Nonlinear Programming

13

Numerous mathematical-programming applications, including many introduced in previous chapters, are cast naturally as linear programs. Linear programming assumptions or approximations may also lead to appropriate problem representations over the range of decision variables being considered. At other times, though, nonlinearities in the form of either nonlinear objective functions or nonlinear constraints are crucial for representing an application properly as a mathematical program. This chapter provides an initial step toward coping with such nonlinearities, first by introducing several characteristics of nonlinear programs and then by treating problems that can be solved using simplex-like pivoting procedures. As a consequence, the techniques to be discussed are primarily algebra-based. The final two sections comment on some techniques that do not involve pivoting.

As our discussion of nonlinear programming unfolds, the reader is urged to reflect upon the linear-programming theory that we have developed previously, contrasting the two theories to understand why the nonlinear problems are intrinsically more difficult to solve. At the same time, we should try to understand the similarities between the two theories, particularly since the nonlinear results often are motivated by, and are direct extensions of, their linear analogs. The similarities will be particularly visible for the material of this chapter where simplex-like techniques predominate.

13.1 NONLINEAR PROGRAMMING PROBLEMS

A general optimization problem is to select n decision variables $x_1, x_2, \ldots, x_n$ from a given feasible region in such a way as to optimize (minimize or maximize) a given objective function

$$f(x_1, x_2, \ldots, x_n)$$

of the decision variables. The problem is called a *nonlinear programming problem* (NLP) if the objective function is nonlinear and/or the feasible region is determined by nonlinear constraints. Thus, in maximization form, the general nonlinear program is stated as:

$$\text{Maximize } f(x_1, x_2, \ldots, x_n),$$

subject to:

$$\begin{aligned} g_1(x_1, x_2, \ldots, x_n) &\leqq b_1, \\ &\vdots \\ g_m(x_1, x_2, \ldots, x_n) &\leqq b_m, \end{aligned}$$

where each of the constraint functions g_1 through g_m is given. A special case is the linear program that has been treated previously. The obvious association for this case is

$$f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} c_j x_j,$$

and

$$g_i(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} a_{ij} x_j \qquad (i = 1, 2, \ldots, m).$$

Note that nonnegativity restrictions on variables can be included simply by appending the additional constraints:

$$g_{m+i}(x_1, x_2, \ldots, x_n) = -x_i \leqq 0 \qquad (i = 1, 2, \ldots, n).$$

Sometimes these constraints will be treated explicitly, just like any other problem constraints. At other times, it will be convenient to consider them implicitly in the same way that nonnegativity constraints are handled implicitly in the simplex method.

For notational convenience, we usually let x denote the vector of n decision variables $x_1, x_2, \ldots, x_n$ — that is, $x = (x_1, x_2, \ldots, x_n)$ — and write the problem more concisely as

$$\text{Maximize } f(x),$$

subject to:

$$g_i(x) \leqq b_i \qquad (i = 1, 2, \ldots, m).$$

As in linear programming, we are not restricted to this formulation. To minimize $f(x)$, we can of course maximize $-f(x)$. Equality constraints $h(x) = b$ can be written as two inequality constraints $h(x) \leqq b$ and $-h(x) \leqq -b$. In addition, if we introduce a slack variable, each inequality constraint is transformed to an equality constraint.

Thus sometimes we will consider an alternative equality form:

Maximize $f(x)$,

subject to:

$$h_i(x) = b_i \qquad (i = 1, 2, \ldots, m)$$

$$x_j \geqq 0 \qquad (j = 1, 2, \ldots, n).$$

Usually the problem context suggests either an equality or inequality formulation (or a formulation with both types of constraints), and we will not wish to force the problem into either form.

The following three simplified examples illustrate how nonlinear programs can arise in practice.

Portfolio Selection An investor has \$5000 and two potential investments. Let x_j for $j = 1$ and $j = 2$ denote his allocation to investment j in thousands of dollars. From historical data, investments 1 and 2 have an expected annual return of 20 and 16 percent, respectively. Also, the total risk involved with investments 1 and 2, as measured by the variance of total return, is given by $2x_1^2 + x_2^2 + (x_1 + x_2)^2$, so that risk increases with total investment and with the amount of each individual investment. The investor would like to maximize his expected return and at the same time minimize his risk. Clearly, both of these objectives cannot, in general, be satisfied simultaneously. There are several possible approaches. For example, he can minimize risk subject to a constraint imposing a lower bound on expected return. Alternatively, expected return and risk can be combined in an objective function, to give the model:

$$\text{Maximize } f(x) = 20x_1 + 16x_2 - \theta[2x_1^2 + x_2^2 + (x_1 + x_2)^2],$$

subject to:

$$g_1(x) = x_1 + x_2 \leqq 5,$$

$$x_1 \geqq 0, \quad x_2 \geqq 0, \qquad (\text{that is, } g_2(x) = -x_1, \quad g_3(x) = -x_2).$$

The nonnegative constant θ reflects his tradeoff between risk and return. If $\theta = 0$, the model is a linear program, and he will invest completely in the investment with greatest expected return. For very large θ, the objective contribution due to expected return becomes negligible and he is essentially minimizing his risk.

Water Resources Planning In regional water planning, sources emitting pollutants might be required to remove waste from the water system. Let x_j be the pounds of Biological Oxygen Demand (an often-used measure of pollution) to be removed at source j.

One model might be to minimize total costs to the region to meet specified pollution standards:

$$\text{Minimize} \sum_{j=1}^{n} f_j(x_j),$$

subject to:

$$\sum_{j=1}^{n} a_{ij}x_j \geqq b_i \qquad (i = 1, 2, \ldots, m)$$

$$0 \leqq x_j \leqq u_j \qquad (j = 1, 2, \ldots, n),$$

where

$f_j(x_j)$ = Cost of removing x_j pounds of Biological Oxygen Demand at source j,

b_i = Minimum desired improvement in water quality at point i in the system,

a_{ij} = Quality response, at point i in the water system, caused by removing one pound of Biological Oxygen Demand at source j,

u_j = Maximum pounds of Biological Oxygen Demand that can be removed at source j.

Constrained Regression A university wishes to assess the job placements of its graduates. For simplicity, it assumes that each graduate accepts either a government, industrial, or academic position. Let

$$N_j = \text{Number of graduates in year } j \qquad (j = 1, 2, \ldots, n),$$

and let G_j, I_j, and A_j denote the number entering government, industry, and academia, respectively, in year j $(G_j + I_j + A_j = N_j)$.

One model being considered assumes that a given fraction of the student population joins each job category each year. If these fractions are denoted as λ_1, λ_2, and λ_3, then the predicted number entering the job categories in year j is given by the expressions

$$\hat{G}_j = \lambda_1 N_j,$$

$$\hat{I}_j = \lambda_2 N_j,$$

$$\hat{A}_j = \lambda_3 N_j.$$

A reasonable performance measure of the model's validity might be the difference between the actual number of graduates G_j, I_j, and A_j entering the three job categories and the predicted numbers $\hat{G}_j$, $\hat{I}_j$, and $\hat{A}_j$, as in the least-squares estimate:

$$\text{Minimize} \sum_{j=1}^{n} [(G_j - \hat{G}_j)^2 + (I_j - \hat{I}_j)^2 + (A_j - \hat{A}_j)^2],$$

subject to the constraint that all graduates are employed in one of the professions. In terms of the fractions entering each profession, the model can be written as:

$$\text{Minimize} \sum_{j=1}^{n} [(G_j - \lambda_1 N_j)^2 + (I_j - \lambda_2 N_j)^2 + (A_j - \lambda_3 N_j)^2],$$

subject to:

$$\lambda_1 + \lambda_2 + \lambda_3 = 1,$$

$$\lambda_1 \geqq 0, \quad \lambda_2 \geqq 0, \quad \lambda_3 \geqq 0.$$

This is a nonlinear program in three variables λ_1, λ_2, and λ_3.

There are alternative ways to approach this problem. For example, the objective function can be changed to:

$$\text{Minimize} \sum_{j=1}^{n} \left[|G_j - \hat{G}_j| + |I_j - \hat{I}_j| + |A_j - \hat{A}_j|\right].^{\dagger}$$

This formulation is appealing since the problem now can be transformed into a linear program. Exercise 28 (see also Exercise 20) from Chapter 1 illustrates this transformation.

The range of nonlinear-programming applications is practically unlimited. For example, it is usually simple to give a nonlinear extension to any linear program. Moreover, the constraint $x = 0$ or 1 can be modeled as $x(1 - x) = 0$ and the constraint x integer as $\sin(\pi x) = 0$. Consequently, *in theory* any application of integer programming can be modeled as a nonlinear program. We should not be overly optimistic about these formulations, however; later we shall explain why nonlinear programming is not attractive for solving these problems.

13.2 LOCAL vs. GLOBAL OPTIMUM

Geometrically, nonlinear programs can behave much differently from linear programs, even for problems with linear constraints. In Fig. 13.1, the portfolio-selection example from the last section has been plotted for several values of the tradeoff parameter θ. For each fixed value of θ, contours of constant objective values are concentric ellipses. As Fig. 13.1 shows, the optimal solution can occur:

a) at an interior point of the feasible region;

b) on the boundary of the feasible region, which is not an extreme point; or

c) at an extreme point of the feasible region.

As a consequence, procedures, such as the simplex method, that search only extreme points may not determine an optimal solution.

† $|\ |$ denotes absolute value; that is, $|x| = x$ if $x \geqq 0$ and $|x| = -x$ if $x < 0$.

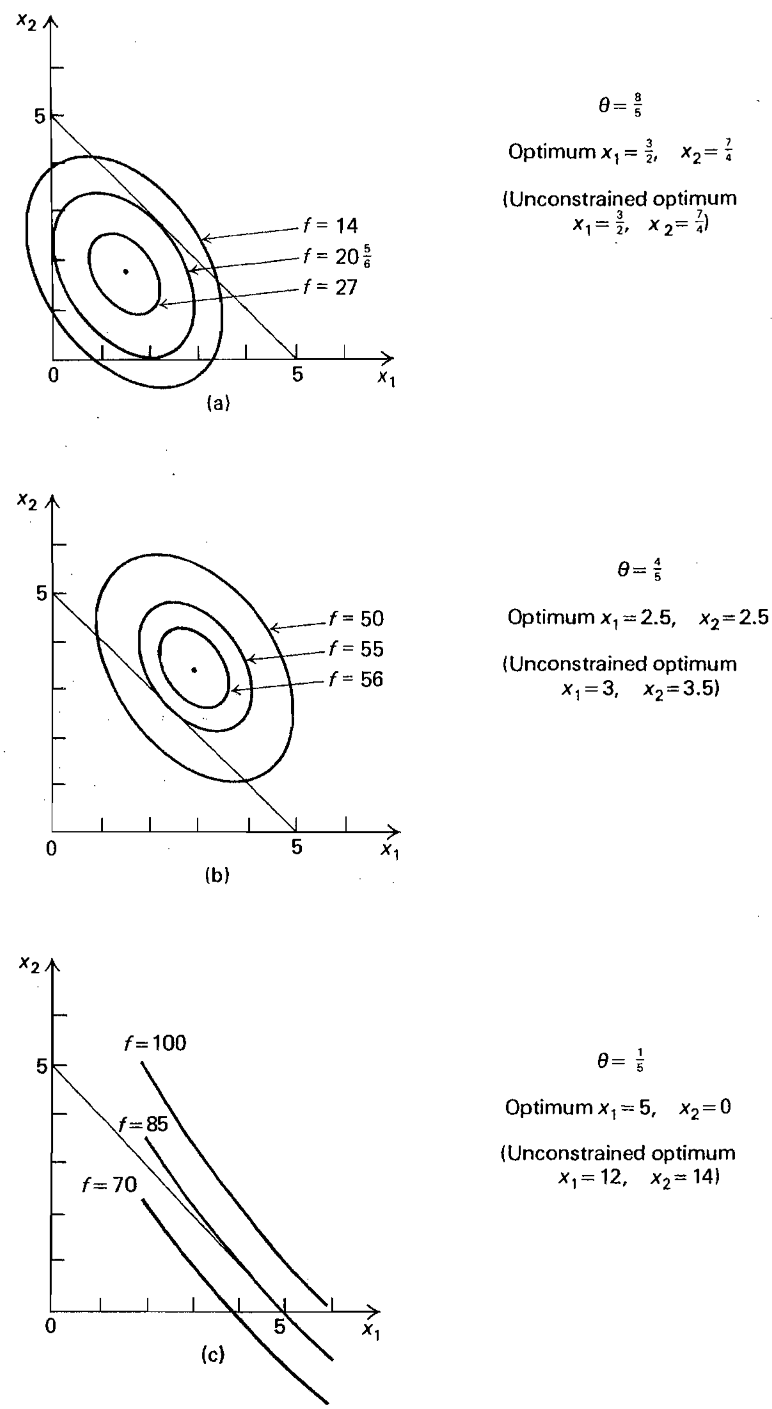

Fig. 13.1 Portfolio-selection example for various values of θ. (Lines are contours of constant objective values.)

Figure 13.2 illustrates another feature of nonlinear-programming problems. Suppose that we are to minimize $f(x)$ in this example, with $0 \leqq x \leqq 10$. The point $x = 7$ is optimal. Note, however, that in the indicated dashed interval, the point $x = 0$ is the best feasible point; i.e., it is an optimal feasible point in the local vicinity of $x = 0$ specified by the dashed interval.

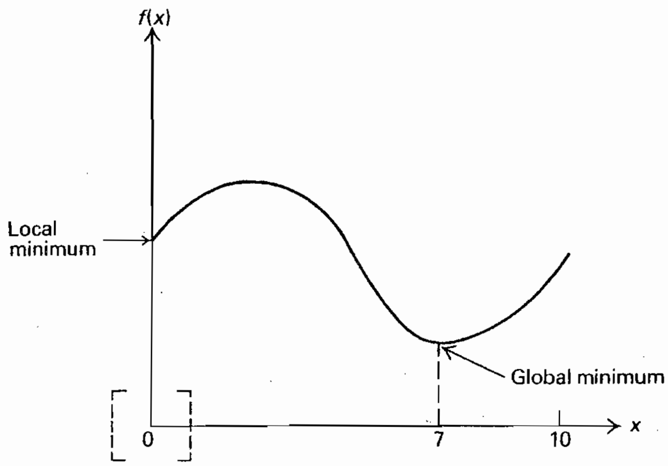

Fig. 13.2 Local and global minima.

The latter example illustrates that a solution optimal in a local sense need not be optimal for the overall problem. Two types of solution must be distinguished. A global optimum is a solution to the overall optimization problem. Its objective value is as good as any other point in the feasible region. A local optimum, on the other hand, is optimum only with respect to feasible solutions close to that point. Points far removed from a local optimum play no role in its definition and may actually be preferred to the local optimum. Stated more formally,

Definition. Let $x = (x_1, x_2, \ldots, x_n)$ be a feasible solution to a maximization problem with objective function $f(x)$. We call x

1. A *global maximum* if $f(x) \geqq f(y)$ for every feasible point $y = (y_1, y_2, \ldots, y_n)$;
2. A *local maximum* if $f(x) \geqq f(y)$ for every feasible point $y = (y_1, y_2, \ldots, y_n)$ sufficiently close to x. That is, if there is a number $\epsilon > 0$ (possibly quite small) so that, whenever each variable y_j is within ϵ of x_j — that is, $x_j - \epsilon \leqq y_j \leqq x_j + \epsilon$ — and y is feasible, then $f(x) \geqq f(y)$.

Global and local minima are defined analogously. The definition of local maximum simply says that if we place an n-dimensional box (e.g., a cube in three dimensions) about x, whose side has length 2ϵ, then $f(x)$ is as small as $f(y)$ for every feasible point y lying within the box. (Equivalently, we can use n-dimensional spheres in this definition.) For instance, if $\epsilon = 1$ in the above example, the one-dimensional box, or interval, is pictured about the local minimum $x = 0$ in Fig. 13.2.

The concept of a local maximum is extremely important. As we shall see, most general-purpose nonlinear-programming procedures are near-sighted and can do no better than determine local maxima. We should point out that, since every global maximum is also a local maximum, the overall optimization problem can be viewed as seeking the best local maxima.

Under certain circumstances, local maxima and minima are known to be global. Whenever a function "curves upward" as in Fig. 13.3(a), local minima will be global. These functions are called *convex*. Whenever a function "curves downward" as in Fig. 13.3(b) a local maximum will be a global maximum. These functions are called *concave*.† For this reason we usually wish to minimize convex functions and maximize concave functions. These observations are formalized below.

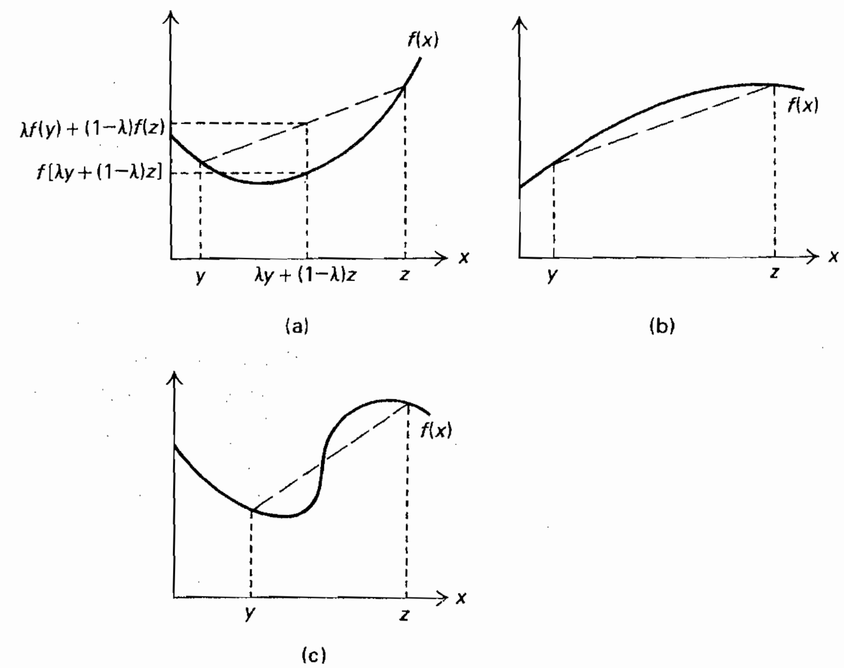

Fig. 13.3 a) Convex function; b) concave function; c) nonconvex, nonconcave function.

13.3 CONVEX AND CONCAVE FUNCTIONS

Because of both their pivotal role in model formulation and their convenient mathematical properties, certain functional forms predominate in mathematical programming. Linear functions are by far the most important. Next in importance are functions which are convex or concave. These functions are so central to the theory that we take some time here to introduce a few of their basic properties.

† As a mnemonic, the "A" in concAve reflects the shape of these functions.

An essential assumption in a linear-programming model for profit maximization is constant returns to scale for each activity. This assumption implies that if the level of one activity doubles, then that activity's profit contribution also doubles; if the first activity level changes from x_1 to $2x_1$, then profit increases proportionally from say \$20 to \$40 [i.e., from $c_1 x_1$ to $c_1(2x_1)$]. In many instances, it is realistic to assume constant returns to scale over the range of the data. At other times, though, due to economies of scale, profit might increase disproportionately, to say \$45; or, due to diseconomies of scale (saturation effects), profit may be only \$35. In the former case, marginal returns are increasing with the activity level, and we say that the profit function is *convex* (Fig. 13.3(a)). In the second case, marginal returns are decreasing with the activity level and we say that the profit function is *concave* (Fig. 13.3(b)). Of course, marginal returns may increase over parts of the data range and decrease elsewhere, giving functions that are neither convex nor concave (Fig. 13.3(c)).

An alternative way to view a convex function is to note that linear interpolation overestimates its values. That is, for any points y and z, the line segment joining $f(y)$ and $f(z)$ lies above the function (see Fig. 13.3). More intuitively, convex functions are "bathtub like" and hold water. Algebraically,

Definition. A function $f(x)$ is called *convex* if, for every y and z and every $0 \leqq \lambda \leqq 1$,

$$f[\lambda y + (1 - \lambda)z] \leqq \lambda f(y) + (1 - \lambda)f(z).$$

It is called *strictly convex* if, for every two distinct points y and z and every $0 < \lambda < 1$,

$$f[\lambda y + (1 - \lambda)z] < \lambda f(y) + (1 - \lambda)f(z).$$

The lefthand side in this definition is the function evaluation on the line joining x and y; the righthand side is the linear interpolation. Strict convexity corresponds to profit functions whose marginal returns are strictly increasing.

Note that although we have pictured f above to be a function of one decision variable, this is not a restriction. If $y = (y_1, y_2, \ldots, y_n)$ and $z = (z_1, z_2, \ldots, z_n)$, we must interpret $\lambda y + (1 - \lambda)z$ only as weighting the decision variables one at a time, i.e., as the decision vector $(\lambda y_1 + (1 - \lambda)z_1, \ldots, \lambda y_n + (1 - \lambda)z_n)$.

Concave functions are simply the negative of convex functions. In this case, linear interpolation underestimates the function. The definition above is altered by reversing the direction of the inequality. Strict concavity is defined analogously. Formally,

Definition. A function $f(x)$ is called *concave* if, for every y and z and every $0 \leqq \lambda \leqq 1$,

$$f[\lambda y + (1 - \lambda)z] \geqq \lambda f(y) + (1 - \lambda)f(z).$$

It is called *strictly concave* if, for every y and z and every $0 < \lambda < 1$,

$$f[\lambda y + (1 - \lambda)z] > \lambda f(y) + (1 - \lambda)f(z).$$

We can easily show that a linear function is both convex and concave. Consider the linear function:

$$f(x) = \sum_{j=1}^{n} c_j x_j,$$

and let $0 \leqq \lambda \leqq 1$. Then

$$\begin{aligned} f(\lambda y + (1 - \lambda)z) &= \sum_{j=1}^{n} c_j(\lambda y_j + (1 - \lambda)z_j) \\ &= \lambda\left[\sum_{j=1}^{n} c_j y_j\right] + (1 - \lambda)\left[\sum_{j=1}^{n} c_j z_j\right] \\ &= \lambda f(y) + (1 - \lambda)f(z). \end{aligned}$$

These manipulations state, quite naturally, that linear interpolation gives exact values for f and consequently, from the definitions, that a linear function is both convex and concave. This property is essential, permitting us to either maximize or minimize linear functions by computationally attractive methods such as the simplex method for linear programming.

Other examples of convex functions are x^2, x^4, e^x, e^{-x} or $-\log x$. Multiplying each example by minus one gives a concave function. The definition of convexity implies that the sum of convex functions is convex and that any nonnegative multiple of a convex function also is convex. Utilizing this fact, we can obtain a large number of convex functions met frequently in practice by combining these simple examples, giving, for instance,

$$2x^2 + e^x, \qquad e^x + 4x,$$

or

$$-3 \log x + x^4.$$

Similarly, we can easily write several concave functions by multiplying these examples by minus one.

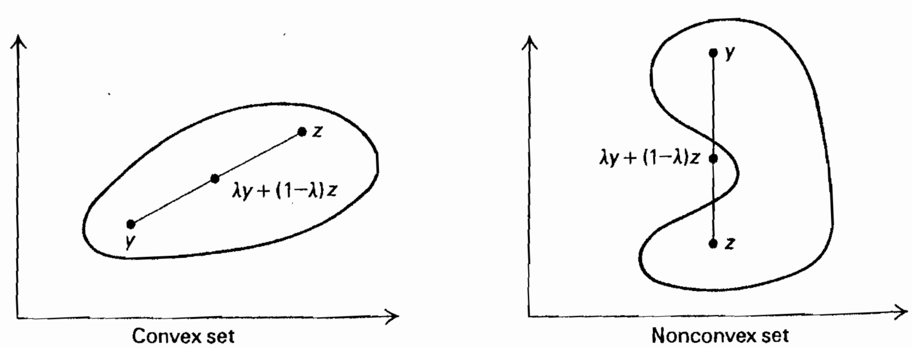

Fig. 13.4 Convex and nonconvex sets.

A notion intimately related to convex and concave functions is that of a *convex set*. These sets are "fat," in the sense that, whenever y and z are contained in the set, every point on the line segment joining these points is also in the set (see Fig. 13.4). Formally,

Definition. A set of points C is called *convex* if, for all λ in the interval $0 \leqq \lambda \leqq 1$, $\lambda y + (1 - \lambda)z$ is contained in C whenever x and y are contained in C.

Again we emphasize that y and z in this definition are decision vectors; in the example, each of these vectors has two components.

We have encountered convex sets frequently before, since the feasible region for a linear program is convex. In fact, the feasible region for a nonlinear program is convex if it is specified by less-than-or-equal-to equalities with convex functions. That is, if $f_i(x)$ for $i = 1, 2, \ldots, m$, are convex functions and if the points $x = y$ and $x = z$ satisfy the inequalities

$$f_i(x) \leqq b_i \qquad (i = 1, 2, \ldots, m),$$

then, for any $0 \leqq \lambda \leqq 1$, $\lambda y + (1 - \lambda)z$ is feasible also, since the inequalities

$$f_i(\lambda y + (1 - \lambda)z) \underset{\substack{\uparrow \\ \text{Convexity}}}{\leqq} \lambda f_i(y) + (1 - \lambda)f_i(z) \underset{\substack{\uparrow \\ \text{Feasibility of } y \text{ and } z}}{\leqq} \lambda b_i + (1 - \lambda)b_i = b_i$$

hold for every constraint. Similarly, if the constraints are specified by greater-than-or-equal-to inequalities and the functions are concave, then the feasible region is convex. In sum, for convex feasible regions we want convex functions for less-than-or-equal-to constraints and concave functions for greater-than-or-equal-to constraints. Since linear functions are both convex and concave, they may be treated as equalities.

An elegant mathematical theory, which is beyond the scope of this chapter, has been developed for convex and concave functions and convex sets. Possibly the most important property for the purposes of nonlinear programming was previewed in the previous section. Under appropriate assumptions, a local optimal can be shown to be a global optimum.

Local Minimum and Local Maximum Property

1. A local $\begin{Bmatrix}\text{minimum}\\\text{maximum}\end{Bmatrix}$ of a $\begin{Bmatrix}\text{convex}\\\text{concave}\end{Bmatrix}$ function on a convex feasible region is also a global $\begin{Bmatrix}\text{minimum}\\\text{maximum}\end{Bmatrix}$.

2. A local $\begin{Bmatrix}\text{minimum}\\\text{maximum}\end{Bmatrix}$ of a strictly $\begin{Bmatrix}\text{convex}\\\text{concave}\end{Bmatrix}$ function on a convex feasible region is the unique global $\begin{Bmatrix}\text{minimum}\\\text{maximum}\end{Bmatrix}$.

We can establish this property easily by reference to Fig. 13.5. The argument is for convex functions; the concave case is handled similarly. Suppose that y is a local minimum. If y is not a global minimum, then, by definition, there is a feasible point z with $f(z) < f(y)$. But then if f is convex, the function must lie on or below the dashed linear interpolation line. Thus, in any box about y, there must be an x on the line segment joining y and z, with $f(x) < f(y)$. Since the feasible region is convex, this x is feasible and we have contradicted the hypothesis that y is a local minimum. Consequently, no such point z can exist and any local minimum such as y must be a global minimum.

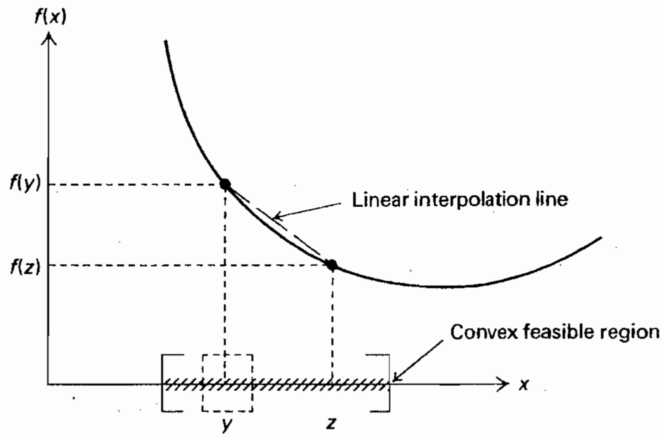

Fig. 13.5 Local minima are global minima for convex functions.

To see the second assertion, suppose that y is a local minimum. By Property 1 it is also a global minimum. If there is another global minimum z (so that $f(z) = f(y)$), then $\frac{1}{2}x + \frac{1}{2}z$ is feasible and, by the definition of strict convexity,

$$f(\tfrac{1}{2}x + \tfrac{1}{2}z) < \tfrac{1}{2}f(y) + \tfrac{1}{2}f(z) = f(y).$$

But this states that $\frac{1}{2}x + \frac{1}{2}z$ is preferred to y, contradicting our premise that y is a global minimum. Consequently, no other global minimum such as z can possibly exist; that is, y must be the unique global minimum.

13.4 PROBLEM CLASSIFICATION

Many of the nonlinear-programming solution procedures that have been developed do not solve the general problem

Maximize $f(x)$,

subject to:

$$g_i(x) \leq b_i \qquad (i = 1, 2, \ldots, m),$$