occasional use of these algorithms in engineering design problems, we have not included them in this book. A detailed discussion of these algorithms can be found elsewhere. 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 [12] [13].

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 various 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 optima 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. 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 described in Chapter 6 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. For example, the truss structure problem described earlier should really be reformulated as the minimization of both the weight of the truss and the deflection at the point C. Multiobjective optimization problems give rise to a set of optimal solutions known as Pareto-optimal solutions (Chankong and Haimes, 1983), 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 described in Chapter 6 demonstrate one way to handle multiple objectives and help find multiple Pareto-optimal solutions simultaneously [13] [14] [18].

At the end of the optimization process, one obvious question may arise: Is the obtained solution a true optimum solution? Unfortunately, there is no easy answer to this question for all optimization problems. In problems where the objective functions and constraints can be written in simple, explicit mathematical forms, the Kuhn-Tucker conditions described in Chapter 4 may be used to check the optimality of the obtained solution. However, those conditions are valid only for a few classes of optimization problems. In a generic problem, this question is answered in a more practical way. 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 [1] [4].

Recent Development in Optimization Algorithms:

Multivariable Optimization Algorithms:

Therefore, a proper understanding of the single-variable function optimization algorithms would be helpful in appreciating multivariable function optimization algorithms presented in this chapter. The algorithms are presented for minimization problems, but they can also be used for maximization problems by using the duality principle. The algorithms can be broadly classified into two categories-direct search methods and gradient-based methods. In direct search methods, only function values at different points are used to constitute a search. In gradient-based methods, derivative information is used to constitute a search.

Optimality Criteria:

The definition of a local, a global, or an inflection point remains the same as that for single-variable functions, but the optimality criteria for multivariable functions are different. In a multivariable function, the gradient of a function is not a scalar quantity; instead it is a vector quantity. The optimality criteria can be derived by using the definition of a local optimal point and by using Taylor's series expansion of a multivariable function (Rao, 1984). Without going into the details of the analysis, we simply present the optimality criteria for a multivariable function [14]. In this chapter and subsequent chapters, we assume that the objective function is a function of N variables represented by zi, £2, ..., XN. The gradient vector at any point x is represented by V/(z) which is an TV-dimensional vector given as follows:

The first-order partial derivatives can be calculated numerically using Equation presented in the previous part. However, the second-order derivatives in multivariable functions form a matrix, V2/ (z) (better known as the Hessian matrix) given as follows:

$$\nabla^2 f(x^{(t)}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_N} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_N \partial x_1} & \frac{\partial^2 f}{\partial x_N \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_N^2} \end{bmatrix}_{x^{(t)}}$$

The second-order partial derivatives can also be calculated numerically. By defining the derivatives, we are now ready to present the optimality criteria (Reklaitis, et al., 1983):
A point x is a stationary point if Vf(x) = 0. Furthermore, the point is a minimum, a maximum, or an inflection point if V2f(x) is positive-definite, negative-definite, or otherwise.
A matrix V2/(z'()) is defined to be positive-definite if for any point y in the search space the quantity TV2f(x}y > 0. The matrix is called a negative definite matrix if at any point the quantity < 0. If at some point y ‘1’ in the search space the quantity y+ V2/(s)y+ is positive and at some other point y the quantity y~ V'2f(x)y~ is negative, then the matrix V2f(x} is neither positive-definite nor negative-definite. There are a number of ways to investigate whether a matrix is positive-definite (Strang, 1980). One common way to do this is to evaluate the eigen values of the matrix. If all eigen values are positive, the matrix is positive-definite. The other way to test the positive-definiteness of a matrix is to calculate the principal determinants of the matrix. If all principal determinants are positive, the matrix is positive-definite. It is worth mentioning here that the negative-definiteness of a matrix A can be verified by testing the positive definiteness of the matrix – A [11] [12] [13].

Unidirectional Search:
Many multivariable optimization techniques use successive unidirectional search techniques to find the minimum point along a particular search direction. Since unidirectional searches will be mentioned and used a number of times in the remaining chapters, we illustrate here how a unidirectional search can be performed on a multivariable function, A unidirectional search is a one-dimensional search performed by comparing function values only along a specified direction. Usually, a unidirectional search is performed from a point x(t) and in a specified direction s(t). That is, only points that lie on a line (in an TV-dimensional space) passing through the point x(t) and oriented along the search direction ‘s’ are allowed to be considered in the search process. Any arbitrary point on that line can be expressed as follows:
x (α) = x(t) + α(t).
The parameter a is a scalar quantity, specifying a relative measure of distance of the point x(α) from x(t). Note, however, that the above equation is a vector equation specifying all design variables xi (α). Thus, for a given value of a, the point x(a) can be known. Any positive and negative value of a will create a point on the desired line. If α = 0, the current point x(t) is obtained. In order to find the minimum point on the specified line, we can rewrite the multivariable objective function in terms of a single variable Q by substituting each variable Xi by the expression xi (α) given in Equation. and by using a suitable single-variable search method.
Consider the objective function:
Minimize f(x1, x2) = (x1 - 10)2 + (x2 - 10)2.
Figure 2 shows the contour plot of this function.

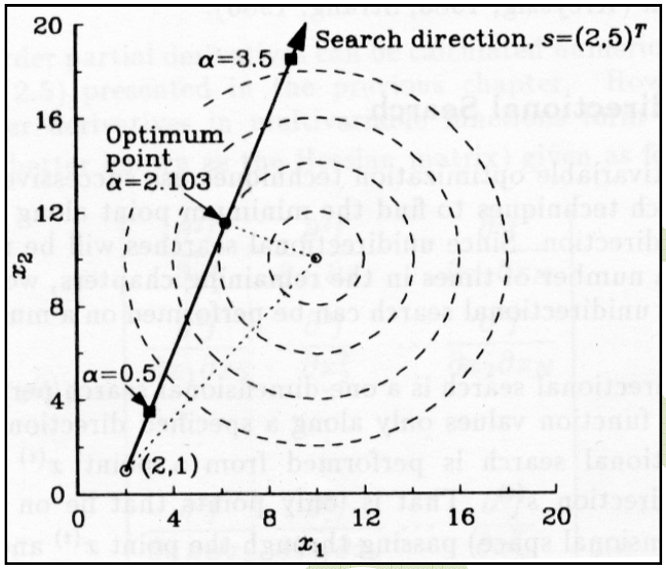

Fig 2 an illustration of a unidirectional search method.

The property of a contour line is that any two points on a contour line have the same function value. Thus, it is convenient to show the optimum point of a function on contour plots. The figure shows that the minimum point lies at the point (10, 10)T. The function value at this point is zero. Let us say that the current point of interest is x (t) = (2, 1) T and we are interested in finding the minimum point and the minimum function value in a search direction s (t) = (2, 5) T from the current point. From the right-angled triangle shown in dotted line, we obtain the optimal point x* = (6.207, 11.517) T [1].

Direct Search Methods:

In this section, we present a number of minimization algorithms that use function values only. Algorithms requiring gradient of the objective function are discussed in the next section. It is important to mention here that if the gradient information is available, a gradient-based method may be more efficient. Unfortunately, many real-world optimization problems require the use of computationally expensive simulation packages to calculate the objective function, thereby making it difficult to compute the derivative of the objective function. In these problems, direct search techniques may be found to be useful,

In a single-variable function optimization, there are only two search directions a point can be modified either in the positive direction or the negative direction. The extent of increment or decrement in each direction depends on the current point and the objective function. In multi-objective function optimization, each variable can be modified either in the positive or in the negative direction, thereby totaling 2 different ways. It is important to mention here that if the gradient information is available, a gradient-based method may be more efficient. Unfortunately, many real-world optimization problems require the use of computationally expensive simulation packages to calculate the objective function, thereby making it difficult to compute the derivative of the objective function. In these problems, direct search techniques may be found to be useful,

Moreover, an algorithm, having searches along each variable one at a time, can only successfully solve linearly separable functions. These algorithms (called one-variable-at-a-time methods) cannot usually solve functions having nonlinear interactions among design variables. Ideally, we require algorithms which either completely eliminate the concept of search direction or manipulate a set of points to create a better set of points or use complex search directions to effectively decouple the nonlinearity of the function [4].

Hooke-Jeeves pattern search method:

The pattern search method works by creating a set of search directions iteratively. The created search directions should be such that they completely span the search space. In other words, they should be such that starting from any point in the search space any other point in the search space can be reached by traversing along these search directions only. In a N-dimensional problem, this requires at least N linearly independent search directions. For example, in a two- variable function, at least two search directions are required to go from any one point to any other point. Among many possible combinations of N search directions, some combinations may be able to reach the destination faster (with lesser iterations), and some may require more iterations.

In the Hooke-Jeeves method, a combination of exploratory moves and heuristic pattern moves is made iteratively. An exploratory move is performed in the vicinity of the current point systematically to find the best point around the current point. Thereafter, two such points are used to make a pattern move. We describe each of these moves in the following paragraphs:

Exploratory move

Assume that the current solution (the base point) is denoted by xc. Assume also that the variable x% is perturbed by A; Set i = 1 and x = xc.

Step 1 Calculate f = f (x), f + = f (xi + Δi) and f - = f (xi - Δi).

Step 2 Find f min = min (f, f +, f -). Set x corresponds to f min.

Step 3 Is i = N? If no, set i = i + 1 and go to Step 1;
Else x is the result and goes to Step 4.

Step 4 If x xc, success; Else failure.

In the exploratory move, the current point is perturbed in positive and negative directions along each variable one at a time and the best point is recorded. The current point is changed to the best point at the end of each variable perturbation. If the point found at the end of all variable perturbations is different than the original point, the exploratory move is a success; otherwise the exploratory move is a failure. In any case, the best point is considered to be the outcome of the exploratory move [1] [4] [19].

- **Pattern move**

A new point is found by jumping from the current best point xc along a direction connecting the previous best point a '1' and the current base point x(k) as follows:

xp (k+1) = x (k) + (x(k) - x(k-1))

The Hooke-Jeeves method comprises of an iterative application of an exploratory move in the locality of the current point and a subsequent jump using the pattern move. If the pattern move does not take the solution to a better region, the pattern move is not accepted and the extent of the exploratory search is reduced. The algorithm works as follows,

- **Algorithm**

Step 1 Choose a starting point x (0), variable increments Δi (i=1, 2... N), a step reduction factor α > 1, and a termination parameter, €. Set k = 0.

Step 2 Perform an exploratory move with x (k) as the base point. Say x is the outcome of the exploratory move. If the exploratory move is a success, set z (k+1) = x and go to Step 4;
Else go to Step 3.

Step 3 Is ‖ Δ ‖ < € ? If yes, Terminate; Else set A; = A/a for i - 1, 2, . . . , N and go to Step 2.

Step 4 Set k = k + 1 and perform the pattern move:
xp (k+1) = x (k) + x(k) – x(k+1).

Step 5 Perform another exploratory move using Xp as the base point. Let the result be x (k+1).

Step 6 Is f(x (k+1)) < f (x (k))? If yes, go to Step 4.
Else go to Step 3.

The search strategy is simple and straightforward. The algorithm requires less storage for variables; only two points (x (k) and x (k+l)) need to be stored at any iteration. The numerical calculations involved in the process are also simple. But, since the search largely depends on the moves along the coordinate directions (xi, x-z, and so on) during the exploratory move, the algorithm may prematurely converge to a wrong solution, especially in the case of functions with highly nonlinear interactions among variables. The algorithm may also get stuck in the loop of generating exploratory moves either between Steps 5 and 6 or between Steps 2 and 3. Another feature of this algorithm is that it terminates only by exhaustively searching the vicinity of the converged point. This requires a large number of function evaluations for convergence to a solution with a reasonable degree of accuracy. The convergence to the optimum point depends on the parameter a. A value a = 2 is recommended. The working principle of the algorithm can be better understood through the following hand-simulation on a numerical exercise problem.

Example

Consider the Himmelblau function again:

Minimize f(x1, x2) = (x12 + x2 - 11)2 + (x1 + x22 - 7)2.

Step 1 Let us choose the initial point to be x(0) = $(0,0)^T$, the increment vector $\Delta = (0.5, 0.5)^T$, and reduction factor $\alpha = 2$. We also set a termination parameter € = 10-3 and an iteration counter k = 0.

Step 2 We perform an iteration of exploratory move using x(0) as the base point. Thus, we set x = xc = $(0, 0)^T$ and i = 1. The steps of the exploratory move are worked out in the following:

Step 1 We first explore the vicinity of the variable x. We calculate the function values at three points:

(x(0) + Δ1,*?))r = $(0.5,0.0)^T$, f+ = f ($(0.5, 0)^T$) = 157.81,

x(0) = $(0.0,0.0)^T$, f = f ($(0,0)^T$) = 170,

$(x1(0) - \Delta 1, x2(0))^T$ - $(-0.5,0.0)^T$, f - = 171.81.

Step 2 The minimum of the above three function values is fmin = 157.81, and the corresponding point is x = $(0.5,0.0)^T$.

Step 3 At this iteration $i \neq 2$ (not all variables are explored yet). Thus, we increment the counter i = 2 and explore the second variable x^- This completes one iteration of the exploratory move.

Step 1 Note that at this point, the base point is $(0.5,0)^T$. We explore variable xi at this point and calculate function values at the following three points:

f + = f ($(0.5,0.5)^T$) = 144.12,

f = f ($(0.5,0.0)^T$) = 157.81,

f - = f ((0.5,-0.5)T) = 165.62.

Step 2 Here, fmin = 144.12 and the corresponding point is x = $(0.5,0.5)^T$.

Step 3 At this step i = 2 and we move to Step 4 of the exploratory move.

Step 4 Since $x \neq x^c$, the exploratory move is a success and we set x = $(0.5,0.5)^T$.

Since the exploratory move is a success, we set art1' = x = $(0.5, 0.5)^T$ and move to Step 4. The successful points in the exploratory move of the Hooke-Jeeves algorithm are marked with a filled circle.

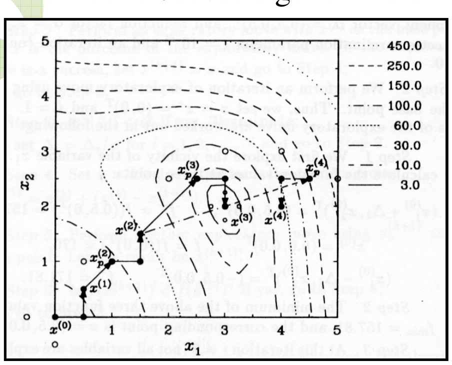

Figure 3 Four iterations of Hooke-Jeeves search method shown on a contour plot of the Himmelblau function.

Step 4 We set k = 1 and perform a pattern move:
xp(2) = x(1) + (x(1) - x(0)) = $2(0.5,0.5)^T - (0,0)^T - (1,1)^T$.
Step 5 We perform another exploratory move using Xp as the base point. After performing the exploratory move as before, we observe that the search is a success and the new point is x = $(1.5,1.5)^T$. We set the new point x (2) = x = $(1.5,1.5)^T$.
Step 6 We observe that f(x (2)) = 63.12, which is smaller than f (x (1)) = 144.12. Thus, we proceed to Step 4 to perform another pattern move. This completes one iteration of the Hooke-Jeeves method.
Step 4 We set k = 2 and create a new point
xp(3) = 2x(2) – x(1) = $(2.5, 2.5)^T$.
It is interesting to note here that since x(2) is better than x(1), a jump along the direction (x(2) – x(1)) is made. This jump takes the search closer towards the true minimum.
Step 5 Perform another exploratory move locally to find out if there is any better point around the new point. After performing an exploratory move on both variables, it is found that the search is a success and the new point is x (3) = $(3.0, 2.0)^T$. This new point is incidentally the true minimum point. In this example, the chosen initial point and the increment vector happen to be such that in two iterations of the Hooke-Jeeves algorithm the minimum point is obtained. But, in general, more iterations may be required. An interesting point to note that even though the minimum point is found in two iterations, the algorithm has no way of knowing whether the optimum is reached or not. The algorithm proceeds until the norm of the increment vector Δ is small. We continue to perform two more iterations to see how the algorithm may finally terminate at the minimum point.
Step 6 Calculating the function value at the new point, we observe that f (x (3)) = 0 < f (x (2)) = 63.12. Thus, we move to Step 4.
Step 4 The iteration counter k = 3 and the new point is xp (4), = 2x (3) – x (2) = $(4.5, 2.5)^T$.
Step 5 By performing an exploratory move with xp (4) as the base point, we find that the search is a success and x — (4.0, 2.0). Thus, we set x (4) = $(4.0, 2.0)^T$.
Step 6 The function value at this point is f(x (4)) = 50, which is larger than f(x (3)) = 0. Thus, we move to Step 3.
Step 3 Since || Δ || = 0.5, we reduce the increment vector
$\Delta = (-0.5, 0.5)^T/2 = (0.25,0.25)^T$
And proceed to Step 2 to perform the next iteration.
Step 2 Perform an exploratory move with x = (3.0, 2.0) r as the current point. The exploratory move on both variables is a failure and we obtain x = (3.0, 2.0) r. Thus we proceed to Step 3.
Step-3 Since ||Δ|| is not small, we reduce the increment vector by a again and move to Step 2. The new" increment vector is A = $(0.125, 0.125)^T$.
The algorithm now continues with Steps 2 and 3 until ||Δ|| are smaller than the termination factor ϵ. Thus, the final solution is x = $(3.0, 2.0)^T$ with a function value f (x) = 0 [1] [4] [19].

CONCLUSION

In the Hooke-Jeeves method, a combination of exploratory moves and heuristic pattern moves is made iteratively. An exploratory move is performed in the vicinity of the current point systematically to find the best point around the current point. Thereafter, two such points are used to make a pattern move. It is important to mention here that if the gradient information is available, a gradient-based method may be more efficient. Unfortunately, many real-world optimization problems require the use of computationally expensive simulation packages to calculate the objective function, thereby making it difficult to compute the derivative of the objective function. In these problems, direct search techniques may be found to be useful, In a single-variable function optimization, there are only two search directions a point can be modified either in the positive direction or the negative direction. The extent of increment or decrement in each direction depends on the current point and the objective function. In multi-objective function optimization, each variable can be modified either in the positive or in the negative direction, thereby totaling 2 different ways.

References:

[1] Kalyanmoy Deb, "*Optimization for Engineering Design- Algorithms and Examples*" PHI Learning Private Limited New Delhi, 10th Edition, May 2009.
[2] W.A.J Chapman, "*Workshop Technology-Part3*",CBS Publishers & Distributors, Third Edition, 1986.
[3] S.K. Hajra Choudhary, A.K.Hajra Choudhary,Nirjhar Roy"*Elements of Workshop Technology*", Vol:II MACHINE TOOLS,Media Promoters & Publishers Pvt. Ltd.,Mumbai,Thirteenth Edition,2010.
[4] G.S.S.,Bhishma Rao,"*Optimization Techniques*", Scitech Publications,India,Pvt. Ltd. ,Second Edition,2006.
[5] WIDIA, "*WIDAX TOOLS for Milling and Drilling*", WIDIA (India) Ltd., Pub. No. WMC-073-94.
[6] A.Bhattacharya,"*Metal Cutting- Theory & Pratice*",New Central Book Agency Pvt. Ltd., Sixth Edition,2008.
[7] Steve F. Krar,Mario Rapisarda, Albert F. Check,"*Machine Tool and Manufacturing Technology*", Delmar Publisher-Cengage Learning India Pvt. Ltd. ,Second Edition,2010.
[8] G.R. Nagpal,"*Machine Tool Engineering*",Khanna Publisher New Delhi,Sixth Edition,2008
Purna Chandra Biswal,"*Optimization in Engineering*",Scitech Publications India Pvt. Ltd.,Revised Edition,2006.
[9] Panos Y. Papalambros, Douglas J.Wilde,"*Principles of Optimal Design- Modeling and Computation*", Cambridge University Press New Delhi,Second Edition,2009.
[10] Kalyanmoy Deb,Partha Chakraborty,*N G R Iyengar, Santosh K Gupta,"Advaces in Computational Optimization and its Applications*", Universities Press(India) Pvt. Ltd.,First Edition,2007.
[11[Jasbir S. Arora,"*Introduction to Optimum Design*",ELSEVIER Academic Press,Second Edition,2004.
[12] Singresu S. Rao,"*Enigneering Optimization Theory and Practice*",New Age International Publishers,2008.
[13] Way Kuo,V.Rajendra Prasad,Frank A. Tillman,Ching Lai Hwang,"*Optimal Reliability Design, Fundamentals and Applications*", Cambridge University Press,Cambridge,U.K.,2001.
[14] Luis W.Hernández González, Roberto Pérez Rodríguez, Patricia C. Zambrano Robledo , Martha P. Guerrero Mata & Lumimita Dumitrescu, "Study of Performance of High Speed Turning of AISI 1045 using the Volumetric Dimension of Cutting Tool Life " , *International Journal of Advances in Machining and Forming Operations ,* Vol.3 , January – June 2011 , pp. 19-32.
[14] S. V. Shvets , " Assessment of Physical Efficiency of the Cutting System ", *International Journal of Advances in Machining and Forming Operations ,* Vol.3 , January – June 2011 , pp. 33-41.

[15] Ajay P. Dhawan and M. T. Telsang , " *Optimization of Cutting Parameters in Multipass Turning Using Taguchi Method* " , *International Journal of Mechanical Engineering and Materials Sciences* , Vol.4 , July-Dec. 2011 , pp. 39-46.

[16] Robert Michael Lewis, ICASE, Hampton, Virginia, Virginia Torczon and Michael W. Trosset, College of William & Mary, Williamsburg, Virginia, "*Direct Search Methods: Then and Now*", NASA/CR-2000-210125 ICASE Report No. 2000-26, May 2000,pp. 1-17.

[17] Robert Hooke and T.A. Jeeves,Westinghouse Research Laboratories,Piitsburgh,Pennsylvania,"Direct Research Solution of Numerical and statistical Problem research",International Journal of Statistics, Vol.16, pp.670-683,1960.

[18] Frank A. Tillman,Ching-lai Hwang,Way Kuo,"Optimization of System Reliability",Marcel Dekker Inc.,New York and Besel,1980.

[19] Indrajit Mukharjee,Pradip Kumar Ray,"A review of optimization process in metal cutting process", Computers and Industrial Engineering 50(2006),pp. 15-34.

Websites:

A. http://www.mathworks.in/products/matlab/whatsnew.html

B. http://matlab-r2014a.software.informer.com/8.3/

C. http://en.wikipedia.org/wiki/MATLAB

D. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6118021&matchBoolean%3Dtrue%26queryText%3DOptimization+in+metal+cutting+process