Chapter III

Basic Theory of Optimization

In engineering economics, and the physical sciences, as well as in other fields, the analyst is frequently faced with the problem of optimizing complex arrangements of equipment, operations, circuits, or processes. He wishes to minimize or maximize some function, termed the objective function, representing cost, weight, throughput, or the like, subject to certain constraints. When formulated as mathematical statements, a broad class of these optimization problems can be grouped together into a category termed the nonlinear programming problem; methods of solving such problems are called nonlinear programming. In this chapter we first describe somewhat formally the general nonlinear programming problem and certain special subproblems. Then the relationship between the goal of optimization of a real process and the mathematical representation of the optimization is characterized by means of an example, Next, certain definitions and terminology are briefly outlined and finally, the conditions for optimality are described.

3.1 THE LINEAR PROGRAMMING PROBLEM

A linear programming problem is one in which a linear function is the criterion to be minimized or maximized, a criterion subject to constraints that are also linear functions. A combination of scalars or vectors denoted in general by x is said to be linear if the scalars or vectors can be assembled in the form

$$c_1 x_1 + c_2 x_2 + \dots + c_n x_n \qquad (3.1\text{-}1)$$

where the c's are constants. For example, the following function is linear in the variables x_1, x_2, and x_3 :

$$4x_1 + 3x_2 + 5x_3 + 2$$

whereas the following function is nonlinear in the same variables:

$$2x^2_1 + x_1 x_2 + 3e^{x3}$$

The x's are said to be linearly dependent if, for some set of c_1's (assuming the c_1's are not all zero), the following is true:

$$c_1 x_1 + c_2 x_2 + \dots + c_n x_n = \sum_{i=1}^{n} c_1 x_1 = 0 \qquad (3.1\text{-}2)$$

On the other hand, if $\sum_{i=1}^{n} c_1 x_1 = 0$ only if the c1's are all zero, the x's are said to be linearly independent.

Linear programming has flourished since World War II, drawing the attention of many mathematicians, economists, and engineers because of its widespread practical applications as well as its mathematical elegance, Fruitful applications have been demonstrated in the areas of:

1. Optimal routing for air transport
2. Time-phased distribution of supply from factories and depots to bases
3. Allocation of electronic equipment to naval vessels
4. Production scheduling
5. Contract awards
6. Communication system design and message routing
7. Personnel assignment
8. Maximal flows in transportation networks
9. Gasoline blending

To some extent the publicity attracted to linear programming has distorted its significance, for it is applicable only when the underlying hypotheses are satisfied, and these are predicated on a linear mathematical representation of the real world, Nonlinear programming avoids such drastic simplifications.

Although the linear programming problem can be stated in many related forms, we shall write it as follows:

$$\text{Minimize :} \quad y = \sum_{i=1}^{n} c_i \, x_i \qquad (3.1\text{-}3a)$$

$$\text{Subject to} \quad \sum_{i=1}^{n} a_{ij} \, x_i - b_i \geq 0 \quad j = 1, \ldots . m \qquad (3.1\text{-}3b)$$

$$x_i \geq 0 \quad i = 1, \ldots . n \qquad (3.1\text{-}3c)$$

where the a' s, b' s, and c's are constants, and the x's are the variables whose values are sought.

Matrix notation provides a compact way of stating mathematical programming problems and describing algorithms for their solution. We will let x and c be n x 1 column vectors in E^n (e.g., in the n-dimensional Euclidean space composed of the variables), a be an n x m matrix of constants, and b be an m x 1 column vector.

$$x = \begin{vmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{vmatrix} \quad a = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{vmatrix} \quad b = \begin{vmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_m \end{vmatrix} \quad c = \begin{vmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_n \end{vmatrix}$$

Then the equivalent of Eqs. (3.1-3) in matrix notation is

$$\text{Minimize} \quad y \equiv f(x) = c^T x \tag{3.1-4a}$$

$$\text{Subject to } a^T x \geq b \tag{3.1-4b}$$

$$x \geq 0 \tag{3.1-4c}$$

where the superscript T denotes transpose. A vector x^* satisfying expression (3.1-4) is the desired solution.

3.2 THE GENERAL NONLINEAR PROGRAMMING PROBLEM

In the broadest sense, the general nonlinear problem is to find an extremum of an objective function subject to equality and/or inequality constraints. The constraints may be linear and/or nonlinear. However, it is customary to be somewhat more restrictive in talking about the general nonlinear programming problem, and to exclude specifically from consideration the following special cases:

1. The variables are restricted to integer values (nonlinear integer programming)

2. The constraints involve the parameter time in the form of differential equations (optimal control; dynamic optimization).

We will let the continuous functions $f(x)$ denote the objective function, $h_1(x),\ldots,h_m(x)$ denote the equality constraints, and $g_{m+1}(x),\ldots,g_p(x)$ denote the inequality constraints, where $x = [x_1,\ldots, x_n]^T$ is a column vector of components $x_1,\ldots,x_n$, in the n-dimensional Euclidean space.

As in linear programming, the variables $x_1, x_2,\ldots,x_n$ may be design parameters, controller adjustments, instrument readings, ect., while the objective function may represent cost, weight, revenue, and so forth, and the constraints represent the technical requirements, operating conditions, flow capacities, or safety factors inherent in the process.

The nonlinear programming problem can be formally stated as:

$$\text{Minimize: } f(x) \quad x \in E^n \tag{3.2-1}$$

subject to m linear and/or nonlinear equality constraints

$$h_j(x) = 0 \qquad j = 1,\ldots,m \quad . \tag{3.2-2}$$

and (p-m) linear and/or nonlinear inequality constraints

$$g_j(x) \geq 0 \qquad j = m+1,\ldots p \tag{3.2-3}$$

Although in some special cases the equality constraints can be explicitly solved for selected variables and those variables eliminated from the problem as independent variables, reducing the problem to one with inequality constraints only, most often the equality constraints can be solved only implicitly and must be retained.

An alternative representation of expressions (3.2-1) through (3.2-3) sometimes encountered is

$$\text{Minimize: } \left[f(x) \,\middle|\, x \in R \right] \tag{3.2-4}$$

where R is the domain of x for which conditions (3.2-2) and (3.2-3) are satisfied, e.g.,

$$R = \left[x \,\middle|\, h_j(x) = 0 \ ; \ g_j(x) \geq 0 \quad \text{for all } j \right] \tag{3.2-5}$$

The inequality sign in $g_j(x) \geq 0$ can be reversed by multiplying through by - 1 without changing the mathematical statement of the problem.

As a simple example of a nonlinear programming problem that can be illustrated graphically, we write

Minimize: $f(x) = x_1^2 + x_2^2 + 2x_2$

Subject to : $h_1(x) = x_1^2 + x_2^2 - 1 = 0$

$g_2(x) = x_1 + 2x_2 - 0.5 \geq 0$

$g_3(x) = x_1 \geq 0 \quad g_4(x) = x_2 \geq 0$

In Fig 3.2-1, the objective function is depicted by dashed lines, the equality constraint by a solid heavy line and the boundary of the region formed by the inequality constraints designated by the shaded lines (with the shading in the interior of the region).

For each point x of the n-dimensional space of variables $x_1, x_2, \ldots, x_n$ the function f(x) has a particular value, and as a consequence, the n-dimensional space is a scalar field for the optimality criterion. A family of contours (equipotential hypersurfaces) can be drawn in the space corresponding to specific values of f(x) as indicated in Fig.3.2-1. The space of variables $X_1, x_2 \ldots, x_n$ is also a scalar field for the constraint functions and equations, and equipotential hypersurfaces for these bounding functions can also be drawn. In general, it is not possible to ascertain beforehand by classical calculus the location of the vector x* that yields the minimum (or maximum) value of f(x), for x* may be located on the intersection of the constraint surfaces or within it.

The linear and the quadratic programming problem can be considered to be two special cases of the general nonlinear programming problem. When the function (3.2-1) and the equalities (3.2-2) and inequalities (3.2-3) are all linear, one obtains a linear programming problem. If the objective function is quadratic and the constraints are linear, one obtains the quadratic programming problem.

Minimize : $f(x) = a_o + c^T x - x^T Qx$ (3.2-6a)

Subject to: $a^T x \geq b$ (3.2-6b)

$x \geq 0$ (3.2-6c)

Where Q is a positive definite or semidefinite symmertric square matrix, and a and b are coefficient matrices previously defined in connection with Eq. (3.1-4). Sometimes linear equality constraints are included as part of the quadratic programming problem.

$a'^T x = b'$ (3.2-6d)

An example of a quadratic programming problem is

Minimize : $f(x) = 0.5x_1^2 + 0.5\, x_2^2 - x_1 - 2x_2$

Subject to: $g_1(x) : 6 - 2x_1 - 3x_2 \geq 0$

$g_2(x) : 5 - x_1 - 4x_2 \geq 0$

$g_3(x) : x_1 \geq 0$

$g_4(x) : x_2 \geq 0$

In all three classes of problems-----nonlinear, linear, and quadratic----one wants to find the vector $x^* = [x_1^* \ldots\ldots x_n^*]^t$ that minimizes (or alternatively, maximizes) $f(x)$ under conditions such that $h_j(x) = 0$, $j = 1,\ldots\ldots,m$ and $g_j(x) \geq 0$, $j = m+1,\ldots\ldots p$.

3.3 RELATION OF THE NONLINEAR PROGRAMMING PROBLEM TO A REAL PROCESS

in certain optimization problems such as least squares ("minimize the sum of the squares of the deviations between the observed and predicted response") or the minimization of an integral generated from a variational calculus problem, the nonlinear programming problem statement is easily connected with the physical problem to be solved. But in other cases this is not so. A simple examination of the mathematical statement of a nonlinear programming problem cannot by itself bring out all the factors involved in the optimization of a real process. In this section we briefly review the interaction between the physical problem and its mathematical representation.

In the optimization of a real process the parameters and or the variables are connected by physical laws, such as the conservation of mass or energy, that must be incorporated in the nonlinear programming problem as equality constraints even if they are only inferred. Thus one group of constraints consists of functional relations that must be taken into account if the optimization is to be physically realizable, a second group of constraints incorporates existing limits on variables or parameters that ensure their physical realizability or compatibility with the process; this second group comprises the inequality constraints. In addition, empirical relations, normally equalities, may be substituted for or added to the constraints. Finally, definitions are often made to simplify the process-model statements, and these comprise additional equality constraints when they cannot be explicitly solved for the defined variable. Figure 3.3-1 illustrates the relationships among the parts of the nonlinear programming problem and nonlinear programming itself.

Each equality constraint absorbs one degree of freedom in the process model and results in one dependent variable being generated. It is usually assumed that the analyst prepares the process-model statements carefully enought so that the equalities are independent, for if by omission or error he includes two redundant or otherwise dependent equations, then the apparent number of degrees of freedom will be different from the actual number. The true residual number of degrees of freedom should correspond to the number of independent variables (often termed decision variables) in the nonlinear programming problem. The number of residual degrees of freedom is an important concept in any type of optimization subject to equality constraints, because if the number of variables equals the number of independent equality constraints, no optimization need take place-the values of all the variables can be determined directly from the simultaneous solution of the system of equality constraints, $hj(x) = 0$ $j = 1,\ldots,n$. If the number of variables exceeds the number of independent equality constraints, m, the only type of solution that can be obtained for a process model is to adjust the (n-m) decision variables until an objective function attains its optimum value. If the number of independent equality constraints exceeds the number of variables, optimization is also required, except that in this case the objective function must consist of some type of statistical criterion, such as the one used in least squares.

Because in nonlinear programming problems derived from physical processes it may not be convenient to substitute nonlinear equalities into the objective function and thus eliminate the dependent variables, one or more of the dependent variables in the problem may be treated as a decision variable along with the true independent variables. For example, in the problem

$$\text{Minimize:}\quad f(x) = x_1^2 - 3x_2^2 + x_1x_2 + x_1x_3 + x_3 + 6$$

$$\text{Subject to:}\quad h_1(x) = (x_1 - x_2 + 2)^3 - (\sqrt{x_2} + x_2 + 2)^2 = 0$$

$$h_2(x) = x_1 + x_2 + x_3 + 3 = 0$$

It is quite easy to solve constraint $h_2(x)$ for x_3, eliminate x_3 from $f(x)$ and $h_1(x)$, and eliminate constraint $h_2(x)$. But to solve $h_1(x)$ for either x_1 or x_2 so that one of these variables can be eliminated from the objective function is much more complicated. It usually proves easier, especially if the problem contains several nonlinear equality constraints, to retain $h_1(x)$ as an equality constraint and treat x as a vector containing two variables for numerical calculations. Nevertheless, the true residual degree of freedom is only one. Some algorithms do take advantage of the reduced number of degrees of freedom, taking into account the equality constraints and active inequality constraints by searching for the optimum in a reduced space.

On eliminating one variable in the objective function by substitution, it is necessary to be careful not to inadvertently change the domain of a variable. For example, if $f(x) = -x_1^2 + x_2$ and $h(x) = -x_1^2 - x_2^2 + 1 = 0$, x_2 must be limited to $-1 \leq x2 \leq 1$, because smaller or larger values of x_2 will yield imaginary values of x_1. Elimination of x_1 by substituting $x_1^2 = 1 - x_2^2$ gives $f(x_2) = f(x) = x_2 + x_2^2 - 1$, and apparently x_2 is unbounded, whereas in fact the constraints $-1 \leq x_2 \leq 1$ must be added to $f(x_2)$ to render the problem unchanged.

As an illustration of the residual degrees of freedom for $n > m$, consider the problem of finding the volume and dimensions of the largest box when length and girth combined cannot exceed 72 in. (The answer is V = 24 x 12 x 12 = 3456 in.^{3}). The problem can be formulated as a maximization problem in which the performance criterion is the volume of the box.

$$\text{Maximize} : f(x) = x_1 \, x_2 \, x_3 \qquad x \in E^3$$

subject to the conditions imposed on the dimensions plus the implicit non-negativity conditions on the dimensions

$$g_1(x) \quad : 72 - x_1 - 2x_2 - 2x_3 \geq 0$$

$$g_{j+1}(x) : x_j \geq 0 \quad j = 1,\ldots.3$$

if one forgets to include the implicit constraints in a problem, a valid mathematical solution may be obtained that bears no correspondence to reality.

Since there are no equality constraints involved, the problem has three residual degrees of freedom, which means that all three variables can be varied independently as long as the inequality constraints are not violated. If an equality constraint is incorporated into the problem, such as to require that the height and width of the box be equal, that is, $h_1(x) = x_2 - x_3 = \emptyset$ then the problem will have only two residual degrees of freedom. Clearly, in this simple case, one variable, either x_2 or x_3 may be eliminated from the objective function, and the problem becomes one of maximizing the volume by adjusting two independent variables only. However, in more realistic problems, elimination of one variable by substitution into the objective function may not prove feasible. Which specific variables become decision variables and which become dependent variables is somewhat arbitrary, but an analyst will have in mind certain natural decision variables associated with any process model, either the controllable variables or certain mathematically convenient variables.

3.4 NOTATION AND TERMINOLOGY

To outline certain aspects of the terminology to be used, a few preliminary remarks are made in this section concerning some of the terms which occur rather frequently in the literature of optimization as well as in this thesis.

3.4.1 Optimal solutions

The column vector $x^* = [x^*_1, \ldots, x^*_n]^T$ which satisfies the expressions (3.2-1), (3.2-2), and (3.2-3), is called the optimal point, and the corresponding value of $f(x^*)$ is termed the optimal value of the objective function. The pair x^* and $f(x^*)$ constitutes an optimal solution. Various categories of optimal solutions can exist if the objective function is not unimodal, i.e., has one extremum, as illustrated by the multimodal function in Fig. 3.4-1. A global optimal solution represents the smallest value of $f(x)$, whereas a local (or relative) optimal solution represents the smallest value of $f(x)$ in the vicinity of some x vector. For both the global and local minimum.

$$f(x^*) \leq f(x)$$

but the global optimal solution refers to all x in E^n, whereas the local optimal solution refers only to a small region where $\|x - x^*\| < \delta$

. If the precision of the optimal solution is to be taken into account, a more exact condition for the optimal solutions is

$$f(x^*) \leq f(x) - \gamma$$

where γ is some small number.

In practice, the prospect of a local extremum being the global extremum can be tested by using a number of starting vectors, but even if only one local solution is found, it cannot be demonstrated in general that the solution is assuredly the global optimum. Fortunately, for problems based on real processes, the objective function usually is a well-behaved function with a single extremum. Therefore, for most practical purposes, the use of numerical procedures that provide a local solution to the programming problem is not a great disadvantage.

3.4.2 Concavity and convexity

The concepts of concavity and convexity assist in determining under what conditions a local optimal solution is also the global optimal solution, a matter of some concern in view of what has been said above concerning multiple optima.

A function $\emptyset(x)$ is called convex over the domain of R if for any two vectors, x_1 and x_2 R,

$$\emptyset\ (\theta x_1 + (1-\theta x_2) \leq \theta\emptyset(x_1) + (1-\theta)\ \emptyset\ (x_2) \qquad (3.4\text{-}1)$$

where θ is a scalar with the range $0 \leq \theta \leq 1$. The function is strictly convex if, for $x_1 \neq x_2$, the sign of (3.4-1) may be replaced with the inequality (<) sign. (A vector inequality $x \geq y$ means $x_i > y_i$ for each element; for $x > y$,$x_i > y_i$, for all i.). Figure 3.4-2 illustrates geometrically a strictly convex function of one independent variable; a convex function cannot have any value larger than the value of the function obtained by linear interpolation between $\emptyset(x_1)$ and $\emptyset(x_2)$. If the reverse inequality of (3.4-1) holds the function is said to be concave. Thus a function $\emptyset(x)$ is concave (strictly concave) if $-\emptyset(x)$ is convex (strictly convex). Linear functions are both convex and concave. A differentiable convex function has the following properties:

(a) $\emptyset(x_2) - \emptyset(x_1) \geq \nabla^T \emptyset(x_1)(x_2) - x_1)$ for all x_1, x_2

(b) The matrix of the second partial derivatives of $\emptyset(x)$ with respect to x (the hessian matrix) is positive definite (or positive semidefinite) for all x if $\emptyset(x)$ is strictly convex (or concave)

(c) Over the domain of R, $\emptyset(x)$ has only one extremum.

The requirement that a function be unimodal is much weaker than the requirement that it be convex or concave, for as Fig. 3.4-3a indicates, unimodality demands neither continuity nor the existence of a unique derivative.

A set of points (or region) is defined as a convex set in n-dimensional space if, for all pairs of two points x_1 and x_2 in the set, the straight-line segment joining them is also entirely in the set. Thus every point x where

$$x = \theta x_1 + (1-\theta) x_2 \quad 0 \leq \theta \leq 1$$

is also in the set. Figure 3.4-4 illustrates a convex and nonconvex set in the two-dimensional space. As an example, the following general set of expressions form a convex region (possibly empty or unbounded):

$$g_j(x) \leq b_j$$

$$a_j^T x = b_j$$

if the $g_j(x)$ are all convex, as illustrated in Fig. 3.4-5.

An important result in mathematical programming evolves from the concepts of convexity. For the nonlinear programming problem known as the convex programming problem,

Minimize : $f(x)$

Subject to : $g_j(x) \geq 0 \quad j = 1, \ldots, p$

$x \geq 0$

in which (1) f(x) is a convex function and (2) each inequality constraint is a concave function (the constraints form a convex set), the following property can be shown to be true: The local minimum is also the global minimum. Analogously, a local maximum is the global maximum if the objective function is concave and the constraints form a convex set.

3.4-3 Feasibility

Any vector x that satisfies both the equality and inequality constraints is called a feasible point or vector. The set of all points which satisfy the constraints constitutes the feasible domain of f(x) and will be represented by R: any point not in R is termed nonfeasible. A constrained optimum is one in which the local optimum lies on the boundary of the feasible region. If the constraints consist only of equalities, a feasible x vector must be on the intersection of all the hypersurfaces corresponding to $h_j(x) = 0$ In relation to just the inequality constraints, a point x may be classified either as an interior point (a feasible point), a boundary point (a feasible point), or an exterior point (a nonfeasible point). An interior point is one for which all the $g_j(x) > 0$; a boundary point satisfies at least one $g_j(x) = 0$ and an exterior point causes at least one $g_j(x) < 0$. The set of points for which the $g_j(x) = 0$, $j = 1,\ldots,p$ defines the boundary surfaces of the inequality constraint set. An active or binding inequality contraint is one for which $g_j(x) = 0$. The region of admissible values of the variables may be simply connected as in Fig. 3.4-6a or nonsimply connected as in Fig.3.4-6b, in which case the nonlinear programming algorithm is likely to miss searching more than one or two feasible regions unless a large number of starting vectors are employed. Fortunately, most nonlinear programming problems based on real processes are formulated so that only a simply connected feasible region exists.

3.4-4 The Gradient

The set of points for which the objective function has a constant value is called a contour of f(x). A few such contours are illustrated in Fig. 3.4-7. The gradient of the objective function f(x) is defined as the column vector of the first partial derivatives of f(x) with respect to x evaluated at some point x. A superscript k, k = 0, 1,..., will be used to denote the point in E^n at which the gradient is evaluated, so that the gradient at $x^{(k)}$ is

$$\nabla f(x^{(k)}) = \begin{vmatrix} \dfrac{\partial f(x^{(k)})}{\partial x_1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \dfrac{\partial f(x^{(k)})}{\partial x_n} \end{vmatrix}$$

The expression $f(x(k))$ will denote a row vector. Most books on vectors and matrices explain how the gradient of a scalar function points in the direction of the greatest increase in the value of the function, that is, in the direction of steepest ascent, and is orthogonal to that contour of $f(x)$ that passes through $x^{(k)}$. The negative of the gradient is in the direction of steepest descent. Any vector v orthogonal to $f(x^{(k)})$, such as the tangent surface to $f(x^{(k)})$ at $x^{(k)}$, can be denoted by $v^T \quad f(x^k) = 0$.

3.5 THE NECESSARY AND SUFFICIENT CONDITIONS FOR A SOLUTION TO BE AN OPTIMAL SOLUTION

Considerable effort has been devoted in the study of nonlinear programming to delineating the necessary and sufficient conditions for an x vector to be a local extremum. Optimality criteria for certain special cases of the general nonlinear programming problem listed in expressions (3.2-1) through (3.2-3) have been formulated, but the structure of the nonlinear programming problem when general functions are involved is such that completely comprehensive optimality criteria have yet to be devised. Consequently, we can only describe certain special cases in this section, but they are of quite common occurrence and of practical importance. The conditions that determine whether or not an x vector solves a nonlinear programming problem will be stated in the form of theorems without the associated proofs.

3.5-1 Nonlinear Programming without Constraints

The problem is

$$\text{Minimize} \ : \ f(x) \text{ for } x \in E^n \qquad (3.5\text{-}1)$$

For an unconstrained nonlinear programming problem the necessary conditions for x* to be a local minimum of problem (3.5-1) are that

1. f(x) be differentiable at x*

2. $\nabla f(x) = 0$, i.e., a stationary point exists at x*. The sufficient conditions for x* to be a local minimum of problem (3.5-1) are, in addition to conditions 1 and 2 above,

3. $\nabla^2 f(x^*) > 0$; i.e., the Hessian matrix is possitive definite. (The corresponding conditions for a maximum are the same, except that the Hessian matrix of f(x*) must be negative definite.)

3.5.2 Nonlinear Programming with Both Equality and Inequality Constraints

The problem is

$$\begin{aligned} &\text{Minimize} &&: f(x) && x \in E^n \\ &\text{Subject to} &&: h_j(x) = 0 && j = 1,\dots, m \qquad (3.5\text{-}2) \\ & && \ \ g_j(x) \geq 0 && j = m + 1,\dots p \end{aligned}$$

We commence with the following concept; if x* is a local minimum, f(x) cannot decrase along any smooth arc directed from x* into the feasible region. Let the vector v be tangent to the arc leading from x*. We partition the nonzero vectors v into three classes, each set V_i comprising the set of v such that:

Class	Inequality constraints		Equality Constraints		Objective Function
V_1	[$v^T\nabla g_j(x^*) \geq 0$ for active constraints]	and	[$v^T\nabla h_j(x^*) = 0$ for all $j = 1,..m$]	and	$[v^T\nabla f(x^*) \geq 0]$
V_2	[$v^T\nabla g_j(x^*) \geq 0$ for active constraints]	and	[$v^T\nabla h_j(x^*) = 0$ for all $j = 1,..m$]	and	$[v^T\nabla f(x^*) < 0]$
V_3	[$v^T \nabla g_j (X^*) < 0$ for at least one active constraint]	or	[$v^T \nabla h_j (X^*) \neq 0$ for at least one constraint]		

All feasible perturbations of x^* fall into the union of V_1 and V_2, and if v is contained in V_2, f(x) decreases. While if v is contained in V_1, f(x) increases or is constant. In essence the first-order necessary conditions pose the requirement that the set V_2 be empty.

More detall of nonlinear optimization may be found elsewhere (23)

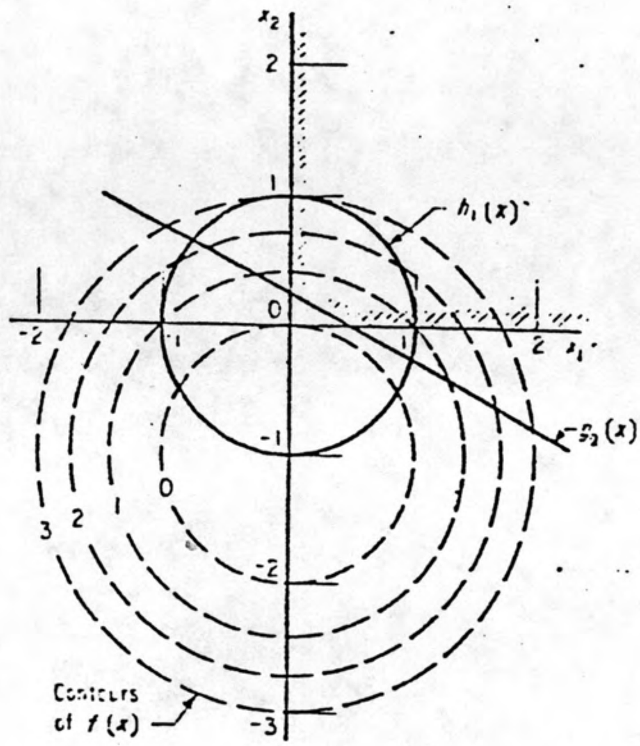

Fig 3.2-1 Geometric representation of the functions in a nonlinear programming problem.

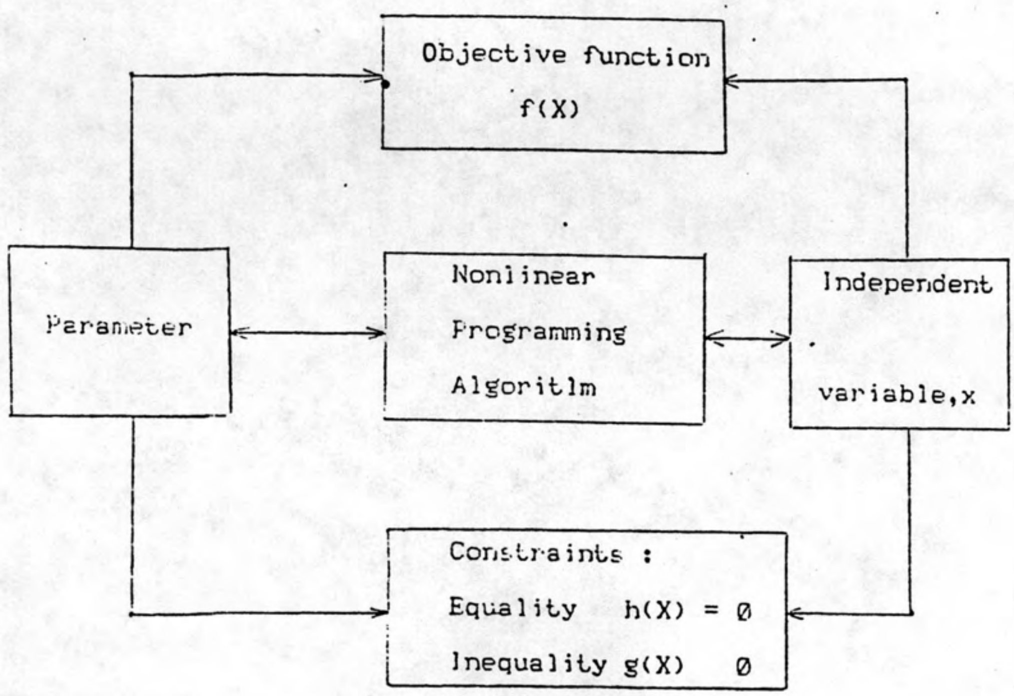

Fig 3.3-1 Relationship in the nonlinear programming problem and nonlinear programming itself.

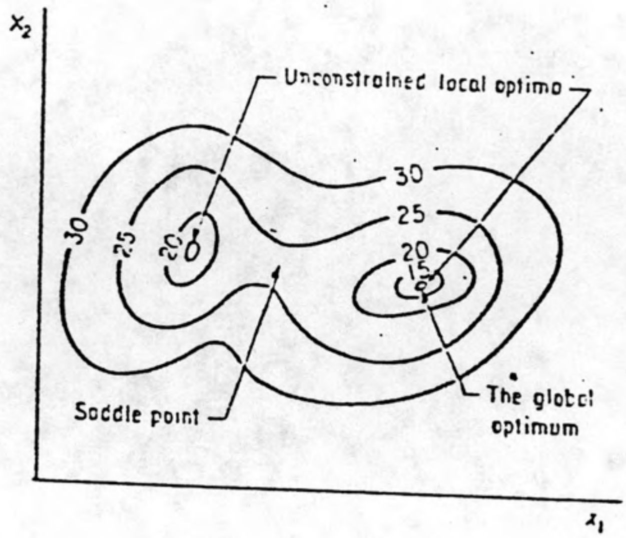

Fig 3.4-1 Classification of optimal solution.

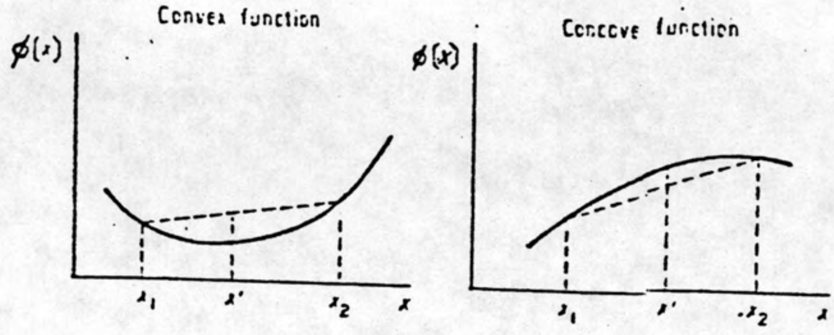

Fig 3.4-2 Convex and concave function (in a given range of x).

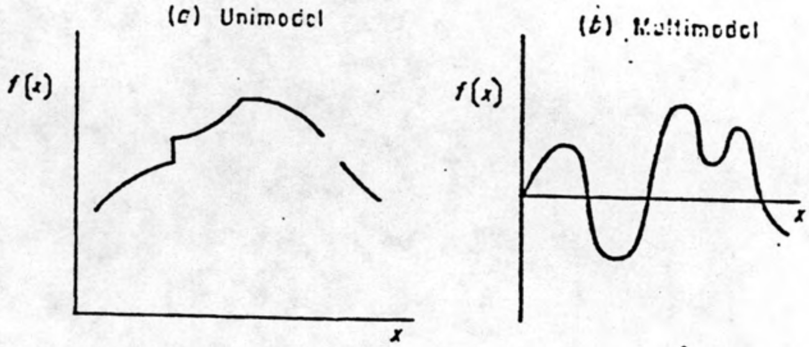

Fig 3-4-3 Unimodal and multimodal functions.

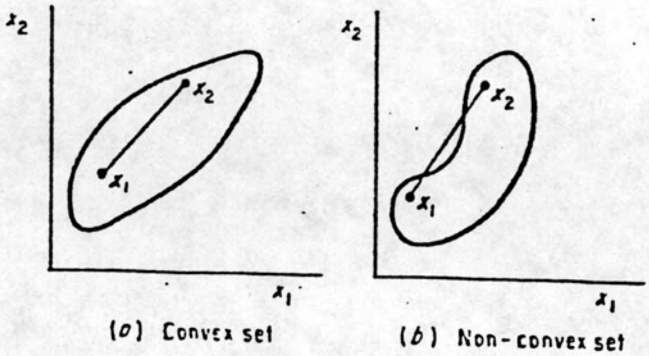

Fig 3-4-4 Convex and nonconvex sets.

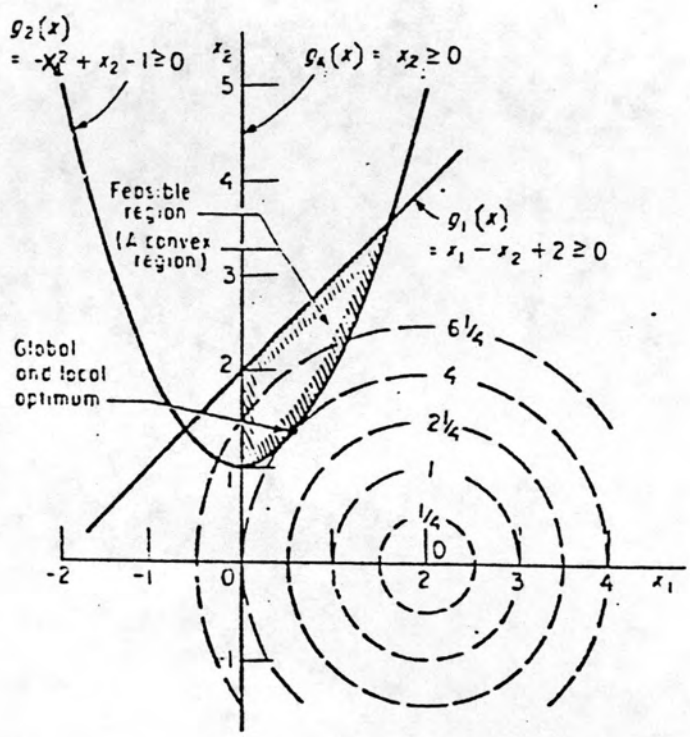

Fig 3-4-5 A convex programming problem illustrating the feasible region (comprising a convex set) and the global optimum.

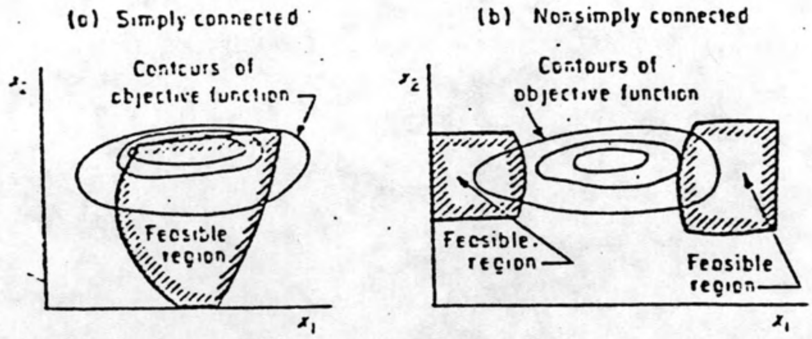

Fig 3-4-6 Example of simply connected and nonsimply connected feasible regions (as pertaining to inequality constraints).

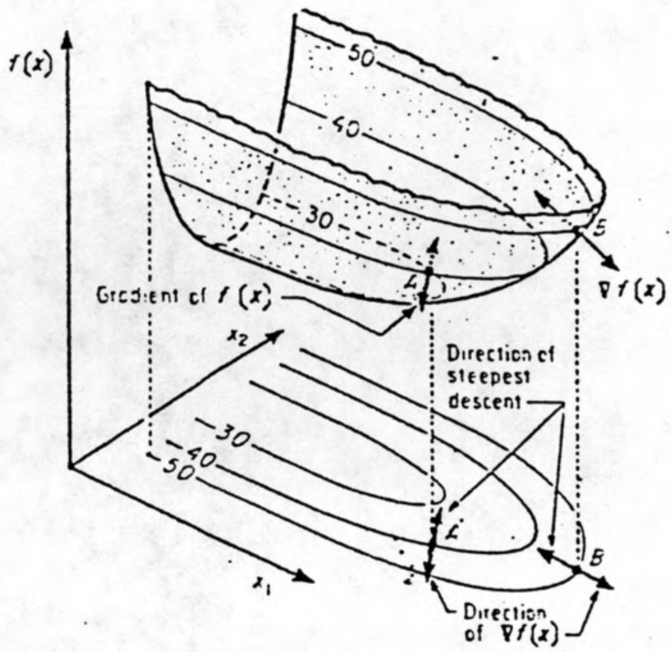

Fig 3.4-7 The gradient (direction of steepest ascent) and the direction of steepest descent at two points.