

4. Consider the problem

$$\text{Maximize } f(\mathbf{X})$$

subject to

$$g(\mathbf{X}) = 0$$

Given  $f(\mathbf{X})$  is concave and  $g_i(\mathbf{X})$  ( $i = 1, 2, \dots, m$ ) is a linear function, show that the KKT necessary conditions are also sufficient. Is this result true if  $g_i(\mathbf{X})$  is a convex nonlinear function for all  $i$ ? Why?

5. Consider the problem

$$\text{Maximize } f(\mathbf{X})$$

subject to

$$g_1(\mathbf{X}) \geq 0, g_2(\mathbf{X}) = 0, g_3(\mathbf{X}) \leq 0$$

Develop the KKT conditions, and give the stipulations under which the conditions are sufficient.

## BIBLIOGRAPHY

- Bazarra, M., H. Sherali, and C. Shetty, *Nonlinear Programming Theory and Algorithms*, 3rd ed., Wiley, New York, 2006.
- Beightler, C., D. Phillips, and D. Wilde, *Foundations of Optimization*, 2nd ed., Prentice Hall, NJ, 1979.
- Fletcher, R., *Practical Methods of Optimization*, 2nd ed., Wiley, New York, 2000.

## CHAPTER 21

## Nonlinear Programming Algorithms

資料來源:

Operations Research

An Introduction 9th ed

Hamdy A. Taha.

## 21.1 UNCONSTRAINED ALGORITHMS

This section presents two types of algorithms for the unconstrained problem: *direct search* and *gradient*.

## 21.1.1 Direct Search Method

Direct search methods apply primarily to strictly unimodal single-variable functions. Although the case may appear trivial, Section 21.1.2 shows that optimization of single-variable functions is key in the development of the more general multivariable algorithm.

The idea of direct search methods is to identify the **interval of uncertainty** known to include the optimum solution point. The procedure locates the optimum by iteratively narrowing the interval of uncertainty to a desired level of accuracy.

Two closely related search algorithms are presented in this section: **dichotomous** and **golden section**. Both algorithms seek the maximization of a unimodal function  $f(x)$  over the interval  $a \leq x \leq b$  that includes the optimum point  $x^*$ . The two methods start with the initial interval of uncertainty  $I_0 = (a, b)$ .

**General step  $i$ .** Let  $I_{i-1} = (x_L, x_R)$  be the current interval of uncertainty (at iteration 0,  $x_L = a$  and  $x_R = b$ ). The following table shows how  $x_1$  and  $x_2$  are determined:

Dichotomous method	Golden section method
$x_1 = \frac{1}{2}(x_R + x_L - \Delta)$	$x_1 = x_R - \left(\frac{\sqrt{5}-1}{2}\right)(x_R - x_L)$
$x_2 = \frac{1}{2}(x_R + x_L + \Delta)$	$x_2 = x_L + \left(\frac{\sqrt{5}-1}{2}\right)(x_R - x_L)$

The selection of  $x_1$  and  $x_2$  guarantees that  $x_L < x_1 < x_2 < x_R$ .

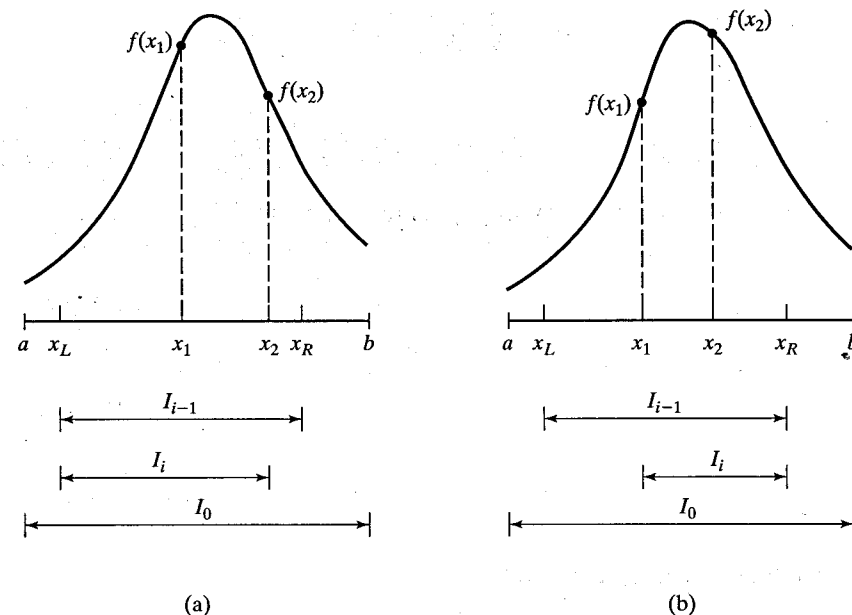


FIGURE 21.1

Illustration of the general step of the dichotomous/golden section search methods

The next interval of uncertainty,  $I_i$ , is determined in the following manner:

1. If  $f(x_1) > f(x_2)$ , then  $x_L < x^* < x_2$ . Let  $x_R = x_2$  and set  $I_i = (x_L, x_2)$  [see Figure 21.1(a)].
2. If  $f(x_1) < f(x_2)$ , then  $x_1 < x^* < x_R$ . Let  $x_L = x_1$  and set  $I_i = (x_1, x_R)$  [see Figure 21.1(b)].
3. If  $f(x_1) = f(x_2)$ , then  $x_1 < x^* < x_2$ . Let  $x_L = x_1$  and  $x_R = x_2$ ; set  $I_i = (x_1, x_2)$ .

The manner in which  $x_1$  and  $x_2$  are determined guarantees that  $I_{i+1} < I_i$ , as will be shown shortly. The algorithm terminates at iteration  $k$  if  $I_k \leq \Delta$ , where  $\Delta$  is a user-specified level of accuracy.

In the dichotomous method, the values  $x_1$  and  $x_2$  sit symmetrically around the midpoint of the current interval of uncertainty. This means that

$$I_{i+1} = .5(I_i + \Delta)$$

Repeated application of the algorithm guarantees that the length of the interval of uncertainty will approach the desired accuracy,  $\Delta$ .

In the golden section method, the idea is more involved. We notice that each iteration of the dichotomous method requires calculating the two values  $f(x_1)$  and  $f(x_2)$ , but ends up discarding one of them. What the golden section method proposes is to save computations by reusing the discarded value in the immediately succeeding iteration.

Define

$$\begin{cases} x_1 = x_R - \alpha(x_R - x_L) \\ x_2 = x_L + \alpha(x_R - x_L) \end{cases} \quad 0 < \alpha < 1$$

Then the interval of uncertainty  $I_i$  at iteration  $i$  equals  $(x_L, x_2)$  or  $(x_1, x_R)$ . Consider the case  $I_i = (x_L, x_2)$ , which means that  $x_1$  is included in  $I_i$ . In iteration  $i + 1$ , we select  $x_2$  equal to  $x_1$  in iteration  $i$ , which leads to the following equation:

$$x_2(\text{iteration } i + 1) = x_1(\text{iteration } i)$$

Substitution yields

$$x_L + \alpha[x_2(\text{iteration } i) - x_L] = x_R - \alpha(x_R - x_L)$$

or

$$x_L + \alpha[x_L + \alpha(x_R - x_L) - x_L] = x_R - \alpha(x_R - x_L)$$

which simplifies to

$$\alpha^2 + \alpha - 1 = 0$$

This equation yields  $\alpha = \frac{-1 \pm \sqrt{5}}{2}$ . The positive root  $\alpha = \frac{-1 + \sqrt{5}}{2} \approx .681$  is selected because  $0 < \alpha < 1$ .

The design of the golden section computations guarantees an  $\alpha$ -reduction in the successive intervals of uncertainty—that is

$$I_{i+1} = \alpha I_i$$

The golden section method converges more rapidly than the dichotomous method because, in the dichotomous method, the narrowing of the interval of uncertainty slows down appreciably as  $I \rightarrow \Delta$ . In addition, the golden section method requires half the computations because it recycles one set of computations from the immediately preceding iteration.

#### Example 21.1-1

$$\text{Maximize } f(x) = \begin{cases} 3x, & 0 \leq x \leq 2 \\ \frac{1}{3}(-x + 20), & 2 \leq x \leq 3 \end{cases}$$

The maximum value of  $f(x)$  occurs at  $x = 2$ . The following table demonstrates the calculations for iterations 1 and 2 using the dichotomous and the golden section methods, with  $\Delta = .1$ .

## Dichotomous method

## Iteration 1

$$I_0 = (0, 3) = (x_L, x_R)$$

$$x_1 = 0 + .5(3 - 0 - .1) = 1.45, f(x_1) = 4.35$$

$$x_2 = 0 + .5(3 - 0 + .1) = 1.55, f(x_2) = 4.65$$

$$f(x_2) > f(x_1) \Rightarrow x_L = 1.45, I_1 = (1.45, 3)$$

## Iteration 2

$$I_1 = (1.45, 3) = (x_L, x_R)$$

$$x_1 = 1.45 + .5(3 - 1.45 - .1) = 2.175, f(x_1) = 5.942$$

$$x_2 = \frac{3+1.45+1}{2} = 2.275, f(x_2) = 5.908$$

$$f(x_1) > f(x_2) \Rightarrow x_R = 2.275, I_2 = (1.45, 2.275)$$

## Golden section method

## Iteration 1

$$I_0 = (0, 3) = (x_L, x_R)$$

$$x_1 = 3 - .618(3 - 0) = 1.146, f(x_1) = 3.438$$

$$x_2 = 0 + .618(3 - 0) = 1.854, f(x_2) = 5.562$$

$$f(x_2) > f(x_1) \Rightarrow x_L = 1.146, I_1 = (1.146, 3)$$

## Iteration 2

$$I_1 = (1.146, 3) = (x_L, x_R)$$

$$x_1 = x_2 \text{ in iteration 0} = 1.854, f(x_1) = 5.562$$

$$x_2 = 1.146 + .618(3 - 1.146) = 2.292, f(x_2) = 5.903$$

$$f(x_2) > f(x_1) \Rightarrow x_L = 1.854, I_2 = (1.854, 3)$$

Continuing in the same manner, the interval of uncertainty will eventually narrow down to the desired  $\Delta$ -tolerance.

## Excel Moment

Excel template *excelDiGold.xls* handles both methods by entering the letter X in either D5 (dichotomous) or F5 (golden section). The input data include  $f(x)$ ,  $a$ ,  $b$ , and  $\Delta$ . The function  $f(x)$  is entered in cell E3 as

$$=IF(C3<=2, 3*C3, (-C3+20)/3)$$

Cell C3 plays the role of  $x$  in  $f(x)$ .

Figure 21.2 compares the two methods. The golden section method requires less than half the iterations of the dichotomous method, in addition to half the calculations at each iteration.

## PROBLEM SET 21.1A

1. Use Excel template *excelDiGold.xls* to solve Example 21.1-1 assuming that  $\Delta = .01$ . Compare the amount of computations and the accuracy of the results with those in Figure 21.2.
2. Find the maximum of each of the following functions by dichotomous search. Assume that  $\Delta = .05$ .

(a)  $f(x) = \frac{1}{|(x-3)^3|}, \quad 2 \leq x \leq 4$

(b)  $f(x) = x \cos x, \quad 0 \leq x \leq \pi$

\*(c)  $f(x) = x \sin \pi x, \quad 1.5 \leq x \leq 2.5$

(d)  $f(x) = -(x-3)^2, \quad 2 \leq x \leq 4$

\*(e)  $f(x) = \begin{cases} 4x, & 0 \leq x \leq 2 \\ 4-x, & 2 \leq x \leq 4 \end{cases}$

	A	B	C	D	E	F
1	Dichotomous/Golden Section Search					
2	Input data: Type f(C3) in E3, where C3 represents x in f(x)					
3	$\Delta =$	0.1	C3		#VALUE!	
4	Minimum x =	0	Maximum x =	3		
5	Solution:	Enter x to select	Dichotomous:	X	GoldenSection:	
6	$x^* =$	2.04001	$f(x^*) =$	5.97002		
7	Perform calculation					
8	Calculations:					
9	xL	xR	x1	x2	f(x1)	f(x2)
10	0.000000	3.000000	1.450000	1.550000	4.350000	4.650000
11	1.450000	3.000000	2.175000	2.275000	5.941667	5.908333
12	1.450000	2.275000	1.812500	1.912500	5.437500	5.737500
13	1.812500	2.275000	1.993750	2.093750	5.981250	5.968750
14	1.812500	2.093750	1.903125	2.003125	5.709375	5.998958
15	1.903125	2.093750	1.948438	2.048438	5.845313	5.983854
16	1.948438	2.093750	1.971094	2.071094	5.913281	5.976302
17	1.971094	2.093750	1.982422	2.082422	5.947266	5.972526
18	1.982422	2.093750	1.988086	2.088086	5.964258	5.970638
19	1.988086	2.093750	1.990918	2.090918	5.972754	5.969694
20	1.988086	2.090918	1.989502	2.089502	5.968506	5.970166
21	1.989502	2.090918	1.990210	2.090210	5.970630	5.969930
22	1.989502	2.090210	1.989856	2.089856	5.969568	5.970048
23	1.989856	2.090210	1.990033	2.090033	5.970099	5.969989
24	1.989856	2.090033	1.989944	2.089944	5.969833	5.970019
25	1.989944	2.090033	1.989989	2.089989	5.969966	5.970004
26	1.989989	2.090033	1.990011	2.090011	5.970033	5.969996
27	1.989989	2.090011	1.990000	2.090000	5.969999	5.970000
28	1.990000	2.090011	1.990005	2.090005	5.970016	5.969998
29	1.990000	2.090005	1.990003	2.090003	5.970008	5.969999
5	Solution:	Enter x to select	Dichotomous:		GoldenSection:	X
6	$x^* =$	2.00909	$f(x^*) =$	5.99290		
7	Perform calculation					
8	Calculations:					
9	xL	xR	x1	x2	f(x1)	f(x2)
10	0.000000	3.000000	1.145898	1.854102	3.437694	5.562306
11	1.145898	3.000000	1.854102	2.291796	5.562306	5.902735
12	1.854102	3.000000	2.291796	2.562306	5.902735	5.812565
13	1.854102	2.562306	2.124612	2.291796	5.958463	5.902735
14	1.854102	2.291796	2.021286	2.124612	5.992905	5.958463
15	1.854102	2.124612	1.957428	2.021286	5.872283	5.992905
16	1.957428	2.124612	2.021286	2.060753	5.992905	5.979749
17	1.957428	2.060753	1.996894	2.021286	5.990683	5.992905
18	1.996894	2.060753	2.021286	2.036361	5.992905	5.987880

FIGURE 21.2

Excel output of the dichotomous and golden section methods applied to Example 21.1-1 (file *excelDiGold.xls*)

## 21.1.2 Gradient Method

This section develops a method for optimizing twice continuously differentiable functions, called the **steepest ascent** method. The idea is to generate successive points in the direction of the gradient of the function.<sup>1</sup> Termination of the gradient method

<sup>1</sup>The Newton-Raphson method in Section 20.1.2 is also a gradient method that locates the optimum indirectly by solving the necessary conditions equations.

occurs at the point where the gradient vector becomes null. This is only a necessary condition for optimality.

Suppose that  $f(\mathbf{X})$  is maximized. Let  $\mathbf{X}_0$  be the initial point from which the procedure starts, and define  $\nabla f(\mathbf{X}_k)$  as the gradient of  $f$  at point  $\mathbf{X}_k$ . The idea is to determine a particular path  $p$  along which  $\frac{\partial f}{\partial p}$  is maximized at a given point. This result is achieved if successive points  $\mathbf{X}_k$  and  $\mathbf{X}_{k+1}$  are selected such that

$$\mathbf{X}_{k+1} = \mathbf{X}_k + r_k \nabla f(\mathbf{X}_k)$$

where  $r_k$  is the optimal **step size** at  $\mathbf{X}_k$ .

The step size  $r_k$  is determined such that the next point,  $\mathbf{X}_{k+1}$ , leads to the largest improvement in  $f$ . This is equivalent to determining  $r = r_k$  that maximizes the function

$$h(r) = f[\mathbf{X}_k + r \nabla f(\mathbf{X}_k)]$$

Because  $h(r)$  is a single-variable function, the search method in Section 21.1.1 may be used to find the optimum, provided that  $h(r)$  is unimodal.

The proposed procedure terminates when two successive trial points  $\mathbf{X}_k$  and  $\mathbf{X}_{k+1}$  are approximately equal. This is equivalent to having  $r_k \nabla f(\mathbf{X}_k) \approx \mathbf{0}$ , or, equivalently,  $\nabla f(\mathbf{X}_k) \approx \mathbf{0}$ .

### Example 21.1-2

Consider the following problem:

$$\text{Maximize } f(x_1, x_2) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

The exact optimum occurs at  $(x_1^*, x_2^*) = (\frac{1}{3}, \frac{4}{3})$ .

The gradient of  $f$  is

$$\nabla f(\mathbf{X}) = (4 - 4x_1 - 2x_2, 6 - 2x_1 - 4x_2)$$

The quadratic nature of the function indicates that the gradients at two successive points are orthogonal (perpendicular to one another).

Suppose that we start at the initial point  $\mathbf{X}_0 = (1, 1)$ . Figure 21.3 shows the successive solution points.

#### Iteration 1

$$\nabla f(\mathbf{X}_0) = (-2, 0)$$

The next point  $\mathbf{X}_1$  is obtained by considering

$$\mathbf{X} = (1, 1) + r(-2, 0) = (1 - 2r, 1)$$

Thus,

$$h(r) = f(1 - 2r, 1) = -2(1 - 2r)^2 + 2(1 - 2r) + 4$$

The optimal step size is obtained using the classical necessary conditions in Chapter 20 (you may also use the search algorithms in Section 21.1.1 to determine the optimum). The maximum value of  $h(r)$  is  $r_1 = \frac{1}{4}$ , which yields the next solution point  $\mathbf{X}_1 = (\frac{1}{2}, 1)$ .

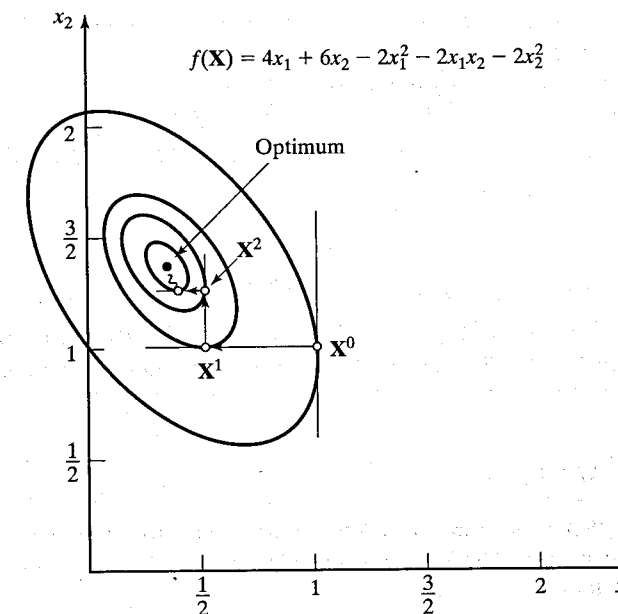


FIGURE 21.3

Maximization of  $f(x_1, x_2) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$  by the steepest-ascent method

#### Iteration 2

$$\nabla f(\mathbf{X}_1) = (0, 1)$$

$$\mathbf{X} = \left(\frac{1}{2}, 1\right) + r(0, 1) = \left(\frac{1}{2}, 1 + r\right)$$

$$h(r) = -2(1 + r)^2 + 5(1 + r) + \frac{3}{2}$$

Thus,  $r_2 = \frac{1}{4}$  and  $\mathbf{X}_2 = (\frac{1}{2}, \frac{5}{4})$ .

#### Iteration 3

$$\nabla f(\mathbf{X}_2) = \left(-\frac{1}{2}, 0\right)$$

$$\mathbf{X} = \left(\frac{1}{2}, \frac{5}{4}\right) + r\left(-\frac{1}{2}, 0\right) = \left(\frac{1-r}{2}, \frac{5}{4}\right)$$

$$h(r) = -\frac{1}{2}(1 - r)^2 + \frac{3}{4}(1 - r) + \frac{35}{8}$$

Hence,  $r_3 = \frac{1}{4}$  and  $\mathbf{X}_3 = (\frac{3}{8}, \frac{5}{4})$ .

#### Iteration 4

$$\nabla f(\mathbf{X}_3) = \left(0, \frac{1}{4}\right)$$

$$\mathbf{X} = \left(\frac{3}{8}, \frac{5}{4}\right) + r\left(0, \frac{1}{4}\right) = \left(\frac{3}{8}, \frac{5+r}{4}\right)$$

$$h(r) = -\frac{1}{8}(5 + r)^2 + \frac{21}{16}(5 + r) + \frac{39}{32}$$

Thus,  $r_4 = \frac{1}{4}$  and  $\mathbf{X}_4 = (\frac{3}{8}, \frac{21}{16})$ .



## Iteration 5

$$\nabla f(\mathbf{X}_4) = \left(-\frac{1}{8}, 0\right)$$

$$\mathbf{X} = \left(\frac{3}{8}, \frac{21}{16}\right) + r\left(-\frac{1}{8}, 0\right) = \left(\frac{3-r}{8}, \frac{21}{16}\right)$$

$$h(r) = -\frac{1}{32}(3-r)^2 + \frac{11}{64}(3-r) + \frac{567}{128}$$

This gives  $r_5 = \frac{1}{4}$  and  $\mathbf{X}_5 = \left(\frac{11}{32}, \frac{21}{16}\right)$ .

## Iteration 6

$$\nabla f(\mathbf{X}_5) = \left(0, \frac{1}{16}\right)$$

The process can be terminated at this point because  $\nabla f(\mathbf{X}_5) \approx \mathbf{0}$ . The approximate maximum point is given by  $\mathbf{X}_5 = (.3438, 1.3125)$ . The exact optimum is  $\mathbf{X}^* = (.3333, 1.3333)$ .

## PROBLEM SET 21.1B

- \*1. Show that, in general, the Newton-Raphson method (Section 20.1.2) when applied to a strictly concave quadratic function will converge in exactly one step. Apply the method to the maximization of

$$f(\mathbf{X}) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

2. Carry out five iterations for each of the following problems using the method of steepest descent (ascent). Assume that  $\mathbf{X}^0 = \mathbf{0}$  in each case.

(a)  $\min f(\mathbf{X}) = \min f(\mathbf{X}) = (x_2 - x_1^2)^2 + (1 - x_1)$

(b)  $\max f(\mathbf{X}) = \mathbf{cX} + \mathbf{X}^T \mathbf{A} \mathbf{X}$   
where

$$\mathbf{c} = (1, 3, 5)$$

$$\mathbf{A} = \begin{pmatrix} -5 & -3 & -\frac{1}{2} \\ -3 & -2 & 0 \\ -\frac{1}{2} & 0 & -\frac{1}{2} \end{pmatrix}$$

(c)  $\min f(\mathbf{X}) = x_1 - x_2 + x_1^2 - x_1x_2$

## 21.2 CONSTRAINED ALGORITHMS

The general constrained nonlinear programming problem is defined as

$$\text{Maximize (or minimize) } z = f(\mathbf{X})$$

subject to

$$\mathbf{g}(\mathbf{X}) \leq \mathbf{0}$$

The nonnegativity conditions,  $\mathbf{X} \geq \mathbf{0}$ , are part of the constraints. Also, at least one of the functions  $f(\mathbf{X})$  and  $\mathbf{g}(\mathbf{X})$  is nonlinear, and all the functions are continuously differentiable.

The erratic behavior of the nonlinear functions precludes the development of a single algorithm for the general nonlinear model. Perhaps the most general result applicable to the problem is the KKT conditions (Section 20.2.2). Table 20.2 shows that the KKT conditions are only necessary, unless  $f(\mathbf{X})$  and  $\mathbf{g}(\mathbf{X})$  are well-behaved functions.

This section presents a number of algorithms that may be classified generally as *indirect* and *direct* methods. Indirect methods solve the nonlinear problem by dealing with one or more *linear* programs derived from the original program. Direct methods deal with the original problem.

The indirect algorithms presented in this section include separable, quadratic, and chance-constrained programming. The direct algorithms include the method of linear combinations and a brief discussion of SUMT, the sequential unconstrained maximization technique. Other important nonlinear techniques can be found in the list of references at the end of the chapter.

## 21.2.1 Separable Programming

A function  $f(x_1, x_2, \dots, x_n)$  is **separable** if it can be expressed as the sum of  $n$  single-variable functions  $f_1(x_1), f_2(x_2), \dots, f_n(x_n)$ —that is,

$$f(x_1, x_1, \dots, x_n) = f_1(x_1) + f_2(x_2) + \dots + f_n(x_n)$$

For example, any linear function is separable. On the other hand, the function

$$h(x_1, x_2, x_3) = x_1^2 + x_1 \sin(x_2 + x_3) + x_2 e^{x_3}$$

is not separable.

Some (convoluted) nonlinear functions can be made separable using appropriate substitutions. Consider, for example, the case of maximizing  $z = x_1x_2$ . Let  $y = x_1x_2$ , then  $\ln y = \ln x_1 + \ln x_2$ , and the equivalent separable problem is

$$\text{Maximize } z = y$$

subject to

$$\ln y = \ln x_1 + \ln x_2$$

The substitution assumes that  $x_1$  and  $x_2$  are *positive* variables because the logarithmic function is undefined for nonpositive values. We can account for the case where  $x_1$  and  $x_2$  can assume zero values by employing the approximations

$$w_1 = x_1 + \delta_1 > 0$$

$$w_2 = x_2 + \delta_2 > 0$$

The constants  $\delta_1$  and  $\delta_2$  are very small positive values.

This section shows how an approximate solution can be obtained for *any* separable problem by using linear approximation and the simplex method of linear programming. The single-variable function  $f(x)$  can be approximated by a piecewise-linear function using mixed integer programming (Chapter 9). Suppose that  $f(x)$  is

approximated over an interval  $[a, b]$ , and define  $a_k, k = 1, 2, \dots, K$ , as the  $k$ th breakpoint on the  $x$ -axis such that  $a_1 < a_2 < \dots < a_K$ . The points  $a_1$  and  $a_K$  coincide with end points  $a$  and  $b$  of the designated interval. Thus,  $f(x)$  is approximated as

$$f(x) \approx \sum_{k=1}^K f(a_k)w_k$$

$$x = \sum_{k=1}^K a_k w_k$$

The nonnegative weights  $w_k$  must satisfy the condition

$$\sum_{k=1}^K w_k = 1, w_k \geq 0, k = 1, 2, \dots, K$$

Mixed integer programming ensures the validity of the approximation by imposing two additional conditions:

1. At most two  $w_k$  are positive.
2. If  $w_k$  is positive, then only an adjacent  $w_{k+1}$  or  $w_{k-1}$  can assume a positive value.

To show how these conditions are satisfied, consider the separable problem

$$\text{Maximize (or minimize) } z = \sum_{j=1}^n f_j(x_j)$$

subject to

$$\sum_{j=1}^n g_{ij}(x_j) \leq b_i, i = 1, 2, \dots, m$$

This problem can be approximated by a mixed integer program as follows. Let<sup>2</sup>

$$\left. \begin{array}{l} a_{jk} = \text{breakpoint } k \text{ for variable } x_j \\ w_{jk} = \text{weight with breakpoint } k \text{ of variable } x_j \end{array} \right\} k = 1, 2, \dots, K_j, j = 1, 2, \dots, n$$

Then the equivalent mixed problem is

$$\text{Maximize (or minimize) } z = \sum_{j=1}^n \sum_{k=1}^{K_j} f_j(a_{jk})w_{jk}$$

subject to

$$\sum_{j=1}^n \sum_{k=1}^{K_j} g_{jk}(a_{jk})w_{jk} \leq b_i, \quad i = 1, 2, \dots, m$$

$$0 \leq w_{j1} \leq y_{j1}, \quad j = 1, 2, \dots, n$$

<sup>2</sup>It is more accurate to replace the index  $k$  with  $k_j$  to correspond uniquely to variable  $j$ . In this instant, we will forsake mathematical accuracy for a simpler notation.

$$0 \leq w_{jk} \leq y_{j,k-1} + y_{jk}, \quad k = 2, 3, \dots, K_j - 1, j = 1, 2, \dots, n$$

$$0 \leq w_{jK_j} \leq y_{j,K_j-1}, \quad j = 1, 2, \dots, n$$

$$\sum_{k=1}^{K_j-1} y_{jk} = 1, \quad j = 1, 2, \dots, n$$

$$\sum_{k=1}^{K_j} w_{jk} = 1, \quad j = 1, 2, \dots, n$$

$$y_{jk} = (0, 1), \quad k = 1, 2, \dots, K_j, j = 1, 2, \dots, n$$

The variables in the approximation problem are  $w_{jk}$  and  $y_{jk}$ .

The formulation shows how any separable problem can be solved, in principle, by mixed integer programming. The difficulty is that the number of constraints increases rather rapidly with the number of breakpoints. In particular, the computational feasibility of the procedure is questionable because there are no consistently reliable computer codes for solving large mixed integer programming problems.

Another method for solving the approximation model is the regular simplex method (Chapter 3) using **restricted basis**. In this case, the additional constraints involving  $y_{jk}$  are dropped. The restricted basis modifies the simplex method optimality condition by selecting the entering variable  $w_j$  with the *best* ( $z_{jk} - c_{jk}$ ) that does not violate the adjacency requirement of the  $w$ -variables with positive values. The process is repeated until the optimality condition is satisfied or until it is impossible to satisfy the restricted basis condition, whichever occurs first.

The mixed integer programming method yields a global optimum to the approximate problem, whereas the restricted basis method can only guarantee a local optimum. Additionally, in the two methods, the approximate solution may not be feasible for the original problem, in which case it may be necessary to refine the approximation by increasing the number of breakpoints.

### Example 21.2-1

Consider the problem

$$\text{Maximize } z = x_1 + x_2^4$$

subject to

$$3x_1 + 2x_2^2 \leq 9$$

$$x_1, x_2 \geq 0$$

The exact optimum solution to this problem, obtained by AMPL or Solver, is  $x_1 = 0$ ,  $x_2 = 2.12132$ , and  $z^* = 20.25$ . To show how the approximating method is used, consider the separable functions

$$f_1(x_1) = x_1$$

$$f_2(x_2) = x_2^4$$

$$g_1(x_1) = 3x_1$$

$$g_2(x_2) = 2x_2^2$$

The variable  $x_1$  is not approximated because the functions  $f_1(x_1)$  and  $g_1(x_1)$  are already linear. Considering  $f_2(x_2)$  and  $g_2(x_2)$ , we assume four breakpoints:  $a_{2k} = 0, 1, 2$ , and  $3$  for  $k = 1, 2, 3$ , and  $4$ , respectively. Given  $x_2 \leq 3$ , it follows that

$k$	$a_{2k}$	$f_2(a_{2k}) = a_{2k}^4$	$g_2(a_{2k}) = 2a_{2k}^2$
1	0	0	0
2	1	1	2
3	2	16	8
4	3	81	18

Thus

$$\begin{aligned} f_2(x_2) &\approx w_{21}f_2(a_{21}) + w_{22}f_2(a_{22}) + w_{23}f_2(a_{23}) + w_{24}f_2(a_{24}) \\ &\approx 0w_{21} + 1w_{22} + 16w_{23} + 81w_{24} = w_{22} + 16w_{23} + 81w_{24} \end{aligned}$$

Similarly,

$$g_2(x_2) \approx 2w_{22} + 8w_{23} + 18w_{24}$$

The approximation problem thus becomes

$$\text{Maximize } z = x_1 + w_{22} + 16w_{23} + 81w_{24}$$

subject to

$$\begin{aligned} 3x_1 + 2w_{22} + 8w_{23} + 18w_{24} &\leq 9 \\ w_{21} + w_{22} + w_{23} + w_{24} &= 1 \\ x_1 \geq 0, w_{2k} \geq 0, k &= 1, 2, 3, 4 \end{aligned}$$

The values of  $w_{2k}$ ,  $k = 1, 2, 3, 4$ , must satisfy the restricted basis condition.

The initial simplex tableau (with rearranged columns to provide a starting solution) is given by

Basic	$x_1$	$w_{22}$	$w_{23}$	$w_{24}$	$s_1$	$w_{21}$	Solution
$z$	-1	-1	-16	-81	0	0	0
$s_1$	3	2	8	18	1	0	9
$w_{21}$	0	1	1	1	0	1	1

The variable  $s_1 (\geq 0)$  is a slack. (This problem happened to have an obvious starting solution. In general, one can use artificial variables, Section 3.4.)

From the  $z$ -row,  $w_{24}$  is the entering variable. Because  $w_{21}$  is currently basic and positive, the restricted basis condition dictates that it must leave before  $w_{24}$  can enter the solution. However, by the feasibility condition,  $s_1$  must be the leaving variable, which means that  $w_{24}$  cannot enter the solution. The next-best entering variable,  $w_{23}$ , requires  $w_{21}$  to leave the basic solution, a condition that happens to be satisfied by the feasibility condition. The new tableau thus becomes

Basic	$x_1$	$w_{22}$	$w_{23}$	$w_{24}$	$s_1$	$w_{21}$	Solution
$z$	-1	15	0	-65	0	16	16
$s_1$	3	-6	0	10	1	-8	1
$w_{23}$	0	1	1	1	0	1	1

Next,  $w_{24}$  is the entering variable, which is admissible because  $w_{23}$  is positive. The simplex method shows that  $s_1$  will leave. Thus,

Basic	$x_1$	$w_{22}$	$w_{23}$	$w_{24}$	$s_1$	$w_{21}$	Solution
$z$	$\frac{37}{2}$	-24	0	0	$\frac{13}{2}$	-36	$22\frac{1}{2}$
$w_{24}$	$\frac{3}{10}$	$-\frac{6}{10}$	0	1	$\frac{1}{10}$	$-\frac{8}{10}$	$\frac{1}{10}$
$w_{23}$	$-\frac{3}{10}$	$\frac{16}{10}$	1	0	$-\frac{1}{10}$	$\frac{18}{10}$	$\frac{9}{10}$

The tableau shows that  $w_{21}$  and  $w_{22}$  are candidates for the entering variable. The variable  $w_{21}$  is not adjacent to basic  $w_{23}$  or  $w_{24}$ , hence it cannot become basic. Similarly,  $w_{22}$  cannot enter because  $w_{24}$  cannot leave. Thus, the last tableau is the best restricted-basis solution for the approximate problem.

The optimum solution to the original problem is

$$\begin{aligned} x_1 &= 0 \\ x_2 &\approx 2w_{23} + 3w_{24} = 2\left(\frac{9}{10}\right) + 3\left(\frac{1}{10}\right) = 2.1 \\ z &= 0 + 2.1^4 = 19.45 \end{aligned}$$

The value  $x_2 = 2.1$  approximately equals the true optimum value ( $=2.12132$ ).

**Separable convex programming.** A special case of separable programming occurs when  $g_{ij}(x_j)$  is convex for all  $i$  and  $j$ , which ensures a convex solution space. Additionally, if  $f_j(x_j)$  is convex (minimization) or concave (maximization) for all  $j$ , then the problem has a global optimum (see Table 20.2, Section 20.2.2). Under such conditions, the following simplified approximation can be used.

Consider a minimization problem, and let  $f_j(x_j)$  be as shown in Figure 21.4. The breakpoints of the function  $f_j(x_j)$  are  $x_j = a_{jk}$ ,  $k = 0, 1, \dots, K_j$ . Let  $x_{jk}$  define the increment of the variable  $x_j$  in the range  $(a_{j,k-1}, a_{jk})$ ,  $k = 1, 2, \dots, K_j$ , and let  $r_{jk}$  be the corresponding rate of change (slope of the line segment) in the same range. Then

$$\begin{aligned} f_j(x_j) &\approx \sum_{k=1}^{K_j} r_{jk}x_{jk} + f_j(a_{j0}) \\ x_j &= \sum_{k=1}^{K_j} x_{jk} \\ 0 &\leq x_{jk} \leq a_{jk} - a_{j,k-1}, k = 1, 2, \dots, K_j \end{aligned}$$

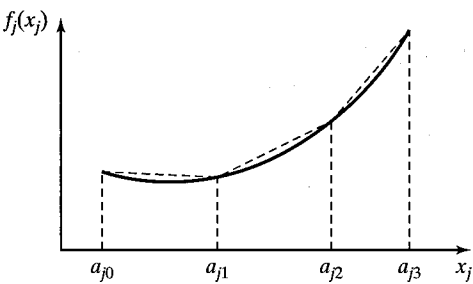


FIGURE 21.4  
Piecewise-linear approximation of a convex function

The fact that  $f_j(x_j)$  is convex ensures that  $r_{j1} < r_{j2} < \dots < r_{jK_j}$ . This means that in the minimization problem the variable  $x_{jp}$  is more attractive than  $x_{jq}$  for  $p < q$ . Consequently,  $x_{jp}$  will always reach its maximum limit before  $x_{jq}$  can assume a positive value.

The convex constraint functions  $g_{ij}(x_j)$  are approximated in essentially the same way. Let  $r_{ijk}$  be the slope of the  $k$ th line segment corresponding to  $g_{ij}(x_j)$ . It follows that

$$g_{ij}(x_j) \approx \sum_{k=1}^{K_j} r_{ijk} x_{jk} + g_{ij}(a_{j0})$$

The complete problem is thus given by

$$\text{Minimize } z = \sum_{j=1}^n \left( \sum_{k=1}^{K_j} r_{jk} x_{jk} + f_j(a_{j0}) \right)$$

subject to

$$\sum_{j=1}^n \left( \sum_{k=1}^{K_j} r_{ijk} x_{jk} + g_{ij}(a_{j0}) \right) \leq b_i, i = 1, 2, \dots, m$$

$$0 \leq x_{jk} \leq a_{jk} - a_{j,k-1}, k = 1, 2, \dots, K_j, j = 1, 2, \dots, n$$

where

$$r_{jk} = \frac{f_j(a_{jk}) - f_j(a_{j,k-1})}{a_{jk} - a_{j,k-1}}$$

$$r_{ijk} = \frac{g_{ij}(a_{jk}) - g_{ij}(a_{j,k-1})}{a_{jk} - a_{j,k-1}}$$

The maximization problem is treated in essentially the same way. In this case,  $r_{j1} > r_{j2} > \dots > r_{jK_j}$ , which means that, for  $p < q$ , the variable  $x_{jp}$  will always reach its maximum value before  $x_{jq}$  is allowed to assume a positive value (see Problem 7, Set 21.2a, for proof).

The new problem can be solved by the simplex method with upper-bounded variables (Section 7.3). The restricted basis concept is not needed because the convexity (concavity) of the functions guarantees correct selection of basic variables.

### Example 21.2-2

Consider the problem

$$\text{Maximize } z = x_1 - x_2$$

subject to

$$3x_1^4 + x_2 \leq 243$$

$$x_1 + 2x_2^2 \leq 32$$

$$x_1 \geq 2.1$$

$$x_2 \geq 3.5$$

The separable functions of this problem are

$$f_1(x_1) = x_1, \quad f_2(x_2) = -x_2$$

$$g_{11}(x_1) = 3x_1^4, \quad g_{12}(x_2) = x_2$$

$$g_{21}(x_1) = x_1, \quad g_{22}(x_2) = 2x_2^2$$

These functions satisfy the convexity condition required for the minimization problems. The functions  $f_1(x_1)$ ,  $f_2(x_2)$ ,  $g_{12}(x_2)$ , and  $g_{21}(x_1)$  are already linear.

The ranges of the variables  $x_1$  and  $x_2$  (estimated from the constraints) are  $0 \leq x_1 \leq 3$  and  $0 \leq x_2 \leq 4$ . Let  $K_1 = 3$  and  $K_2 = 4$ . The slopes corresponding to the separable functions are determined as follows.

For  $j = 1$ ,

$k$	$a_{1k}$	$g_{11}(a_{1k}) = 3a_{1k}^4$	$r_{11k}$	$x_{1k}$
0	0	0	—	—
1	1	3	3	$x_{11}$
2	2	48	45	$x_{12}$
3	3	243	195	$x_{13}$

For  $j = 2$ ,

$k$	$a_{2k}$	$g_{22}(a_{2k}) = 2a_{2k}^2$	$r_{22k}$	$x_{2k}$
0	0	0	—	—
1	1	2	2	$x_{21}$
2	2	8	6	$x_{22}$
3	3	18	10	$x_{23}$
4	4	32	14	$x_{24}$

The complete problem then becomes

$$\text{Maximize } z = x_1 - x_2$$

subject to

$$3x_{11} + 45x_{12} + 195x_{13} + x_2 \leq 243 \quad (1)$$

$$x_1 + 2x_{21} + 6x_{22} + 10x_{23} + 14x_{24} \leq 32 \quad (2)$$

$$x_1 \geq 2.1 \quad (3)$$

$$x_2 \geq 3.5 \quad (4)$$

$$x_{11} + x_{12} + x_{13} - x_1 = 0 \quad (5)$$

$$x_{21} + x_{22} + x_{23} + x_{24} - x_2 = 0 \quad (6)$$

$$0 \leq x_{1k} \leq 1, \quad k = 1, 2, 3 \quad (7)$$

$$0 \leq x_{2k} \leq 1, \quad k = 1, 2, 3, 4 \quad (8)$$

$$x_1, x_2 \geq 0$$



Constraints 5 and 6 are needed to maintain the relationship between the original and new variables. The optimum solution is

$$z = -.52, x_1 = 2.98, x_2 = 3.5, x_{11} = x_{12} = 1, x_{13} = .98, x_{21} = x_{22} = x_{23} = 1, x_{24} = .5$$

### AMPL Moment

AMPL modeling of the *original* nonlinear problem of Example 21.2-2 is very much the same as in linear problems. Obtaining the solution is an entirely different matter because of the "unpredictable" behavior of the nonlinear functions. File *amplEx21.2-2.txt* provides the model. An explanation of the model is given in Appendix C on the website (see Figure C.17).

### PROBLEM SET 21.2A

1. Approximate the following problem as a mixed integer program.

$$\text{Maximize } z = e^{-x_1} + x_1 + (x_2 + 1)^2$$

subject to

$$x_1^2 + x_2 \leq 3$$

$$x_1, x_2 \geq 0$$

- \*2. Repeat Problem 1 using the restricted basis method. Then find the optimal solution.

3. Consider the problem

$$\text{Maximize } z = x_1 x_2 x_3$$

subject to

$$x_1^2 + x_2 + x_3 \leq 4$$

$$x_1, x_2, x_3 \geq 0$$

Approximate the problem as a linear program for use with the restricted basis method.

- \*4. Show how the following problem can be made separable.

$$\text{Maximize } z = x_1 x_2 + x_3 + x_1 x_3$$

subject to

$$x_1 x_2 + x_2 + x_1 x_3 \leq 10$$

$$x_1, x_2 x_3 \geq 0$$

5. Show how the following problem can be made separable.

$$\text{Minimize } z = e^{2x_1 + x_2^2} + (x_3 - 2)^2$$

subject to

$$x_1 + x_2 + x_3 \leq 6$$

$$x_1, x_2 x_3 \geq 0$$

6. Show how the following problem can be made separable.

$$\text{Maximize } z = e^{x_1 x_2} + x_2^2 x_3 + x_4$$

subject to

$$x_1 + x_2 x_3 + x_3 \leq 10$$

$$x_1, x_2, x_3 \geq 0$$

$$x_4 \text{ unrestricted in sign}$$

7. Show that in separable convex programming, it is never optimal to have  $x_{ki} > 0$  when  $x_{k-1, i}$  is not at its upper bound.

8. Solve as a separable convex programming problem.

$$\text{Minimize } z = x_1^4 + x_2 + x_3^2$$

subject to

$$x_1^2 + x_2 + x_3^2 \leq 4$$

$$|x_1 + x_2| \leq 0$$

$$x_1, x_3 \geq 0$$

$$x_2 \text{ unrestricted in sign}$$

9. Solve the following as a separate convex programming problem.

$$\text{Minimize } z = (x_1 - 2)^2 + 4(x_2 - 6)^2$$

subject to

$$6x_1 + 3(x_2 + 1)^2 \leq 12$$

$$x_1, x_2 \geq 0$$

### 21.2.2 Quadratic Programming

A quadratic programming model is defined as

$$\text{Maximize } z = \mathbf{C}\mathbf{X} + \mathbf{X}^T \mathbf{D} \mathbf{X}$$

subject to

$$\mathbf{A}\mathbf{X} \leq \mathbf{b}, \mathbf{X} \geq 0$$

where

$$\mathbf{X} = (x_1, x_2, \dots, x_n)^T$$

$$\mathbf{C} = (c_1, c_2, \dots, c_n)$$

$$\mathbf{b} = (b_1, b_2, \dots, b_m)^T$$

$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{pmatrix}$$

The function  $\mathbf{X}^T \mathbf{D} \mathbf{X}$  defines a quadratic form (see Section D.3 on the website). The matrix  $\mathbf{D}$  is assumed symmetric and negative definite—meaning that  $z$  is strictly concave. The constraints are linear, which guarantees a convex solution space.

The solution to this problem is based on the KKT necessary conditions. These conditions (as shown in Table 20.2, Section 20.2.2) are also sufficient because  $z$  is concave and the solution space is a convex set.

The quadratic programming problem will be treated for the maximization case. Conversion to minimization is straightforward. The problem may be written as

$$\text{Maximize } z = \mathbf{C}\mathbf{X} + \mathbf{X}^T \mathbf{D} \mathbf{X}$$

subject to

$$\mathbf{G}(\mathbf{X}) = \begin{pmatrix} \mathbf{A} \\ -\mathbf{I} \end{pmatrix} \mathbf{X} - \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \leq \mathbf{0}$$

Let

$$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)^T$$

$$\mathbf{U} = (\mu_1, \mu_2, \dots, \mu_n)^T$$

be the Lagrange multipliers corresponding to constraints  $\mathbf{A}\mathbf{X} - \mathbf{b} \leq \mathbf{0}$  and  $-\mathbf{X} \leq \mathbf{0}$ , respectively. Application of the KKT conditions yields

$$\boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{U} \geq \mathbf{0}$$

$$\nabla z - (\boldsymbol{\lambda}^T, \mathbf{U}^T) \nabla \mathbf{G}(\mathbf{X}) = \mathbf{0}$$

$$\lambda_i \left( b_i - \sum_{j=1}^n a_{ij} x_j \right) = 0, i = 1, 2, \dots, m$$

$$\mu_j x_j = 0, j = 1, 2, \dots, n$$

$$\mathbf{A}\mathbf{X} \leq \mathbf{b}$$

$$-\mathbf{X} \leq \mathbf{0}$$

Now

$$\nabla z = \mathbf{C} + 2\mathbf{X}^T \mathbf{D}$$

$$\nabla \mathbf{G}(\mathbf{X}) = \begin{pmatrix} \mathbf{A} \\ -\mathbf{I} \end{pmatrix}$$

Let  $\mathbf{S} = \mathbf{b} - \mathbf{A}\mathbf{X} \geq \mathbf{0}$  be the slack variables of the constraints. The conditions reduce to

$$-2\mathbf{X}^T \mathbf{D} + \boldsymbol{\lambda}^T \mathbf{A} - \mathbf{U}^T = \mathbf{C}$$

$$\mathbf{A}\mathbf{X} + \mathbf{S} = \mathbf{b}$$

$$\mu_j x_j = 0 = \lambda_i s_i \text{ for all } i \text{ and } j$$

$$\boldsymbol{\lambda}, \mathbf{U}, \mathbf{X}, \mathbf{S} \geq \mathbf{0}$$

Because  $\mathbf{D}^T = \mathbf{D}$ , the transpose of the first set of equations can be written as

$$-2\mathbf{D}\mathbf{X} + \mathbf{A}^T \boldsymbol{\lambda} - \mathbf{U} = \mathbf{C}^T$$

Hence, the necessary conditions may be combined as

$$\begin{pmatrix} -2\mathbf{D} & \mathbf{A}^T & -\mathbf{I} & \mathbf{0} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ \boldsymbol{\lambda} \\ \mathbf{U} \\ \mathbf{S} \end{pmatrix} = \begin{pmatrix} \mathbf{C}^T \\ \mathbf{b} \end{pmatrix}$$

$$\mu_j x_j = 0 = \lambda_i s_i, \text{ for all } i \text{ and } j$$

$$\boldsymbol{\lambda}, \mathbf{U}, \mathbf{X}, \mathbf{S} \geq \mathbf{0}$$

Except for the conditions  $\mu_j x_j = 0 = \lambda_i s_i$ , the remaining equations are linear in  $\mathbf{X}$ ,  $\boldsymbol{\lambda}$ ,  $\mathbf{U}$ , and  $\mathbf{S}$ . Thus, the problem is equivalent to solving a set of linear equations with the additional conditions  $\mu_j x_j = 0 = \lambda_i s_i$ .

The solution of the system is obtained by using phase I of the two-phase method (Section 3.4.2), with the added restrictions  $\lambda_i s_i = 0$  and  $\mu_j x_j = 0$ . This means that  $\lambda_i$  and  $s_i$  cannot be positive simultaneously, and neither can  $\mu_j$  and  $x_j$ . This is the same idea of the **restricted basis** used in Section 21.2.1.

Phase I will render all the artificial variables equal to zero provided the problem has a feasible solution space.

### Example 21.2-3

Consider the problem

$$\text{Maximize } z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

subject to

$$x_1 + 2x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

This problem can be put in the following matrix form:

$$\text{Maximize } z = (4, 6) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (x_1, x_2) \begin{pmatrix} -2 & -1 \\ -1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

subject to

$$(1, 2) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq 2$$

$$x_1, x_2 \geq 0$$

The KKT conditions are given as

$$\begin{pmatrix} 4 & 2 & 1 & -1 & 0 & 0 \\ 2 & 4 & 2 & 0 & -1 & 0 \\ 1 & 2 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \mu_1 \\ \mu_2 \\ s_1 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \\ 2 \end{pmatrix}, \mu_1 x_1 = \mu_2 x_2 = \lambda_1 s_1 = 0$$

The initial tableau for phase 1 is obtained by introducing the artificial variables  $R_1$  and  $R_2$  and updating the objective row.

Basic	$x_1$	$x_2$	$\lambda_1$	$\mu_1$	$\mu_2$	$R_1$	$R_2$	$s_1$	Solution
$r$	6	6	3	-1	-1	0	0	0	10
$R_1$	4	2	1	-1	0	1	0	0	4
$R_2$	2	4	2	0	-1	0	1	0	6
$s_1$	1	2	0	0	0	0	0	1	2

**Iteration 1.** The most promising entering variable  $x_1$  can be made basic because  $\mu_1 = 0$ .

Basic	$x_1$	$x_2$	$\lambda_1$	$\mu_1$	$\mu_2$	$R_1$	$R_2$	$s_1$	Solution
$R$	0	3	$\frac{3}{2}$	$\frac{1}{2}$	-1	$-\frac{3}{2}$	0	0	4
$x_1$	1	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{1}{4}$	0	$\frac{1}{4}$	0	0	1
$R_2$	0	3	$\frac{3}{2}$	$\frac{1}{2}$	-1	$-\frac{1}{2}$	1	0	4
$s_1$	0	$\frac{3}{2}$	$-\frac{1}{4}$	$\frac{1}{4}$	0	$-\frac{1}{4}$	0	1	1

**Iteration 2.** The most promising variable  $x_2$  can be made basic because  $\mu_2 = 0$ .

Basic	$x_1$	$x_2$	$\lambda_1$	$\mu_1$	$\mu_2$	$R_1$	$R_2$	$s_1$	Solution
$r$	0	0	2	0	-1	-1	0	-2	2
$x_1$	1	0	$\frac{1}{3}$	$-\frac{1}{3}$	0	$\frac{1}{3}$	0	$-\frac{1}{3}$	$\frac{2}{3}$
$R_1$	0	0	2	0	-1	0	1	-2	2
$x_2$	0	1	$-\frac{1}{6}$	$\frac{1}{6}$	0	$-\frac{1}{6}$	0	$\frac{2}{3}$	$\frac{2}{3}$

**Iteration 3.** The multiplier  $\lambda_1$  can be made basic because  $s_1 = 0$ .

Basic	$x_1$	$x_2$	$\lambda_1$	$\mu_1$	$\mu_2$	$R_1$	$R_2$	$s_1$	Solution
$r$	0	0	0	0	0	-1	-1	0	0
$x_1$	1	0	0	$-\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{3}$	$-\frac{1}{6}$	0	$\frac{1}{3}$
$\lambda_1$	0	0	1	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	-1	1
$x_2$	0	1	0	$\frac{1}{6}$	$-\frac{1}{12}$	$-\frac{1}{6}$	$\frac{1}{12}$	$\frac{1}{2}$	$\frac{5}{6}$

The last tableau gives the optimal feasible solution ( $x_1^* = \frac{1}{3}$ ,  $x_2^* = \frac{5}{6}$ ). The associated optimal value of  $z$  is 4.16.

### Solver Moment

Solver template—*excelQP.xls*—solves Example 21.2-3. The data are entered in a manner similar to linear programming (see Section 2.3.1). The main difference occurs in the way the nonlinear functions are entered. Specifically, the nonlinear objective function is entered in target cell D5 as

$$=4*B10+6*C10-2*B10^2-2*B10*C10-2*C10^2$$

The changing cells are B10:C10 [ $\equiv (x_1, x_2)$ ]. Notice that cells B5:C5 are not used at all in the model. For readability, we entered the symbol NL to indicate that the associated constraint is nonlinear. Also, you can specify the nonnegativity of the variables either in the Options dialogue box or by adding explicit nonnegativity constraints.

### PROBLEM SET 21.2B

\*1. Consider the problem

$$\text{Maximize } z = 6x_1 + 3x_2 - 4x_1x_2 - 2x_1^2 - 3x_2^2$$

subject to

$$x_1 + x_2 \leq 1$$

$$2x_1 + 3x_2 \leq 4$$

$$x_1, x_2 \geq 0$$

Show that  $z$  is strictly concave, and then solve the problem using the quadratic programming algorithm.

\*2. Consider the problem:

$$\text{Minimize } z = 2x_1^2 + 2x_2^2 + 3x_3^2 + 2x_1x_2 + 2x_2x_3 + x_1 - 3x_2 - 5x_3$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &\geq 1 \\ 3x_1 + 2x_2 + x_3 &\leq 6 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

Show that  $z$  is strictly convex, and then solve by the quadratic programming algorithm.

### 21.2.3 Chance-Constrained Programming

Chance-constrained programming deals with situations in which the parameters of the constraints are random variables and the constraints are realized with a minimum probability. Mathematically, the problem is defined as

$$\text{Maximize } z = \sum_{j=1}^n c_j x_j$$

subject to

$$P\left\{\sum_{j=1}^n a_{ij}x_j \leq b_i\right\} \geq 1 - \alpha_i, i = 1, 2, \dots, m, x_j \geq 0, \text{ for all } j$$

The parameters  $a_{ij}$  and  $b_i$  are random variables, and constraint  $i$  is realized with a minimum probability of  $1 - \alpha_i$ ,  $0 < \alpha_i < 1$ .

Three cases are considered:

1. Only  $a_{ij}$  is random for all  $i$  and  $j$ .
2. Only  $b_i$  is random for all  $i$ .
3. Both  $a_{ij}$  and  $b_i$  are random for all  $i$  and  $j$ .

In all three cases, it is assumed that the parameters are normally distributed with known means and variances.

**Case 1.** Each  $a_{ij}$  is normally distributed with mean  $E\{a_{ij}\}$ , variance  $\text{var}\{a_{ij}\}$ , and  $\text{cov}\{a_{ij}, a_{i'j'}\}$  of  $a_{ij}$  and  $a_{i'j'}$ .

Consider

$$P\left\{\sum_{j=1}^n a_{ij}x_j \leq b_i\right\} \geq 1 - \alpha_i$$

Define

$$h_i = \sum_{j=1}^n a_{ij}x_j$$

The random variable  $h_i$  is normally distributed with

$$\begin{aligned} E\{h_i\} &= \sum_{j=1}^n E\{a_{ij}\}x_j \\ \text{var}\{h_i\} &= \mathbf{X}^T \mathbf{D}_i \mathbf{X} \end{aligned}$$

where

$$\begin{aligned} \mathbf{X} &= (x_1, \dots, x_n)^T \\ \mathbf{D}_i &= i\text{th covariance matrix} \\ &= \begin{pmatrix} \text{var}\{a_{i1}\} & \dots & \text{cov}\{a_{i1}, a_{in}\} \\ \vdots & \ddots & \vdots \\ \text{cov}\{a_{in}, a_{i1}\} & \dots & \text{var}\{a_{in}\} \end{pmatrix} \end{aligned}$$

Now

$$P\{h_i \leq b_i\} = P\left\{\frac{h_i - E\{h_i\}}{\sqrt{\text{var}\{h_i\}}} \leq \frac{b_i - E\{h_i\}}{\sqrt{\text{var}\{h_i\}}}\right\} \geq 1 - \alpha_i$$

Letting  $F$  be the CDF of the standard normal distribution, it follows that

$$P\{h_i \leq b_i\} = F\left(\frac{b_i - E\{h_i\}}{\sqrt{\text{var}\{h_i\}}}\right)$$

Let  $K_{\alpha_i}$  be the standard normal value such that

$$F(K_{\alpha_i}) = 1 - \alpha_i$$

Then the statement  $P\{h_i \leq b_i\} \geq 1 - \alpha_i$  is realized if, and only if,

$$\frac{b_i - E\{h_i\}}{\sqrt{\text{var}\{h_i\}}} \geq K_{\alpha_i}$$

This yields the following nonlinear deterministic constraint:

$$\sum_{j=1}^n E\{a_{ij}\}x_j + K_{\alpha_i} \sqrt{\mathbf{X}^T \mathbf{D}_i \mathbf{X}} \leq b_i$$

For the special case where the parameters  $a_{ij}$  are independent,  $\text{cov}\{a_{ij}, a_{i'j'}\} = 0$ , and the last constraint reduces to

$$\sum_{j=1}^n E\{a_{ij}\}x_j + K_{\alpha_i} \sqrt{\sum_{j=1}^n \text{var}\{a_{ij}\}x_j^2} \leq b_i$$

This constraint can be put in the separable programming form (Section 21.2.1) by using the substitution

$$y_i = \sqrt{\sum_{j=1}^n \text{var}\{a_{ij}\}x_j^2}, \text{ for all } i$$

Thus, the original constraint is equivalent to

$$\sum_{j=1}^n E\{a_{ij}\}x_j + K_{\alpha_i} y_i \leq b_i$$

and

$$\sum_{j=1}^n \text{var}\{a_{ij}\}x_j^2 - y_i^2 = 0$$

**Case 2.** Only  $b_i$  is normal with mean  $E\{b_i\}$  and variance  $\text{var}\{b_i\}$ . Consider the stochastic constraint

$$P\left\{b_i \geq \sum_{j=1}^n a_{ij}x_j\right\} \geq \alpha_i$$

As in case 1,

$$P\left\{\frac{b_i - E\{b_i\}}{\sqrt{\text{var}\{b_i\}}} \geq \frac{\sum_{j=1}^n a_{ij}x_j - E\{b_i\}}{\sqrt{\text{var}\{b_i\}}}\right\} \geq \alpha_i$$

This can hold true only if

$$\frac{\sum_{j=1}^n a_{ij}x_j - E\{b_i\}}{\sqrt{\text{var}\{b_i\}}} \leq K_{\alpha_i}$$

Thus, the stochastic constraint is equivalent to the deterministic linear constraint

$$\sum_{j=1}^n a_{ij}x_j \leq E\{b_i\} + K_{\alpha_i}\sqrt{\text{var}\{b_i\}}$$

**Case 3.** All  $a_{ij}$  and  $b_i$  are normal random variables. Consider the constraint

$$\sum_{j=1}^n a_{ij}x_j \leq b_i$$

This may be written

$$\sum_{j=1}^n a_{ij}x_j - b_i \leq 0$$

Because all  $a_{ij}$  and  $b_i$  are normal,  $\sum_{j=1}^n a_{ij}x_j - b_i$  is also normal. This shows that the chance constraint reduces to the situation in case 1 and is treated in a similar manner.

#### Example 21.2-4

Consider the chance-constrained problem

$$\text{Maximize } z = 5x_1 + 6x_2 + 3x_3$$

subject to

$$P\{a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \leq 8\} \geq .95$$

$$P\{5x_1 + x_2 + 6x_3 \leq b_2\} \geq .10$$

$$x_1, x_2, x_3 \geq 0$$

Assume that the parameters  $a_{1j}$ ,  $j = 1, 2, 3$ , are independent and normally distributed random variables with the following means and variances:

$$E\{a_{11}\} = 1, E\{a_{12}\} = 3, E\{a_{13}\} = 9$$

$$\text{var}\{a_{11}\} = 25, \text{var}\{a_{12}\} = 16, \text{var}\{a_{13}\} = 4$$

The parameter  $b_2$  is normally distributed with mean 7 and variance 9.

From standard normal tables in Appendix B (or *excelStatTables.xls*),

$$K_{\alpha_1} = K_{.05} \approx 1.645, \quad K_{\alpha_2} = K_{.10} \approx 1.285$$

For the first constraint, the equivalent deterministic constraint is

$$x_1 + 3x_2 + 9x_3 + 1.645\sqrt{25x_1^2 + 16x_2^2 + 4x_3^2} \leq 8$$

and for the second constraint

$$5x_1 + x_2 + 6x_3 \leq 7 + 1.285(3) = 10.855$$

The resulting problem can be solved as a nonlinear program (using AMPL or Solver), or it can be converted to a separable program as follows:

$$y^2 = 25x_1^2 + 16x_2^2 + 4x_3^2$$

The problem becomes

$$\text{Maximize } z = 5x_1 + 6x_2 + 3x_3$$

subject to

$$x_1 + 3x_2 + 9x_3 + 1.645y \leq 8$$

$$25x_1^2 + 16x_2^2 + 4x_3^2 - y^2 = 0$$

$$5x_1 + x_2 + 6x_3 \leq 10.855$$

$$x_1, x_2, x_3, y \geq 0$$

The problem can be solved by separable programming. Also, Excel file *excelCCP.xls* can be used to solve the nonlinear problem directly.

#### PROBLEM SET 21.2C

\*1. Convert the following stochastic problem into an equivalent deterministic model.

$$\text{Maximize } z = x_1 + 2x_2 + 5x_3$$



subject to

$$P\{a_1x_1 + 3x_2 + a_3x_3 \leq 10\} \geq 0.9$$

$$P\{7x_1 + 5x_2 + x_3 \leq b_2\} \geq 0.1$$

$$x_1, x_2, x_3 \geq 0$$

Assume that  $a_1$  and  $a_3$  are independent and normally distributed random variables with means  $E\{a_1\} = 2$  and  $E\{a_3\} = 5$  and variances  $\text{var}\{a_1\} = 9$  and  $\text{var}\{a_3\} = 16$ , and that  $b_2$  is normally distributed with mean 15 and variance 25.

2. Consider the following stochastic programming model:

$$\text{Maximize } z = x_1 + x_2^2 + x_3$$

subject to

$$P\{x_1^2 + a_2x_2^3 + a_3\sqrt{x_3} \leq 10\} \geq 0.9$$

$$x_1, x_2, x_3 \geq 0$$

The parameters  $a_2$  and  $a_3$  are independent and normally distributed random variables with means 5 and 2, and variance 16 and 25, respectively. Convert the problem into a (deterministic) separable programming form.

#### 21.2.4 Linear Combinations Method

This method deals with the following problem in which all constraints are linear:

$$\text{Maximize } z = f(\mathbf{X})$$

subject to

$$\mathbf{AX} \leq \mathbf{b}, \mathbf{X} \geq 0$$

The procedure is based on the steepest-ascent (gradient) method (Section 21.1.2). However, the direction specified by the gradient vector may not yield a feasible solution for the constrained problem. Also, the gradient vector will not necessarily be null at the optimum (constrained) point. The steepest ascent method thus must be modified to handle the constrained case.

Let  $\mathbf{X}_k$  be the feasible trial point at iteration  $k$ . The objective function  $f(\mathbf{X})$  can be expanded in the neighborhood of  $\mathbf{X}_k$  using Taylor's series. This gives

$$f(\mathbf{X}) \approx f(\mathbf{X}_k) + \nabla f(\mathbf{X}_k)(\mathbf{X} - \mathbf{X}_k) = (f(\mathbf{X}_k) - \nabla f(\mathbf{X}_k)\mathbf{X}_k) + \nabla f(\mathbf{X}_k)\mathbf{X}$$

The procedure calls for determining a feasible point  $\mathbf{X} = \mathbf{X}^*$  such that  $f(\mathbf{X})$  is maximized subject to the (linear) constraints of the problem. Because  $f(\mathbf{X}_k) - \nabla f(\mathbf{X}_k)\mathbf{X}_k$  is a constant, the problem for determining  $\mathbf{X}^*$  reduces to solving the following linear program:

$$\text{Maximize } w_k(\mathbf{X}) = \nabla f(\mathbf{X}_k)\mathbf{X}$$

subject to

$$\mathbf{AX} \leq \mathbf{b}, \mathbf{X} \geq 0$$

Given that  $w_k$  is constructed from the gradient of  $f(\mathbf{X})$  at  $\mathbf{X}_k$ , an improved solution point can be secured if and only if  $w_k(\mathbf{X}^*) > w_k(\mathbf{X}_k)$ . From Taylor's expansion, the condition does not guarantee that  $f(\mathbf{X}^*) > f(\mathbf{X}_k)$  unless  $\mathbf{X}^*$  is in the neighborhood of  $\mathbf{X}_k$ . However, given  $w_k(\mathbf{X}^*) > w_k(\mathbf{X}_k)$ , there must exist a point  $\mathbf{X}_{k+1}$  on the line segment  $(\mathbf{X}_k, \mathbf{X}^*)$  such that  $f(\mathbf{X}_{k+1}) > f(\mathbf{X}_k)$ . The objective is to determine  $\mathbf{X}_{k+1}$ . Define

$$\mathbf{X}_{k+1} = (1 - r)\mathbf{X}_k + r\mathbf{X}^* = \mathbf{X}^k + r(\mathbf{X}^* - \mathbf{X}_k), 0 < r \leq 1$$

This means that  $\mathbf{X}_{k+1}$  is a linear combination of  $\mathbf{X}_k$  and  $\mathbf{X}^*$ . Because  $\mathbf{X}_k$  and  $\mathbf{X}^*$  are two feasible points in a convex solution space,  $\mathbf{X}_{k+1}$  is also feasible. In terms of the steepest-ascent method (Section 21.1.2), the parameter  $r$  represents step size.

The point  $\mathbf{X}_{k+1}$  is determined such that  $f(\mathbf{X})$  is maximized. Because  $\mathbf{X}_{k+1}$  is a function of  $r$  only,  $\mathbf{X}_{k+1}$  is determined by maximizing

$$h(r) = f(\mathbf{X}_k + r(\mathbf{X}^* - \mathbf{X}_k))$$

The procedure is repeated until, at the  $k$ th iteration,  $w_k(\mathbf{X}^*) \leq w_k(\mathbf{X}_k)$ . At this point, no further improvements are possible, and the process terminates with  $\mathbf{X}_k$  as the best solution point.

The linear programming problems generated at the successive iterations differ only in the coefficients of the objective function. Post-optimal analysis procedures presented in Section 4.5 thus may be used to carry out calculations efficiently.

#### Example 21.2-5

Consider the quadratic programming of Example 21.2-3.

$$\text{Maximize } f(\mathbf{X}) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

subject to

$$x_1 + 2x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

Let the initial trial point be  $\mathbf{X}_0 = (\frac{1}{2}, \frac{1}{2})$ , which is feasible. Now

$$\nabla f(\mathbf{X}) = (4 - 4x_1 - 2x_2, 6 - 2x_1 - 4x_2)$$

#### Iteration 1

$$\nabla f(\mathbf{X}_0) = (1, 3)$$

The associated linear program maximizes  $w_1 = x_1 + 3x_2$  subject to the constraints of the original problem. This gives the optimal solution  $\mathbf{X}^* = (0, 1)$ . The values of  $w_1$  at  $\mathbf{X}_0$  and  $\mathbf{X}^*$  equal 2 and 3, respectively. Hence, a new trial point is determined as

$$\mathbf{X}_1 = (\frac{1}{2}, \frac{1}{2}) + r[(0, 1) - (\frac{1}{2}, \frac{1}{2})] = (\frac{1-r}{2}, \frac{1+r}{2})$$

The maximization of

$$h(r) = f\left(\frac{1-r}{2}, \frac{1+r}{2}\right)$$

yields  $r_1 = 1$ . Thus  $\mathbf{X}_1 = (0, 1)$  with  $f(\mathbf{X}_1) = 4$ .

### Iteration 2

$$\nabla f(\mathbf{X}_1) = (2, 2)$$

The objective function of the new linear programming problem is  $w_2 = 2x_1 + 2x_2$ . The optimum solution to this problem yields  $\mathbf{X}^* = (2, 0)$ . Because the values of  $w_2$  at  $\mathbf{X}_1$  and  $\mathbf{X}^*$  are 2 and 4, respectively, a new trial point must be determined. Thus

$$\mathbf{X}_2 = (0, 1) + r[(2, 0) - (0, 1)] = (2r, 1 - r)$$

The maximization of

$$h(r) = f(2r, 1 - r)$$

yields  $r_2 = \frac{1}{6}$ . Thus  $\mathbf{X}_2 = \left(\frac{1}{3}, \frac{5}{6}\right)$  with  $f(\mathbf{X}_2) \approx 4.16$ .

### Iteration 3

$$\nabla f(\mathbf{X}_2) = (1, 2)$$

The corresponding objective function is  $w_3 = x_1 + 2x_2$ . The optimum solution of this problem yields the alternative solutions  $\mathbf{X}^* = (0, 1)$  and  $\mathbf{X}^* = (2, 0)$ . The value of  $w_3$  for both points equals its value at  $\mathbf{X}_2$ . Consequently, no further improvements are possible. The *approximate* optimum solution is  $\mathbf{X}_2 = \left(\frac{1}{3}, \frac{5}{6}\right)$  with  $f(\mathbf{X}_2) \approx 4.16$ . This happens to be the exact optimum.

## PROBLEM SET 21.2D

1. Solve the following problem by the linear combinations method.

$$\text{Minimize } f(\mathbf{X}) = x_1^3 + x_2^3 - 3x_1x_2$$

subject to

$$3x_1 + x_2 \leq 3$$

$$5x_1 - 3x_2 \leq 5$$

$$x_1, x_2 \geq 0$$

## 21.2.5 SUMT Algorithm

In this section, a more general gradient method is presented. It is assumed that the objective function  $f(\mathbf{X})$  is concave and each constraint function  $g_i(\mathbf{X})$  is convex. Moreover, the solution space must have an interior. This rules out both implicit and explicit use of *equality* constraints.

The SUMT (Sequential Unconstrained Maximization Technique) algorithm is based on transforming the constrained problem into an equivalent *unconstrained* problem. The procedure is more or less similar to the Lagrange multipliers method. The transformed problem can then be solved using the steepest-ascent method (Section 21.1.2).

To clarify the concept, consider the new function

$$p(\mathbf{X}, t) = f(\mathbf{X}) + t\left(\sum_{i=1}^m \frac{1}{g_i(\mathbf{X})} - \sum_{j=1}^n \frac{1}{x_j}\right)$$

where  $t$  is a nonnegative parameter. The second summation sign accounts for the non-negativity constraints, which must be put in the form  $-x_j \leq 0$  to be consistent with the original constraints. Because  $g_i(\mathbf{X})$  is convex,  $\frac{1}{g_i(\mathbf{X})}$  is concave. This means that  $p(\mathbf{X}, t)$  is concave in  $\mathbf{X}$ . Consequently,  $p(\mathbf{X}, t)$  possesses a unique maximum. Optimization of the original constrained problem is equivalent to optimization of  $p(\mathbf{X}, t)$ .

The algorithm is initiated by arbitrarily selecting an initial *nonnegative* value for  $t$ . An initial point  $\mathbf{X}_0$  is selected as the first trial solution. This point must be an interior point—that is, it must not lie on the boundaries of the solution space. Given the value of  $t$ , the steepest-ascent method is used to determine the corresponding optimal solution (maximum) of  $p(\mathbf{X}, t)$ .

The new solution point will always be an interior point, because if the solution point is close to the boundaries, at least one of the functions  $\frac{1}{g_i(\mathbf{X})}$  or  $-\frac{1}{x_i}$  will acquire a very large negative value. Because the objective is to maximize  $p(\mathbf{X}, t)$ , such solution points are automatically excluded. The main result is that successive solution points will always be interior points. Consequently, the problem can always be treated as an unconstrained case.

Once the optimum solution corresponding to a given value of  $t$  is obtained, a new value of  $t$  is generated, and the optimization process (using the steepest-ascent method) is repeated. If  $t'$  is the current value of  $t$ , the next value,  $t''$ , must be selected such that  $0 < t'' < t'$ .

The SUMT algorithm ends when, for two successive values of  $t$ , the corresponding *optimum* values of  $\mathbf{X}$  obtained by maximizing  $p(\mathbf{X}, t)$  are approximately the same. At this point, further trials will produce little improvement.

Actual implementation of SUMT involves more details than have been presented here. Specifically, the selection of an initial value of  $t$  is an important factor that can affect the speed of convergence. Further, the determination of an initial interior point may require special techniques. These details can be found in Fiacco and McCormick (1968).

## BIBLIOGRAPHY

- Bazaraa, M., H. Sherali, and C. Shetty, *Nonlinear Programming, Theory and Algorithms*, 2nd ed., Wiley, New York, 1993.
- Beightler, C., D. Phillips, and D. Wilde, *Foundations of Optimization*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1979.

Fiacco, A., and G. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968.

Fletcher, R., *Practical Methods of Optimization*, 2nd ed., Wiley, New York, 2000.

Luenberger, D., *Linear and Nonlinear Programming*, Kluwer Academic Publishers, Boston, 2003.

Rardin, D., *Optimization in Operations Research*, Prentice Hall, NJ, 1998.

APPENDIX A

Statistical Tables<sup>1</sup>

TABLE A.1 Normal Distribution Function

$$F(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{t^2}{2}} dt$$

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767

<sup>1</sup>Spreadsheet *excelStatTable.xls* replaces the (hard-copy) statistical tables of 12 common distributions, including the ones presented in this appendix.