

PART III

LINEAR PROGRAMMING

CHAPTER 15

INTRODUCTION TO LINEAR PROGRAMMING

15.1 Brief History of Linear Programming

The goal of linear programming is to determine the values of decision variables that maximize or minimize a linear objective function, where the decision variables are subject to linear constraints. A linear programming problem is a special case of a general constrained optimization problem. In the general setting, the goal is to find a point that minimizes the objective function and at the same time satisfies the constraints. We refer to any point that satisfies the constraints as a *feasible point*. In a linear programming problem, the objective function is linear, and the set of feasible points is determined by a set of linear equations and/or inequalities.

In this part we study methods for solving linear programming problems. Linear programming methods provide a way of choosing the best feasible point among the many possible feasible points. In general, the number of feasible points is infinitely large. However, as we shall see, the solution to a linear programming problem can be found by searching through a particular finite number of feasible points, known as *basic feasible solutions*. Therefore, in principle, we can solve a linear programming problem simply by comparing

the finite number of basic feasible solutions and finding one that minimizes or maximizes the objective function—we refer to this approach as the *brute-force approach*. For most practical decision problems, even this finite number of basic feasible solutions is so large that the method of choosing the best solution by comparing them to each other is impractical. To get a feel for the amount of computation needed in a brute-force approach, consider the following example. Suppose that we have a small factory with 20 different machines producing 20 different parts. Assume that any of the machines can produce any part. We also assume that the time for producing each part on each machine is known. The problem then is to assign a part to each machine so that the overall production time is minimized. We see that there are $20!$ (20 factorial) possible assignments. The brute-force approach to solving this assignment problem would involve writing down all the possible assignments and then choosing the best one by comparing them. Suppose that we have at our disposal a computer that takes $1 \mu\text{s}$ (10^{-6} second) to determine each assignment. Then, to find the best (optimal) assignment this computer would need 77,147 years (working 24 hours a day, 365 days a year) to find the best solution. An alternative approach to solving this problem is to use experienced planners to optimize this assignment problem. Such an approach relies on heuristics. Heuristics come close, but give suboptimal answers. Heuristics that do reasonably well, with an error of, say, 10%, may still not be good enough. For example, in a business that operates on large volumes and a small profit margin, a 10% error could mean the difference between loss and profit.

Efficient methods for solving linear programming problems became available in the late 1930s. In 1939, Kantorovich presented a number of solutions to some problems related to production and transportation planning. During World War II, Koopmans contributed significantly to the solution of transportation problems. Kantorovich and Koopmans were awarded a Nobel Prize in Economics in 1975 for their work on the theory of optimal allocation of resources. In 1947, Dantzig developed a new method for solving linear programs, known today as the *simplex method* (see [34] for Dantzig's own treatment of the algorithm). In the following chapters we discuss the simplex method in detail. The simplex method is efficient and elegant and has been declared one of the 10 algorithms with the greatest influence on the development and practice of science and engineering in the twentieth century [40].

The simplex method has the undesirable property that in the worst case, the number of steps (and hence total time) required to find a solution grows exponentially with the number of variables. Thus, the simplex method is said to have *exponential worst-case complexity*. This led to an interest in devising algorithms for solving linear programs that have polynomial complexity—algorithms that find a solution in an amount of time that is bounded by a polynomial in the number of variables. Khachiyan, in 1979, was the first to devise such an algorithm. However, his algorithm gained more theoretical than practical interest. Then, in 1984, Karmarkar proposed a new linear program-

ming algorithm that has polynomial complexity and appears to solve some complicated real-world problems of scheduling, routing, and planning more efficiently than the simplex method. Karmarkar's work led to the development of many other nonsimplex methods commonly referred to as *interior-point methods*. This approach is currently still an active research area. For more details on Karmarkar's and related algorithms, see [42], [55], [71], [119], and [124]. Some basic ideas illustrating Khachiyan's and Karmarkar's algorithms are presented in Chapter 18.

15.2 Simple Examples of Linear Programs

Formally, a linear program is an optimization problem of the form

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{m \times n}$. The vector inequality $\mathbf{x} \geq \mathbf{0}$ means that each component of \mathbf{x} is nonnegative. Several variations of this problem are possible; for example, instead of minimizing, we can maximize, or the constraints may be in the form of inequalities, such as $\mathbf{Ax} \geq \mathbf{b}$ or $\mathbf{Ax} \leq \mathbf{b}$. We also refer to these variations as linear programs. In fact, as we shall see later, these variations can all be rewritten into the standard form shown above.

The purpose of this section is to give some simple examples of linear programming problems illustrating the importance and the various applications of linear programming methods.

Example 15.1 This example is adapted from [123]. A manufacturer produces four different products: X_1 , X_2 , X_3 , and X_4 . There are three inputs to this production process: labor in person-weeks, kilograms of raw material A, and boxes of raw material B. Each product has different input requirements. In determining each week's production schedule, the manufacturer cannot use more than the available amounts of labor and the two raw materials. The relevant information is presented in Table 15.1. Every production decision must satisfy the restrictions on the availability of inputs. These constraints can be written using the data in Table 15.1. In particular, we have

$$\begin{aligned} x_1 + 2x_2 + x_3 + 2x_4 &\leq 20 \\ 6x_1 + 5x_2 + 3x_3 + 2x_4 &\leq 100 \\ 3x_1 + 4x_2 + 9x_3 + 12x_4 &\leq 75. \end{aligned}$$

Because negative production levels are not meaningful, we must impose the following nonnegativity constraints on the production levels:

$$x_i \geq 0, \quad i = 1, 2, 3, 4.$$

Table 15.1 Data for Example 15.1

Inputs	Product				Input Availabilities
	X_1	X_2	X_3	X_4	
Person-weeks	1	2	1	2	20
Kilograms of material A	6	5	3	2	100
Boxes of material B	3	4	9	12	75
Production levels	x_1	x_2	x_3	x_4	

Now, suppose that one unit of product X_1 sells for \$6, and X_2 , X_3 , and X_4 sell for \$4, \$7, and \$5, respectively. Then, the total revenue for any production decision (x_1, x_2, x_3, x_4) is

$$f(x_1, x_2, x_3, x_4) = 6x_1 + 4x_2 + 7x_3 + 5x_4.$$

The problem is then to maximize f subject to the given constraints (the three inequalities and four nonnegativity constraints). Using vector notation with

$$\mathbf{x} = [x_1, x_2, x_3, x_4]^\top,$$

the problem can be written in the compact form

$$\begin{aligned} & \text{maximize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where

$$\begin{aligned} \mathbf{c}^\top &= [6, 4, 7, 5], \\ \mathbf{A} &= \begin{bmatrix} 1 & 2 & 1 & 2 \\ 6 & 5 & 3 & 2 \\ 3 & 4 & 9 & 12 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 20 \\ 100 \\ 75 \end{bmatrix}. \end{aligned}$$

■

Another example that illustrates linear programming involves determining the most economical diet that satisfies the basic minimum requirements for good health.

Example 15.2 Diet Problem. This example is adapted from [88]. Assume that n different food types are available. The j th food sells at a price c_j per unit. In addition, there are m basic nutrients. To achieve a balanced diet, you must receive at least b_i units of the i th nutrient per day. Assume that each

unit of food j contains a_{ij} units of the i th nutrient. Denote by x_j the number of units of food j in the diet. The objective is to select the x_j to minimize the total cost of the diet:

$$\text{minimize } c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

subject to the nutritional constraints

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq b_2$$

 \vdots

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq b_m,$$

and the nonnegativity constraints

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0.$$

In the more compact vector notation, this problem becomes

$$\text{minimize } \mathbf{c}^\top \mathbf{x}$$

$$\text{subject to } \mathbf{A}\mathbf{x} \geq \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0},$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top$ is an n -dimensional column vector, \mathbf{c}^\top is an n -dimensional row vector, \mathbf{A} is an $m \times n$ matrix, and \mathbf{b} is an m -dimensional column vector. We call this problem the *diet problem* and will return to it in Chapter 17. ■

In the next example we consider a linear programming problem that arises in manufacturing.

Example 15.3 A manufacturer produces two different products, X_1 and X_2 , using three machines: M_1 , M_2 , and M_3 . Each machine can be used for only a limited amount of time. Production times of each product on each machine are given in Table 15.2. The objective is to maximize the combined time of utilization of all three machines.

Every production decision must satisfy the constraints on the available time. These restrictions can be written down using data from Table 15.2. In particular, we have

$$x_1 + x_2 \leq 8,$$

$$x_1 + 3x_2 \leq 18,$$

$$2x_1 + x_2 \leq 14,$$

where x_1 and x_2 denote the production levels. The combined production time of all three machines is

$$f(x_1, x_2) = 4x_1 + 5x_2.$$

Table 15.2 Data for Example 15.3

Machine	Production time (hours/unit)		(hours)
	X_1	X_2	
M_1	1	1	8
M_2	1	3	18
M_3	2	1	14
Total	4	5	

Thus, writing $\mathbf{x} = [x_1, x_2]^\top$, the problem in compact notation has the form

$$\begin{aligned} & \text{maximize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where

$$\begin{aligned} \mathbf{c}^\top &= [4, 5], \\ \mathbf{A} &= \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 2 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 8 \\ 18 \\ 14 \end{bmatrix}. \end{aligned}$$

■

In the following example we discuss an application of linear programming in transportation.

Example 15.4 A manufacturing company has plants in cities A, B, and C. The company produces and distributes its product to dealers in various cities. On a particular day, the company has 30 units of its product in A, 40 in B, and 30 in C. The company plans to ship 20 units to D, 20 to E, 25 to F, and 35 to G, following orders received from dealers. The transportation costs per unit of each product between the cities are given in Table 15.3. In the table, the quantities supplied and demanded appear at the right and along the bottom of the table. The quantities to be transported from the plants to different destinations are represented by the decision variables.

Table 15.3 Data for Example 15.4

To From	D	E	F	G	Supply
A	\$7	\$10	\$14	\$8	30
B	\$7	\$11	\$12	\$6	40
C	\$5	\$8	\$15	\$9	30
Demand	20	20	25	35	100

This problem can be stated in the form

$$\begin{aligned} \text{minimize} \quad & 7x_{11} + 10x_{12} + 14x_{13} + 8x_{14} + 7x_{21} + 11x_{22} + 12x_{23} \\ & + 6x_{24} + 5x_{31} + 8x_{32} + 15x_{33} + 9x_{34} \end{aligned}$$

$$\text{subject to} \quad x_{11} + x_{12} + x_{13} + x_{14} = 30$$

$$x_{21} + x_{22} + x_{23} + x_{24} = 40$$

$$x_{31} + x_{32} + x_{33} + x_{34} = 30$$

$$x_{11} + x_{21} + x_{31} = 20$$

$$x_{12} + x_{22} + x_{32} = 20$$

$$x_{13} + x_{23} + x_{33} = 25$$

$$x_{14} + x_{24} + x_{34} = 35$$

$$x_{11}, x_{12}, \dots, x_{34} \geq 0.$$

In this problem one of the constraint equations is redundant because it can be derived from the rest of the constraint equations. The mathematical formulation of the transportation problem is then in a linear programming form with twelve (3×4) decision variables and six ($3 + 4 - 1$) linearly independent constraint equations. Obviously, we also require nonnegativity of the decision variables, since a negative shipment is impossible and does not have a valid interpretation. ■

Next, we give an example of a linear programming problem arising in electrical engineering.

Example 15.5 This example is adapted from [100]. Figure 15.1 shows an electric circuit that is designed to use a 30-V source to charge 10-V, 6-V, and 20-V batteries connected in parallel. Physical constraints limit the currents I_1, I_2, I_3, I_4 , and I_5 to a maximum of 4 A, 3 A, 3 A, 2 A, and 2 A, respectively. In addition, the batteries must not be discharged; that is, the currents I_1, I_2, I_3, I_4 , and I_5 must not be negative. We wish to find the values of the currents I_1, \dots, I_5 such that the total power transferred to the batteries is maximized.

The total power transferred to the batteries is the sum of the powers transferred to each battery and is given by $10I_2 + 6I_4 + 20I_5$ W. From the circuit in

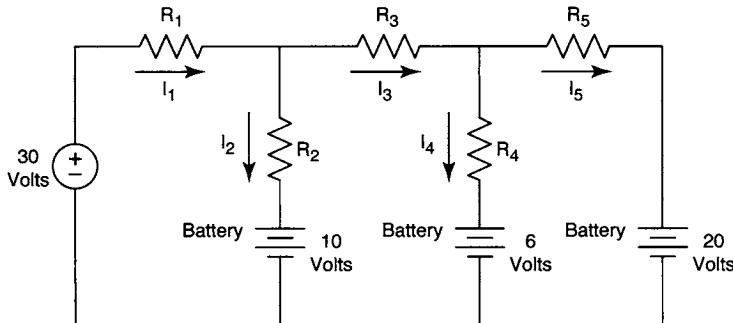


Figure 15.1 Battery charger circuit for Example 15.5.

Figure 15.1, we observe that the currents satisfy the constraints $I_1 = I_2 + I_3$ and $I_3 = I_4 + I_5$. Therefore, the problem can be posed as the following linear program:

$$\begin{aligned} & \text{maximize} && 10I_2 + 6I_4 + 20I_5 \\ & \text{subject to} && I_1 = I_2 + I_3 \\ & && I_3 = I_4 + I_5 \\ & && I_1 \leq 4 \\ & && I_2 \leq 3 \\ & && I_3 \leq 3 \\ & && I_4 \leq 2 \\ & && I_5 \leq 2 \\ & && I_1, I_2, I_3, I_4, I_5 \geq 0. \end{aligned}$$

■

Finally, we present an example from wireless communications.

Example 15.6 Consider the wireless communication system shown in Figure 15.2. There are n “mobile” users. For each $i = 1, \dots, n$, user i transmits a signal to the base station with power p_i and an attenuation factor of h_i (i.e., the actual signal power received at the base station from user i is $h_i p_i$). When the base station is receiving from user i , the total power received from all other users is considered *interference* (i.e., the interference for user i is $\sum_{j \neq i} h_j p_j$). For the communication with user i to be reliable, the signal-to-interference ratio must exceed a threshold γ_i , where the “signal” is the power received from user i .

We are interested in minimizing the total power transmitted by all users subject to having reliable communications for all users. We can formulate the

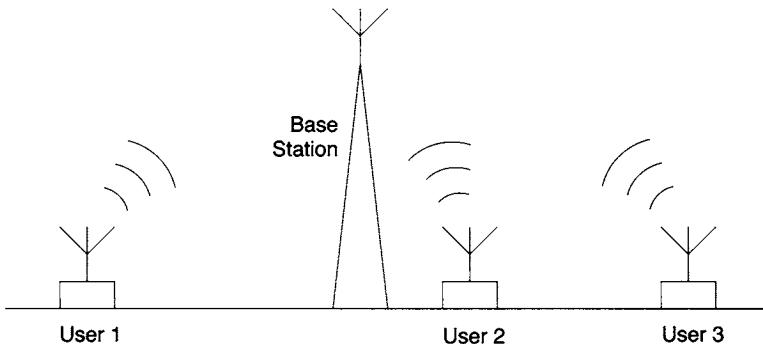


Figure 15.2 Wireless communication system in Example 15.6.

problem as a linear programming problem of the form

$$\begin{aligned} & \text{minimize } \mathbf{c}^\top \mathbf{x} \\ & \text{subject to } \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

We proceed as follows. The total power transmitted is $p_1 + \dots + p_n$. The signal-to-interference ratio for user i is

$$\frac{h_i p_i}{\sum_{j \neq i} h_j p_j}.$$

Hence, the problem can be written as

$$\begin{aligned} & \text{minimize } p_1 + \dots + p_n \\ & \text{subject to } \frac{h_i p_i}{\sum_{j \neq i} h_j p_j} \geq \gamma_i, \quad i = 1, \dots, n \\ & \quad p_1, \dots, p_n \geq 0. \end{aligned}$$

We can write the above as the linear programming problem

$$\begin{aligned} & \text{minimize } p_1 + \dots + p_n \\ & \text{subject to } h_i p_i - \gamma_i \sum_{j \neq i} h_j p_j \geq 0, \quad i = 1, \dots, n \\ & \quad p_1, \dots, p_n \geq 0. \end{aligned}$$

In matrix form, we have

$$\mathbf{c} = [1, \dots, 1]^\top$$

$$\mathbf{A} = \begin{bmatrix} h_1 & -\gamma_1 h_2 & \cdots & -\gamma_1 h_n \\ -\gamma_2 h_1 & h_2 & \cdots & -\gamma_2 h_n \\ \vdots & \ddots & \ddots & \vdots \\ -\gamma_n h_1 & -\gamma_n h_2 & \cdots & h_n \end{bmatrix}, \quad \mathbf{b} = \mathbf{0}.$$

■

For more examples of linear programming and their applications in a variety of engineering problems, we refer the reader to [1], [34], [35], [46], and [109]. For applications of linear programming to the design of control systems, see [33]. Linear programming also provides the basis for theoretical applications, as, for example, in matrix game theory (discussed in [18]).

15.3 Two-Dimensional Linear Programs

Many fundamental concepts of linear programming are easily illustrated in two-dimensional space. Therefore, we consider linear problems in \mathbb{R}^2 before discussing general linear programming problems.

Consider the following linear program (adapted from [123]):

$$\begin{aligned} & \text{maximize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{x} = [x_1, x_2]^\top$ and

$$\mathbf{c}^\top = [1, 5],$$

$$\mathbf{A} = \begin{bmatrix} 5 & 6 \\ 3 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 30 \\ 12 \end{bmatrix}.$$

First, we note that the set of equations $\{\mathbf{c}^\top \mathbf{x} = x_1 + 5x_2 = f, f \in \mathbb{R}\}$ specifies a family of straight lines in \mathbb{R}^2 . Each member of this family can be obtained by setting f equal to some real number. Thus, for example, $x_1 + 5x_2 = -5$, $x_1 + 5x_2 = 0$, and $x_1 + 5x_2 = 3$ are three parallel lines belonging to the family. Now, suppose that we try to choose several values for x_1 and x_2 and observe how large we can make f while still satisfying the constraints on x_1 and x_2 . We first try $x_1 = 1$ and $x_2 = 3$. This point satisfies the constraints. For this point, $f = 16$. If we now select $x_1 = 0$ and $x_2 = 5$, then $f = 25$ and this point yields a larger value for f than does $\mathbf{x} = [1, 3]^\top$. There are infinitely many points $[x_1, x_2]^\top$ satisfying the constraints. Therefore, we need a better

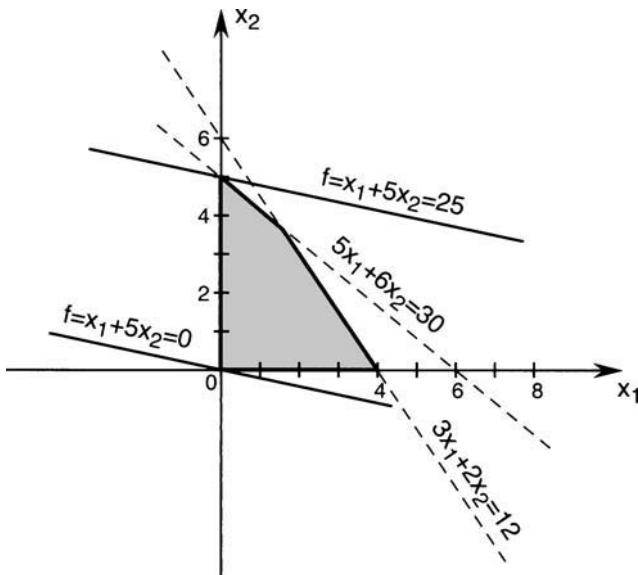


Figure 15.3 Geometric solution of a linear program in \mathbb{R}^2 .

method than trial and error to solve the problem. In the following sections we develop a systematic approach that simplifies considerably the process of solving linear programming problems.

For the example above we can easily solve the problem using geometric arguments. First let us sketch the constraints in \mathbb{R}^2 . The region of feasible points (the set of points \mathbf{x} satisfying the constraints $A\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$) is depicted by the shaded region in Figure 15.3.

Geometrically, maximizing $\mathbf{c}^\top \mathbf{x} = x_1 + 5x_2$ subject to the constraints can be thought of as finding the straight line $f = x_1 + 5x_2$ that intersects the shaded region and has the largest f . The coordinates of the point of intersection will then yield a maximum value of $\mathbf{c}^\top \mathbf{x}$. In our example, the point $[0, 5]^\top$ is the solution (see Figure 15.3).

Example 15.7 Suppose that you are given two different types of concrete. The first type contains 30% cement, 40% gravel, and 30% sand (all percentages of weight). The second type contains 10% cement, 20% gravel, and 70% sand. The first type of concrete costs \$5 per pound and the second type costs \$1 per pound. How many pounds of each type of concrete should you buy and mix together so that your cost is minimized but you get a concrete mixture that has at least a total of 5 pounds of cement, 3 pounds of gravel, and 4 pounds of sand?

The problem can be represented as

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where

$$\begin{aligned} \mathbf{c}^\top &= [5, 1], \\ \mathbf{A} &= \begin{bmatrix} 0.3 & 0.1 \\ 0.4 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 5 \\ 3 \\ 4 \end{bmatrix}. \end{aligned}$$

Using the graphical method described above, we get a solution of $[0, 50]^\top$, which means that we should purchase 50 pounds of the second type of concrete. (For a variation of this problem solved using a different method, see Example 12.1.) ■

In some cases, when using the graphical method, there may be more than one point of intersection of the optimal straight line $f = \mathbf{c}^\top \mathbf{x}$ with the boundary of the feasible region. In this case all of the intersection points will yield the same value for the objective function $\mathbf{c}^\top \mathbf{x}$, and therefore any one of them is a solution.

15.4 Convex Polyhedra and Linear Programming

The goal of linear programming is to minimize (or maximize) a linear objective function

$$\mathbf{c}^\top \mathbf{x} = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$$

subject to constraints that are represented by linear equalities and/or inequalities. For the time being, let us consider only constraints of the form $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$. In this section we discuss linear programs from a geometric point of view (for a review of geometric concepts used in the section, see Chapter 4). The set of points satisfying these constraints can be represented as the intersection of a finite number of closed half-spaces. Thus, the constraints define a convex polytope. We assume, for simplicity, that this polytope is nonempty and bounded. In other words, the equations of constraints define a polyhedron M in \mathbb{R}^n . Let H be a hyperplane of support of this polyhedron. If the dimension of M is less than n , then the set of all points common to the hyperplane H and the polyhedron M coincides with M . If the dimension of M is equal to n , then the set of all points common to the hyperplane H and the polyhedron M is a face of the polyhedron. If this face is $(n - 1)$ -dimensional, then there exists only one hyperplane of support, namely, the carrier of this face. If the

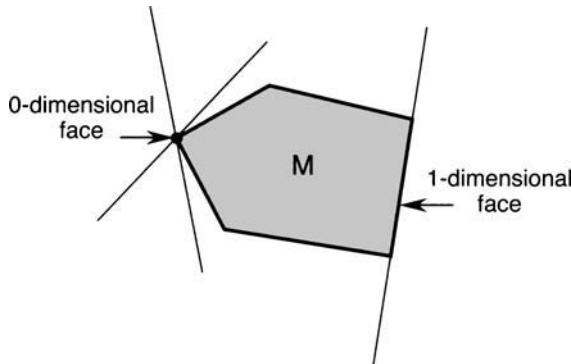


Figure 15.4 Hyperplanes of support at different boundary points of the polyhedron M .

dimension of the face is less than $n - 1$, then there exist an infinite number of hyperplanes of support whose intersection with this polyhedron yields this face (see Figure 15.4).

The goal of our linear programming problem is to maximize a linear objective function $f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x} = c_1 x_1 + \cdots + c_n x_n$ on the convex polyhedron M . Next, let H be the hyperplane defined by the equation

$$\mathbf{c}^\top \mathbf{x} = 0.$$

Draw a hyperplane of support \tilde{H} to the polyhedron M , which is parallel to H and positioned such that the vector \mathbf{c} points in the direction of the half-space that does not contain M (see Figure 15.5). The equation of the hyperplane \tilde{H} has the form

$$\mathbf{c}^\top \mathbf{x} = \beta,$$

and for all $\mathbf{x} \in M$ we have $\mathbf{c}^\top \mathbf{x} \leq \beta$. Denote by \tilde{M} the convex polyhedron that is the intersection of the hyperplane of support \tilde{H} with the polyhedron M . We now show that f is constant on \tilde{M} and that \tilde{M} is the set of all points in M for which f attains its maximum value. To this end, let \mathbf{y} and \mathbf{z} be two arbitrary points in \tilde{M} . This implies that both \mathbf{y} and \mathbf{z} belong to \tilde{H} . Hence,

$$f(\mathbf{y}) = \mathbf{c}^\top \mathbf{y} = \beta = \mathbf{c}^\top \mathbf{z} = f(\mathbf{z}),$$

which means that f is constant on \tilde{M} .

Let \mathbf{y} be a point of \tilde{M} , and let \mathbf{x} be a point of $M \setminus \tilde{M}$; that is, \mathbf{x} is a point of M that does not belong to \tilde{M} (see Figure 15.5). Then,

$$\mathbf{c}^\top \mathbf{x} < \beta = \mathbf{c}^\top \mathbf{y},$$

which implies that

$$f(\mathbf{x}) < f(\mathbf{y}).$$

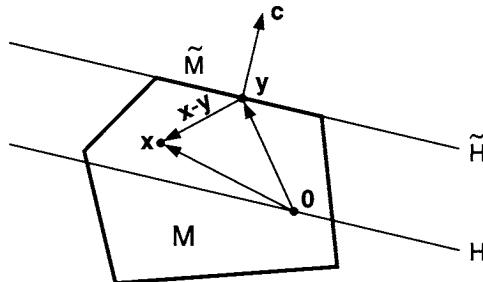


Figure 15.5 Maximization of a linear function on the polyhedron M .

Thus, the values of f at the points of M that do not belong to \tilde{M} are smaller than the values at points of \tilde{M} . Hence, f achieves its maximum on M at points in \tilde{M} .

It may happen that \tilde{M} contains only a single point, in which case f achieves its maximum at a unique point. This occurs when the the hyperplane of support passes through an extreme point of M (see Figure 15.6).

15.5 Standard Form Linear Programs

We refer to a linear program of the form

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \end{aligned}$$

as a linear program in *standard form*. Here \mathbf{A} is an $m \times n$ matrix composed of real entries, $m < n$, $\text{rank } \mathbf{A} = m$. Without loss of generality, we assume that

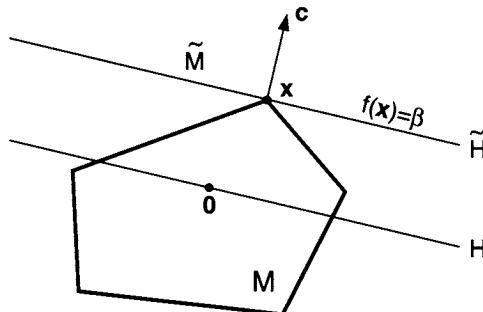


Figure 15.6 Unique maximum point of f on the polyhedron M .

$\mathbf{b} \geq \mathbf{0}$. If a component of \mathbf{b} is negative, say the i th component, we multiply the i th constraint by -1 to obtain a positive right-hand side.

Theorems and solution techniques for linear programs are usually stated for problems in standard form. Other forms of linear programs can be converted to the standard form, as we now show. If a linear program is in the form

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

then by introducing *surplus variables* y_i , we can convert the original problem into the standard form

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - y_i = b_i, \quad i = 1, \dots, m \\ & && x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0 \\ & && y_1 \geq 0, y_2 \geq 0, \dots, y_m \geq 0. \end{aligned}$$

In more compact notation, the formulation above can be represented as

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{Ax} - \mathbf{I}_m \mathbf{y} = [\mathbf{A}, -\mathbf{I}_m] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}, \end{aligned}$$

where \mathbf{I}_m is the $m \times m$ identity matrix.

If, on the other hand, the constraints have the form

$$\begin{aligned} & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

then we introduce *slack variables* y_i to convert the constraints into the form

$$\begin{aligned} & \mathbf{Ax} + \mathbf{I}_m \mathbf{y} = [\mathbf{A}, \mathbf{I}_m] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}, \end{aligned}$$

where \mathbf{y} is the vector of slack variables. Note that neither surplus nor slack variables contribute to the objective function $\mathbf{c}^\top \mathbf{x}$.

At first glance, it may appear that the two problems

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{Ax} \geq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \end{aligned}$$

and

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{Ax} - \mathbf{I}_m \mathbf{y} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \\ & && \mathbf{y} \geq \mathbf{0} \end{aligned}$$

are different, in that the first problem refers to the intersection of half-spaces in the n -dimensional space, whereas the second problem refers to an intersection of half-spaces and hyperplanes in the $(n+m)$ -dimensional space. It turns out that both formulations are algebraically equivalent in the sense that a solution to one of the problems implies a solution to the other. To illustrate this equivalence, we consider the following examples.

Example 15.8 Suppose that we are given the inequality constraint

$$x_1 \leq 7.$$

We convert this to an equality constraint by introducing a slack variable $x_2 \geq 0$ to obtain

$$\begin{aligned} x_1 + x_2 &= 7 \\ x_2 &\geq 0. \end{aligned}$$

Consider the sets $C_1 = \{x_1 : x_1 \leq 7\}$ and $C_2 = \{x_1 : x_1 + x_2 = 7, x_2 \geq 0\}$. Are the sets C_1 and C_2 equal? It is clear that indeed they are; in this example, we give a geometric interpretation for their equality. Consider a third set $C_3 = \{[x_1, x_2]^\top : x_1 + x_2 = 7, x_2 \geq 0\}$. From Figure 15.7 we can see that the set C_3 consists of all points on the line to the left and above the point of intersection of the line with the x_1 -axis. This set, being a subset of \mathbb{R}^2 , is of course not the same set as the set C_1 (a subset of \mathbb{R}). However, we can project the set C_3 onto the x_1 -axis (see Figure 15.7). We can associate with each point $x_1 \in C_1$ a point $[x_1, 0]^\top$ on the orthogonal projection of C_3 onto the x_1 -axis, and vice versa. Note that $C_2 = \{x_1 : [x_1, x_2]^\top \in C_3\} = C_1$. ■

Example 15.9 Consider the inequality constraints

$$\begin{aligned} a_1 x_1 + a_2 x_2 &\leq b \\ x_1, x_2 &\geq 0, \end{aligned}$$

where a_1 , a_2 , and b are positive numbers. Again, we introduce a slack variable $x_3 \geq 0$ to get

$$\begin{aligned} a_1 x_1 + a_2 x_2 + x_3 &= b \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

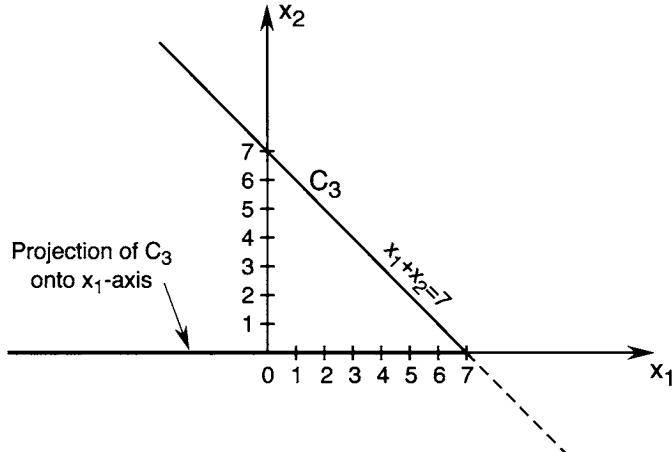


Figure 15.7 Projection of the set C_3 onto the x_1 -axis.

Define the sets

$$\begin{aligned}C_1 &= \{[x_1, x_2]^\top : a_1 x_1 + a_2 x_2 \leq b, x_1, x_2 \geq 0\}, \\C_2 &= \{[x_1, x_2]^\top : a_1 x_1 + a_2 x_2 + x_3 = b, x_1, x_2, x_3 \geq 0\}, \\C_3 &= \{[x_1, x_2, x_3]^\top : a_1 x_1 + a_2 x_2 + x_3 = b, x_1, x_2, x_3 \geq 0\}.\end{aligned}$$

We again see that C_3 is not the same as C_1 . However, the orthogonal projection of C_3 onto the (x_1, x_2) -plane allows us to associate the resulting set with the set C_1 . We associate the points $[x_1, x_2, 0]^\top$ resulting from the orthogonal projection of C_3 onto the (x_1, x_2) -plane with the points in C_1 (see Figure 15.8). Note that $C_2 = \{[x_1, x_2]^\top : [x_1, x_2, x_3]^\top \in C_3\} = C_1$. ■

Example 15.10 Suppose that we wish to maximize

$$f(x_1, x_2) = c_1 x_1 + c_2 x_2$$

subject to the constraints

$$\begin{aligned}a_{11} x_1 + a_{12} x_2 &\leq b_1 \\a_{21} x_1 + a_{22} x_2 &= b_2 \\x_1, x_2 &\geq 0,\end{aligned}$$

where, for simplicity, we assume that each $a_{ij} > 0$ and $b_1, b_2 \geq 0$. The set of feasible points is depicted in Figure 15.9. Let $C_1 \subset \mathbb{R}^2$ be the set of points satisfying the constraints.

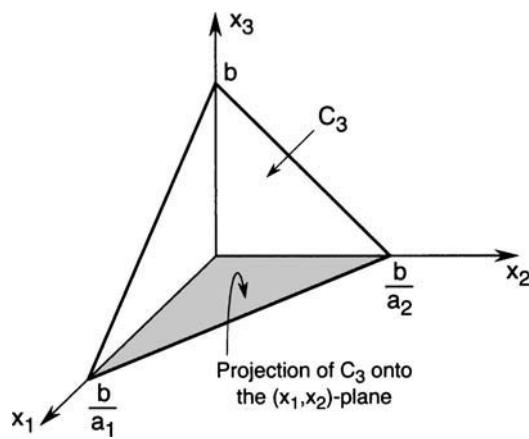


Figure 15.8 Projection of the set C_3 onto the (x_1, x_2) -plane.

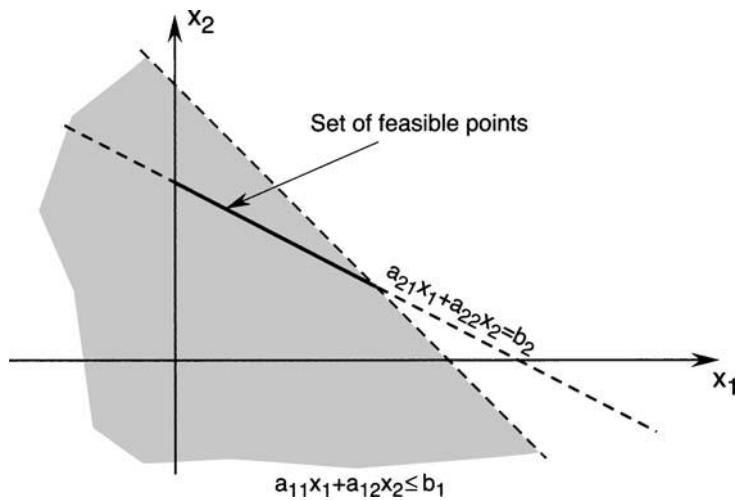


Figure 15.9 The feasible set for Example 15.10.

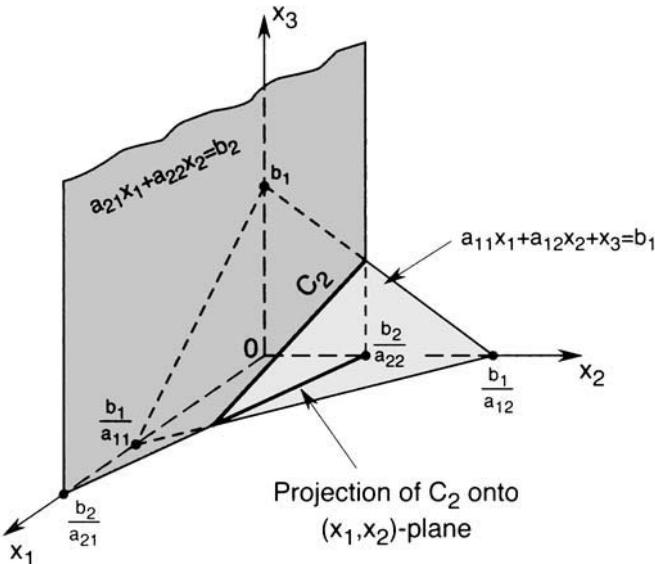


Figure 15.10 Projection of C_2 onto the (x_1, x_2) -plane.

Introducing a slack variable, we convert the constraints into standard form:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2 \\ x_i &\geq 0, \quad i = 1, 2, 3. \end{aligned}$$

Let $C_2 \subset \mathbb{R}^3$ be the set of points satisfying the constraints. As illustrated in Figure 15.10, this set is a line segment (in \mathbb{R}^3). We now project C_2 onto the (x_1, x_2) -plane. The projected set consists of the points $[x_1, x_2, 0]^\top$, with $[x_1, x_2, x_3]^\top \in C_2$ for some $x_3 \geq 0$. In Figure 15.10 this set is marked by a heavy line in the (x_1, x_2) -plane. We can associate the points on the projection with the corresponding points in the set C_1 . ■

In the following example we convert an optimization problem into a standard form linear programming problem.

Example 15.11 Consider the following optimization problem

$$\begin{aligned} &\text{maximize} && x_2 - x_1 \\ &\text{subject to} && 3x_1 = x_2 - 5 \\ & && |x_2| \leq 2 \\ & && x_1 \leq 0. \end{aligned}$$

To convert the problem into a standard form linear programming problem, we perform the following steps:

1. Change the objective function to: minimize $x_1 - x_2$.
2. Substitute $x_1 = -x'_1$.
3. Write $|x_2| \leq 2$ as $x_2 \leq 2$ and $-x_2 \leq 2$.
4. Introduce slack variables x_3 and x_4 , and convert the inequalities above to $x_2 + x_3 = 2$ and $-x_2 + x_4 = 2$.
5. Write $x_2 = u - v$, $u, v \geq 0$.

Hence, we obtain

$$\begin{aligned} & \text{minimize} && -x'_1 - u + v \\ & \text{subject to} && 3x'_1 + u - v = 5 \\ & && u - v + x_3 = 2 \\ & && v - u + x_4 = 2 \\ & && x'_1, u, v, x_3, x_4 \geq 0. \end{aligned}$$

■

15.6 Basic Solutions

We have seen in Section 15.5 that any linear programming problem involving inequalities can be converted to *standard form*, that is, a problem involving linear equations with nonnegative variables:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $m < n$, $\text{rank } \mathbf{A} = m$, and $\mathbf{b} \geq \mathbf{0}$. In the following discussion we only consider linear programming problems in standard form.

Consider the system of equalities

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

where $\text{rank } \mathbf{A} = m$. In dealing with this system of equations, we frequently need to consider a subset of columns of the matrix \mathbf{A} . For convenience, we often reorder the columns of \mathbf{A} so that the columns we are interested in appear first. Specifically, let \mathbf{B} be a square matrix whose columns are m linearly independent columns of \mathbf{A} . If necessary, we reorder the columns of \mathbf{A} so that the columns in \mathbf{B} appear first: \mathbf{A} has the form $\mathbf{A} = [\mathbf{B}, \mathbf{D}]$, where

\mathbf{D} is an $m \times (n - m)$ matrix whose columns are the remaining columns of \mathbf{A} . The matrix \mathbf{B} is nonsingular, and thus we can solve the equation

$$\mathbf{B}\mathbf{x}_B = \mathbf{b}$$

for the m -vector \mathbf{x}_B . The solution is $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$. Let \mathbf{x} be the n -vector whose first m components are equal to \mathbf{x}_B and the remaining components are equal to zero; that is, $\mathbf{x} = [\mathbf{x}_B^\top, \mathbf{0}^\top]^\top$. Then, \mathbf{x} is a solution to $\mathbf{Ax} = \mathbf{b}$.

Definition 15.1 We call $[\mathbf{x}_B^\top, \mathbf{0}^\top]^\top$ a *basic solution* to $\mathbf{Ax} = \mathbf{b}$ with respect to the basis \mathbf{B} . We refer to the components of the vector \mathbf{x}_B as *basic variables* and the columns of \mathbf{B} as *basic columns*.

If some of the basic variables of a basic solution are zero, then the basic solution is said to be a *degenerate basic solution*.

A vector \mathbf{x} satisfying $\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, is said to be a *feasible solution*.

A feasible solution that is also basic is called a *basic feasible solution*.

If the basic feasible solution is a degenerate basic solution, then it is called a *degenerate basic feasible solution*. ■

Note that in any basic feasible solution, $\mathbf{x}_B \geq \mathbf{0}$.

Example 15.12 Consider the equation $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4] = \begin{bmatrix} 1 & 1 & -1 & 4 \\ 1 & -2 & -1 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 8 \\ 2 \end{bmatrix},$$

where \mathbf{a}_i denotes the i th column of the matrix \mathbf{A} .

Then, $\mathbf{x} = [6, 2, 0, 0]^\top$ is a basic feasible solution with respect to the basis $\mathbf{B} = [\mathbf{a}_1, \mathbf{a}_2]$, $\mathbf{x} = [0, 0, 0, 2]^\top$ is a degenerate basic feasible solution with respect to the basis $\mathbf{B} = [\mathbf{a}_3, \mathbf{a}_4]$ (as well as $[\mathbf{a}_1, \mathbf{a}_4]$ and $[\mathbf{a}_2, \mathbf{a}_4]$), $\mathbf{x} = [3, 1, 0, 1]^\top$ is a feasible solution that is not basic, and $\mathbf{x} = [0, 2, -6, 0]^\top$ is a basic solution with respect to the basis $\mathbf{B} = [\mathbf{a}_2, \mathbf{a}_3]$, but is not feasible. ■

Example 15.13 As another example, consider the system of linear equations $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} 2 & 3 & -1 & -1 \\ 4 & 1 & 1 & -2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -1 \\ 9 \end{bmatrix}.$$

We now find all solutions of this system. Note that every solution \mathbf{x} of $\mathbf{Ax} = \mathbf{b}$ has the form $\mathbf{x} = \mathbf{v} + \mathbf{h}$, where \mathbf{v} is a particular solution of $\mathbf{Ax} = \mathbf{b}$ and \mathbf{h} is a solution to $\mathbf{Ax} = \mathbf{0}$.

We form the augmented matrix $[\mathbf{A}, \mathbf{b}]$ of the system:

$$[\mathbf{A}, \mathbf{b}] = \begin{bmatrix} 2 & 3 & -1 & -1 & -1 \\ 4 & 1 & 1 & -2 & 9 \end{bmatrix}.$$

Using elementary row operations, we transform this matrix into the form (see Chapter 16) given by

$$\begin{bmatrix} 1 & 0 & \frac{2}{5} & -\frac{1}{2} & \frac{14}{5} \\ 0 & 1 & -\frac{3}{5} & 0 & -\frac{11}{5} \end{bmatrix}.$$

The corresponding system of equations is given by

$$\begin{aligned} x_1 + \frac{2}{5}x_3 - \frac{1}{2}x_4 &= \frac{14}{5} \\ x_2 - \frac{3}{5}x_3 &= -\frac{11}{5}. \end{aligned}$$

Solving for the leading unknowns x_1 and x_2 , we obtain

$$\begin{aligned} x_1 &= \frac{14}{15} - \frac{2}{5}x_3 + \frac{1}{2}x_4 \\ x_2 &= -\frac{11}{5} + \frac{3}{5}x_3, \end{aligned}$$

where x_3 and x_4 are arbitrary real numbers. If $[x_1, x_2, x_3, x_4]^\top$ is a solution, then we have

$$\begin{aligned} x_1 &= \frac{14}{5} - \frac{2}{5}s + \frac{1}{2}t, \\ x_2 &= -\frac{11}{5} + \frac{3}{5}s, \\ x_3 &= s, \\ x_4 &= t, \end{aligned}$$

where we have substituted s and t for x_3 and x_4 , respectively, to indicate that they are arbitrary real numbers.

Using vector notation, we may write the system of equations above as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{14}{5} \\ -\frac{11}{5} \\ 0 \\ 0 \end{bmatrix} + s \begin{bmatrix} -\frac{2}{5} \\ \frac{3}{5} \\ 1 \\ 0 \end{bmatrix} + t \begin{bmatrix} \frac{1}{2} \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Note that we have infinitely many solutions, parameterized by $s, t \in \mathbb{R}$. For the choice $s = t = 0$ we obtain a particular solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$, given by

$$\mathbf{v} = \begin{bmatrix} \frac{14}{5} \\ -\frac{11}{5} \\ 0 \\ 0 \end{bmatrix}.$$

Any other solution has the form $\mathbf{v} + \mathbf{h}$, where

$$\mathbf{h} = s \begin{bmatrix} -\frac{2}{5} \\ \frac{3}{5} \\ 1 \\ 0 \end{bmatrix} + t \begin{bmatrix} \frac{1}{2} \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

The total number of possible basic solutions is at most

$$\binom{n}{m} = \frac{n!}{m!(n-m)!} = \frac{4!}{2!(4-2)!} = 6.$$

To find basic solutions that are feasible, we check each of the basic solutions for feasibility.

Our first candidate for a basic feasible solution is obtained by setting $x_3 = x_4 = 0$, which corresponds to the basis $\mathbf{B} = [\mathbf{a}_1, \mathbf{a}_2]$. Solving $\mathbf{Bx}_B = \mathbf{b}$, we obtain $\mathbf{x}_B = [14/5, -11/5]^\top$, and hence $\mathbf{x} = [14/5, -11/5, 0, 0]^\top$ is a basic solution that is not feasible.

For our second candidate basic feasible solution, we set $x_2 = x_4 = 0$. We have the basis $\mathbf{B} = [\mathbf{a}_1, \mathbf{a}_3]$. Solving $\mathbf{Bx}_B = \mathbf{b}$ yields $\mathbf{x}_B = [4/3, 11/3]^\top$. Hence, $\mathbf{x} = [4/3, 0, 11/3, 0]^\top$ is a basic feasible solution.

A third candidate basic feasible solution is obtained by setting $x_2 = x_3 = 0$. However, the matrix

$$\mathbf{B} = [\mathbf{a}_1, \mathbf{a}_4] = \begin{bmatrix} 2 & -1 \\ 4 & -2 \end{bmatrix}$$

is singular. Therefore, \mathbf{B} cannot be a basis, and we do not have a basic solution corresponding to $\mathbf{B} = [\mathbf{a}_1, \mathbf{a}_4]$.

We get our fourth candidate for a basic feasible solution by setting $x_1 = x_4 = 0$. We have a basis $\mathbf{B} = [\mathbf{a}_2, \mathbf{a}_3]$, resulting in $\mathbf{x} = [0, 2, 7, 0]^\top$, which is a basic feasible solution.

Our fifth candidate for a basic feasible solution corresponds to setting $x_1 = x_3 = 0$, with the basis $\mathbf{B} = [\mathbf{a}_2, \mathbf{a}_4]$. This results in $\mathbf{x} = [0, -11/5, 0, -28/5]^\top$, which is a basic solution that is not feasible.

Finally, the sixth candidate for a basic feasible solution is obtained by setting $x_1 = x_2 = 0$. This results in the basis $\mathbf{B} = [\mathbf{a}_3, \mathbf{a}_4]$, and $\mathbf{x} = [0, 0, 11/3, -8/3]^\top$, which is a basic solution but is not feasible. ■

15.7 Properties of Basic Solutions

In this section we discuss the importance of basic feasible solutions in solving linear programming (LP) problems. We first prove the fundamental theorem of LP, which states that when solving an LP problem, we need only consider

basic feasible solutions. This is because the optimal value (if it exists) is always achieved at a basic feasible solution. We need the following definitions.

Definition 15.2 Any vector \mathbf{x} that yields the minimum value of the objective function $\mathbf{c}^\top \mathbf{x}$ over the set of vectors satisfying the constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, is said to be an *optimal feasible solution*.

An optimal feasible solution that is basic is said to be an *optimal basic feasible solution*. ■

Theorem 15.1 Fundamental Theorem of LP. Consider a linear program in standard form.

1. If there exists a feasible solution, then there exists a basic feasible solution.
2. If there exists an optimal feasible solution, then there exists an optimal basic feasible solution. □

Proof. We first prove part 1. Suppose that $\mathbf{x} = [x_1, \dots, x_n]^\top$ is a feasible solution and it has p positive components. Without loss of generality, we can assume that the first p components are positive, whereas the remaining components are zero. Then, in terms of the columns of $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_p, \dots, \mathbf{a}_n]$, this solution satisfies

$$x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_p\mathbf{a}_p = \mathbf{b}.$$

There are now two cases to consider.

Case 1: If $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ are linearly independent, then $p \leq m$. If $p = m$, then the solution \mathbf{x} is basic and the proof is done. If, on the other hand, $p < m$, then, since $\text{rank } \mathbf{A} = m$, we can find $m - p$ columns of \mathbf{A} from the remaining $n - p$ columns so that the resulting set of m columns forms a basis. Hence, the solution \mathbf{x} is a (degenerate) basic feasible solution corresponding to the basis above.

Case 2: Assume that $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ are linearly dependent. Then, there exist numbers y_i , $i = 1, \dots, p$, not all zero, such that

$$y_1\mathbf{a}_1 + y_2\mathbf{a}_2 + \cdots + y_p\mathbf{a}_p = \mathbf{0}.$$

We can assume that there exists at least one y_i that is positive, for if all the y_i are nonpositive, we can multiply the equation above by -1 . Multiply the equation by a scalar ε and subtract the resulting equation from $x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_p\mathbf{a}_p = \mathbf{b}$ to obtain

$$(x_1 - \varepsilon y_1)\mathbf{a}_1 + (x_2 - \varepsilon y_2)\mathbf{a}_2 + \cdots + (x_p - \varepsilon y_p)\mathbf{a}_p = \mathbf{b}.$$

Let

$$\mathbf{y} = [y_1, \dots, y_p, 0, \dots, 0]^\top.$$

Then, for any ε we can write

$$\mathbf{A}[\mathbf{x} - \varepsilon \mathbf{y}] = \mathbf{b}.$$

Let $\varepsilon = \min\{x_i/y_i : i = 1, \dots, p, y_i > 0\}$. Then, the first p components of $\mathbf{x} - \varepsilon \mathbf{y}$ are nonnegative, and at least one of these components is zero. We then have a feasible solution with at most $p - 1$ positive components. We can repeat this process until we get linearly independent columns of \mathbf{A} , after which we are back to case 1. Therefore, part 1 is proved.

We now prove part 2. Suppose that $\mathbf{x} = [x_1, \dots, x_n]^\top$ is an optimal feasible solution and only the first p variables are nonzero. Then, we have two cases to consider. The first case is exactly the same as in part 1. The second case follows the same arguments as in part 1, but in addition we must show that $\mathbf{x} - \varepsilon \mathbf{y}$ is optimal for any ε . We do this by showing that $\mathbf{c}^\top \mathbf{y} = 0$. To this end, assume that $\mathbf{c}^\top \mathbf{y} \neq 0$. Note that for ε of sufficiently small magnitude ($|\varepsilon| \leq \min\{|x_i/y_i| : i = 1, \dots, p, y_i \neq 0\}$), the vector $\mathbf{x} - \varepsilon \mathbf{y}$ is feasible. We can choose ε such that $\mathbf{c}^\top \mathbf{x} > \mathbf{c}^\top \mathbf{x} - \varepsilon \mathbf{c}^\top \mathbf{y} = \mathbf{c}^\top (\mathbf{x} - \varepsilon \mathbf{y})$. This contradicts the optimality of \mathbf{x} . We can now use the procedure from part 1 to obtain an optimal basic feasible solution from a given optimal feasible solution. ■

Example 15.14 Consider the system of equations given in Example 15.13. Find a nonbasic feasible solution to this system and use the method in the proof of the fundamental theorem of LP to find a basic feasible solution.

Recall that solutions for the system given in Example 15.13 have the form

$$\mathbf{x} = \begin{bmatrix} \frac{14}{5} \\ -\frac{11}{5} \\ 0 \\ 0 \end{bmatrix} + s \begin{bmatrix} -\frac{2}{5} \\ \frac{3}{5} \\ 1 \\ 0 \end{bmatrix} + t \begin{bmatrix} \frac{1}{2} \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

where $s, t \in \mathbb{R}$. Note that if $s = 4$ and $t = 0$, then

$$\mathbf{x}_0 = \begin{bmatrix} \frac{6}{5} \\ \frac{1}{5} \\ \frac{4}{5} \\ 0 \end{bmatrix}$$

is a nonbasic feasible solution.

There are constants y_i , $i = 1, 2, 3$, such that

$$y_1 \mathbf{a}_1 + y_2 \mathbf{a}_2 + y_3 \mathbf{a}_3 = \mathbf{0}.$$

For example, let

$$y_1 = -\frac{2}{5},$$

$$y_2 = \frac{3}{5},$$

$$y_3 = 1.$$

Note that

$$\mathbf{A}(\mathbf{x}_0 - \varepsilon \mathbf{y}) = \mathbf{b},$$

where

$$\mathbf{y} = \begin{bmatrix} -\frac{2}{5} \\ \frac{3}{5} \\ 1 \\ 0 \end{bmatrix}.$$

If $\varepsilon = 1/3$, then

$$\mathbf{x}_1 = \mathbf{x}_0 - \varepsilon \mathbf{y} = \begin{bmatrix} \frac{4}{3} \\ 0 \\ \frac{11}{3} \\ 0 \end{bmatrix}$$

is a basic feasible solution. ■

Observe that the fundamental theorem of LP reduces the task of solving a linear programming problem to that of searching over a finite number of basic feasible solutions. That is, we need only check basic feasible solutions for optimality. As mentioned before, the total number of basic solutions is at most

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}.$$

Although this number is finite, it may be quite large. For example, if $m = 5$ and $n = 50$, then

$$\binom{n}{m} = \binom{50}{5} = 2,118,760.$$

This is potentially the number of basic feasible solutions to be checked for optimality. Therefore, a more efficient method of solving linear programs is needed. To this end, in the next section we analyze a geometric interpretation of the fundamental theorem of LP. This leads us to the simplex method for solving linear programs, which we discuss in Chapter 16.

15.8 Geometric View of Linear Programs

Recall that a set $\Theta \subset \mathbb{R}^n$ is said to be *convex* if, for every $\mathbf{x}, \mathbf{y} \in \Theta$ and every real number α , $0 < \alpha < 1$, the point $\alpha\mathbf{x} + (1 - \alpha)\mathbf{y} \in \Theta$. In other words, a set is convex if given two points in the set, every point on the line segment joining these two points is also a member of the set.

Note that the set of points satisfying the constraints

$$\mathbf{Ax} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}$$

is convex. To see this, let \mathbf{x}_1 and \mathbf{x}_2 satisfy the constraints, that is, $A\mathbf{x}_i = \mathbf{b}$, $\mathbf{x}_i \geq \mathbf{0}$, $i = 1, 2$. Then, for all $\alpha \in (0, 1)$, $A(\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2) = \alpha A\mathbf{x}_1 + (1 - \alpha)A\mathbf{x}_2 = \mathbf{b}$. Also, for $\alpha \in (0, 1)$, we have $\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2 \geq \mathbf{0}$.

Recall that a point \mathbf{x} in a convex set Θ is said to be an *extreme point* of Θ if there are no two distinct points \mathbf{x}_1 and \mathbf{x}_2 in Θ such that $\mathbf{x} = \alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2$ for some $\alpha \in (0, 1)$. In other words, an extreme point is a point that does not lie strictly within the line segment connecting two other points of the set. Therefore, if \mathbf{x} is an extreme point, and $\mathbf{x} = \alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2$ for some $\mathbf{x}_1, \mathbf{x}_2 \in \Theta$ and $\alpha \in (0, 1)$, then $\mathbf{x}_1 = \mathbf{x}_2$. In the following theorem we show that extreme points of the constraint set are equivalent to basic feasible solutions.

Theorem 15.2 *Let Ω be the convex set consisting of all feasible solutions, that is, all n -vectors \mathbf{x} satisfying*

$$A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0},$$

where $A \in \mathbb{R}^{m \times n}$, $m < n$. Then, \mathbf{x} is an extreme point of Ω if and only if \mathbf{x} is a basic feasible solution to $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$. \square

Proof. \Rightarrow : Suppose that \mathbf{x} satisfies $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, and has p positive components. As before, without loss of generality, we can assume that the first p components are positive and the remaining components are zero. We have

$$x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_p\mathbf{a}_p = \mathbf{b}.$$

Let y_i , $i = 1, \dots, p$, be numbers such that

$$y_1\mathbf{a}_1 + y_2\mathbf{a}_2 + \cdots + y_p\mathbf{a}_p = \mathbf{0}.$$

We show that each $y_i = 0$. To begin, multiply this equation by $\varepsilon > 0$, then add and subtract the result from the equation $x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_p\mathbf{a}_p = \mathbf{b}$ to get

$$\begin{aligned} (x_1 + \varepsilon y_1)\mathbf{a}_1 + (x_2 + \varepsilon y_2)\mathbf{a}_2 + \cdots + (x_p + \varepsilon y_p)\mathbf{a}_p &= \mathbf{b}, \\ (x_1 - \varepsilon y_1)\mathbf{a}_1 + (x_2 - \varepsilon y_2)\mathbf{a}_2 + \cdots + (x_p - \varepsilon y_p)\mathbf{a}_p &= \mathbf{b}. \end{aligned}$$

Because each $x_i > 0$, $\varepsilon > 0$ can be chosen such that each $x_i + \varepsilon y_i, x_i - \varepsilon y_i \geq 0$ (e.g., $\varepsilon = \min\{|x_i/y_i| : i = 1, \dots, p, y_i \neq 0\}$). For such a choice of ε , the vectors

$$\begin{aligned} \mathbf{z}_1 &= [x_1 + \varepsilon y_1, x_2 + \varepsilon y_2, \dots, x_p + \varepsilon y_p, 0, \dots, 0]^\top, \\ \mathbf{z}_2 &= [x_1 - \varepsilon y_1, x_2 - \varepsilon y_2, \dots, x_p - \varepsilon y_p, 0, \dots, 0]^\top \end{aligned}$$

belong to Ω . Observe that $\mathbf{x} = \frac{1}{2}\mathbf{z}_1 + \frac{1}{2}\mathbf{z}_2$. Because \mathbf{x} is an extreme point, $\mathbf{z}_1 = \mathbf{z}_2$. Hence, each $y_i = 0$, which implies that the \mathbf{a}_i are linearly independent.

\Leftarrow : Let $\mathbf{x} \in \Omega$ be a basic feasible solution. Let $\mathbf{y}, \mathbf{z} \in \Omega$ be such that

$$\mathbf{x} = \alpha\mathbf{y} + (1 - \alpha)\mathbf{z}$$

for some $\alpha \in (0, 1)$. We show that $\mathbf{y} = \mathbf{z}$ and conclude that \mathbf{x} is an extreme point. Because $\mathbf{y}, \mathbf{z} \geq \mathbf{0}$, and the last $n - m$ components of \mathbf{x} are zero, the last $n - m$ components of \mathbf{y} and \mathbf{z} are zero as well. Furthermore, since $\mathbf{A}\mathbf{y} = \mathbf{A}\mathbf{z} = \mathbf{b}$,

$$y_1\mathbf{a}_1 + \cdots + y_m\mathbf{a}_m = \mathbf{b}$$

and

$$z_1\mathbf{a}_1 + \cdots + z_m\mathbf{a}_m = \mathbf{b}.$$

Combining these two equations yields

$$(y_1 - z_1)\mathbf{a}_1 + \cdots + (y_m - z_m)\mathbf{a}_m = \mathbf{0}.$$

Because the columns $\mathbf{a}_1, \dots, \mathbf{a}_m$ are linearly independent, we have $y_i = z_i$, $i = 1, \dots, m$. Therefore, $\mathbf{y} = \mathbf{z}$, and hence \mathbf{x} is an extreme point of Ω . ■

From Theorem 15.2 it follows that the set of extreme points of the constraint set $\Omega = \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is equal to the set of basic feasible solutions to $\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$. Combining this observation with the fundamental theorem of LP (Theorem 15.1), we can see that in solving linear programming problems we need only examine the extreme points of the constraint set.

Example 15.15 Consider the following LP problem:

$$\begin{aligned} & \text{maximize} && 3x_1 + 5x_2 \\ & \text{subject to} && x_1 + 5x_2 \leq 40 \\ & && 2x_1 + x_2 \leq 20 \\ & && x_1 + x_2 \leq 12 \\ & && x_1, x_2 \geq 0. \end{aligned}$$

We introduce slack variables x_3, x_4, x_5 to convert this LP problem into standard form:

$$\begin{aligned} & \text{minimize} && -3x_1 - 5x_2 \\ & \text{subject to} && x_1 + 5x_2 + x_3 = 40 \\ & && 2x_1 + x_2 + x_4 = 20 \\ & && x_1 + x_2 + x_5 = 12 \\ & && x_1, \dots, x_5 \geq 0. \end{aligned}$$

In the remainder of the example we consider only the problem in standard form. We can represent the constraints above as

$$\begin{aligned} x_1 \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} + x_2 \begin{bmatrix} 5 \\ 1 \\ 1 \end{bmatrix} + x_3 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + x_5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} &= \begin{bmatrix} 40 \\ 20 \\ 12 \end{bmatrix}, \\ x_1, \dots, x_5 \geq 0, \end{aligned}$$

that is, $x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + x_3\mathbf{a}_3 + x_4\mathbf{a}_4 + x_5\mathbf{a}_5 = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$. Note that

$$\mathbf{x} = [0, 0, 40, 20, 12]^\top$$

is a feasible solution. But for this \mathbf{x} , the value of the objective function is zero. We already know that the minimum of the objective function (if it exists) is achieved at an extreme point of the constraint set Ω defined by the constraints. The point $[0, 0, 40, 20, 12]^\top$ is an extreme point of the set of feasible solutions, but it turns out that it does not minimize the objective function. Therefore, we need to seek the solution among the other extreme points. To do this we move from one extreme point to an adjacent extreme point such that the value of the objective function decreases. Here, we define two extreme points to be adjacent if the corresponding basic columns differ by only one vector. We begin with $\mathbf{x} = [0, 0, 40, 20, 12]^\top$. We have

$$0\mathbf{a}_1 + 0\mathbf{a}_2 + 40\mathbf{a}_3 + 20\mathbf{a}_4 + 12\mathbf{a}_5 = \mathbf{b}.$$

To select an adjacent extreme point, let us choose to include \mathbf{a}_1 as a basic column in the new basis. We need to remove either \mathbf{a}_3 , \mathbf{a}_4 , or \mathbf{a}_5 from the old basis. We proceed as follows. We first express \mathbf{a}_1 as a linear combination of the old basic columns:

$$\mathbf{a}_1 = 1\mathbf{a}_3 + 2\mathbf{a}_4 + 1\mathbf{a}_5.$$

Multiplying both sides of this equation by $\varepsilon_1 > 0$, we get

$$\varepsilon_1\mathbf{a}_1 = \varepsilon_1\mathbf{a}_3 + 2\varepsilon_1\mathbf{a}_4 + \varepsilon_1\mathbf{a}_5.$$

We now add this equation to the equation $0\mathbf{a}_1 + 0\mathbf{a}_2 + 40\mathbf{a}_3 + 20\mathbf{a}_4 + 12\mathbf{a}_5 = \mathbf{b}$. Collecting terms yields

$$\varepsilon_1\mathbf{a}_1 + 0\mathbf{a}_2 + (40 - \varepsilon_1)\mathbf{a}_3 + (20 - 2\varepsilon_1)\mathbf{a}_4 + (12 - \varepsilon_1)\mathbf{a}_5 = \mathbf{b}.$$

We want to choose ε_1 in such a way that each of the coefficients above is nonnegative and at the same time, one of the coefficients \mathbf{a}_3 , \mathbf{a}_4 , or \mathbf{a}_5 becomes zero. Clearly, $\varepsilon_1 = 10$ does the job. The result is

$$10\mathbf{a}_1 + 30\mathbf{a}_3 + 2\mathbf{a}_5 = \mathbf{b}.$$

The corresponding basic feasible solution (extreme point) is

$$[10, 0, 30, 0, 2]^\top.$$

For this solution, the objective function value is -30 , which is an improvement relative to the objective function value at the old extreme point.

We now apply the same procedure as above to move to another adjacent extreme point, which hopefully further decreases the value of the objective function. This time, we choose \mathbf{a}_2 to enter the new basis. We have

$$\mathbf{a}_2 = \frac{1}{2}\mathbf{a}_1 + \frac{9}{2}\mathbf{a}_3 + \frac{1}{2}\mathbf{a}_5$$

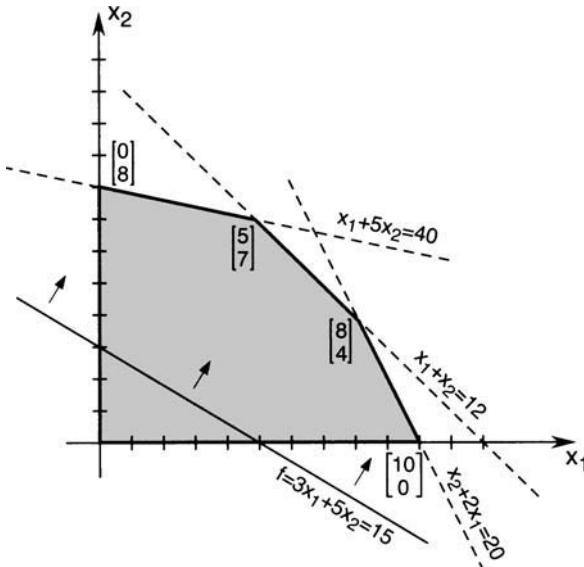


Figure 15.11 Graphical solution to the LP problem in Example 15.15.

and

$$\left(10 - \frac{1}{2}\varepsilon_2\right)\mathbf{a}_1 + \varepsilon_2\mathbf{a}_2 + \left(30 - \frac{9}{2}\varepsilon_2\right)\mathbf{a}_3 + \left(2 - \frac{1}{2}\varepsilon_2\right)\mathbf{a}_5 = \mathbf{b}.$$

Substituting $\varepsilon_2 = 4$, we obtain

$$8\mathbf{a}_1 + 4\mathbf{a}_2 + 12\mathbf{a}_3 = \mathbf{b}.$$

The solution is $[8, 4, 12, 0, 0]^\top$ and the corresponding value of the objective function is -44 , which is smaller than the value at the previous extreme point. To complete the example we repeat the procedure once more. This time, we select \mathbf{a}_4 and express it as a combination of the vectors in the previous basis, \mathbf{a}_1 , \mathbf{a}_2 , and \mathbf{a}_3 :

$$\mathbf{a}_4 = \mathbf{a}_1 - \mathbf{a}_2 + 4\mathbf{a}_3,$$

and hence

$$(8 - \varepsilon_3)\mathbf{a}_1 + (4 + \varepsilon_3)\mathbf{a}_2 + (12 - 4\varepsilon_3)\mathbf{a}_3 + \varepsilon_3\mathbf{a}_4 = \mathbf{b}.$$

The largest permissible value for ε_3 is 3 . The corresponding basic feasible solution is $[5, 7, 0, 3, 0]^\top$, with an objective function value of -50 . The solution $[5, 7, 0, 3, 0]^\top$ turns out to be an optimal solution to our problem in standard form. Hence, the solution to the original problem is $[5, 7]^\top$, which we can easily obtain graphically (see Figure 15.11). ■

The technique used in this example for moving from one extreme point to an adjacent extreme point is also used in the simplex method for solving LP problems. The simplex method is essentially a refined method of performing these manipulations.

EXERCISES

15.1 Convert the following linear programming problem to *standard form*:

$$\begin{aligned} & \text{maximize} && 2x_1 + x_2 \\ & \text{subject to} && 0 \leq x_1 \leq 2 \\ & && x_1 + x_2 \leq 3 \\ & && x_1 + 2x_2 \leq 5 \\ & && x_2 \geq 0. \end{aligned}$$

15.2 Consider a discrete-time linear system $x_{k+1} = ax_k + bu_k$, where u_k is the input at time k , x_k is the output at time k , and $a, b \in \mathbb{R}$ are system parameters. Given an initial condition $x_0 = 1$, consider the problem of minimizing the output x_2 at time 2 subject to the constraint that $|u_i| \leq 1$, $i = 0, 1$.

Formulate the problem as a linear programming problem, and convert it into standard form.

15.3 Consider the optimization problem

$$\begin{aligned} & \text{minimize} && c_1|x_1| + c_2|x_2| + \cdots + c_n|x_n| \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b}, \end{aligned}$$

where $c_i \neq 0$, $i = 1, \dots, n$. Convert this problem into an equivalent standard form linear programming problem.

Hint: Given any $x \in \mathbb{R}$, we can find unique numbers $x^+, x^- \in \mathbb{R}$, $x^+, x^- \geq 0$, such that $|x| = x^+ + x^-$ and $x = x^+ - x^-$.

15.4 Does every linear programming problem in standard form have a nonempty feasible set? If “yes,” provide a proof. If “no,” give a specific example.

Does every linear programming problem in standard form (assuming a nonempty feasible set) have an optimal solution? If “yes,” provide a proof. If “no,” give a specific example.

15.5 Suppose that a computer supplier has two warehouses, one located in city A and another in city B. The supplier receives orders from two customers, one in city C and another in city D. The customer in city C orders 50 units,

and the customer in city D orders 60 units. The number of units at the warehouse in city A is 70, and the number of units at the warehouse in city B is 80. The cost of shipping each unit from A to C is 1, from A to D is 2, from B to C is 3, and from B to D is 4.

Formulate the problem of deciding how many units from each warehouse should be shipped to each customer to minimize the total shipping cost (assuming that the values of units to be shipped are real numbers). Express the problem as an equivalent standard form linear programming problem.

15.6 Consider a computer network consisting of six computers, A through F . The computers are connected according to the following links, with maximum data rates (in Mbps) shown: AC (10), BC (7), BF (3), CD (8), DE (12), DF (4). For example, “ AC (10)” means that computers A and C are connected with a link that supports data rates up to 10 Mbps.

Suppose that A and B need to send data to E and F , respectively (no other communication is taking place in the network). Any path through the given links above may be used as long as the path has no loop. Also, multiple paths (say from A to E) can be used simultaneously. Link bandwidth can be shared as long as the total data rate through the link does not exceed its maximum (the total data rate through a link is the sum of the data rates of communication in both directions).

For every Mbps of data rate the network can support for transmission from A to E , we receive 2 dollars. For every Mbps of data rate the network can support for transmission from B to F , we receive 3 dollars. Formulate a linear programming problem to represent the goal of maximizing the total revenue. Then, convert this problem into standard form.

Hint: Draw a picture of the network, then label each link with the maximum data rate and the paths that share that link.

15.7 A cereal manufacturer wishes to produce 1000 pounds of a cereal that contains exactly 10% fiber, 2% fat, and 5% sugar (by weight). The cereal is to be produced by combining four items of raw food material in appropriate proportions. These four items have certain combinations of fiber, fat, and sugar content, and are available at various prices per pound:

Item	1	2	3	4
% fiber	3	8	16	4
% fat	6	46	9	9
% sugar	20	5	4	0
Price/lb	2	4	1	2

The manufacturer wishes to find the amounts of each item to be used to produce the cereal in the least expensive way. Formulate the problem as a linear programming problem. What can you say about the existence of a solution to this problem?

15.8 Suppose that a wireless broadcast system has n transmitters. Transmitter j broadcasts at a power of $p_j \geq 0$. There are m locations where the broadcast is to be received. The path gain from transmitter j to location i is $g_{i,j}$; that is, the power of the signal transmitted from transmitter j received at location i is $g_{i,j}p_j$. The total power received at location i is the sum of the powers received from all the transmitters. Formulate the problem of finding the minimum sum of the powers transmitted subject to the requirement that the power received at each location is at least P .

15.9 Consider the system of equations

$$\begin{bmatrix} 2 & -1 & 2 & -1 & 3 \\ 1 & 2 & 3 & 1 & 0 \\ 1 & 0 & -2 & 0 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 14 \\ 5 \\ -10 \end{bmatrix}.$$

Check if the system has basic solutions. If yes, find all basic solutions.

15.10 Solve the following linear program graphically:

$$\begin{aligned} &\text{maximize} && 2x_1 + 5x_2 \\ &\text{subject to} && 0 \leq x_1 \leq 4 \\ & && 0 \leq x_2 \leq 6 \\ & && x_1 + x_2 \leq 8. \end{aligned}$$

15.11 The optimization toolbox in MATLAB provides a function, `linprog`, for solving linear programming problems. Use the function `linprog` to solve the problem in Example 15.5. Use the initial condition **0**.

CHAPTER 16

SIMPLEX METHOD

16.1 Solving Linear Equations Using Row Operations

The examples in previous chapters illustrate that solving linear programs involves the solution of systems of linear simultaneous algebraic equations. In this section we describe a method for solving a system of n linear equations in n unknowns that we use in subsequent sections. The method uses elementary row operations and corresponding elementary matrices. For a discussion of numerical issues involved in solving a system of simultaneous linear algebraic equations, we refer the reader to [41] and [53].

An elementary row operation on a given matrix is an algebraic manipulation of the matrix that corresponds to one of the following:

1. Interchanging any two rows of the matrix
2. Multiplying one of its rows by a real nonzero number
3. Adding a scalar multiple of one row to another row

An elementary row operation on a matrix is equivalent to premultiplying the matrix by a corresponding *elementary matrix*, which we define next.

Definition 16.1 We call \mathbf{E} an *elementary matrix of the first kind* if \mathbf{E} is obtained from the identity matrix \mathbf{I} by interchanging any two of its rows. ■

An elementary matrix of the first kind formed from \mathbf{I} by interchanging the i th and the j th rows has the form

$$\mathbf{E} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & 0 & \cdots & 1 \\ & & & & 1 & \\ & & \vdots & & \ddots & \vdots \\ & & & & & 1 \\ & & & & & & 1 \\ & & & & & & & 1 \\ & & & & & & & & 1 \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \end{bmatrix} \quad \begin{array}{l} \leftarrow i\text{th row} \\ \leftarrow j\text{th row} \end{array}$$

Note that \mathbf{E} is invertible and $\mathbf{E} = \mathbf{E}^{-1}$.

Definition 16.2 We call \mathbf{E} an *elementary matrix of the second kind* if \mathbf{E} is obtained from the identity matrix \mathbf{I} by multiplying one of its rows by a real number $\alpha \neq 0$. ■

The elementary matrix of the second kind formed from \mathbf{I} by multiplying the i th row by $\alpha \neq 0$ has the form

$$\mathbf{E} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & 0 & \\ & & & \alpha & & & \\ & & & & 1 & & \\ & & 0 & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \quad \leftarrow i\text{th row}$$

Note that \mathbf{E} is invertible and

$$\mathbf{E}^{-1} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & 0 \\ & & 1 & & & \\ & & & 1/\alpha & & \\ & & & & 1 & \\ 0 & & & & & \ddots \\ & & & & & & 1 \end{bmatrix} \quad \leftarrow \text{ith row}$$

Definition 16.3 We call \mathbf{E} an *elementary matrix of the third kind* if \mathbf{E} is obtained from the identity matrix \mathbf{I} by adding β times one row to another row of \mathbf{I} . ■

An elementary matrix of the third kind obtained from \mathbf{I} by adding β times the j th row to the i th row has the form

$$\mathbf{E} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & 0 \\ & & 1 & \cdots & \beta & \\ & & & \ddots & \vdots & \\ & & & & 1 & \\ 0 & & & & & \ddots \\ & & & & & & 1 \end{bmatrix} \quad \begin{array}{l} \leftarrow \text{ith row} \\ \leftarrow \text{jth row} \end{array}$$

Observe that \mathbf{E} is the identity matrix with an extra β in the (i,j) th location. Note that \mathbf{E} is invertible and

$$\mathbf{E}^{-1} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & 0 \\ & & 1 & \cdots & -\beta & & \\ & & & \ddots & \vdots & & \\ & & & & 1 & & \\ 0 & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \quad \begin{array}{l} \leftarrow \text{ith row} \\ \leftarrow \text{jth row} \end{array}$$

Definition 16.4 An *elementary row operation* (of first, second, or third kind) on a given matrix is a premultiplication of the given matrix by a corresponding elementary matrix of the respective kind. ■

Because elementary matrices are invertible, we can define the corresponding inverse elementary row operations.

Consider a system of n linear equations in n unknowns x_1, x_2, \dots, x_n with right-hand sides b_1, b_2, \dots, b_n . In matrix form this system may be written as

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}, \quad \mathbf{A} \in \mathbb{R}^{n \times n}.$$

If \mathbf{A} is invertible, then

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}.$$

Thus, the problem of solving the system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, with $\mathbf{A} \in \mathbb{R}^{n \times n}$ invertible, is related to the problem of computing \mathbf{A}^{-1} . We now show that \mathbf{A}^{-1} can be computed effectively using elementary row operations. In particular, we prove the following theorem.

Theorem 16.1 *Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a given matrix. Then, \mathbf{A} is nonsingular (invertible) if and only if there exist elementary matrices \mathbf{E}_i , $i = 1, \dots, t$, such that*

$$\mathbf{E}_t \cdots \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} = \mathbf{I}.$$

□

Proof. \Rightarrow : If \mathbf{A} is nonsingular, then its first column must have at least one nonzero element, say $a_{j1} \neq 0$. Premultiplying \mathbf{A} by an elementary matrix of the first kind of the form

$$\mathbf{E}_1 = \left[\begin{array}{cccccc} 0 & & & 1 & & & \\ & 1 & & & & & \\ & & \ddots & & \vdots & & \\ & & & 1 & & & \\ 1 & \dots & 0 & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{array} \right] \quad \text{← } j\text{th row}$$

brings the nonzero element a_{j1} to the location $(1, 1)$. Hence, in the matrix $\mathbf{E}_1 \mathbf{A}$, the element $a_{11} \neq 0$. Note that since \mathbf{E}_1 is nonsingular, $\mathbf{E}_1 \mathbf{A}$ is also nonsingular.

Next, we premultiply $\mathbf{E}_1\mathbf{A}$ by an elementary matrix of the second kind of the form

$$\mathbf{E}_2 = \begin{bmatrix} 1/a_{11} & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}.$$

The result of this operation is the matrix $\mathbf{E}_2\mathbf{E}_1\mathbf{A}$ with unity in the location $(1, 1)$. We next apply a sequence of elementary row operations of the third kind on the matrix $\mathbf{E}_2\mathbf{E}_1\mathbf{A}$. Specifically, we premultiply $\mathbf{E}_2\mathbf{E}_1\mathbf{A}$ by $n - 1$ elementary matrices of the form

$$\mathbf{E}_3 = \begin{bmatrix} 1 & & & \\ -a_{21} & 1 & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{bmatrix}, \dots, \mathbf{E}_r = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ \vdots & & 1 & \\ -a_{n1} & & & \ddots \\ & & & & 1 \end{bmatrix},$$

where $r = 2 + n - 1 = n + 1$. The result of these operations is the nonsingular matrix

$$\mathbf{E}_r\mathbf{E}_{r-1}\cdots\mathbf{E}_2\mathbf{E}_1\mathbf{A} = \begin{bmatrix} 1 & \bar{a}_{12} & \cdots & \bar{a}_{1n} \\ 0 & \bar{a}_{22} & \cdots & \bar{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \bar{a}_{n2} & \cdots & \bar{a}_{nn} \end{bmatrix}.$$

Because the matrix $\mathbf{E}_r\cdots\mathbf{E}_1\mathbf{A}$ is nonsingular, its submatrix

$$\begin{bmatrix} \bar{a}_{22} & \cdots & \bar{a}_{2n} \\ \vdots & & \vdots \\ \bar{a}_{n2} & \cdots & \bar{a}_{nn} \end{bmatrix}$$

must also be nonsingular. This implies that there is a nonzero element \bar{a}_{j2} , where $2 \leq j \leq n$. Using an elementary operation of the first kind, we bring this element to the location $(2, 2)$. Thus, in the matrix

$$\mathbf{E}_{r+1}\mathbf{E}_r\cdots\mathbf{E}_1\mathbf{A}$$

the $(2, 2)$ th element is nonzero. Premultiplying the matrix by an elementary matrix of the second kind yields the matrix

$$\mathbf{E}_{r+2}\mathbf{E}_{r+1}\cdots\mathbf{E}_1\mathbf{A},$$

in which the element in the location $(2, 2)$ is unity. As before, we premultiply this matrix by $n - 1$ elementary row operations of the third kind, to get a

matrix of the form

$$\mathbf{E}_s \cdots \mathbf{E}_r \cdots \mathbf{E}_1 \mathbf{A} = \begin{bmatrix} 1 & 0 & \tilde{a}_{13} & \cdots & \tilde{a}_{1n} \\ 0 & 1 & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ 0 & 0 & \tilde{a}_{33} & \cdots & \tilde{a}_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \tilde{a}_{n3} & \cdots & \tilde{a}_{nn} \end{bmatrix},$$

where $s = r + 2 + n - 1 = 2(n + 1)$. This matrix is nonsingular. Hence, there is a nonzero element \tilde{a}_{j3} , $3 \leq j \leq n$. Proceeding in a similar fashion as before, we obtain

$$\mathbf{E}_t \cdots \mathbf{E}_s \cdots \mathbf{E}_r \cdots \mathbf{E}_1 \mathbf{A} = \mathbf{I},$$

where $t = n(n + 1)$.

\Leftarrow : If there exist elementary matrices $\mathbf{E}_1, \dots, \mathbf{E}_t$ such that

$$\mathbf{E}_t \cdots \mathbf{E}_1 \mathbf{A} = \mathbf{I},$$

then clearly \mathbf{A} is invertible, with

$$\mathbf{A}^{-1} = \mathbf{E}_t \cdots \mathbf{E}_1.$$

■

Theorem 16.1 suggests the following procedure for finding \mathbf{A}^{-1} , if it exists. We first form the matrix

$$[\mathbf{A}, \mathbf{I}].$$

We then apply elementary row operations to $[\mathbf{A}, \mathbf{I}]$ so that \mathbf{A} is transformed into \mathbf{I} ; that is, we obtain

$$\mathbf{E}_t \cdots \mathbf{E}_1 [\mathbf{A}, \mathbf{I}] = [\mathbf{I}, \mathbf{B}].$$

It then follows that

$$\mathbf{B} = \mathbf{E}_t \cdots \mathbf{E}_1 = \mathbf{A}^{-1}.$$

Example 16.1 Let

$$\mathbf{A} = \begin{bmatrix} 2 & 5 & 10 & 0 \\ 1 & 1 & 1 & 0 \\ -2 & -10 & -30 & 1 \\ -1 & -2 & -3 & 0 \end{bmatrix}.$$

Find \mathbf{A}^{-1} .

We form the matrix

$$[\mathbf{A}, \mathbf{I}] = \begin{bmatrix} 2 & 5 & 10 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ -2 & -10 & -30 & 1 & 0 & 0 & 1 & 0 \\ -1 & -2 & -3 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and perform row operations on this matrix. Applying row operations of the first and third kinds yields

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 3 & 8 & 0 & 1 & -2 & 0 & 0 \\ 0 & -8 & -28 & 1 & 0 & 2 & 1 & 0 \\ 0 & -1 & -2 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

We then interchange the second and fourth rows and apply elementary row operations of the second and third kinds to get

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 2 & 0 & 1 \\ 0 & 1 & 2 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 2 & 0 & 1 & 1 & 0 & 3 \\ 0 & 0 & -12 & 1 & 0 & -6 & 1 & -8 \end{bmatrix}.$$

Now multiply the third row by 1/2 and then perform a sequence of elementary operations of the third kind to obtain

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \frac{1}{2} & \frac{5}{2} & 0 & \frac{5}{2} \\ 0 & 1 & 0 & 0 & -1 & -2 & 0 & -4 \\ 0 & 0 & 1 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \frac{3}{2} \\ 0 & 0 & 0 & 1 & 6 & 0 & 1 & 10 \end{bmatrix}.$$

Hence,

$$\mathbf{A}^{-1} = \begin{bmatrix} \frac{1}{2} & \frac{5}{2} & 0 & \frac{5}{2} \\ -1 & -2 & 0 & -4 \\ \frac{1}{2} & \frac{1}{2} & 0 & \frac{3}{2} \\ 6 & 0 & 1 & 10 \end{bmatrix}.$$

■

We now return to the general problem of solving the system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{n \times n}$. If \mathbf{A}^{-1} exists, then the solution is $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. However, we do not need an explicit expression for \mathbf{A}^{-1} to find the solution. Indeed, let \mathbf{A}^{-1} be expressed as a product of elementary matrices

$$\mathbf{A}^{-1} = \mathbf{E}_t \mathbf{E}_{t-1} \cdots \mathbf{E}_1.$$

Thus,

$$\mathbf{E}_t \cdots \mathbf{E}_1 \mathbf{A} \mathbf{x} = \mathbf{E}_t \cdots \mathbf{E}_1 \mathbf{b}$$

and hence

$$\mathbf{x} = \mathbf{E}_t \cdots \mathbf{E}_1 \mathbf{b}.$$

The discussion above leads to the following procedure for solving the system $\mathbf{A}\mathbf{x} = \mathbf{b}$. Form an augmented matrix

$$[\mathbf{A}, \mathbf{b}].$$

Then, perform a sequence of row elementary operations on this augmented matrix until we obtain

$$[\mathbf{I}, \tilde{\mathbf{b}}].$$

From the above we have that if \mathbf{x} is a solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$, then it is also a solution to $\mathbf{E}\mathbf{A}\mathbf{x} = \mathbf{E}\mathbf{b}$, where $\mathbf{E} = \mathbf{E}_t \cdots \mathbf{E}_1$ represents a sequence of elementary row operations. Because $\mathbf{E}\mathbf{A} = \mathbf{I}$, and $\mathbf{E}\mathbf{b} = \tilde{\mathbf{b}}$, it follows that $\mathbf{x} = \tilde{\mathbf{b}}$ is the solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{n \times n}$ invertible.

Suppose now that $\mathbf{A} \in \mathbb{R}^{m \times n}$ where $m < n$, and $\text{rank } \mathbf{A} = m$. Then, \mathbf{A} is not a square matrix. Clearly, in this case the system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ has infinitely many solutions. Without loss of generality, we can assume that the first m columns of \mathbf{A} are linearly independent. Then, if we perform a sequence of elementary row operations on the augmented matrix $[\mathbf{A}, \mathbf{b}]$ as before, we obtain

$$[\mathbf{I}, \mathbf{D}, \tilde{\mathbf{b}}],$$

where \mathbf{D} is an $m \times (n - m)$ matrix. Let $\mathbf{x} \in \mathbb{R}^n$ be a solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ and write $\mathbf{x} = [\mathbf{x}_B^\top, \mathbf{x}_D^\top]^\top$, where $\mathbf{x}_B \in \mathbb{R}^m$, $\mathbf{x}_D \in \mathbb{R}^{(n-m)}$. Then, $[\mathbf{I}, \mathbf{D}]\mathbf{x} = \tilde{\mathbf{b}}$, which we can rewrite as $\mathbf{x}_B + \mathbf{D}\mathbf{x}_D = \tilde{\mathbf{b}}$, or $\mathbf{x}_B = \tilde{\mathbf{b}} - \mathbf{D}\mathbf{x}_D$. Note that for an arbitrary $\mathbf{x}_D \in \mathbb{R}^{(n-m)}$, if $\mathbf{x}_B = \tilde{\mathbf{b}} - \mathbf{D}\mathbf{x}_D$, then the resulting vector $\mathbf{x} = [\mathbf{x}_B^\top, \mathbf{x}_D^\top]^\top$ is a solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$. In particular, $[\tilde{\mathbf{b}}^\top, \mathbf{0}^\top]^\top$ is a solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$. We often refer to the basic solution $[\tilde{\mathbf{b}}^\top, \mathbf{0}^\top]^\top$ as a *particular solution* to $\mathbf{A}\mathbf{x} = \mathbf{b}$. Note that $[-(\mathbf{D}\mathbf{x}_D)^\top, \mathbf{x}_D^\top]^\top$ is a solution to $\mathbf{A}\mathbf{x} = \mathbf{0}$. Any solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ has the form

$$\mathbf{x} = \begin{bmatrix} \tilde{\mathbf{b}} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\mathbf{D}\mathbf{x}_D \\ \mathbf{x}_D \end{bmatrix}$$

for some $\mathbf{x}_D \in \mathbb{R}^{(n-m)}$.

16.2 The Canonical Augmented Matrix

Consider the system of simultaneous linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\text{rank } \mathbf{A} = m$. Using a sequence of elementary row operations and reordering the variables

if necessary, we transform the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ into the following *canonical form*:

$$\begin{aligned} x_1 &+ y_{1m+1}x_{m+1} + \cdots + y_{1n}x_n = y_{10} \\ x_2 &+ y_{2m+1}x_{m+1} + \cdots + y_{2n}x_n = y_{20} \\ &\vdots \\ x_m &+ y_{mm+1}x_{m+1} + \cdots + y_{mn}x_n = y_{m0}. \end{aligned}$$

This can be represented in matrix notation as

$$[\mathbf{I}_m, \mathbf{Y}_{m,n-m}] \mathbf{x} = \mathbf{y}_0.$$

Formally, we define the canonical form as follows.

Definition 16.5 A system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is said to be in *canonical form* if among the n variables there are m variables with the property that each appears in only one equation, and its coefficient in that equation is unity. ■

A system is in canonical form if by some reordering of the equations and the variables it takes the form $[\mathbf{I}_m, \mathbf{Y}_{m,n-m}] \mathbf{x} = \mathbf{y}_0$. If a system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ is not in canonical form, we can transform the system into canonical form by a sequence of elementary row operations. The system in canonical form has the same solution as the original system $\mathbf{A}\mathbf{x} = \mathbf{b}$ and is called the *canonical representation* of the system with respect to the basis $\mathbf{a}_1, \dots, \mathbf{a}_m$. There are, in general, many canonical representations of a given system, depending on which columns of \mathbf{A} we transform into the columns of \mathbf{I}_m (i.e., basic columns). We call the augmented matrix $[\mathbf{I}_m, \mathbf{Y}_{m,n-m}, \mathbf{y}_0]$ of the canonical representation of a given system the *canonical augmented matrix* of the system with respect to the basis $\mathbf{a}_1, \dots, \mathbf{a}_m$. Of course, there may be many canonical augmented matrices of a given system, depending on which columns of \mathbf{A} are chosen as basic columns.

The variables corresponding to basic columns in a canonical representation of a given system are the basic variables, whereas the other variables are the nonbasic variables. In particular, in the canonical representation $[\mathbf{I}_m, \mathbf{Y}_{m,n-m}] \mathbf{x} = \mathbf{y}_0$ of a given system, the variables x_1, \dots, x_m are the basic variables and the other variables are the nonbasic variables. Note that in general the basic variables need not be the first m variables. However, in the following discussion we assume, for convenience and without loss of generality, that the basic variables are indeed the first m variables in the system.

Having done so, the corresponding basic solution is

$$x_1 = y_{10},$$

$$\vdots$$

$$x_m = y_{m0},$$

$$x_{m+1} = 0,$$

$$\vdots$$

$$x_n = 0;$$

that is,

$$\mathbf{x} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{0} \end{bmatrix}.$$

Given a system of equations $\mathbf{Ax} = \mathbf{b}$, consider the associated canonical augmented matrix

$$[\mathbf{I}_m, \mathbf{Y}_{m,n-m}, \mathbf{y}_0] = \begin{bmatrix} 1 & 0 & \cdots & 0 & y_{1m+1} & \cdots & y_{1n} & y_{10} \\ 0 & 1 & \cdots & 0 & y_{2m+1} & \cdots & y_{2n} & y_{20} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & y_{mm+1} & \cdots & y_{mn} & y_{m0} \end{bmatrix}.$$

From the arguments above we conclude that

$$\mathbf{b} = y_{10}\mathbf{a}_1 + y_{20}\mathbf{a}_2 + \cdots + y_{m0}\mathbf{a}_m.$$

In other words, the entries in the last column of the canonical augmented matrix are the coordinates of the vector \mathbf{b} with respect to the basis $\{\mathbf{a}_1, \dots, \mathbf{a}_m\}$. The entries of all the other columns of the canonical augmented matrix have a similar interpretation. Specifically, the entries of the j th column of the canonical augmented matrix, $j = 1, \dots, n$, are the coordinates of \mathbf{a}_j with respect to the basis $\{\mathbf{a}_1, \dots, \mathbf{a}_m\}$. To see this, note that the first m columns of the augmented matrix form a basis (the standard basis). Every other vector in the augmented matrix can be expressed as a linear combination of these basis vectors by reading the coefficients down the corresponding column. Specifically, let \mathbf{a}'_i , $i = 1, \dots, n+1$, be the i th column in the augmented matrix above. Clearly, since $\mathbf{a}'_1, \dots, \mathbf{a}'_m$ form the standard basis, then for $m < j \leq n$,

$$\mathbf{a}'_j = y_{1j}\mathbf{a}'_1 + y_{2j}\mathbf{a}'_2 + \cdots + y_{mj}\mathbf{a}'_m.$$

Let \mathbf{a}_i , $i = 1, \dots, n$, be the i th column of \mathbf{A} , and $\mathbf{a}_{n+1} = \mathbf{b}$. Now, $\mathbf{a}'_i = \mathbf{E}\mathbf{a}_i$, $i = 1, \dots, n+1$, where \mathbf{E} is a nonsingular matrix that represents the elementary row operations needed to transform $[\mathbf{A}, \mathbf{b}]$ into $[\mathbf{I}_m, \mathbf{Y}_{m,n-m}, \mathbf{y}_0]$. Therefore, for $m < j \leq n$, we also have

$$\mathbf{a}_j = y_{1j}\mathbf{a}_1 + y_{2j}\mathbf{a}_2 + \cdots + y_{mj}\mathbf{a}_m.$$

16.3 Updating the Augmented Matrix

To summarize Section 16.2, the canonical augmented matrix of a given system $\mathbf{A}\mathbf{x} = \mathbf{b}$ specifies the representations of the columns \mathbf{a}_j , $m < j \leq n$, in terms of the basic columns $\mathbf{a}_1, \dots, \mathbf{a}_m$. Thus, the elements of the j th column of the canonical augmented matrix are the coordinates of the vector \mathbf{a}_j with respect to the basis $\mathbf{a}_1, \dots, \mathbf{a}_m$. The coordinates of \mathbf{b} are given in the last column.

Suppose that we are given the canonical representation of a system $\mathbf{A}\mathbf{x} = \mathbf{b}$. We now consider the following question: If we replace a basic variable by a nonbasic variable, what is the new canonical representation corresponding to the new set of basic variables? Specifically, suppose that we wish to replace the basis vector \mathbf{a}_p , $1 \leq p \leq m$, by the vector \mathbf{a}_q , $m < q \leq n$. Provided that the first m vectors with \mathbf{a}_p replaced by \mathbf{a}_q are linearly independent, these vectors constitute a basis and every vector can be expressed as a linear combination of the new basic columns.

Let us now find the coordinates of the vectors $\mathbf{a}_1, \dots, \mathbf{a}_n$ with respect to the new basis. These coordinates form the entries of the canonical augmented matrix of the system with respect to the new basis. In terms of the old basis, we can express \mathbf{a}_q as

$$\mathbf{a}_q = \sum_{i=1}^m y_{iq} \mathbf{a}_i = \sum_{\substack{i=1 \\ i \neq p}}^m y_{iq} \mathbf{a}_i + y_{pq} \mathbf{a}_p.$$

Note that the set of vectors $\{\mathbf{a}_1, \dots, \mathbf{a}_{p-1}, \mathbf{a}_q, \mathbf{a}_{p+1}, \dots, \mathbf{a}_m\}$ is linearly independent if and only if $y_{pq} \neq 0$. Solving the equation above for \mathbf{a}_p , we get

$$\mathbf{a}_p = \frac{1}{y_{pq}} \mathbf{a}_q - \sum_{\substack{i=1 \\ i \neq p}}^m \frac{y_{iq}}{y_{pq}} \mathbf{a}_i.$$

Recall that in terms of the old augmented matrix, any vector \mathbf{a}_j , $m < j \leq n$, can be expressed as

$$\mathbf{a}_j = y_{1j} \mathbf{a}_1 + y_{2j} \mathbf{a}_2 + \cdots + y_{mj} \mathbf{a}_m.$$

Combining the last two equations yields

$$\mathbf{a}_j = \sum_{\substack{i=1 \\ i \neq p}}^m \left(y_{ij} - \frac{y_{pj}}{y_{pq}} y_{iq} \right) \mathbf{a}_i + \frac{y_{pj}}{y_{pq}} \mathbf{a}_q.$$

Denoting the entries of the new augmented matrix by y'_{ij} , we obtain

$$y'_{ij} = y_{ij} - \frac{y_{pj}}{y_{pq}} y_{iq}, \quad i \neq p,$$

$$y'_{pj} = \frac{y_{pj}}{y_{pq}}.$$

Therefore, the entries of the new canonical augmented matrix can be obtained from the entries of the old canonical augmented matrix via the formulas above. These equations are often called the *pivot equations*, and y_{pq} , the *pivot element*.

We refer to the operation on a given matrix by the formulas above as *pivoting about the (p, q) th element*. Note that pivoting about the (p, q) th element results in a matrix whose q th column has all zero entries, except the (p, q) th entry, which is unity. The pivoting operation can be accomplished via a sequence of elementary row operations, as was done in the proof of Theorem 16.1.

16.4 The Simplex Algorithm

The essence of the simplex algorithm is to move from one basic feasible solution to another until an optimal basic feasible solution is found. The canonical augmented matrix discussed in Section 16.3 plays a central role in the simplex algorithm.

Suppose that we are given the basic feasible solution

$$\mathbf{x} = [x_1, \dots, x_m, 0, \dots, 0]^\top, \quad x_i \geq 0, \quad i = 1, \dots, m$$

or equivalently

$$x_1 \mathbf{a}_1 + \cdots + x_m \mathbf{a}_m = \mathbf{b}.$$

In Section 16.3 we saw how to update the canonical augmented matrix if we wish to replace a basic column by a nonbasic column, that is, if we wish to change from one basis to another by replacing a single basic column. The values of the basic variables in a basic solution corresponding to a given basis are given in the last column of the canonical augmented matrix with respect to that basis; that is, $x_i = y_{i0}$, $i = 1, \dots, m$. Basic solutions are not necessarily feasible—the values of the basic variables may be negative. In the simplex method we want to move from one basic feasible solution to another. This means that we want to change basic columns in such a way that the last column of the canonical augmented matrix remains nonnegative. In this section we discuss a systematic method for doing this.

In the remainder of this chapter we assume that every basic feasible solution of

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b}, \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

is a nondegenerate basic feasible solution. We make this assumption primarily for convenience—all arguments can be extended to include degeneracy.

Let us start with the basic columns $\mathbf{a}_1, \dots, \mathbf{a}_m$, and assume that the corresponding basic solution $\mathbf{x} = [y_{10}, \dots, y_{m0}, 0, \dots, 0]^\top$ is feasible; that is, the

entries y_{i0} , $i = 1, \dots, m$, in the last column of the canonical augmented matrix are positive. Suppose that we now decide to make the vector \mathbf{a}_q , $q > m$, a basic column. We first represent \mathbf{a}_q in terms of the current basis as

$$\mathbf{a}_q = y_{1q}\mathbf{a}_1 + y_{2q}\mathbf{a}_2 + \cdots + y_{mq}\mathbf{a}_m.$$

Multiplying the above by $\varepsilon > 0$ yields

$$\varepsilon\mathbf{a}_q = \varepsilon y_{1q}\mathbf{a}_1 + \varepsilon y_{2q}\mathbf{a}_2 + \cdots + \varepsilon y_{mq}\mathbf{a}_m.$$

We combine this equation with

$$y_{10}\mathbf{a}_1 + \cdots + y_{m0}\mathbf{a}_m = \mathbf{b}$$

to get

$$(y_{10} - \varepsilon y_{1q})\mathbf{a}_1 + (y_{20} - \varepsilon y_{2q})\mathbf{a}_2 + \cdots + (y_{m0} - \varepsilon y_{mq})\mathbf{a}_m + \varepsilon\mathbf{a}_q = \mathbf{b}.$$

Note that the vector

$$\begin{bmatrix} y_{10} - \varepsilon y_{1q} \\ \vdots \\ y_{m0} - \varepsilon y_{mq} \\ 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{bmatrix},$$

where ε appears in the q th position, is a solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$. If $\varepsilon = 0$, then we obtain the old basic feasible solution. As ε is increased from zero, the q th component of the vector above increases. All other entries of this vector will increase or decrease linearly as ε is increased, depending on whether the corresponding y_{iq} is negative or positive. For small enough ε , we have a feasible but nonbasic solution. If any of the components decreases as ε increases, we choose ε to be the smallest value where one (or more) of the components vanishes. That is,

$$\varepsilon = \min_i \{y_{i0}/y_{iq} : y_{iq} > 0\}.$$

With this choice of ε we have a new basic feasible solution, with the vector \mathbf{a}_q replacing \mathbf{a}_p , where p corresponds to the minimizing index $p = \arg \min_i \{y_{i0}/y_{iq} : y_{iq} > 0\}$. So, we now have a new basis $\mathbf{a}_1, \dots, \mathbf{a}_{p-1}, \mathbf{a}_{p+1}, \dots, \mathbf{a}_m, \mathbf{a}_q$. As we can see, \mathbf{a}_p was replaced by \mathbf{a}_q in the new basis. We say that \mathbf{a}_q enters the basis and \mathbf{a}_p leaves the basis. If the minimum in $\min_i \{y_{i0}/y_{iq} : y_{iq} > 0\}$ is achieved by more than a single index, then the new solution is degenerate and any of the zero components can

be regarded as the component corresponding to the basic column that leaves the basis. If none of the y_{iq} are positive, then all components in the vector $[y_{10} - \varepsilon y_{1q}, \dots, y_{m0} - \varepsilon y_{mq}, 0, \dots, \varepsilon, \dots, 0]^\top$ increase (or remain constant) as ε is increased, and no new basic feasible solution is obtained, no matter how large we make ε . In this case there are feasible solutions having arbitrarily large components, which means that the set Ω of feasible solutions is unbounded.

So far, we have discussed how to change from one basis to another, while preserving feasibility of the corresponding basic solution, assuming that we have already chosen a nonbasic column to enter the basis. To complete our development of the simplex method, we need to consider two more issues. The first issue concerns the choice of which nonbasic column should enter the basis. The second issue is to find a stopping criterion, that is, a way to determine if a basic feasible solution is optimal or is not. To this end, suppose that we have found a basic feasible solution. The main idea of the simplex method is to move from one basic feasible solution (extreme point of the set Ω) to another basic feasible solution at which the value of the objective function is smaller. Because there is only a finite number of extreme points of the feasible set, the optimal point will be reached after a finite number of steps.

We already know how to move from one extreme point of the set Ω to a neighboring one by updating the canonical augmented matrix. To see which neighboring solution we should move to and when to stop moving, consider the following basic feasible solution:

$$[\mathbf{x}_B^\top, \mathbf{0}^\top]^\top = [y_{10}, \dots, y_{m0}, 0, \dots, 0]^\top$$

together with the corresponding canonical augmented matrix, having an identity matrix appearing in the first m columns. The value of the objective function for any solution \mathbf{x} is

$$z = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n.$$

For our basic solution, the value of the objective function is

$$z = z_0 = \mathbf{c}_B^\top \mathbf{x}_B = c_1 y_{10} + \cdots + c_m y_{m0},$$

where

$$\mathbf{c}_B^\top = [c_1, c_2, \dots, c_m].$$

To see how the value of the objective function changes when we move from one basic feasible solution to another, suppose that we choose the q th column, $m < q \leq n$, to enter the basis. To update the canonical augmented matrix, let $p = \arg \min_i \{y_{i0}/y_{iq} : y_{iq} > 0\}$ and $\varepsilon = y_{p0}/y_{pq}$. The new basic feasible

solution is

$$\begin{bmatrix} y_{10} - \varepsilon y_{1q} \\ \vdots \\ y_{m0} - \varepsilon y_{mq} \\ 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{bmatrix}.$$

Note that the single ε appears in the q th component, whereas the p th component is zero. Observe that we could have arrived at the basic feasible solution above simply by updating the canonical augmented matrix using the pivot equations from the previous Section 16.3:

$$\begin{aligned} y'_{ij} &= y_{ij} - \frac{y_{pj}}{y_{pq}} y_{iq}, \quad i \neq p, \\ y'_{pj} &= \frac{y_{pj}}{y_{pq}}, \end{aligned}$$

where the q th column enters the basis and the p th column leaves [i.e., we pivot about the (p, q) th element]. The values of the basic variables are entries in the last column of the updated canonical augmented matrix.

The cost for this new basic feasible solution is

$$\begin{aligned} z &= c_1(y_{10} - y_{1q}\varepsilon) + \cdots + c_m(y_{m0} - y_{mq}\varepsilon) + c_q\varepsilon \\ &= z_0 + [c_q - (c_1y_{1q} + \cdots + c_my_{mq})]\varepsilon, \end{aligned}$$

where $z_0 = c_1y_{10} + \cdots + c_my_{m0}$. Let

$$z_q = c_1y_{1q} + \cdots + c_my_{mq}.$$

Then,

$$z = z_0 + (c_q - z_q)\varepsilon.$$

Thus, if

$$z - z_0 = (c_q - z_q)\varepsilon < 0,$$

then the objective function value at the new basic feasible solution above is smaller than the objective function value at the original solution (i.e., $z < z_0$). Therefore, if $c_q - z_q < 0$, then the new basic feasible solution with a_q entering the basis has a lower objective function value.

On the other hand, if the given basic feasible solution is such that for all $q = m+1, \dots, n$,

$$c_q - z_q \geq 0,$$

then we can show that this solution is in fact an optimal solution. To show this, recall from Section 16.1 that any solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ can be represented as

$$\mathbf{x} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\mathbf{Y}_{m,n-m} \mathbf{x}_D \\ \mathbf{x}_D \end{bmatrix}$$

for some $\mathbf{x}_D = [x_{m+1}, \dots, x_n]^\top \in \mathbb{R}^{(n-m)}$. Using manipulations similar to the above, we obtain

$$\mathbf{c}^\top \mathbf{x} = z_0 + \sum_{i=m+1}^n (c_i - z_i)x_i,$$

where $z_i = c_1 y_{1i} + \dots + c_m y_{mi}$, $i = m+1, \dots, n$. For a feasible solution we have $x_i \geq 0$, $i = 1, \dots, n$. Therefore, if $c_i - z_i \geq 0$ for all $i = m+1, \dots, n$, then any feasible solution \mathbf{x} will have objective function value $\mathbf{c}^\top \mathbf{x}$ no smaller than z_0 .

Let $r_i = 0$ for $i = 1, \dots, m$ and $r_i = c_i - z_i$ for $i = m+1, \dots, n$. We call r_i the *i*th *reduced cost coefficient* or *relative cost coefficient*. Note that the reduced cost coefficients corresponding to basic variables are zero.

We summarize the discussion above with the following result.

Theorem 16.2 *A basic feasible solution is optimal if and only if the corresponding reduced cost coefficients are all nonnegative.* \square

At this point we have all the necessary steps for the simplex algorithm.

Simplex Algorithm

1. Form a canonical augmented matrix corresponding to an initial basic feasible solution.
2. Calculate the reduced cost coefficients corresponding to the nonbasic variables.
3. If $r_j \geq 0$ for all j , stop—the current basic feasible solution is optimal.
4. Select a q such that $r_q < 0$.
5. If no $y_{iq} > 0$, stop—the problem is unbounded; else, calculate $p = \arg \min_i \{y_{i0}/y_{iq} : y_{iq} > 0\}$. (If more than one index i minimizes y_{i0}/y_{iq} , we let p be the smallest such index.)
6. Update the canonical augmented matrix by pivoting about the (p, q) th element.
7. Go to step 2.

We state the following result for the simplex algorithm, which we have already proved in the foregoing discussion.

Theorem 16.3 Suppose that we have an LP problem in standard form that has an optimal feasible solution. If the simplex method applied to this problem terminates and the reduced cost coefficients in the last step are all nonnegative, then the resulting basic feasible solution is optimal. \square

Example 16.2 Consider the following linear program (see also Exercise 15.10):

$$\begin{aligned} & \text{maximize} && 2x_1 + 5x_2 \\ & \text{subject to} && x_1 \leq 4 \\ & && x_2 \leq 6 \\ & && x_1 + x_2 \leq 8 \\ & && x_1, x_2 \geq 0. \end{aligned}$$

We solve this problem using the simplex method.

Introducing slack variables, we transform the problem into standard form:

$$\begin{aligned} & \text{minimize} && -2x_1 - 5x_2 - 0x_3 - 0x_4 - 0x_5 \\ & \text{subject to} && x_1 + x_3 = 4 \\ & && x_2 + x_4 = 6 \\ & && x_1 + x_2 + x_5 = 8 \\ & && x_1, x_2, x_3, x_4, x_5 \geq 0. \end{aligned}$$

The starting canonical augmented matrix for this problem is

$$\begin{array}{ccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{b} \\ 1 & 0 & 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 1 & 0 & 6 \\ 1 & 1 & 0 & 0 & 1 & 8 \end{array}$$

Observe that the columns forming the identity matrix in the canonical augmented matrix above do not appear at the beginning. We could rearrange the augmented matrix so that the identity matrix would appear first. However, this is not essential from the computational point of view.

The starting basic feasible solution to the problem in standard form is

$$\mathbf{x} = [0, 0, 4, 6, 8]^T.$$

The columns \mathbf{a}_3 , \mathbf{a}_4 , and \mathbf{a}_5 corresponding to x_3 , x_4 , and x_5 are basic, and they form the identity matrix. The basis matrix is $\mathbf{B} = [\mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5] = \mathbf{I}_3$.

The value of the objective function corresponding to this basic feasible solution is $z = 0$. We next compute the reduced cost coefficients corresponding to the nonbasic variables x_1 and x_2 . They are

$$\begin{aligned} r_1 &= c_1 - z_1 = c_1 - (c_3 y_{11} + c_4 y_{21} + c_5 y_{31}) = -2, \\ r_2 &= c_2 - z_2 = c_2 - (c_3 y_{12} + c_4 y_{22} + c_5 y_{32}) = -5. \end{aligned}$$

We would like now to move to an adjacent basic feasible solution for which the objective function value is lower. Naturally, if there is more than one such solution, it is desirable to move to the adjacent basic feasible solution with the lowest objective value. A common practice is to select the most negative value of r_j and then to bring the corresponding column into the basis (see Exercise 16.18 for an alternative rule for choosing the column to bring into the basis). In our example, we bring a_2 into the basis; that is, we choose a_2 as the new basic column. We then compute $p = \arg \min\{y_{i0}/y_{i2} : y_{i2} > 0\} = 2$. We now update the canonical augmented matrix by pivoting about the (2, 2)th entry using the pivot equations:

$$\begin{aligned} y'_{ij} &= y_{ij} - \frac{y_{2j}}{y_{22}} y_{i2}, \quad i \neq 2, \\ y'_{2j} &= \frac{y_{2j}}{y_{22}}. \end{aligned}$$

The resulting updated canonical augmented matrix is

$$\begin{array}{cccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{b} \\ \hline 1 & 0 & 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 1 & 0 & 6 \\ 1 & 0 & 0 & -1 & 1 & 2 \end{array}$$

Note that a_2 entered the basis and a_4 left the basis. The corresponding basic feasible solution is $\mathbf{x} = [0, 6, 4, 0, 2]^\top$. We now compute the reduced cost coefficients for the nonbasic columns:

$$\begin{aligned} r_1 &= c_1 - z_1 = -2, \\ r_4 &= c_4 - z_4 = 5. \end{aligned}$$

Because $r_1 = -2 < 0$, the current solution is not optimal, and a lower objective function value can be obtained by bringing a_1 into the basis. Proceeding to update the canonical augmented matrix by pivoting about the (3, 1)th element, we obtain

$$\begin{array}{cccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{b} \\ \hline 0 & 0 & 1 & 1 & -1 & 2 \\ 0 & 1 & 0 & 1 & 0 & 6 \\ 1 & 0 & 0 & -1 & 1 & 2 \end{array}$$

The corresponding basic feasible solution is $\mathbf{x} = [2, 6, 2, 0, 0]^\top$. The reduced cost coefficients are

$$\begin{aligned} r_4 &= c_4 - z_4 = 3, \\ r_5 &= c_5 - z_5 = 2. \end{aligned}$$

Because no reduced cost coefficient is negative, the current basic feasible solution $\mathbf{x} = [2, 6, 2, 0, 0]^\top$ is optimal. The solution to the original problem is therefore $x_1 = 2$, $x_2 = 6$, and the objective function value is 34. ■

We can see from Example 16.2 that we can solve a linear programming problem of any size using the simplex algorithm. To make the calculations in the algorithm more efficient, we discuss the matrix form of the simplex method in the next section.

16.5 Matrix Form of the Simplex Method

Consider a linear programming problem in standard form:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Let the first m columns of \mathbf{A} be the basic columns. The columns form a square $m \times m$ nonsingular matrix \mathbf{B} . The nonbasic columns of \mathbf{A} form an $m \times (n-m)$ matrix \mathbf{D} . We partition the cost vector correspondingly as $\mathbf{c}^\top = [\mathbf{c}_B^\top, \mathbf{c}_D^\top]$. Then, the original linear program can be represented as follows:

$$\begin{aligned} & \text{minimize} && \mathbf{c}_B^\top \mathbf{x}_B + \mathbf{c}_D^\top \mathbf{x}_D \\ & \text{subject to} && [\mathbf{B}, \mathbf{D}] \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_D \end{bmatrix} = \mathbf{B}\mathbf{x}_B + \mathbf{D}\mathbf{x}_D = \mathbf{b} \\ & && \mathbf{x}_B \geq \mathbf{0}, \mathbf{x}_D \geq \mathbf{0}. \end{aligned}$$

If $\mathbf{x}_D = \mathbf{0}$, then the solution $\mathbf{x} = [\mathbf{x}_B^\top, \mathbf{x}_D^\top]^\top = [\mathbf{x}_B^\top, \mathbf{0}^\top]^\top$ is the basic feasible solution corresponding to the basis \mathbf{B} . It is clear that for this to be a solution, we need $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$; that is, the basic feasible solution is

$$\mathbf{x} = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}.$$

The corresponding objective function value is

$$z_0 = \mathbf{c}_B^\top \mathbf{B}^{-1}\mathbf{b}.$$

If, on the other hand, $\mathbf{x}_D \neq \mathbf{0}$, then the solution $\mathbf{x} = [\mathbf{x}_B^\top, \mathbf{x}_D^\top]^\top$ is not basic. In this case \mathbf{x}_B is given by

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{D}\mathbf{x}_D,$$

and the corresponding objective function value is

$$\begin{aligned} z &= \mathbf{c}_B^\top \mathbf{x}_B + \mathbf{c}_D^\top \mathbf{x}_D \\ &= \mathbf{c}_B^\top (\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{D}\mathbf{x}_D) + \mathbf{c}_D^\top \mathbf{x}_D \\ &= \mathbf{c}_B^\top \mathbf{B}^{-1}\mathbf{b} + (\mathbf{c}_D^\top - \mathbf{c}_B^\top \mathbf{B}^{-1}\mathbf{D})\mathbf{x}_D. \end{aligned}$$

Defining

$$\mathbf{r}_D^\top = \mathbf{c}_D^\top - \mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{D},$$

we obtain

$$z = z_0 + \mathbf{r}_D^\top \mathbf{x}_D.$$

The elements of the vector \mathbf{r}_D are the reduced cost coefficients corresponding to the nonbasic variables.

If $\mathbf{r}_D \geq \mathbf{0}$, then the basic feasible solution corresponding to the basis \mathbf{B} is optimal. If, on the other hand, a component of \mathbf{r}_D is negative, then the value of the objective function can be reduced by increasing a corresponding component of \mathbf{x}_D , that is, by changing the basis.

We now use the foregoing observations to develop a matrix form of the simplex method. To this end we first add the cost coefficient vector \mathbf{c}^\top to the bottom of the augmented matrix $[\mathbf{A}, \mathbf{b}]$ as follows:

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{c}^\top & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{B} & \mathbf{D} & \mathbf{b} \\ \mathbf{c}_B^\top & \mathbf{c}_D^\top & 0 \end{bmatrix}.$$

We refer to this matrix as the *tableau* of the given LP problem. The tableau contains all relevant information about the linear program.

Suppose that we now apply elementary row operations to the tableau such that the top part of the tableau corresponding to the augmented matrix $[\mathbf{A}, \mathbf{b}]$ is transformed into canonical form. This corresponds to premultiplying the tableau by the matrix

$$\begin{bmatrix} \mathbf{B}^{-1} & 0 \\ \mathbf{0}^\top & 1 \end{bmatrix}.$$

The result of this operation is

$$\begin{bmatrix} \mathbf{B}^{-1} & 0 \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{B} & \mathbf{D} & \mathbf{b} \\ \mathbf{c}_B^\top & \mathbf{c}_D^\top & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_m & \mathbf{B}^{-1}\mathbf{D} & \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{c}_B^\top & \mathbf{c}_D^\top & 0 \end{bmatrix}.$$

We now apply elementary row operations to the tableau above so that the entries of the last row corresponding to the basic columns become zero. Specifically, this corresponds to premultiplication of the tableau by the matrix

$$\begin{bmatrix} \mathbf{I}_m & 0 \\ -\mathbf{c}_B^\top & 1 \end{bmatrix}.$$

The result is

$$\begin{bmatrix} \mathbf{I}_m & 0 \\ -\mathbf{c}_B^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_m & \mathbf{B}^{-1}\mathbf{D} & \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{c}_B^\top & \mathbf{c}_D^\top & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_m & \mathbf{B}^{-1}\mathbf{D} & \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0}^\top & \mathbf{c}_D^\top - \mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{D} & -\mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{b} \end{bmatrix}.$$

We refer to the resulting tableau as the *canonical tableau corresponding to the basis \mathbf{B}* . Note that the first m entries of the last column of the canonical

tableau, $\mathbf{B}^{-1}\mathbf{b}$, are the values of the basic variables corresponding to the basis \mathbf{B} . The entries of $\mathbf{c}_D^\top - \mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{D}$ in the last row are the reduced cost coefficients. The last element in the last row of the tableau, $-\mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{b}$, is the negative of the value of the objective function corresponding to the basic feasible solution.

Given an LP problem, we can in general construct many different canonical tableaus, depending on which columns are basic. Suppose that we have a canonical tableau corresponding to a particular basis. Consider the task of computing the tableau corresponding to another basis that differs from the previous basis by a single vector. This can be accomplished by applying elementary row operations to the tableau in a similar fashion as discussed above. We refer to this operation as *updating* the canonical tableau. Note that updating of the tableau involves using exactly the same update equations as we used before in updating the canonical augmented matrix, namely, for $i = 1, \dots, m+1$,

$$\begin{aligned} y'_{ij} &= y_{ij} - \frac{y_{pj}}{y_{pq}} y_{iq}, \quad i \neq p, \\ y'_{pj} &= \frac{y_{pj}}{y_{pq}}, \end{aligned}$$

where y_{ij} and y'_{ij} are the (i, j) th entries of the original and updated canonical tableaus, respectively.

Working with the tableau is a convenient way of implementing the simplex algorithm, since updating the tableau immediately gives us the values of both the basic variables and the reduced cost coefficients. In addition, the (negative of the) value of the objective function can be found in the lower right-hand corner of the tableau. We illustrate the use of the tableau in the following example.

Example 16.3 Consider the following linear programming problem:

$$\begin{aligned} &\text{maximize} && 7x_1 + 6x_2 \\ &\text{subject to} && 2x_1 + x_2 \leq 3 \\ & && x_1 + 4x_2 \leq 4 \\ & && x_1, x_2 \geq 0. \end{aligned}$$

We first transform the problem into standard form so that the simplex method can be applied. To do this we change the maximization to minimization by multiplying the objective function by -1 . We then introduce two nonnegative slack variables, x_3 and x_4 , and construct the tableau for the problem:

$$\begin{array}{ccccc} & \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{b} \\ & 2 & 1 & 1 & 0 & 3 \\ & 1 & 4 & 0 & 1 & 4 \\ \mathbf{c}^\top & -7 & -6 & 0 & 0 & 0 \end{array}$$

Notice that this tableau is already in canonical form with respect to the basis $[a_3, a_4]$. Hence, the last row contains the reduced cost coefficients, and the rightmost column contains the values of the basic variables. Because $r_1 = -7$ is the most negative reduced cost coefficient, we bring a_1 into the basis. We then compute the ratios $y_{10}/y_{11} = 3/2$ and $y_{20}/y_{21} = 4$. Because $y_{10}/y_{11} < y_{20}/y_{21}$, we get $p = \arg \min_i \{y_{i0}/y_{i1} : y_{i1} > 0\} = 1$. We pivot about the $(1, 1)$ th element of the tableau to obtain

$$\begin{array}{ccccc} 1 & \frac{1}{2} & \frac{1}{2} & 0 & \frac{3}{2} \\ 0 & \frac{7}{2} & -\frac{1}{2} & 1 & \frac{5}{2} \\ 0 & -\frac{5}{2} & \frac{7}{2} & 0 & \frac{21}{2} \end{array}$$

In the second tableau above, only r_2 is negative. Therefore, $q = 2$ (i.e., we bring a_2 into the basis). Because

$$\frac{y_{10}}{y_{12}} = 3, \quad \frac{y_{20}}{y_{22}} = \frac{5}{7}$$

we have $p = 2$. We thus pivot about the $(2, 2)$ th element of the second tableau to obtain the third tableau:

$$\begin{array}{ccccc} 1 & 0 & \frac{4}{7} & -\frac{1}{7} & \frac{8}{7} \\ 0 & 1 & -\frac{1}{7} & \frac{2}{7} & \frac{5}{7} \\ 0 & 0 & \frac{22}{7} & \frac{5}{7} & \frac{86}{7} \end{array}$$

Because the last row of the third tableau above has no negative elements, we conclude that the basic feasible solution corresponding to the third tableau is optimal. Thus, $x_1 = 8/7$, $x_2 = 5/7$, $x_3 = 0$, $x_4 = 0$ is the solution to our LP in standard form, and the corresponding objective value is $-86/7$. The solution to the original problem is simply $x_1 = 8/7$, $x_2 = 5/7$, and the corresponding objective value is $86/7$. ■

Degenerate basic feasible solutions may arise in the course of applying the simplex algorithm. In such a situation, the minimum ratio y_{i0}/y_{iq} is 0. Therefore, even though the basis changes after we pivot about the (p, q) th element, the basic feasible solution does not (and remains degenerate). It is possible that if we start with a basis corresponding to a degenerate solution, several iterations of the simplex algorithm will involve the same degenerate solution, and eventually the original basis will occur. The entire process will then repeat indefinitely, leading to what is called *cycling*. Such a scenario, although rare in practice, is clearly undesirable. Fortunately, there is a simple rule for choosing q and p , due to Bland, that eliminates the cycling problem (see Exercise 16.18):

$$\begin{aligned} q &= \min\{i : r_i < 0\}, \\ p &= \min\{j : y_{j0}/y_{jq} = \min_i \{y_{i0}/y_{iq} : y_{iq} > 0\}\}. \end{aligned}$$

16.6 Two-Phase Simplex Method

The simplex method requires starting with a tableau for the problem in canonical form; that is, we need an initial basic feasible solution. A brute-force approach to finding a starting basic feasible solution is to choose m basic columns arbitrarily and transform the tableau for the problem into canonical form. If the rightmost column is positive, then we have a legitimate (initial) basic feasible solution. Otherwise, we would have to pick another candidate basis. Potentially, this brute-force procedure requires $\binom{n}{m}$ tries, and is therefore not practical.

Certain LP problems have obvious initial basic feasible solutions. For example, if we have constraints of the form $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and we add m slack variables z_1, \dots, z_m , then the constraints in standard form become

$$[\mathbf{A}, \mathbf{I}_m] \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} = \mathbf{b}, \quad \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} \geq \mathbf{0},$$

where $\mathbf{z} = [z_1, \dots, z_m]^\top$. The obvious initial basic feasible solution is

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix},$$

and the basic variables are the slack variables. This was the case in the example in Section 16.5.

Suppose that we are given a linear program in standard form:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

In general, an initial basic feasible solution is not always apparent. We therefore need a systematic method for finding an initial basic feasible solution for general LP problems so that the simplex method can be initialized. For this purpose, suppose that we are given an LP problem in standard form. Consider the following associated *artificial problem*:

$$\begin{aligned} & \text{minimize} && y_1 + y_2 + \cdots + y_m \\ & \text{subject to} && [\mathbf{A}, \mathbf{I}_m] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \mathbf{b} \\ & && \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{y} = [y_1, \dots, y_m]^\top$. We call \mathbf{y} the vector of *artificial variables*. Note that the artificial problem has an obvious initial basic feasible solution:

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix}.$$

We can therefore solve this problem by the simplex method.

Proposition 16.1 *The original LP problem has a basic feasible solution if and only if the associated artificial problem has an optimal feasible solution with objective function value zero.* \square

Proof. \Rightarrow : If the original problem has a basic feasible solution \mathbf{x} , then the vector $[\mathbf{x}^\top, \mathbf{0}^\top]^\top$ is a basic feasible solution to the artificial problem. Clearly, this solution has an objective function value of zero. This solution is therefore optimal for the artificial problem, since there can be no feasible solution with negative objective function value.

\Leftarrow : Suppose that the artificial problem has an optimal feasible solution with objective function value zero. Then, this solution must have the form $[\mathbf{x}^\top, \mathbf{0}^\top]^\top$, where $\mathbf{x} \geq \mathbf{0}$. Hence, we have $A\mathbf{x} = \mathbf{b}$, and \mathbf{x} is a feasible solution to the original problem. By the fundamental theorem of LP, there also exists a basic feasible solution. ■

Assume that the original LP problem has a basic feasible solution. Suppose that the simplex method applied to the associated artificial problem has terminated with an objective function value of zero. Then, as indicated in the proof above, the solution to the artificial problem will have all $y_i = 0$, $i = 1, \dots, m$. Hence, assuming nondegeneracy, the basic variables are in the first n components; that is, none of the artificial variables are basic. Therefore, the first n components form a basic feasible solution to the original problem. We can then use this basic feasible solution (resulting from the artificial problem) as the initial basic feasible solution for the original LP problem (after deleting the components corresponding to artificial variables). Thus, using artificial variables, we can attack a general linear programming problem by applying the *two-phase simplex method*. In phase I we introduce artificial variables and the artificial objective function and find a basic feasible solution. In phase II we use the basic feasible solution resulting from phase I to initialize the simplex algorithm to solve the original LP problem. The two-phase simplex method is illustrated in Figure 16.1.

Example 16.4 Consider the following linear programming problem:

$$\begin{aligned} &\text{minimize} && 2x_1 + 3x_2 \\ &\text{subject to} && 4x_1 + 2x_2 \geq 12 \\ & && x_1 + 4x_2 \geq 6 \\ & && x_1, x_2 \geq 0. \end{aligned}$$

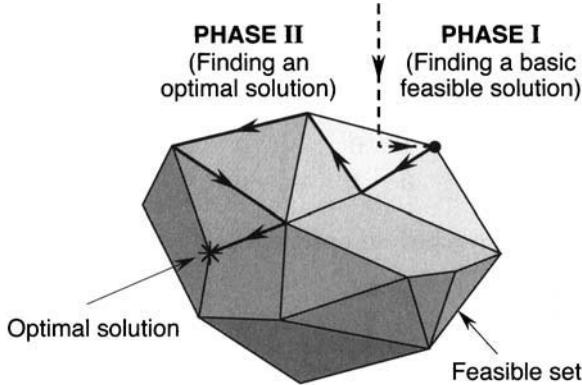


Figure 16.1 Illustration of the two-phase simplex method.

First, we express the problem in standard form by introducing surplus variables:

$$\begin{aligned} & \text{minimize} && 2x_1 + 3x_2 \\ & \text{subject to} && 4x_1 + 2x_2 - x_3 = 12 \\ & && x_1 + 4x_2 - x_4 = 6 \\ & && x_1, \dots, x_4 \geq 0. \end{aligned}$$

For the LP problem above there is no obvious basic feasible solution that we can use to initialize the simplex method. Therefore, we use the two-phase method.

Phase I. We introduce artificial variables $x_5, x_6 \geq 0$, and an artificial objective function $x_5 + x_6$. We form the corresponding tableau for the problem:

	a_1	a_2	a_3	a_4	a_5	a_6	b
	4	2	-1	0	1	0	12
	1	4	0	-1	0	1	6
c^T	0	0	0	0	1	1	0

To initiate the simplex procedure, we must update the last row of this tableau to transform it into canonical form. We obtain

	a_1	a_2	a_3	a_4	a_5	a_6	b
	4	2	-1	0	1	0	12
	1	4	0	-1	0	1	6
	-5	-6	1	1	0	0	-18

The basic feasible solution corresponding to this tableau is not optimal. Therefore, we proceed with the simplex method to obtain the next tableau:

$$\begin{array}{ccccccc} \frac{7}{2} & 0 & -1 & \frac{1}{2} & 1 & -\frac{1}{2} & 9 \\ \frac{1}{4} & 1 & 0 & -\frac{1}{4} & 0 & \frac{1}{4} & \frac{3}{2} \\ -\frac{7}{2} & 0 & 1 & -\frac{1}{2} & 0 & \frac{3}{2} & -9 \end{array}$$

We still have not yet reached an optimal basic feasible solution. Performing another iteration, we get

$$\begin{array}{ccccccc} 1 & 0 & -\frac{2}{7} & \frac{1}{7} & \frac{2}{7} & -\frac{1}{7} & \frac{18}{7} \\ 0 & 1 & \frac{1}{14} & -\frac{2}{7} & -\frac{1}{14} & \frac{2}{7} & \frac{6}{7} \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

Both of the artificial variables have been driven out of the basis, and the current basic feasible solution is optimal. We now proceed to phase II.

Phase II. We start by deleting the columns corresponding to the artificial variables in the last tableau in phase I and revert back to the original objective function. We obtain

$$\begin{array}{ccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{b} \\ 1 & 0 & -\frac{2}{7} & \frac{1}{7} & \frac{18}{7} \\ 0 & 1 & \frac{1}{14} & -\frac{2}{7} & \frac{6}{7} \\ \mathbf{c}^\top & 2 & 3 & 0 & 0 \end{array}$$

We transform the last row so that the zeros appear in the basis columns; that is, we transform the tableau above into canonical form:

$$\begin{array}{ccccc} 1 & 0 & -\frac{2}{7} & \frac{1}{7} & \frac{18}{7} \\ 0 & 1 & \frac{1}{14} & -\frac{2}{7} & \frac{6}{7} \\ 0 & 0 & \frac{5}{14} & \frac{4}{7} & -\frac{54}{7} \end{array}$$

All the reduced cost coefficients are nonnegative. Hence, the optimal solution is

$$\mathbf{x} = \left[\frac{18}{7}, \frac{6}{7}, 0, 0 \right]^\top$$

and the optimal cost is $54/7$. ■

16.7 Revised Simplex Method

Consider an LP problem in standard form with a matrix \mathbf{A} of size $m \times n$. Suppose that we use the simplex method to solve the problem. Experience suggests that if m is much smaller than n , then, in most instances, pivots will

occur in only a small fraction of the columns of the matrix \mathbf{A} . The operation of pivoting involves updating all the columns of the tableau. However, if a particular column of \mathbf{A} never enters any basis during the entire simplex procedure, then computations performed on this column are never used. Therefore, if m is much smaller than n , the effort expended on performing operations on many of the columns of \mathbf{A} may be wasted. The *revised simplex method* reduces the amount of computation leading to an optimal solution by eliminating operations on columns of \mathbf{A} that do not enter the bases.

To be specific, suppose that we are at a particular iteration in the simplex algorithm. Let \mathbf{B} be the matrix composed of the columns of \mathbf{A} forming the current basis, and let \mathbf{D} be the matrix composed of the remaining columns of \mathbf{A} . The sequence of elementary row operations on the tableau leading to this iteration (represented by matrices $\mathbf{E}_1, \dots, \mathbf{E}_k$) corresponds to premultiplying \mathbf{B} , \mathbf{D} , and \mathbf{b} by $\mathbf{B}^{-1} = \mathbf{E}_k \cdots \mathbf{E}_1$. In particular, the vector of current values of the basic variables is $\mathbf{B}^{-1}\mathbf{b}$. Observe that computation of the current basic feasible solution does not require computation of $\mathbf{B}^{-1}\mathbf{D}$; all we need is the matrix \mathbf{B}^{-1} . In the revised simplex method we do not compute $\mathbf{B}^{-1}\mathbf{D}$. Instead, we only keep track of the basic variables and the revised tableau, which is the tableau $[\mathbf{B}^{-1}, \mathbf{B}^{-1}\mathbf{b}]$. Note that this tableau is only of size $m \times (m+1)$ [compared to the tableau in the original simplex method, which is $m \times (n+1)$]. To see how to update the revised tableau, suppose that we choose the column \mathbf{a}_q to enter the basis. Let $\mathbf{y}_q = \mathbf{B}^{-1}\mathbf{a}_q$, $\mathbf{y}_0 = [y_{01}, \dots, y_{0m}]^\top = \mathbf{B}^{-1}\mathbf{b}$, and $p = \arg \min_i \{y_{i0}/y_{iq} : y_{iq} > 0\}$ (as in the original simplex method). Then, to update the revised tableau, we form the augmented revised tableau $[\mathbf{B}^{-1}, \mathbf{y}_0, \mathbf{y}_q]$ and pivot about the p th element of the last column. We claim that the first $m+1$ columns of the resulting matrix comprise the updated revised tableau (i.e., we simply remove the last column of the updated augmented revised tableau to obtain the updated revised tableau). To see this, write \mathbf{B}^{-1} as $\mathbf{B}^{-1} = \mathbf{E}_k \cdots \mathbf{E}_1$, and let the matrix \mathbf{E}_{k+1} represent the pivoting operation above (i.e., $\mathbf{E}_{k+1}\mathbf{y}_q = \mathbf{e}_p$, the p th column of the $m \times m$ identity matrix). The matrix \mathbf{E}_{k+1} is given by

$$\mathbf{E}_{k+1} = \begin{bmatrix} 1 & -y_{1q}/y_{pq} & 0 \\ \ddots & \vdots & \\ & 1/y_{pq} & \\ & \vdots & \ddots \\ 0 & -y_{mq}/y_{pq} & 1 \end{bmatrix}.$$

Then, the updated augmented tableau resulting from the pivoting operation above is $[\mathbf{E}_{k+1}\mathbf{B}^{-1}, \mathbf{E}_{k+1}\mathbf{y}_0, \mathbf{e}_p]$. Let \mathbf{B}_{new} be the new basis. Then, we have $\mathbf{B}_{\text{new}}^{-1} = \mathbf{E}_{k+1} \cdots \mathbf{E}_1$. But notice that $\mathbf{B}_{\text{new}}^{-1} = \mathbf{E}_{k+1}\mathbf{B}^{-1}$, and the values of the basic variables corresponding to \mathbf{B}_{new} are given by $\mathbf{y}_{0\text{new}} = \mathbf{E}_{k+1}\mathbf{y}_0$. Hence, the updated tableau is indeed $[\mathbf{B}_{\text{new}}^{-1}, \mathbf{y}_{0\text{new}}] = [\mathbf{E}_{k+1}\mathbf{B}^{-1}, \mathbf{E}_{k+1}\mathbf{y}_0]$.

We summarize the foregoing discussion in the following algorithm.

Revised Simplex Method

1. Form a revised tableau corresponding to an initial basic feasible solution $[\mathbf{B}^{-1}, \mathbf{y}_0]$.
2. Calculate the current reduced cost coefficients vector via

$$\mathbf{r}_D^\top = \mathbf{c}_D^\top - \boldsymbol{\lambda}^\top \mathbf{D},$$

where

$$\boldsymbol{\lambda}^\top = \mathbf{c}_B^\top \mathbf{B}^{-1}.$$

3. If $r_j \geq 0$ for all j , stop—the current basic feasible solution is optimal.
4. Select a q such that $r_q < 0$ (e.g., the q corresponding to the most negative r_q), and compute

$$\mathbf{y}_q = \mathbf{B}^{-1} \mathbf{a}_q.$$

5. If no $y_{iq} > 0$, stop—the problem is unbounded; else, compute $p = \arg \min_i \{y_{io}/y_{iq} : y_{iq} > 0\}$.
6. Form the augmented revised tableau $[\mathbf{B}^{-1}, \mathbf{y}_0, \mathbf{y}_q]$, and pivot about the p th element of the last column. Form the updated revised tableau by taking the first $m+1$ columns of the resulting augmented revised tableau (i.e., remove the last column).
7. Go to step 2.

The reason for computing \mathbf{r}_D in two steps as indicated in step 2 is as follows. We first note that $\mathbf{r}_D = \mathbf{c}_D^\top - \mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{D}$. To compute $\mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{D}$, we can do the multiplication in the order either $(\mathbf{c}_B^\top \mathbf{B}^{-1}) \mathbf{D}$ or $\mathbf{c}_B^\top (\mathbf{B}^{-1} \mathbf{D})$. The former involves two vector-matrix multiplications, whereas the latter involves a matrix-matrix multiplication followed by a vector-matrix multiplication. Clearly, the former is more efficient.

As in the original simplex method, we can use the two-phase method to solve a given LP problem using the revised simplex method. In particular, we use the revised tableau from the final step of phase I as the initial revised tableau in phase II. We illustrate the method in the following example.

Example 16.5 Consider solving the following LP problem using the revised simplex method:

$$\begin{aligned} & \text{maximize} && 3x_1 + 5x_2 \\ & \text{subject to} && x_1 + x_2 \leq 4 \\ & && 5x_1 + 3x_2 \geq 8 \\ & && x_1, x_2 \geq 0. \end{aligned}$$

First, we express the problem in standard form by introducing one slack and one surplus variable, to obtain

$$\begin{aligned} \text{minimize} \quad & -3x_1 - 5x_2 \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 4 \\ & 5x_1 + 3x_2 - x_4 = 8 \\ & x_1, \dots, x_4 \geq 0. \end{aligned}$$

There is no obvious basic feasible solution to this LP problem. Therefore, we use the two-phase method.

Phase I. We introduce one artificial variable x_5 and an artificial objective function x_5 . The tableau for the artificial problem is

$$\begin{array}{ccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{b} \\ 1 & 1 & 1 & 0 & 0 & 4 \\ 5 & 3 & 0 & -1 & 1 & 8 \\ \mathbf{c}^\top & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

We start with an initial basic feasible solution and corresponding \mathbf{B}^{-1} , as shown in the following revised tableau:

$$\begin{array}{c|ccc} \text{Variable} & \mathbf{B}^{-1} & \mathbf{y}_0 \\ \hline x_3 & 1 & 0 & 4 \\ x_5 & 0 & 1 & 8 \end{array}$$

We compute

$$\begin{aligned} \boldsymbol{\lambda}^\top &= \mathbf{c}_B^\top \mathbf{B}^{-1} = [0, 1], \\ \mathbf{r}_D^\top &= \mathbf{c}_D^\top - \boldsymbol{\lambda}^\top \mathbf{D} = [0, 0, 0] - [5, 3, -1] = [-5, -3, 1] = [r_1, r_2, r_4]. \end{aligned}$$

Because r_1 is the most negative reduced cost coefficient, we bring \mathbf{a}_1 into the basis. To do this, we first compute $\mathbf{y}_1 = \mathbf{B}^{-1} \mathbf{a}_1$. In this case, $\mathbf{y}_1 = \mathbf{a}_1$. We get the augmented revised tableau:

$$\begin{array}{c|ccc|c} \text{Variable} & \mathbf{B}^{-1} & \mathbf{y}_0 & \mathbf{y}_1 \\ \hline x_3 & 1 & 0 & 4 & 1 \\ x_5 & 0 & 1 & 8 & 5 \end{array}$$

We then compute $p = \arg \min_i \{y_{i0}/y_{iq} : y_{iq} > 0\} = 2$ and pivot about the second element of the last column to get the updated revised tableau:

$$\begin{array}{c|ccc|c} \text{Variable} & \mathbf{B}^{-1} & \mathbf{y}_0 \\ \hline x_3 & 1 & -\frac{1}{5} & \frac{12}{5} \\ x_1 & 0 & \frac{1}{5} & \frac{8}{5} \end{array}$$

We next compute

$$\begin{aligned}\boldsymbol{\lambda}^\top &= \mathbf{c}_B^\top \mathbf{B}^{-1} = [0, 0], \\ \mathbf{r}_D^\top &= \mathbf{c}_D^\top - \boldsymbol{\lambda}^\top \mathbf{D} = [0, 0, 1] = [r_2, r_4, r_5] \geq \mathbf{0}^\top.\end{aligned}$$

The reduced cost coefficients are all nonnegative. Hence, the solution to the artificial problem is $[8/5, 0, 12/5, 0, 0]^\top$. The initial basic feasible solution for phase II is therefore $[8/5, 0, 12/5, 0]^\top$.

Phase II. The tableau for the original problem (in standard form) is

$$\begin{array}{ccccc|c} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{b} \\ 1 & 1 & 1 & 0 & 4 \\ 5 & 3 & 0 & -1 & 8 \\ \mathbf{c}^\top & -3 & -5 & 0 & 0 & 0 \end{array}$$

As the initial revised tableau for phase II, we take the final revised tableau from phase I. We then compute

$$\begin{aligned}\boldsymbol{\lambda}^\top &= \mathbf{c}_B^\top \mathbf{B}^{-1} = [0, -3] \begin{bmatrix} 1 & -\frac{1}{5} \\ 0 & \frac{1}{5} \end{bmatrix} = \left[0, -\frac{3}{5}\right], \\ \mathbf{r}_D^\top &= \mathbf{c}_D^\top - \boldsymbol{\lambda}^\top \mathbf{D} = [-5, 0] - \left[0, -\frac{3}{5}\right] \begin{bmatrix} 1 & 0 \\ 3 & -1 \end{bmatrix} = \left[-\frac{16}{5}, -\frac{3}{5}\right] = [r_2, r_4].\end{aligned}$$

We bring \mathbf{a}_2 into the basis, and compute $\mathbf{y}_2 = \mathbf{B}^{-1} \mathbf{a}_2$ to get

$$\begin{array}{ccccc} \text{Variable} & \mathbf{B}^{-1} & \mathbf{y}_0 & \mathbf{y}_2 \\ \hline x_3 & 1 & -\frac{1}{5} & \frac{12}{5} & \frac{2}{5} \\ x_1 & 0 & \frac{1}{5} & \frac{8}{5} & \frac{3}{5} \end{array}$$

In this case we get $p = 2$. We update this tableau by pivoting about the second element of the last column to get

$$\begin{array}{ccccc} \text{Variable} & \mathbf{B}^{-1} & \mathbf{y}_0 \\ \hline x_3 & 1 & -\frac{1}{3} & \frac{4}{3} \\ x_2 & 0 & \frac{1}{3} & \frac{8}{3} \end{array}$$

We compute

$$\begin{aligned}\boldsymbol{\lambda}^\top &= \mathbf{c}_B^\top \mathbf{B}^{-1} = [0, -5] \begin{bmatrix} 1 & -\frac{1}{3} \\ 0 & \frac{1}{3} \end{bmatrix} = \left[0, -\frac{5}{3}\right], \\ \mathbf{r}_D^\top &= \mathbf{c}_D^\top - \boldsymbol{\lambda}^\top \mathbf{D} = [-3, 0] - \left[0, -\frac{5}{3}\right] \begin{bmatrix} 1 & 0 \\ 5 & -1 \end{bmatrix} = \left[\frac{16}{3}, -\frac{5}{3}\right] = [r_1, r_4].\end{aligned}$$

We now bring a_4 into the basis:

Variable	\mathbf{B}^{-1}	\mathbf{y}_0	\mathbf{y}_4
x_3	1	$-\frac{1}{3}$	$\frac{4}{3}$
x_2	0	$\frac{1}{3}$	$-\frac{1}{3}$

We update the tableau to obtain

Variable	\mathbf{B}^{-1}	\mathbf{y}_0
x_4	3	-1
x_2	1	0

We compute

$$\boldsymbol{\lambda}^\top = \mathbf{c}_B^\top \mathbf{B}^{-1} = [0, -5] \begin{bmatrix} 3 & -1 \\ 1 & 0 \end{bmatrix} = [-5, 0],$$

$$\mathbf{r}_D^\top = \mathbf{c}_D^\top - \boldsymbol{\lambda}^\top \mathbf{D} = [-3, 0] - [-5, 0] \begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix} = [2, 5] = [r_1, r_3].$$

The reduced cost coefficients are all positive. Hence, $[0, 4, 0, 4]^\top$ is optimal. The optimal solution to the original problem is $[0, 4]^\top$. ■

EXERCISES

16.1 This question is concerned with elementary row operations and rank.

- a. For the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & -1 & 3 & 2 \\ 2 & -1 & 3 & 0 & 1 \\ 3 & 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 1 & 1 \end{bmatrix},$$

find its rank by first transforming the matrix using elementary row operations into an upper triangular form.

- b. Find the rank of the following matrix for different values of the parameter γ by first transforming the matrix using elementary row operations into an upper triangular form:

$$\mathbf{A} = \begin{bmatrix} 1 & \gamma & -1 & 2 \\ 2 & -1 & \gamma & 5 \\ 1 & 10 & -6 & 1 \end{bmatrix}.$$

16.2 Consider the following standard form LP problem:

$$\begin{aligned} & \text{minimize} && 2x_1 - x_2 - x_3 \\ & \text{subject to} && 3x_1 + x_2 + x_4 = 4 \\ & && 6x_1 + 2x_2 + x_3 + x_4 = 5 \\ & && x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

- a. Write down the \mathbf{A} , \mathbf{b} , and \mathbf{c} matrices/vectors for the problem.
- b. Consider the basis consisting of the third and fourth columns of \mathbf{A} , ordered according to $[a_4, a_3]$. Compute the canonical tableau corresponding to this basis.
- c. Write down the basic feasible solution corresponding to the basis above, and its objective function value.
- d. Write down the values of the reduced cost coefficients (for all the variables) corresponding to the basis.
- e. Is the basic feasible solution in part c an optimal feasible solution? If yes, explain why. If not, determine which element of the canonical tableau to pivot about so that the new basic feasible solution will have a lower objective function value.
- f. Suppose that we apply the two-phase method to the problem, and at the end of phase I, the tableau for the artificial problem is

$$\begin{array}{ccccccc} 0 & 0 & -1 & 1 & 2 & -1 & 3 \\ 1 & \frac{1}{3} & \frac{1}{3} & 0 & -\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

Does the original problem have a basic feasible solution? Explain.

- g. From the final tableau for phase I in part f, find the initial canonical tableau for phase II.

16.3 Use the simplex method to solve the following linear program:

$$\begin{aligned} & \text{maximize} && x_1 + x_2 + 3x_3 \\ & \text{subject to} && x_1 + x_3 = 1 \\ & && x_2 + x_3 = 2 \\ & && x_1, x_2, x_3 \geq 0. \end{aligned}$$

16.4 Consider the linear program

$$\begin{aligned} & \text{maximize} && 2x_1 + x_2 \\ & \text{subject to} && 0 \leq x_1 \leq 5 \\ & && 0 \leq x_2 \leq 7 \\ & && x_1 + x_2 \leq 9. \end{aligned}$$

Convert the problem to standard form and solve it using the simplex method.

16.5 Consider a standard form linear programming problem with

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} ? & ? & 0 & 1 \\ ? & ? & 1 & 0 \end{bmatrix}, & \mathbf{b} &= \begin{bmatrix} 5 \\ 6 \end{bmatrix}, \\ \mathbf{c}^\top &= \begin{bmatrix} 8 & 7 & ? & ? \end{bmatrix}, \end{aligned}$$

where the “?” symbols signify unknowns to be determined. Suppose that the canonical tableau corresponding to some basis is

$$\left[\begin{array}{ccccc} 0 & 1 & 1 & 2 & ? \\ 1 & 0 & 3 & 4 & ? \\ 0 & 0 & -1 & 1 & ? \end{array} \right].$$

- a. Find all entries of \mathbf{A} .
- b. Find all entries of \mathbf{c} .
- c. Find the basic feasible solution corresponding to the canonical tableau above.
- d. Find all entries in the rightmost column of the tableau.

16.6 Consider the optimization problem

$$\begin{aligned} & \text{minimize} && c_1|x_1| + c_2|x_2| + \cdots + c_n|x_n| \\ & \text{subject to} && \mathbf{Ax} = \mathbf{b}. \end{aligned}$$

We can convert this problem into an equivalent standard form linear programming problem by introducing the new variables

$$x_i = x_i^+ - x_i^- \text{ where } x_i^+ \geq 0, \quad x_i^- \geq 0, \quad i = 1, 2, \dots, n$$

and

$$|x_i| = x_i^+ + x_i^-, \quad i = 1, 2, \dots, n$$

(See also Exercise 15.3.) Then we can apply the simplex method to solve the equivalent problem. Explain, in two or three sentences, why we will always

have that only either x_i^+ or x_i^- can be positive but never both x_i^+ and x_i^- can be positive. In other words, we will always have, $x_i^+ x_i^- = 0$.

16.7 Suppose that we are given a linear programming problem in standard form (written in the usual notation) and are told that the vector $\mathbf{x} = [1, 0, 2, 3, 0]^\top$ is a basic feasible solution with corresponding relative cost coefficient vector $\mathbf{r} = [0, 1, 0, 0, -1]^\top$ and objective function value 6. We are also told that the vector $[-2, 0, 0, 0, 4]^\top$ lies in the nullspace of \mathbf{A} .

- Write down the canonical tableau corresponding to the given basic feasible solution above, filling in as many values of entries as possible (use the symbol * for entries that cannot be determined from the information given). Clearly indicate the dimensions of the tableau.
- Find a feasible solution with an objective function value that is strictly less than 6.

16.8 Consider a standard form linear programming problem (with the usual \mathbf{A} , \mathbf{b} , and \mathbf{c}). Suppose that it has the following canonical tableau:

$$\begin{array}{ccccccc} 0 & 1 & 0 & 1 & -1 & 5 \\ 1 & 2 & 0 & 0 & -2 & 6 \\ 0 & 3 & 1 & 0 & -3 & 7 \\ 0 & 4 & 0 & 0 & -4 & 8 \end{array}$$

- Find the basic feasible solution corresponding to this canonical tableau and the corresponding value of the objective function.
- Find all the reduced cost coefficient values associated with the tableau.
- Does the given linear programming problem have feasible solutions with arbitrarily negative objective function values?
- Suppose that column a_2 enters the basis. Find the canonical tableau for the new basis.
- Find a feasible solution with objective function value equal to -100 .
- Find a basis for the nullspace of \mathbf{A} .

16.9 Consider the problem

$$\begin{aligned} \text{maximize } & -x_1 - 2x_2 \\ \text{subject to } & x_1 \geq 0 \\ & x_2 \geq 1. \end{aligned}$$

- a. Convert the problem into a standard form linear programming problem.
- b. Use the two-phase simplex method to compute the solution to this problem and the value of the objective function at the optimal solution of the problem.

16.10 Consider the linear programming problem

$$\begin{aligned} & \text{minimize} && -x_1 \\ & \text{subject to} && x_1 - x_2 = 1 \\ & && x_1, x_2 \geq 0. \end{aligned}$$

- a. Write down the basic feasible solution for x_1 as a basic variable.
- b. Compute the canonical augmented matrix corresponding to the basis in part a.
- c. If we apply the simplex algorithm to this problem, under what circumstance does it terminate? (In other words, which stopping criterion in the simplex algorithm is satisfied?)
- d. Show that in this problem, the objective function can take arbitrarily negative values over the constraint set.

16.11 Find the solution and the value of the optimal cost for the following problem using the revised simplex method:

$$\begin{aligned} & \text{minimize} && x_1 + x_2 \\ & \text{subject to} && x_1 + 2x_2 \geq 3 \\ & && 2x_1 + x_2 \geq 3 \\ & && x_1, x_2 \geq 0. \end{aligned}$$

Hint: Start with x_1 and x_2 as basic variables.

16.12 Solve the following linear programs using the revised simplex method:

- a. Maximize $-4x_1 - 3x_2$ subject to

$$\begin{aligned} & 5x_1 + x_2 \geq 11 \\ & -2x_1 - x_2 \leq -8 \\ & x_1 + 2x_2 \geq 7 \\ & x_1, x_2 \geq 0. \end{aligned}$$

- b. Maximize $6x_1 + 4x_2 + 7x_3 + 5x_4$ subject to

$$\begin{aligned}x_1 + 2x_2 + x_3 + 2x_4 &\leq 20 \\6x_1 + 5x_2 + 3x_3 + 2x_4 &\leq 100 \\3x_1 + 4x_2 + 9x_3 + 12x_4 &\leq 75 \\x_1, x_2, x_3, x_4 &\geq 0.\end{aligned}$$

- 16.13** Consider a standard form linear programming problem with

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 6 \\ c_2 \\ 4 \\ 5 \end{bmatrix}.$$

Suppose that we are told that the reduced cost coefficient vector corresponding to some basis is $\mathbf{r}^\top = [0, 1, 0, 0]$.

- a. Find an optimal feasible solution to the problem.
- b. Find c_2 .

- 16.14** Consider the linear programming problem

$$\begin{aligned}\text{minimize} \quad & c_1 x_1 + c_2 x_2 \\ \text{subject to} \quad & 2x_1 + x_2 = 2 \\ & x_1, x_2 \geq 0,\end{aligned}$$

where $c_1, c_2 \in \mathbb{R}$. Suppose that the problem has an optimal feasible solution that is not basic.

- a. Find all basic feasible solutions.
- b. Find all possible values of c_1 and c_2 .
- c. At each basic feasible solution, compute the reduced cost coefficients for all nonbasic variables.

- 16.15** Suppose that we apply the simplex method to a given linear programming problem and obtain the following canonical tableau:

$$\begin{array}{ccccc|c} 0 & \beta & 0 & 1 & 4 \\ 1 & \gamma & 0 & 0 & 5 \\ 0 & -3 & 1 & 0 & 6 \\ 0 & 2 - \alpha & 0 & 0 & \delta \end{array}$$

For each of the following conditions, find the set of all parameter values $\alpha, \beta, \gamma, \delta$ that satisfy the condition.

- The problem has no solution because the objective function values are unbounded.
- The current basic feasible solution is optimal, and the corresponding objective function value is 7.
- The current basic feasible solution is not optimal, and the objective function value strictly decreases if we remove the first column of \mathbf{A} from the basis.

16.16 You are given a linear programming problem in standard form. Suppose that you use the two-phase simplex method and arrive at the following canonical tableau in phase I:

$$\left[\begin{array}{ccccccc|c} ? & 0 & 1 & 1 & ? & ? & 0 & 6 \\ ? & 0 & 0 & ? & ? & ? & 1 & \alpha \\ ? & 1 & 0 & ? & ? & ? & 0 & 5 \\ \gamma & 0 & 0 & \delta & ? & ? & \beta & 0 \end{array} \right].$$

The variables α, β, γ , and δ are unknowns to be determined. Those entries marked with “?” are unspecified. The only thing you are told is that the value of γ is either 2 or -1 .

- Determine the values of α, β, γ , and δ .
- Does the given linear programming problem have a feasible solution? If yes, find it. If not, explain why.

16.17 Suppose we are given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a vector $\mathbf{b} \in \mathbb{R}^m$ such that $\mathbf{b} \geq \mathbf{0}$. We are interested in an algorithm that, given this \mathbf{A} and \mathbf{b} , is guaranteed to produce one of following two outputs: (1) If there exists \mathbf{x} such that $\mathbf{Ax} \geq \mathbf{b}$, then the algorithm produces one such \mathbf{x} . (2) If no such \mathbf{x} exists, then the algorithm produces an output to declare so.

Describe in detail how to design this algorithm based on the simplex method.

16.18 Consider the following linear programming problem (attributed to Beale—see [42, p. 43]):

$$\begin{aligned} \text{minimize} \quad & -\frac{3}{4}x_4 + 20x_5 - \frac{1}{2}x_6 + 6x_7 \\ \text{subject to} \quad & x_1 + \frac{1}{4}x_4 - 8x_5 - x_6 + 9x_7 = 0 \\ & x_2 + \frac{1}{2}x_4 - 12x_5 - \frac{1}{2}x_6 + 3x_7 = 0 \\ & x_3 + x_6 = 1 \\ & x_1, \dots, x_7 \geq 0. \end{aligned}$$

- a. Apply the simplex algorithm to the problem using the rule that q is the index corresponding to the most negative r_q . (As usual, if more than one index i minimizes y_{i0}/y_{iq} , let p be the smallest such index.) Start with x_1 , x_2 , and x_3 as initial basic variables. Notice that cycling occurs.
- b. Repeat part a using *Bland's rule* for choosing q and p :

$$\begin{aligned} q &= \min\{i : r_i < 0\}, \\ p &= \min\{j : y_{j0}/y_{jq} = \min_i\{y_{i0}/y_{iq} : y_{iq} > 0\}\}. \end{aligned}$$

Note that Bland's rule for choosing p corresponds to our usual rule that if more than one index i minimizes y_{i0}/y_{iq} , we let p be the smallest such index.

16.19 Consider a standard form LP problem. Suppose that we start with an initial basic feasible solution $\mathbf{x}^{(0)}$ and apply one iteration of the simplex algorithm to obtain $\mathbf{x}^{(1)}$.

We can express $\mathbf{x}^{(1)}$ in terms of $\mathbf{x}^{(0)}$ as

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)},$$

where α_0 minimizes $\phi(\alpha) = f(\mathbf{x}^{(0)} + \alpha \mathbf{d}^{(0)})$ over all $\alpha > 0$ such that $\mathbf{x}^{(0)} + \alpha \mathbf{d}^{(0)}$ is feasible.

- a. Show that $\mathbf{d}^{(0)} \in \mathcal{N}(\mathbf{A})$.
- b. As usual, assume that the initial basis is the first m columns of \mathbf{A} , and the first iteration involves inserting \mathbf{a}_q into the basis, where $q > m$. Let the q th column of the canonical augmented matrix be $\mathbf{y}_q = [y_{1q}, \dots, y_{mq}]^\top$. Express $\mathbf{d}^{(0)}$ in terms of \mathbf{y}_q .

16.20 Write a simple MATLAB function that implements the simplex algorithm. The inputs are \mathbf{c} , \mathbf{A} , \mathbf{b} , and \mathbf{v} , where \mathbf{v} is the vector of indices of basic

columns. Assume that the augmented matrix $[\mathbf{A}, \mathbf{b}]$ is already in canonical form; that is, the v_i th column of \mathbf{A} is $[0, \dots, 1, \dots, 0]^\top$, where 1 occurs in the i th position. The function should output the final solution and the vector of indices of basic columns. Test the MATLAB function on the problem in Example 16.2.

16.21 Write a MATLAB routine that implements the two-phase simplex method. It may be useful to use the MATLAB function of Exercise 16.20. Test the routine on the problem in Example 16.5.

16.22 Write a simple MATLAB function that implements the revised simplex algorithm. The inputs are \mathbf{c} , \mathbf{A} , \mathbf{b} , \mathbf{v} , and \mathbf{B}^{-1} , where \mathbf{v} is the vector of indices of basic columns; that is, the i th column of \mathbf{B} is the v_i th column of \mathbf{A} . The function should output the final solution, the vector of indices of basic columns, and the final \mathbf{B}^{-1} . Test the MATLAB function on the problem in Example 16.2.

16.23 Write a MATLAB routine that implements the two-phase revised simplex method. It may be useful to use the MATLAB function of Exercise 16.22. Test the routine on the problem in Example 16.5.

CHAPTER 17

DUALITY

17.1 Dual Linear Programs

Associated with every linear programming problem is a corresponding dual linear programming problem. The dual problem is constructed from the cost and constraints of the original, or primal, problem. Being an LP problem, the dual can be solved using the simplex method. However, as we shall see, the solution to the dual can also be obtained from the solution of the primal problem, and vice versa. Solving an LP problem via its dual may be simpler in certain cases, and also often provides further insight into the nature of the problem. In this chapter we study basic properties of duality and provide an interpretive example of duality. Duality can be used to improve the performance of the simplex algorithm (leading to the primal-dual algorithm), as well as to develop nonsimplex algorithms for solving LP problems (such as Khachiyan's algorithm and Karmarkar's algorithm). We do not discuss this aspect of duality further in this chapter. For an in-depth discussion of the primal-dual method, as well as other aspects of duality, see, for example, [88]. For a description of Khachiyan's algorithm and Karmarkar's algorithm, see Chapter 18.

Suppose that we are given a linear programming problem of the form

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

We refer to the above as the *primal problem*. We define the corresponding *dual problem* as

$$\begin{aligned} & \text{maximize} && \boldsymbol{\lambda}^\top \mathbf{b} \\ & \text{subject to} && \boldsymbol{\lambda}^\top \mathbf{A} \leq \mathbf{c}^\top \\ & && \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

We refer to the variable $\boldsymbol{\lambda} \in \mathbb{R}^m$ as the *dual vector*. Note that the cost vector \mathbf{c} in the primal has moved to the constraints in the dual. The vector \mathbf{b} on the right-hand side of $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ becomes part of the cost in the dual. Thus, the roles of \mathbf{b} and \mathbf{c} are reversed. The form of duality defined above is called the *symmetric form of duality*.

To define the dual of an arbitrary linear programming problem, we use the following procedure. First, we convert the given linear programming problem into an equivalent problem of the primal form shown above. Then, using the symmetric form of duality, we construct the dual to the equivalent problem. We call the resulting problem the dual of the original problem.

Note that based on the definition of duality above, the dual of the dual problem is the primal problem. To see this, we first represent the dual problem in the form

$$\begin{aligned} & \text{minimize} && \boldsymbol{\lambda}^\top (-\mathbf{b}) \\ & \text{subject to} && \boldsymbol{\lambda}^\top (-\mathbf{A}) \geq -\mathbf{c}^\top \\ & && \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

Therefore, by the symmetric form of duality, the dual to the above is

$$\begin{aligned} & \text{maximize} && (-\mathbf{c}^\top) \mathbf{x} \\ & \text{subject to} && (-\mathbf{A}) \mathbf{x} \leq -\mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Upon rewriting, we get the original primal problem.

Now consider an LP problem in standard form. This form has equality constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$. To formulate the corresponding dual problem, we first convert the equality constraints into equivalent inequality constraints. Specifically, observe that $\mathbf{A}\mathbf{x} = \mathbf{b}$ is equivalent to

$$\begin{aligned} & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & -\mathbf{A}\mathbf{x} \geq -\mathbf{b}. \end{aligned}$$

Thus, the original problem with the equality constraints can be written in the form

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix} \mathbf{x} \geq \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

The LP problem above is in the form of the primal problem in the symmetric form of duality. The corresponding dual is therefore

$$\begin{aligned} & \text{maximize} && [\mathbf{u}^\top \mathbf{v}^\top] \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \\ & \text{subject to} && [\mathbf{u}^\top \mathbf{v}^\top] \begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix} \leq \mathbf{c}^\top \\ & && \mathbf{u}, \mathbf{v} \geq \mathbf{0}. \end{aligned}$$

After simple manipulation the dual above can be represented as

$$\begin{aligned} & \text{maximize} && (\mathbf{u} - \mathbf{v})^\top \mathbf{b} \\ & \text{subject to} && (\mathbf{u} - \mathbf{v})^\top \mathbf{A} \leq \mathbf{c}^\top \\ & && \mathbf{u}, \mathbf{v} \geq \mathbf{0}. \end{aligned}$$

Let $\boldsymbol{\lambda} = \mathbf{u} - \mathbf{v}$. Then, the dual problem becomes

$$\begin{aligned} & \text{maximize} && \boldsymbol{\lambda}^\top \mathbf{b} \\ & \text{subject to} && \boldsymbol{\lambda}^\top \mathbf{A} \leq \mathbf{c}^\top. \end{aligned}$$

Note that since $\boldsymbol{\lambda} = \mathbf{u} - \mathbf{v}$ and $\mathbf{u}, \mathbf{v} \geq \mathbf{0}$, the dual vector $\boldsymbol{\lambda}$ is not restricted to be nonnegative. We have now derived the dual for a primal in standard form. The form of duality above is referred to as the *asymmetric form of duality*.

We summarize the forms of duality in Tables 17.1 and 17.2. Note that in the asymmetric form of duality, the dual of the dual is also the primal. We can show this by reversing the arguments we used to arrive at the asymmetric form of duality and using the symmetric form of duality.

Recall that at the beginning of this chapter we defined the dual of an arbitrary linear programming problem by first transforming the problem into an equivalent problem of the form of the primal in the symmetric form of duality. We then derived the asymmetric form of duality based on the symmetric form. In both forms of duality the dual of the dual is the primal. Therefore, we now have four forms of primal-dual linear programming pairs: Each of the four linear programming problems in Tables 17.1 and 17.2 is a primal in these four pairs. So, given an arbitrary linear programming problem, we can obtain its

dual by converting the problem into any of the four problems in Tables 17.1 and 17.2.

Example 17.1 Suppose that we are given the given linear programming problem

$$\begin{aligned} & \text{minimize } \mathbf{c}^\top \mathbf{x} \\ & \text{subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b}. \end{aligned}$$

This problem is already close to the form of the dual in Table 17.2. In particular, let us rewrite the above as

$$\begin{aligned} & \text{maximize } \mathbf{x}^\top (-\mathbf{c}) \\ & \text{subject to } \mathbf{x}^\top \mathbf{A}^\top \leq \mathbf{b}^\top. \end{aligned}$$

Its associated dual is then given by the primal in Table 17.2, which has the form

$$\begin{aligned} & \text{minimize } \mathbf{b}^\top \boldsymbol{\lambda} \\ & \text{subject to } \mathbf{A}^\top \boldsymbol{\lambda} = -\mathbf{c} \\ & \boldsymbol{\lambda} \geq \mathbf{0}, \end{aligned}$$

which can be written in the equivalent form

$$\begin{aligned} & \text{maximize } -\boldsymbol{\lambda}^\top \mathbf{b} \\ & \text{subject to } \boldsymbol{\lambda}^\top \mathbf{A} = -\mathbf{c}^\top \\ & \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

If we change the sign of the dual variable, we can rewrite the above in a more “natural” form:

$$\begin{aligned} & \text{maximize } \boldsymbol{\lambda}^\top \mathbf{b} \\ & \text{subject to } \boldsymbol{\lambda}^\top \mathbf{A} = \mathbf{c}^\top \\ & \boldsymbol{\lambda} \leq \mathbf{0}. \end{aligned}$$

■

Example 17.2 This example is adapted from [88]. Recall the diet problem (see Example 15.2). We have n different types of food. Our goal is to create the most economical diet and at the same time meet or exceed nutritional requirements. Specifically, let a_{ij} be the amount of the i th nutrient per unit of the j th food, b_i the amount of the i th nutrient required, $1 \leq i \leq m$, c_j the

Table 17.1 Symmetric Form of Duality

Primal		Dual	
minimize	$c^\top x$	maximize	$\lambda^\top b$
subject to	$Ax \geq b$	subject to	$\lambda^\top A \leq c^\top$
	$x \geq 0$		$\lambda \geq 0$

Table 17.2 Asymmetric Form of Duality

Primal		Dual	
minimize	$c^\top x$	maximize	$\lambda^\top b$
subject to	$Ax = b$	subject to	$\lambda^\top A \leq c^\top$
	$x \geq 0$		

cost per unit of the j th food, and x_i the number of units of food i in the diet. Then, the diet problem can be stated as follows:

$$\begin{aligned} & \text{minimize} && c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ & \text{subject to} && a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq b_1 \\ & && a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq b_2 \\ & && \vdots \\ & && a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq b_m \\ & && x_1, \dots, x_n \geq 0. \end{aligned}$$

Now, consider a health food store that sells nutrient pills (all m types of nutrients are available). Let λ_i be the price of a unit of the i th nutrient in the form of nutrient pills. Suppose that we purchase nutrient pills from the health food store at this price such that we exactly meet our nutritional requirements. Then, $\lambda^\top b$ is the total revenue to the store. Note that since prices are nonnegative, we have $\lambda \geq 0$. Consider now the task of substituting nutrient pills for natural food. The cost of buying pills to create the nutritional equivalent of the i th food synthetically is simply $\lambda_1a_{1i} + \cdots + \lambda_ma_{mi}$. Because c_i is the cost per unit of the i th food, if

$$\lambda_1a_{1i} + \cdots + \lambda_ma_{mi} \leq c_i,$$

then the cost of the unit of the i th food made synthetically from nutrient pills is less than or equal to the market price of a unit of the real food. Therefore,

for the health food store to be competitive, the following must hold:

$$\lambda_1 a_{11} + \cdots + \lambda_m a_{m1} \leq c_1$$

⋮

$$\lambda_1 a_{1n} + \cdots + \lambda_m a_{mn} \leq c_n.$$

The problem facing the health food store is to choose the prices $\lambda_1, \dots, \lambda_m$ such that its revenue is maximized. This problem can be stated as

$$\begin{aligned} & \text{maximize} && \boldsymbol{\lambda}^\top \mathbf{b} \\ & \text{subject to} && \boldsymbol{\lambda}^\top \mathbf{A} \leq \mathbf{c}^\top \\ & && \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

Note that this is simply the dual of the diet problem. ■

Example 17.3 Consider the following linear programming problem:

$$\begin{aligned} & \text{maximize} && 2x_1 + 5x_2 + x_3 \\ & \text{subject to} && 2x_1 - x_2 + 7x_3 \leq 6 \\ & && x_1 + 3x_2 + 4x_3 \leq 9 \\ & && 3x_1 + 6x_2 + x_3 \leq 3 \\ & && x_1, x_2, x_3 \geq 0. \end{aligned}$$

Find the corresponding dual problem and solve it.

We first write the primal problem in standard form by introducing slack variables x_4, x_5, x_6 . This primal problem in standard form is

$$\begin{aligned} & \text{minimize} && [\mathbf{c}^\top, \mathbf{0}^\top] \mathbf{x} \\ & \text{subject to} && [\mathbf{A}, \mathbf{I}] \mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{x} = [x_1, \dots, x_6]^\top$ and

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 7 \\ 1 & 3 & 4 \\ 3 & 6 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 6 \\ 9 \\ 3 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} -2 \\ -5 \\ -1 \end{bmatrix}.$$

The corresponding dual problem (asymmetric form) is

$$\begin{aligned} & \text{maximize} && \boldsymbol{\lambda}^\top \mathbf{b} \\ & \text{subject to} && \boldsymbol{\lambda}^\top [\mathbf{A}, \mathbf{I}] \leq [\mathbf{c}^\top, \mathbf{0}^\top]. \end{aligned}$$

Note that the constraints in the dual can be written as

$$\begin{aligned}\boldsymbol{\lambda}^\top \mathbf{A} &\leq \mathbf{c}^\top \\ \boldsymbol{\lambda} &\leq \mathbf{0}.\end{aligned}$$

To solve the dual problem above, we use the simplex method. For this, we need to express the problem in standard form. We substitute $\boldsymbol{\lambda}$ by $-\boldsymbol{\lambda}$ and introduce surplus variables to get

$$\begin{aligned}\text{minimize} \quad & 6\lambda_1 + 9\lambda_2 + 3\lambda_3 \\ \text{subject to} \quad & 2\lambda_1 + \lambda_2 + 3\lambda_3 - \lambda_4 = 2 \\ & -\lambda_1 + 3\lambda_2 + 6\lambda_3 - \lambda_5 = 5 \\ & 7\lambda_1 + 4\lambda_2 + \lambda_3 - \lambda_6 = 1 \\ & \lambda_1, \dots, \lambda_6 \geq 0.\end{aligned}$$

There is no obvious basic feasible solution. Thus, we use the two-phase simplex method to solve the problem.

Phase I. We introduce artificial variables $\lambda_7, \lambda_8, \lambda_9$ and the artificial objective function $\lambda_7 + \lambda_8 + \lambda_9$. The tableau for the artificial problem is

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	\mathbf{c}
	2	1	3	-1	0	0	1	0	0	2
	-1	3	6	0	-1	0	0	1	0	5
	7	4	1	0	0	-1	0	0	1	1
Cost	0	0	0	0	0	0	1	1	1	0

We start with an initial feasible solution and corresponding \mathbf{B}^{-1} :

Variable	\mathbf{B}^{-1}			\mathbf{y}_0
λ_7	1	0	0	2
λ_8	0	1	0	5
λ_9	0	0	1	1

We compute

$$\begin{aligned}\mathbf{r}_D^\top &= [0, 0, 0, 0, 0, 0] - [8, 8, 10, -1, -1, -1] = [-8, -8, -10, 1, 1, 1] \\ &= [r_1, r_2, r_3, r_4, r_5, r_6].\end{aligned}$$

Because r_3 is the most negative reduced cost coefficient, we bring the third column into the basis. In this case, $\mathbf{y}_3 = [3, 6, 1]^\top$. We have

Variable	\mathbf{B}^{-1}			\mathbf{y}_0	\mathbf{y}_3
λ_7	1	0	0	2	3
λ_8	0	1	0	5	6
λ_9	0	0	1	1	1

By inspection, $p = 1$, so we pivot about the first element of the last column. The updated tableau is

Variable	\mathbf{B}^{-1}			\mathbf{y}_0
λ_3	$\frac{1}{3}$	0	0	$\frac{2}{3}$
λ_8	-2	1	0	1
λ_9	$-\frac{1}{3}$	0	1	$\frac{1}{3}$

We compute

$$\mathbf{r}_D^\top = \left[-\frac{4}{3}, -\frac{14}{3}, -\frac{7}{3}, 1, 1, \frac{10}{3} \right] = [r_1, r_2, r_4, r_5, r_6, r_7].$$

We bring the second column into the basis to get

Variable	\mathbf{B}^{-1}			\mathbf{y}_0	\mathbf{y}_2
λ_3	$\frac{1}{3}$	0	0	$\frac{2}{3}$	$\frac{1}{3}$
λ_8	-2	1	0	1	1
λ_9	$-\frac{1}{3}$	0	1	$\frac{1}{3}$	$\frac{11}{3}$

We update the tableau to get

Variable	\mathbf{B}^{-1}			\mathbf{y}_0
λ_3	$\frac{4}{11}$	0	$-\frac{1}{11}$	$\frac{7}{11}$
λ_8	$-\frac{21}{11}$	1	$-\frac{3}{11}$	$\frac{10}{11}$
λ_2	$-\frac{1}{11}$	0	$\frac{3}{11}$	$\frac{1}{11}$

We compute

$$\mathbf{r}_D^\top = \left[\frac{74}{11}, -\frac{21}{11}, 1, -\frac{3}{11}, \frac{32}{11}, \frac{14}{11} \right] = [r_1, r_4, r_5, r_6, r_7, r_9].$$

We bring the fourth column into the basis:

Variable	\mathbf{B}^{-1}			\mathbf{y}_0	\mathbf{y}_4
λ_3	$\frac{4}{11}$	0	$-\frac{1}{11}$	$\frac{7}{11}$	$-\frac{4}{11}$
λ_8	$-\frac{21}{11}$	1	$-\frac{3}{11}$	$\frac{10}{11}$	$\frac{21}{11}$
λ_2	$-\frac{1}{11}$	0	$\frac{3}{11}$	$\frac{1}{11}$	$\frac{1}{11}$

The updated tableau becomes

Variable	\mathbf{B}^{-1}			\mathbf{y}_0
λ_3	0	$\frac{4}{21}$	$-\frac{3}{21}$	$\frac{17}{21}$
λ_4	-1	$\frac{11}{21}$	$-\frac{3}{21}$	$\frac{10}{21}$
λ_2	0	$-\frac{1}{21}$	$\frac{6}{21}$	$\frac{1}{21}$

We compute

$$\mathbf{r}_D^\top = [0, 0, 0, 1, 1, 1] = [r_1, r_5, r_6, r_7, r_8, r_9].$$

Because all the reduced cost coefficients are nonnegative, we terminate phase I.

Phase II. We use the last tableau in phase I (where none of the artificial variables are basic) as the initial tableau in phase II. Note that we now revert back to the original cost of the dual problem in standard form. We compute

$$\mathbf{r}_D^\top = \left[-\frac{62}{7}, \frac{1}{7}, \frac{15}{7} \right] = [r_1, r_5, r_6].$$

We bring the first column into the basis to obtain the augmented revised tableau

Variable		\mathbf{B}^{-1}	\mathbf{y}_0	\mathbf{y}_1
λ_3	0	$\frac{4}{21}$	$-\frac{3}{21}$	$\frac{17}{21}$
λ_4	-1	$\frac{11}{21}$	$-\frac{3}{21}$	$\frac{10}{21}$
λ_2	0	$-\frac{1}{21}$	$\frac{6}{21}$	$\frac{1}{21}$

We update the tableau to get

Variable		\mathbf{B}^{-1}	\mathbf{y}_0
λ_3	0	$\frac{7}{43}$	$\frac{1}{43}$
λ_4	-1	$\frac{19}{43}$	$\frac{15}{43}$
λ_1	0	$-\frac{1}{43}$	$\frac{6}{43}$

We compute

$$\mathbf{r}_D^\top = \left[\frac{186}{43}, \frac{15}{43}, \frac{39}{43} \right] = [r_2, r_5, r_6].$$

Because all the reduced cost coefficients are nonnegative, the current basic feasible solution is optimal for the dual in standard form. Thus, an optimal solution to the original dual problem is

$$\boldsymbol{\lambda} = \left[-\frac{1}{43}, 0, -\frac{36}{43} \right]^\top.$$

■

17.2 Properties of Dual Problems

In this section we present some basic results on dual linear programs. We begin with the weak duality lemma.

Lemma 17.1 Weak Duality Lemma. Suppose that \mathbf{x} and $\boldsymbol{\lambda}$ are feasible solutions to primal and dual LP problems, respectively (either in the symmetric or asymmetric form). Then, $\mathbf{c}^\top \mathbf{x} \geq \boldsymbol{\lambda}^\top \mathbf{b}$. □

Proof. We prove this lemma only for the asymmetric form of duality. The proof for the symmetric form involves only a slight modification (see Exercise 17.1).

Because \mathbf{x} and $\boldsymbol{\lambda}$ are feasible, we have $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, and $\boldsymbol{\lambda}^\top \mathbf{A} \leq \mathbf{c}^\top$. Postmultiplying both sides of the inequality $\boldsymbol{\lambda}^\top \mathbf{A} \leq \mathbf{c}^\top$ by $\mathbf{x} \geq \mathbf{0}$ yields $\boldsymbol{\lambda}^\top \mathbf{A}\mathbf{x} \leq \mathbf{c}^\top \mathbf{x}$. But $\mathbf{A}\mathbf{x} = \mathbf{b}$, hence $\boldsymbol{\lambda}^\top \mathbf{b} \leq \mathbf{c}^\top \mathbf{x}$. ■

The weak duality lemma states that a feasible solution to either problem yields a bound on the optimal cost of the other problem. The cost in the dual is never above the cost in the primal. In particular, the optimal cost of the dual is less than or equal to the optimal cost of the primal, that is, “maximum \leq minimum.” Hence, if the cost of one of the problems is unbounded, then the other problem has no feasible solution. In other words, if “minimum = $-\infty$ ” or “maximum = $+\infty$,” then the feasible set in the other problem must be empty.

Example 17.4 Consider the problem

$$\begin{aligned} & \text{minimize } x \\ & \text{subject to } x \leq 1, \end{aligned}$$

which is clearly unbounded. By Example 17.1, the dual is

$$\begin{aligned} & \text{maximize } \lambda \\ & \text{subject to } \lambda = 1 \\ & \quad \lambda \leq 0, \end{aligned}$$

which is clearly infeasible. ■

It follows from the weak duality lemma that if we are given feasible primal and dual solutions with equal cost, then these solutions must be optimal in their respective problems.

Theorem 17.1 Suppose that \mathbf{x}_0 and $\boldsymbol{\lambda}_0$ are feasible solutions to the primal and dual, respectively (either in symmetric or asymmetric form). If $\mathbf{c}^\top \mathbf{x}_0 = \boldsymbol{\lambda}_0^\top \mathbf{b}$, then \mathbf{x}_0 and $\boldsymbol{\lambda}_0$ are optimal solutions to their respective problems. □

Proof. Let \mathbf{x} be an arbitrary feasible solution to the primal problem. Because $\boldsymbol{\lambda}_0$ is a feasible solution to the dual, by the weak duality lemma, $\mathbf{c}^\top \mathbf{x} \geq \boldsymbol{\lambda}_0^\top \mathbf{b}$. So, if $\mathbf{c}^\top \mathbf{x}_0 = \boldsymbol{\lambda}_0^\top \mathbf{b}$, then $\mathbf{c}^\top \mathbf{x}_0 = \boldsymbol{\lambda}_0^\top \mathbf{b} \leq \mathbf{c}^\top \mathbf{x}$. Hence, \mathbf{x}_0 is optimal for the primal.

On the other hand, let $\boldsymbol{\lambda}$ be an arbitrary feasible solution to the dual problem. Because \mathbf{x}_0 is a feasible solution to the primal, by the weak duality lemma, $\mathbf{c}^\top \mathbf{x}_0 \geq \boldsymbol{\lambda}^\top \mathbf{b}$. Therefore, if $\mathbf{c}^\top \mathbf{x}_0 = \boldsymbol{\lambda}_0^\top \mathbf{b}$, then $\boldsymbol{\lambda}^\top \mathbf{b} \leq \mathbf{c}^\top \mathbf{x}_0 = \boldsymbol{\lambda}_0^\top \mathbf{b}$. Hence, $\boldsymbol{\lambda}_0$ is optimal for the dual. ■

We can interpret Theorem 17.1 as follows. The primal seeks to minimize its cost, and the dual seeks to maximize its cost. Because the weak duality

lemma states that “maximum \leq minimum,” each problem “seeks to reach the other.” When their costs are equal for a pair of feasible solutions, both solutions are optimal, and we have “maximum = minimum.”

It turns out that the converse of Theorem 17.1 is also true, that is, “maximum = minimum” always holds. In fact, we can prove an even stronger result, known as the duality theorem.

Theorem 17.2 Duality Theorem. *If the primal problem (either in symmetric or asymmetric form) has an optimal solution, then so does the dual, and the optimal values of their respective objective functions are equal.* \square

Proof. We first prove the result for the asymmetric form of duality. Assume that the primal has an optimal solution. Then, by the fundamental theorem of LP, there exists an optimal basic feasible solution. As is our usual notation, let \mathbf{B} be the matrix of the corresponding m basic columns, \mathbf{D} the matrix of the $n-m$ nonbasic columns, \mathbf{c}_B the vector of elements of \mathbf{c} corresponding to basic variables, \mathbf{c}_D the vector of elements of \mathbf{c} corresponding to nonbasic variables, and \mathbf{r}_D the vector of reduced cost coefficients. Then, by Theorem 16.2,

$$\mathbf{r}_D^\top = \mathbf{c}_D^\top - \mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{D} \geq \mathbf{0}^\top.$$

Hence,

$$\mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{D} \leq \mathbf{c}_D^\top.$$

Define

$$\boldsymbol{\lambda}^\top = \mathbf{c}_B^\top \mathbf{B}^{-1}.$$

Then,

$$\mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{D} = \boldsymbol{\lambda}^\top \mathbf{D} \leq \mathbf{c}_D^\top.$$

We claim that $\boldsymbol{\lambda}$ is a feasible solution to the dual. To see this, assume for convenience (and without loss of generality) that the basic columns are the first m columns of \mathbf{A} . Then,

$$\boldsymbol{\lambda}^\top \mathbf{A} = \boldsymbol{\lambda}^\top [\mathbf{B}, \mathbf{D}] = [\mathbf{c}_B^\top, \boldsymbol{\lambda}^\top \mathbf{D}] \leq [\mathbf{c}_B^\top, \mathbf{c}_D^\top] = \mathbf{c}^\top.$$

Hence, $\boldsymbol{\lambda}^\top \mathbf{A} \leq \mathbf{c}^\top$ and thus $\boldsymbol{\lambda}^\top = \mathbf{c}_B^\top \mathbf{B}^{-1}$ is feasible.

We claim that $\boldsymbol{\lambda}$ is also an optimal feasible solution to the dual. To see this, note that

$$\boldsymbol{\lambda}^\top \mathbf{b} = \mathbf{c}_B^\top \mathbf{B}^{-1} \mathbf{b} = \mathbf{c}_B^\top \mathbf{x}_B.$$

Thus, by Theorem 17.1, $\boldsymbol{\lambda}$ is optimal.

We now prove the symmetric case. First, we convert the primal problem for the symmetric form into the equivalent standard form by adding surplus

variables:

$$\begin{aligned} \text{minimize} \quad & [\mathbf{c}^\top, \mathbf{0}^\top] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \\ \text{subject to} \quad & [\mathbf{A}, -\mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \mathbf{b} \\ & \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \geq \mathbf{0}. \end{aligned}$$

Note that \mathbf{x} is optimal for the original primal problem if and only if $[\mathbf{x}^\top, (\mathbf{Ax} - \mathbf{b})^\top]^\top$ is optimal for the primal in standard form. The dual to the primal in standard form is equivalent to the dual to the original primal in symmetric form. Therefore, the result above for the asymmetric case applies also to the symmetric case.

This completes the proof. ■

Example 17.5 Recall Example 17.2, where we formulated the dual of the diet problem. From the duality theorem, the maximum revenue for the health food store is the *same* as the minimum cost of a diet that satisfies all of the nutritional requirements; that is, $\mathbf{c}^\top \mathbf{x} = \boldsymbol{\lambda}^\top \mathbf{b}$. ■

Consider a primal-dual pair in asymmetric form. Suppose that we solve the primal problem using the simplex method. The proof of the duality theorem suggests a way of obtaining an optimal solution to the dual by using the last row of the final simplex tableau for the primal. First, we write the tableau for the primal problem:

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{c}^\top & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{B} & \mathbf{D} & \mathbf{b} \\ \mathbf{c}_B^\top & \mathbf{c}_D^\top & 0 \end{bmatrix}.$$

Suppose that the matrix \mathbf{B} is the basis for an optimal basic feasible solution. Then, the final simplex tableau is

$$\begin{bmatrix} \mathbf{I} & \mathbf{B}^{-1}\mathbf{D} & \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0}^\top & \mathbf{r}_D^\top & -\mathbf{c}_B^\top \mathbf{B}^{-1}\mathbf{b} \end{bmatrix},$$

where $\mathbf{r}_D^\top = \mathbf{c}_D^\top - \mathbf{c}_B^\top \mathbf{B}^{-1}\mathbf{D}$. In the proof of the duality theorem we have shown that $\boldsymbol{\lambda}^\top = \mathbf{c}_B^\top \mathbf{B}^{-1}$ is an optimal solution to the dual. The vector $\boldsymbol{\lambda}$ can be obtained from the final tableau above. Specifically, if $\text{rank } \mathbf{D} = m$, then we can solve for $\boldsymbol{\lambda}$ using the vector \mathbf{r}_D , via the equation

$$\boldsymbol{\lambda}^\top \mathbf{D} = \mathbf{c}_D^\top - \mathbf{r}_D^\top.$$

Of course, it may turn out that $\text{rank } \mathbf{D} < m$. In this case as we now show, we have additional linear equations that allow us to solve for $\boldsymbol{\lambda}$. To this end, recall that $\boldsymbol{\lambda}^\top \mathbf{B} = \mathbf{c}_B^\top$. Therefore, if we define $\mathbf{r}^\top = [\mathbf{0}^\top, \mathbf{r}_D^\top]$, then combining the equations $\boldsymbol{\lambda}^\top \mathbf{D} = \mathbf{c}_D^\top - \mathbf{r}_D^\top$ and $\boldsymbol{\lambda}^\top \mathbf{B} = \mathbf{c}_B^\top$ yields

$$\boldsymbol{\lambda}^\top \mathbf{A} = \mathbf{c}^\top - \mathbf{r}^\top.$$

The vector $\boldsymbol{\lambda}$ may be easy to obtain from the equation $\boldsymbol{\lambda}^\top \mathbf{D} = \mathbf{c}_D^\top - \mathbf{r}_D^\top$ if \mathbf{D} takes certain special forms. In particular, this is the case if \mathbf{D} has an $m \times m$ identity matrix embedded in it; that is, by rearranging the positions of the columns of \mathbf{D} , if necessary, \mathbf{D} has the form $\mathbf{D} = [\mathbf{I}_m, \mathbf{G}]$, where \mathbf{G} is an $m \times (n - 2m)$ matrix. In this case we can write the equation $\boldsymbol{\lambda}^\top \mathbf{D} = \mathbf{c}_D^\top - \mathbf{r}_D^\top$ as

$$[\boldsymbol{\lambda}^\top, \boldsymbol{\lambda}^\top \mathbf{G}] = [\mathbf{c}_I^\top, \mathbf{c}_G^\top] - [\mathbf{r}_I^\top, \mathbf{r}_G^\top].$$

Hence, $\boldsymbol{\lambda}$ is given by

$$\boldsymbol{\lambda}^\top = \mathbf{c}_I^\top - \mathbf{r}_I^\top.$$

In other words, the solution to the dual is obtained by subtracting the reduced costs coefficients corresponding to the identity matrix in \mathbf{D} from the corresponding elements in the vector \mathbf{c} (i.e., \mathbf{c}_I).

For example, if we have a problem where we introduced slack variables, and the basic variables for the optimal basic feasible solution do not include any of the slack variables, then the matrix \mathbf{D} has an identity matrix embedded in it. In addition, in this case we have $\mathbf{c}_I = \mathbf{0}$. Therefore, $\boldsymbol{\lambda} = -\mathbf{r}_I$ is an optimal solution to the dual.

Example 17.6 In Example 17.3, the tableau for the primal in standard form is

a_1	a_2	a_3	a_4	a_5	a_6	b
2	-1	7	1	0	0	6
1	3	4	0	1	0	9
3	6	1	0	0	1	3
\mathbf{c}^\top	-2	-5	-1	0	0	0

If we now solve the problem using the simplex method, we get the following final simplex tableau:

$\frac{15}{43}$	0	1	$\frac{6}{43}$	0	$\frac{1}{43}$	$\frac{39}{43}$
$-\frac{74}{43}$	0	0	$-\frac{21}{43}$	1	$-\frac{25}{43}$	$\frac{186}{43}$
$\frac{19}{43}$	1	0	$-\frac{1}{43}$	0	$\frac{7}{43}$	$\frac{15}{43}$
\mathbf{r}^\top	$\frac{24}{43}$	0	0	$\frac{1}{43}$	0	$\frac{36}{43}$
						$\frac{114}{43}$

We can now find the solution of the dual from the above simplex tableau using the equation $\boldsymbol{\lambda}^\top \mathbf{D} = \mathbf{c}_D^\top - \mathbf{r}_D^\top$:

$$[\lambda_1, \lambda_2, \lambda_3] \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 3 & 0 & 1 \end{bmatrix} = [-2, 0, 0] - \left[\frac{24}{43}, \frac{1}{43}, \frac{36}{43} \right].$$

Solving the above, we get

$$\boldsymbol{\lambda}^\top = \left[-\frac{1}{43}, 0, -\frac{36}{43} \right],$$

which agrees with our solution in Example 17.3. ■

We now summarize our discussion relating the solutions of the primal and dual problems. If one has unbounded objective function values, then the other has no feasible solution. If one has an optimal feasible solution, then so does the other (and their objective function values are equal). One final case remains: What can we say if one (the primal, say) has no feasible solution? In this case clearly the other (the dual, say) cannot have an optimal solution. However, is it necessarily the case that the dual is unbounded? The answer is no: If one of the problems has no feasible solution, then the other may or may not have a feasible solution. The following example shows that there exists a primal-dual pair of problems for which both have no feasible solution.

Example 17.7 Consider the primal problem

$$\begin{aligned} & \text{minimize} && [1, -2]\mathbf{x} \\ & \text{subject to} && \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \mathbf{x} \geq \begin{bmatrix} 2 \\ -1 \end{bmatrix} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

The problem has no feasible solution, because the constraints require that $x_1 - x_2 \geq 2$ and $x_1 - x_2 \leq 1$. Based on symmetric duality, the dual is

$$\begin{aligned} & \text{maximize} && \boldsymbol{\lambda}^\top \begin{bmatrix} 2 \\ -1 \end{bmatrix} \\ & \text{subject to} && \boldsymbol{\lambda}^\top \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \leq [1, -2] \\ & && \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

The dual also has no feasible solution, because the constraints require that $\lambda_1 - \lambda_2 \leq 1$ and $\lambda_1 - \lambda_2 \geq 2$. ■

We end this chapter by presenting the following theorem, which describes an alternative form of the relationship between the optimal solutions to the primal and dual problems.

Theorem 17.3 Complementary Slackness Condition. *The feasible solutions \mathbf{x} and $\boldsymbol{\lambda}$ to a dual pair of problems (either in symmetric or asymmetric form) are optimal if and only if:*

$$1. (\mathbf{c}^\top - \boldsymbol{\lambda}^\top \mathbf{A})\mathbf{x} = 0.$$

$$2. \boldsymbol{\lambda}^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) = 0.$$

□

Proof. We first prove the result for the asymmetric case. Note that condition 2 holds trivially for this case. Therefore, we consider only condition 1.

⇒: If the two solutions are optimal, then by Theorem 17.2, $\mathbf{c}^\top \mathbf{x} = \boldsymbol{\lambda}^\top \mathbf{b}$. Because $\mathbf{A}\mathbf{x} = \mathbf{b}$, we also have $(\mathbf{c}^\top - \boldsymbol{\lambda}^\top \mathbf{A})\mathbf{x} = 0$.

⇐: If $(\mathbf{c}^\top - \boldsymbol{\lambda}^\top \mathbf{A})\mathbf{x} = 0$, then $\mathbf{c}^\top \mathbf{x} = \boldsymbol{\lambda}^\top \mathbf{A}\mathbf{x} = \boldsymbol{\lambda}^\top \mathbf{b}$. Therefore, by Theorem 17.1, \mathbf{x} and $\boldsymbol{\lambda}$ are optimal.

We now prove the result for the symmetric case.

⇒: We first show condition 1. If the two solutions are optimal, then by Theorem 17.2, $\mathbf{c}^\top \mathbf{x} = \boldsymbol{\lambda}^\top \mathbf{b}$. Because $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ and $\boldsymbol{\lambda} \geq \mathbf{0}$, we have

$$(\mathbf{c}^\top - \boldsymbol{\lambda}^\top \mathbf{A})\mathbf{x} = \mathbf{c}^\top \mathbf{x} - \boldsymbol{\lambda}^\top \mathbf{A}\mathbf{x} = \boldsymbol{\lambda}^\top \mathbf{b} - \boldsymbol{\lambda}^\top \mathbf{A}\mathbf{x} = \boldsymbol{\lambda}^\top (\mathbf{b} - \mathbf{A}\mathbf{x}) \leq 0.$$

On the other hand, since $\boldsymbol{\lambda}^\top \mathbf{A} \leq \mathbf{c}^\top$ and $\mathbf{x} \geq \mathbf{0}$, we have $(\mathbf{c}^\top - \boldsymbol{\lambda}^\top \mathbf{A})\mathbf{x} \geq 0$. Hence, $(\mathbf{c}^\top - \boldsymbol{\lambda}^\top \mathbf{A})\mathbf{x} = 0$. To show condition 2, note that since $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ and $\boldsymbol{\lambda} \geq \mathbf{0}$, we have $\boldsymbol{\lambda}^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) \geq 0$. On the other hand, since $\boldsymbol{\lambda}^\top \mathbf{A} \leq \mathbf{c}^\top$ and $\mathbf{x} \geq \mathbf{0}$, we have $\boldsymbol{\lambda}^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) = (\boldsymbol{\lambda}^\top \mathbf{A} - \mathbf{c}^\top)\mathbf{x} \leq 0$.

⇐: Combining conditions 1 and 2, we get $\mathbf{c}^\top \mathbf{x} = \boldsymbol{\lambda}^\top \mathbf{A}\mathbf{x} = \boldsymbol{\lambda}^\top \mathbf{b}$. Hence, by Theorem 17.1, \mathbf{x} and $\boldsymbol{\lambda}$ are optimal. ■

Note that if \mathbf{x} and $\boldsymbol{\lambda}$ are feasible solutions for the dual pair of problems, we can write condition 1, that is, $(\mathbf{c}^\top - \boldsymbol{\lambda}^\top \mathbf{A})\mathbf{x} = 0$, as “ $x_i > 0$ implies that $\boldsymbol{\lambda}^\top \mathbf{a}_i = c_i$, $i = 1, \dots, n$,” that is, for any component of \mathbf{x} that is positive, the corresponding constraint for the dual must be an equality at $\boldsymbol{\lambda}$. Also, observe that the statement “ $x_i > 0$ implies that $\boldsymbol{\lambda}^\top \mathbf{a}_i = c_i$ ” is equivalent to “ $\boldsymbol{\lambda}^\top \mathbf{a}_i < c_i$ implies that $x_i = 0$.” A similar representation can be written for condition 2.

Consider the asymmetric form of duality. Recall that for the case of an optimal basic feasible solution \mathbf{x} , $\mathbf{r}^\top = \mathbf{c}^\top - \boldsymbol{\lambda}^\top \mathbf{A}$ is the vector of reduced cost coefficients. Therefore, in this case, the complementary slackness condition can be written as $\mathbf{r}^\top \mathbf{x} = 0$.

Example 17.8 Suppose that you have 26 dollars and you wish to purchase some gold. You have a choice of four vendors, with prices (in dollars per ounce) of $1/2$, 1 , $1/7$, and $1/4$, respectively. You wish to spend your entire 26 dollars by purchasing gold from these four vendors, where x_i is the dollars you spend on vendor i , $i = 1, 2, 3, 4$.

- a. Formulate the linear programming problem (in standard form) that reflects your desire to obtain the maximum weight in gold.
- b. Write down the dual of the linear programming problem in part a, and find the solution to the dual.

- c. Use the complementary slackness condition together with part b to find the optimal values of x_1, \dots, x_4 .

Solution:

- a. The corresponding linear programming problem is

$$\begin{aligned} & \text{minimize} && -(2x_1 + x_2 + 7x_3 + 4x_4) \\ & \text{subject to} && x_1 + x_2 + x_3 + x_4 = 26 \\ & && x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

- b. The dual problem is

$$\begin{aligned} & \text{maximize} && 26\lambda \\ & \text{subject to} && \lambda \leq -2 \\ & && \lambda \leq -1 \\ & && \lambda \leq -7 \\ & && \lambda \leq -4. \end{aligned}$$

The solution is clearly $\lambda = -7$. (*Note:* It is equally valid to have a dual problem with variable $\lambda' = -\lambda$.)

- c. By the complementary slackness condition, we know that if we can find a vector \mathbf{x} that is feasible in the primal and satisfies $(-[2, 1, 7, 4] - (-7)[1, 1, 1, 1])\mathbf{x} = 0$, then this \mathbf{x} is optimal in the primal (original) problem. We can rewrite the conditions above as

$$[1, 1, 1, 1]\mathbf{x} = 26, \quad \mathbf{x} \geq \mathbf{0}, \quad [5, 6, 0, 3]\mathbf{x} = 0.$$

By $\mathbf{x} \geq \mathbf{0}$ and $[5, 6, 0, 3]\mathbf{x} = 0$, we conclude that $x_1 = x_2 = x_4 = 0$, and by $[1, 1, 1, 1]\mathbf{x} = 26$ we then conclude that $\mathbf{x} = [0, 0, 26, 0]^T$. ■

EXERCISES

17.1 Prove the weak duality lemma for the symmetric form of duality.

17.2 Find the dual of the optimization problem in Exercise 15.8.

17.3 Consider the following linear program:

$$\begin{aligned} & \text{maximize} && 2x_1 + 3x_2 \\ & \text{subject to} && x_1 + 2x_2 \leq 4 \\ & && 2x_1 + x_2 \leq 5 \\ & && x_1, x_2 \geq 0. \end{aligned}$$

- a. Use the simplex method to solve the problem.
- b. Write down the dual of the linear program and solve the dual.

17.4 Consider the linear program

$$\begin{aligned} & \text{minimize} && 4x_1 + 3x_2 \\ & \text{subject to} && 5x_1 + x_2 \geq 11 \\ & && 2x_1 + x_2 \geq 8 \\ & && x_1 + 2x_2 \geq 7 \\ & && x_1, x_2 \geq 0. \end{aligned}$$

Write down the corresponding dual problem and find the solution to the dual.
(Compare this problem with the one in Exercise 16.12, part a.)

17.5 Consider the following primal problem:

$$\begin{aligned} & \text{maximize} && x_1 + 2x_2 \\ & \text{subject to} && -2x_1 + x_2 + x_3 = 2 \\ & && -x_1 + 2x_2 + x_4 = 7 \\ & && x_1 + x_5 = 3 \\ & && x_i \geq 0, \quad i = 1, 2, 3, 4, 5. \end{aligned}$$

- a. Construct the dual problem corresponding to the primal problem above.
- b. It is known that the solution to the primal above is $\mathbf{x}^* = [3, 5, 3, 0, 0]^\top$. Find the solution to the dual.

17.6 Consider the linear programming problem

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b}. \end{aligned}$$

- a. Find the dual to this problem.
- b. Suppose that $\mathbf{b} = \mathbf{0}$ and there exists a vector $\mathbf{y} \geq \mathbf{0}$ such that $\mathbf{y}^\top \mathbf{A} + \mathbf{c}^\top = \mathbf{0}^\top$. Does this problem have an optimal feasible solution? If yes, find it. If no, explain why not. Give complete explanations.

17.7 Convert the following optimization problem into a linear programming problem and solve it:

$$\begin{aligned} & \text{maximize} && -|x_1| - |x_2| - |x_3| \\ & \text{subject to} && \begin{bmatrix} 1 & 1 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}. \end{aligned}$$

Then construct its dual program and solve it.

Hint: Introduce two sets of nonnegative variables: $x_i^+ \geq 0$, $x_i^- \geq 0$. Then represent the optimization problem using the variables above. Note that only one x_i^+ and x_i^- can be nonzero at a time. If $x_i \geq 0$ then $x_i^+ = x_i$ and $x_i^- = 0$. On the other hand, if $x_i < 0$ then $x_i^+ = 0$ and $x_i = -x_i^-$. See Exercise 16.6.

17.8 Consider the linear program

$$\begin{aligned} &\text{minimize} && x_1 + \cdots + x_n, \quad x_1, \dots, x_n \in \mathbb{R} \\ &\text{subject to} && a_1 x_1 + \cdots + a_n x_n = 1 \\ & && x_1, \dots, x_n \geq 0, \end{aligned}$$

where $0 < a_1 < a_2 < \cdots < a_n$.

- a. Write down the dual to the problem and find a solution to the dual in terms of a_1, \dots, a_n .
- b. State the duality theorem and use it to find a solution to the primal problem above.
- c. Suppose that we apply the simplex algorithm to the primal problem. Show that if we start at a nonoptimal initial basic feasible solution, the algorithm terminates in one step if and only if we use the rule where the next nonbasic column to enter the basis is the one with the most negative reduced cost coefficient.

17.9 You are given the following linear programming problem:

$$\begin{aligned} &\text{maximize} && c_1 x_1 + \cdots + c_n x_n \\ &\text{subject to} && x_1 + \cdots + x_n = 1 \\ & && x_1, \dots, x_n \geq 0, \end{aligned}$$

where $c_1, \dots, c_n \in \mathbb{R}$ are constants.

- a. Write down the dual linear program for the primal problem.
- b. Suppose you know that $c_4 > c_i$ for all $i \neq 4$. Use this information to solve the dual.
- c. Use part b to solve the linear programming problem.

17.10 Consider the linear programming problem

$$\begin{aligned} &\text{maximize} && \mathbf{c}^\top \mathbf{x} \\ &\text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{0} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{c} = [1, 1, \dots, 1]^\top$. Assume that the problem has a solution.

- a. Write down the dual of this problem.
- b. Find the solution to the problem.
- c. What can you say about the constraint set for the problem?

17.11 Consider a given linear programming problem in standard form (written in the usual notation).

- a. Write down the associated artificial problem for the problem (used in the two-phase method).
- b. Write down the dual to the artificial problem from part a.
- c. Prove that if the original linear programming problem has a feasible solution, then the dual problem in part b has an optimal feasible solution.

17.12 Consider a pair of primal and dual linear programming problems (either in symmetric or asymmetric form). Identify which of the following situations are possible (depending on the particular primal-dual pair) and which are impossible (regardless of the primal-dual pair). In each case, justify your answer (citing results such as the weak duality lemma and the duality theorem whenever needed).

- a. The primal has a feasible solution, and the dual has no feasible solution.
- b. The primal has an optimal feasible solution, and the dual has no optimal feasible solution.
- c. The primal has a feasible solution but no optimal feasible solution, and the dual has an optimal feasible solution.

17.13 Consider an LP problem in standard form. Suppose that \mathbf{x} is a feasible solution to the problem. Show that if there exist $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ such that

$$\begin{aligned}\mathbf{A}^\top \boldsymbol{\lambda} + \boldsymbol{\mu} &= \mathbf{c} \\ \boldsymbol{\mu}^\top \mathbf{x} &= 0 \\ \boldsymbol{\mu} &\geq 0,\end{aligned}$$

then \mathbf{x} is an optimal feasible solution to the problem and $\boldsymbol{\lambda}$ is an optimal feasible solution to the dual. The conditions above, called the *Karush-Kuhn-Tucker optimality conditions for LP*, are discussed in detail in Chapters 21 and 22.

17.14 Consider the linear program

$$\begin{aligned} & \text{maximize } \mathbf{c}^\top \mathbf{x}, \\ & \text{subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b}, \end{aligned}$$

where $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{m \times n}$. Use the symmetric form of duality to derive the dual of this linear program and show that the constraint in the dual involving \mathbf{A} can be written as an equality constraint.

Hint: Write $\mathbf{x} = \mathbf{u} - \mathbf{v}$, with $\mathbf{u}, \mathbf{v} \geq \mathbf{0}$.

17.15 Consider the linear program

$$\begin{aligned} & \text{minimize } x_1 + x_2 \\ & \text{subject to } x_1 + 2x_2 \geq 3 \\ & \quad 2x_1 + x_2 \geq 3 \\ & \quad x_1, x_2 \geq 0. \end{aligned}$$

The solution to the problem is $[1, 1]^\top$ (see Exercise 16.11). Write down the dual to the problem, solve the dual, and verify that the duality theorem holds.

17.16 Consider the problem

$$\begin{aligned} & \text{minimize } \mathbf{c}^\top \mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n \\ & \text{subject to } \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

For this problem we have the following theorem.

Theorem: A solution to the foregoing problem exists if and only if $\mathbf{c} \geq \mathbf{0}$. Moreover, if a solution exists, $\mathbf{0}$ is a solution.

Use the duality theorem to prove this theorem (see also Exercise 22.15).

17.17 Let \mathbf{A} be a given matrix and \mathbf{b} a given vector. Show that there exists a vector \mathbf{x} such that $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$ if and only if for any vector \mathbf{y} satisfying $\mathbf{A}^\top \mathbf{y} \leq \mathbf{0}$ and $\mathbf{y} \geq \mathbf{0}$, we have $\mathbf{b}^\top \mathbf{y} \leq 0$.

17.18 Let \mathbf{A} be a given matrix and \mathbf{b} a given vector. We wish to prove the following result: There exists a vector \mathbf{x} such that $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$ if and only if for any given vector \mathbf{y} satisfying $\mathbf{A}^\top \mathbf{y} \leq \mathbf{0}$, we have $\mathbf{b}^\top \mathbf{y} \leq 0$. This result is known as *Farkas's transposition theorem*. Our argument is based on duality theory, consisting of the following parts.

a. Consider the primal linear program

$$\begin{aligned} & \text{minimize } \mathbf{0}^\top \mathbf{x} \\ & \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \quad \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

Write down the dual of this problem using the notation \mathbf{y} for the dual variable.

- b. Show that the feasible set of the dual problem is guaranteed to be nonempty.

Hint: Think about an obvious feasible point.

- c. Suppose that for any \mathbf{y} satisfying $\mathbf{A}^\top \mathbf{y} \leq \mathbf{0}$, we have $\mathbf{b}^\top \mathbf{y} \leq 0$. In this case what can you say about whether or not the dual has an optimal feasible solution?

Hint: Think about the obvious feasible point in part b.

- d. Suppose that for any \mathbf{y} satisfying $\mathbf{A}^\top \mathbf{y} \leq \mathbf{0}$, we have $\mathbf{b}^\top \mathbf{y} \leq 0$. Use parts b and c to show that there exists \mathbf{x} such that $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$. (This proves one direction of Farkas's transposition theorem.)

- e. Suppose that \mathbf{x} satisfies $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$. Let \mathbf{y} be an arbitrary vector satisfying $\mathbf{A}^\top \mathbf{y} \leq \mathbf{0}$. Show that $\mathbf{b}^\top \mathbf{y} \leq 0$. (This proves the other direction of Farkas's transposition theorem.)

17.19 Let \mathbf{A} be a given matrix and \mathbf{b} a given vector. Show that there exists a vector \mathbf{x} such that $\mathbf{Ax} \leq \mathbf{b}$ if and only if for any given vector \mathbf{y} satisfying $\mathbf{A}^\top \mathbf{y} = \mathbf{0}$ and $\mathbf{y} \geq \mathbf{0}$, we have $\mathbf{b}^\top \mathbf{y} \geq 0$. This result is known as *Gale's transposition theorem*.

17.20 Let \mathbf{A} be a given matrix. Show that there exists a vector \mathbf{x} such that $\mathbf{Ax} < \mathbf{0}$ if and only if for any given vector \mathbf{y} satisfying $\mathbf{A}^\top \mathbf{y} = \mathbf{0}$ and $\mathbf{y} \geq \mathbf{0}$, we have $\mathbf{y} = \mathbf{0}$ (i.e., $\mathbf{y} = \mathbf{0}$ is the only vector satisfying $\mathbf{A}^\top \mathbf{y} = \mathbf{0}$ and $\mathbf{y} \geq \mathbf{0}$). This result is known as *Gordan's transposition theorem*.

17.21 Let $\mathbf{P} \in \mathbb{R}^{n \times n}$ be a matrix with the property that each element is in the real interval $[0, 1]$, and the sum of the elements of each row is equal to 1; call such a matrix a *stochastic matrix*. Now consider a vector $\mathbf{x} \geq \mathbf{0}$ such that $\mathbf{x}^\top \mathbf{e} = 1$, where $\mathbf{e} = [1, \dots, 1]^\top$; call such a vector \mathbf{x} a *probability vector*.

We wish to prove the following result: For any stochastic matrix \mathbf{P} , there exists a probability vector \mathbf{x} such that $\mathbf{x}^\top \mathbf{P} = \mathbf{x}^\top$. Although this is a key result in probability theory (under the topic of *Markov chains*), our argument is based on duality theory (for linear programming), consisting of the following parts.

- a. Consider the primal linear program:

$$\begin{aligned} &\text{maximize} && \mathbf{x}^\top \mathbf{e} \\ &\text{subject to} && \mathbf{x}^\top \mathbf{P} = \mathbf{x}^\top \\ &&& \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Write down the dual of this problem.

- b. Show that the dual is not feasible (i.e., there does not exist a feasible solution to the dual).
- Hint:* Derive a contradiction based on $\mathbf{P}\mathbf{y} > \mathbf{y}$; think about the largest element of \mathbf{y} (call it y_i).
- c. Is the primal feasible? What can you deduce about whether or not the primal is unbounded?
 - d. Use part c to deduce the desired result: that there exists a vector $\mathbf{x} \geq \mathbf{0}$ such that $\mathbf{x}^\top \mathbf{P} = \mathbf{x}^\top$ and $\mathbf{x}^\top \mathbf{e} = 1$.

17.22 Suppose that you are presented with a “black box” that implements a function ϕ defined as follows: Given positive integers m and n , a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, and a vector $\mathbf{b} \in \mathbb{R}^m$, the value of $\phi(m, n, \mathbf{A}, \mathbf{b})$ is a vector $\mathbf{x} = \phi(m, n, \mathbf{A}, \mathbf{b})$ that satisfies $\mathbf{Ax} \geq \mathbf{b}$, if such a vector exists. In other words, the black box solves a linear *feasibility problem*.

Now, given $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$, consider the linear programming problem

$$\begin{aligned} &\text{minimize} && \mathbf{c}^\top \mathbf{x} \\ &\text{subject to} && \mathbf{Ax} \geq \mathbf{b} \\ &&& \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Express a solution to this problem in terms of the function ϕ given above. In other words, show how we can use the black box to solve this linear programming problem.

Hint: Find the appropriate inputs to the black box such that the output immediately gives a solution to the linear programming problem. You should use the black box only once.

17.23 This exercise illustrates the use of duality to compute the sensitivity of the optimal objective function value with respect to perturbations in the constraint.

Consider a primal linear programming problem and its dual (in either symmetric or asymmetric form). Let us view the \mathbf{b} vector in the primal as a parameter that we can vary, and that we wish to calculate the change in the optimal objective function value if we perturb \mathbf{b} by a small perturbation $\Delta\mathbf{b}$ (i.e., replace \mathbf{b} by $\mathbf{b} + \Delta\mathbf{b}$).

- a. To make the problem precise, let $z(\mathbf{b})$ be the optimal value of the primal objective function. Let $\boldsymbol{\lambda}$ denote the corresponding optimal dual vector. Calculate the gradient of z at \mathbf{b} : $\nabla z(\mathbf{b})$. Write the answer in terms of $\boldsymbol{\lambda}$. You may assume that the optimal dual vector remains fixed in a neighborhood of \mathbf{b} ; but if you do, you must explain why this assumption

is reasonable.

Hint: Use the duality theorem to see how $z(\mathbf{b})$ depends on \mathbf{b} .

- b. Suppose that the first component of the optimal dual vector is $\lambda_1 = 3$. Now suppose that we increase b_1 by a very small amount Δb_1 . Determine the amount by which the optimal objective function value will change.

17.24 Consider the *quadratic programming* problem

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \mathbf{x}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Call this problem the *primal problem*.

Consider the associated *dual* quadratic programming problem

$$\begin{aligned} & \text{maximize} && -\frac{1}{2} \mathbf{y}^\top (\mathbf{A}\mathbf{A}^\top) \mathbf{y} - \mathbf{b}^\top \mathbf{y} \\ & \text{subject to} && \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

Let f_1 and f_2 be the objective functions of the primal and dual, respectively.

- a. State and prove a weak duality lemma in this setting.
- b. Show that if \mathbf{x}_0 and \mathbf{y}_0 are feasible points in the primal and dual, and $f_1(\mathbf{x}_0) = f_2(\mathbf{y}_0)$, then \mathbf{x}_0 and \mathbf{y}_0 are optimal solutions to the primal and dual, respectively.

Hint: The techniques used in the linear programming duality results are applicable in this exercise.

CHAPTER 18

NONSIMPLEX METHODS

18.1 Introduction

In previous chapters we studied the simplex method and its variant, the revised simplex method, for solving linear programming problems. The method remains widely used in practice for solving LP problems. However, the amount of time required to compute a solution using the simplex method grows rapidly as the number of components n of the variable $\mathbf{x} \in \mathbb{R}^n$ increases. Specifically, it turns out that the relationship between the required amount of time for the algorithm to find a solution and the size n of \mathbf{x} is exponential in the worst case. An example of an LP problem for which this relationship is evident was devised by Klee and Minty in 1972 [76]. Below, we give a version of the

Klee-Minty example, taken from [9]. Let n be given. Let

$$\begin{aligned}\mathbf{c} &= [10^{n-1}, 10^{n-2}, \dots, 10^1, 1]^\top, \\ \mathbf{b} &= [1, 10^2, 10^4, \dots, 10^{2(n-1)}]^\top, \\ \mathbf{A} &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 2 \times 10^1 & 1 & 0 & \cdots & 0 \\ 2 \times 10^2 & 2 \times 10^1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 2 \times 10^{n-1} & 2 \times 10^{n-2} & \cdots & 2 \times 10^1 & 1 \end{bmatrix}.\end{aligned}$$

Consider the following LP problem:

$$\begin{aligned}&\text{maximize} \quad \mathbf{c}^\top \mathbf{x} \\ &\text{subject to} \quad \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ &\quad \mathbf{x} \geq \mathbf{0}.\end{aligned}$$

The simplex algorithm applied to the LP problem above requires $2^n - 1$ steps to find the solution. Clearly, in this example the relationship between the required amount of time for the simplex algorithm to find a solution and the size n of the variable \mathbf{x} is exponential. This relationship is also called the *complexity* of the algorithm. The simplex algorithm is therefore said to have *exponential complexity*. The complexity of the simplex algorithm is also often written as $O(2^n - 1)$.

Naturally, we would expect that any algorithm that solves LP problems would have the property that the time required to arrive at a solution increases with the size n of the variable \mathbf{x} . However, the issue at hand is the rate at which this increase occurs. As we have seen above, the simplex algorithm has the property that this rate of increase is exponential. For a number of years, computer scientists have distinguished between *exponential complexity* and *polynomial complexity*. If an algorithm for solving LP problems has polynomial complexity, then the time required to obtain the solution is bounded by a polynomial in n . Obviously, polynomial complexity is more desirable than exponential complexity. Therefore, the existence of an algorithm for solving LP problems with polynomial complexity is an important issue. This issue was partially resolved in 1979 by Khachiyan (also transliterated as Hacijan) [74], who proposed an algorithm that has a complexity $O(n^4 L)$, where, roughly speaking, L represents the number of bits used in the computations. The reason that we consider Khachiyan's algorithm (also called the *ellipsoid algorithm*) as only a partial resolution of this issue is that the complexity depends on L , which implies that the time required to solve a given LP problem increases with the required accuracy of the computations. The existence of a method for solving LP problems with a polynomial complexity bound based only on the size of the variable n (and possibly the number of constraints) remains a difficult open problem [55]. In any case, computational

experience with Khachiyan's algorithm has shown that it is not a practical alternative to the simplex method [14]. The theoretical complexity advantage of Khachiyan's method relative to the simplex method remains to be demonstrated in practice.

Another nonsimplex algorithm for solving LP problems was proposed in 1984 by Karmarkar [71]. Karmarkar's algorithm has a complexity of $O(n^{3.5}L)$, which is lower than that of Khachiyan's algorithm. The algorithm is superior to the simplex algorithm from a complexity viewpoint, but has its drawbacks. Improved methods along similar lines, called *interior-point methods*, have received considerable interest since Karmarkar's original paper. Well-implemented versions of these methods are very efficient, especially when the problem involves a large number of variables [55].

This chapter is devoted to a discussion of nonsimplex methods for solving LP problems. In the next section we discuss some ideas underlying Khachiyan's algorithm. We then present Karmarkar's algorithm in the section to follow.

18.2 Khachiyan's Method

Our description of the Khachiyan's algorithm is based on [8] and [9]. The method relies on the concept of duality (see Chapter 17). Our exposition of Khachiyan's algorithm is geared toward a basic understanding of the method. For a detailed rigorous treatment of the method, we refer the reader to [101].

Consider the (primal) linear programming problem

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

We write the corresponding dual problem,

$$\begin{aligned} & \text{maximize} && \boldsymbol{\lambda}^\top \mathbf{b} \\ & \text{subject to} && \boldsymbol{\lambda}^\top \mathbf{A} \leq \mathbf{c}^\top \\ & && \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

Recall that the two LP problems above constitute the symmetric form of duality. From Theorem 17.1, if \mathbf{x} and $\boldsymbol{\lambda}$ are feasible solutions to the primal and dual problems, respectively, and $\mathbf{c}^\top \mathbf{x} = \boldsymbol{\lambda}^\top \mathbf{b}$, then \mathbf{x} and $\boldsymbol{\lambda}$ are optimal solutions to their respective problems. Using this result, we see that to solve the primal problem it is enough to find a vector $[\mathbf{x}^\top, \boldsymbol{\lambda}^\top]^\top$ that satisfies the

following set of relations:

$$\begin{aligned} \mathbf{c}^\top \mathbf{x} &= \mathbf{b}^\top \boldsymbol{\lambda}, \\ \mathbf{A}\mathbf{x} &\geq \mathbf{b}, \\ \mathbf{A}^\top \boldsymbol{\lambda} &\leq \mathbf{c}, \\ \mathbf{x} &\geq \mathbf{0}, \\ \boldsymbol{\lambda} &\geq \mathbf{0}. \end{aligned}$$

Note that the equality $\mathbf{c}^\top \mathbf{x} = \mathbf{b}^\top \boldsymbol{\lambda}$ is equivalent to the two inequalities

$$\begin{aligned} \mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \boldsymbol{\lambda} &\leq 0, \\ -\mathbf{c}^\top \mathbf{x} + \mathbf{b}^\top \boldsymbol{\lambda} &\leq 0. \end{aligned}$$

Taking this into account, we can represent the previous set of relations as

$$\begin{bmatrix} \mathbf{c}^\top & -\mathbf{b}^\top \\ -\mathbf{c}^\top & \mathbf{b}^\top \\ -\mathbf{A} & \mathbf{0} \\ -\mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top \\ \mathbf{0} & -\mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ -\mathbf{b} \\ \mathbf{0} \\ \mathbf{c} \\ \mathbf{0} \end{bmatrix}.$$

Therefore, we have reduced the problem of finding an optimal solution to the primal-dual pair into one of finding a vector $[\mathbf{x}^\top, \boldsymbol{\lambda}^\top]^\top$ that satisfies the system of inequalities above. In other words, if we can find a vector that satisfies the system of inequalities, then this vector gives an optimal solution to the primal-dual pair. On the other hand, if there does not exist a vector satisfying the system of inequalities, then the primal-dual pair has no optimal feasible solution. In the subsequent discussion, we simply represent the system of inequalities as

$$\mathbf{P}\mathbf{z} \leq \mathbf{q},$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{c}^\top & -\mathbf{b}^\top \\ -\mathbf{c}^\top & \mathbf{b}^\top \\ -\mathbf{A} & \mathbf{0} \\ -\mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top \\ \mathbf{0} & -\mathbf{I}_m \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{b} \\ \mathbf{0} \\ \mathbf{c} \\ \mathbf{0} \end{bmatrix}.$$

In our discussion of Khachiyan's algorithm, we will not be using these forms of \mathbf{P} , \mathbf{q} , and \mathbf{z} specifically; we simply treat $\mathbf{P}\mathbf{z} \leq \mathbf{q}$ as a generic matrix inequality, with \mathbf{P} , \mathbf{q} , and \mathbf{z} as generic entities. Let r and s be the sizes of \mathbf{q} and \mathbf{z} , respectively; that is, $\mathbf{P} \in \mathbb{R}^{r \times s}$, $\mathbf{z} \in \mathbb{R}^s$, and $\mathbf{q} \in \mathbb{R}^r$.

Khachiyan's method solves the LP problem by first determining if there exists a vector \mathbf{z} that satisfies the inequality $\mathbf{P}\mathbf{z} \leq \mathbf{q}$; that is, the algorithm decides if the system of linear inequalities above is *consistent*. If the system is consistent, then the algorithm finds a vector \mathbf{z} satisfying the system. In the following we refer to any vector satisfying the system as a *solution* to the system. We assume that the entries in \mathbf{P} and \mathbf{q} are all rational numbers. This is not a restriction in practice, since any representation of our LP problem on a digital computer will involve only rational numbers. In fact, we assume further that the entries in \mathbf{P} and \mathbf{q} are all integers. We can do this without loss of generality since we can always multiply both sides of the inequality $\mathbf{P}\mathbf{z} \leq \mathbf{q}$ by a sufficiently large number to get only integer entries on both sides.

Before discussing Khachiyan's algorithm, we introduce the idea of an *ellipsoid*. To this end, let $\mathbf{z} \in \mathbb{R}^s$ be a given vector and let \mathbf{Q} be an $s \times s$ nonsingular matrix. Then, the *ellipsoid* associated with \mathbf{Q} centered at \mathbf{z} is defined as the set

$$E_{\mathbf{Q}}(\mathbf{z}) = \{\mathbf{z} + \mathbf{Q}\mathbf{y} : \mathbf{y} \in \mathbb{R}^s, \|\mathbf{y}\| \leq 1\}.$$

The main idea underlying Khachiyan's algorithm is as follows. Khachiyan's algorithm is an iterative procedure, where at each iteration we update a vector $\mathbf{z}^{(k)}$ and a matrix \mathbf{Q}_k . Associated with $\mathbf{z}^{(k)}$ and \mathbf{Q}_k is an ellipsoid $E_{\mathbf{Q}_k}(\mathbf{z}^{(k)})$. At each step of the algorithm, the associated ellipsoid contains a solution to the given system of linear inequalities. The algorithm updates $\mathbf{z}^{(k)}$ and \mathbf{Q}_k in such a way that the ellipsoid at the next step is "smaller" than that of the current step, but at the same time is guaranteed to contain a solution to the given system of inequalities, if one exists. If we find that the current point $\mathbf{z}^{(k)}$ satisfies $\mathbf{P}\mathbf{z}^{(k)} \leq \mathbf{q}$, then we terminate the algorithm and conclude that $\mathbf{z}^{(k)}$ is a solution. Otherwise, we continue to iterate. The algorithm has a fixed prespecified maximum number of iterations N to be performed, where N is a number that depends on L and s . Note that we are not free to choose N —it is computed using a formula that uses the values of L and s . The constant L is itself a quantity that we have to compute beforehand using a formula that involves \mathbf{P} and \mathbf{q} . When we have completed N iterations without finding a solution in an earlier step, we terminate the algorithm. The associated ellipsoid will then have shrunk to the extent that it is smaller than the precision of computation. At this stage, we will either discover a solution inside the ellipsoid, if indeed a solution exists, or we will find no solution inside the ellipsoid, in which case we conclude that no solution exists.

As we can see from the description above, Khachiyan's approach is a radical departure from the classical simplex method for solving LP problems. The method has attracted a lot of attention, and many studies have been devoted to it. However, as we pointed out earlier, the algorithm is of little practical value for solving real-world LP problems. Therefore, we do not delve any further into the details of Khachiyan's algorithm. We refer the interested reader to [101].

Despite its practical drawbacks, Khachiyan's method has inspired other researchers to pursue the development of computationally efficient algorithms for solving LP problems with polynomial complexity. One such algorithm is attributed to Karmarkar, which we discuss in Section 18.4.

18.3 Affine Scaling Method

Basic Algorithm

In this section we describe a simple algorithm, called the *affine scaling method*, for solving linear programming problems. This description is to prepare the reader for our discussion of Karmarkar's method in the next section. The affine scaling method is a *an interior-point method*. Such methods differ fundamentally from the classical simplex method in one main respect: An interior-point method starts inside the feasible set and moves within it toward an optimal vertex. In contrast, the simplex method jumps from vertex to vertex of the feasible set seeking an optimal vertex.

To begin our description of the affine scaling method, consider the LP problem

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Note that the feasibility constraints have two parts: $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$. Suppose that we have a feasible point $\mathbf{x}^{(0)}$ that is *strictly interior* (by strictly interior we mean that all of the components of $\mathbf{x}^{(0)}$ are strictly positive). We wish to find a new point $\mathbf{x}^{(1)}$ by searching in a direction $\mathbf{d}^{(0)}$ that decreases the objective function. In other words, we set

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)},$$

where α_0 is a step size. In the gradient method (Chapter 8) we used the negative gradient of the objective function for the search direction. For the LP problem, the negative gradient of the objective function is $-\mathbf{c}$. However, if we set $\mathbf{d}^{(0)} = -\mathbf{c}$, the point $\mathbf{x}^{(1)}$ may not lie inside the feasible set. For $\mathbf{x}^{(1)}$ to lie inside the feasible set, it is necessary that $\mathbf{d}^{(0)}$ be a vector in the nullspace of \mathbf{A} . Indeed, because $\mathbf{x}^{(0)}$ is feasible, we have $\mathbf{A}\mathbf{x}^{(0)} = \mathbf{b}$. We also require that $\mathbf{A}\mathbf{x}^{(1)} = \mathbf{b}$. Combining these two equations yields

$$\mathbf{A} \left(\mathbf{x}^{(1)} - \mathbf{x}^{(0)} \right) = \alpha_0 \mathbf{A} \mathbf{d}^{(0)} = \mathbf{0}.$$

To choose a direction $\mathbf{d}^{(0)}$ that lies in the nullspace of \mathbf{A} but is still “close” to $-\mathbf{c}$, we orthogonally project $-\mathbf{c}$ onto the nullspace of \mathbf{A} and take the resulting

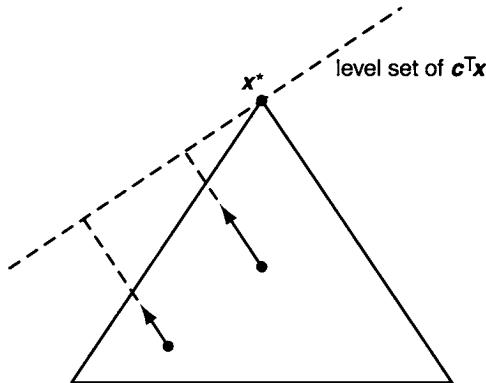


Figure 18.1 Results of projected gradient step from center and noncenter points.

projection as $\mathbf{d}^{(0)}$. The orthogonal projection of any vector onto the nullspace of \mathbf{A} involves multiplication by the following matrix \mathbf{P} , called the *orthogonal projector* (see Section 3.3 and Example 12.5):

$$\mathbf{P} = \mathbf{I}_n - \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1} \mathbf{A}.$$

We set $\mathbf{d}^{(0)}$ to be in the direction of the orthogonal projection of $-\mathbf{c}$ onto the nullspace of \mathbf{A} :

$$\mathbf{d}^{(0)} = -\mathbf{P}\mathbf{c}.$$

It is easy to check that $\mathbf{A}\mathbf{P}\mathbf{c} = \mathbf{0}$ and hence $\mathbf{A}\mathbf{x}^{(1)} = \mathbf{b}$. In summary, given a feasible point $\mathbf{x}^{(0)}$, we find a new feasible point $\mathbf{x}^{(1)}$ using

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \alpha_0 \mathbf{P}\mathbf{c},$$

where the choice of the step size α_0 is discussed later in the section. The choice of $\mathbf{x}^{(1)}$ above can be viewed as one iteration of a *projected gradient algorithm*, discussed in Section 23.3.

We now make the observation that the point $\mathbf{x}^{(0)}$ should be chosen close to the center of the feasible set. Figure 18.1 illustrates this observation. Comparing the center and noncenter starting points in the figure, we can see that if we start at the center of the feasible set, we can take a larger step in the search direction. This larger step from the center point should yield a lower-cost value for the new point compared with the step originating from the noncenter point.

Suppose that we are given a point $\mathbf{x}^{(0)}$ that is feasible but is not a center point. We can transform the point to the center by applying what is called an *affine scaling*. For simplicity, suppose that $\mathbf{A} = [1, 1, \dots, 1]/n$ and $\mathbf{b} = [1]$. It is easy to see that the center of this feasible set is the point $\mathbf{e} = [1, \dots, 1]^\top$. To transform $\mathbf{x}^{(0)}$ to \mathbf{e} , we use the affine-scaling transformation

$$\mathbf{e} = \mathbf{D}_0^{-1} \mathbf{x}^{(0)},$$

where \mathbf{D}_0 is a diagonal matrix whose diagonal entries are the components of the vector $\mathbf{x}^{(0)}$:

$$\mathbf{D}_0 = \text{diag}[x_1^{(0)}, \dots, x_n^{(0)}] = \begin{bmatrix} x_1^{(0)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x_n^{(0)} \end{bmatrix}.$$

Note that \mathbf{D}_0 is invertible because we assumed that $\mathbf{x}^{(0)}$ is *strictly interior*. For general \mathbf{A} and \mathbf{b} we will still use the same affine-scaling transformation as above. In general, we may not be at precisely the center of the feasible set, but we hope that the transformed point will be “close” to the center. At least the point \mathbf{e} is equidistant from the boundaries of the positive orthant $\{\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}$.

Once the starting point is at (or close to) the center of the feasible set after performing the affine-scaling transformation, we can proceed as described before. Because we have transformed the original vector $\mathbf{x}^{(0)}$ via premultiplication by \mathbf{D}_0^{-1} , effectively changing the coordinate system, we also need to represent the original LP problem in the new coordinates. Specifically, the LP problem in the transformed coordinates takes the form

$$\begin{aligned} & \text{minimize} && \bar{\mathbf{c}}_0^\top \bar{\mathbf{x}} \\ & \text{subject to} && \bar{\mathbf{A}}_0 \bar{\mathbf{x}} = \mathbf{b} \\ & && \bar{\mathbf{x}} \geq \mathbf{0}, \end{aligned}$$

where

$$\begin{aligned} \bar{\mathbf{c}}_0 &= \mathbf{D}_0 \mathbf{c}, \\ \bar{\mathbf{A}}_0 &= \mathbf{A} \mathbf{D}_0. \end{aligned}$$

In the new $(\bar{\mathbf{x}})$ coordinate system we construct the orthogonal projector

$$\bar{\mathbf{P}}_0 = \mathbf{I}_n - \bar{\mathbf{A}}_0^\top (\bar{\mathbf{A}}_0 \bar{\mathbf{A}}_0^\top)^{-1} \bar{\mathbf{A}}_0$$

and set $\bar{\mathbf{d}}^{(0)}$ to be in the direction of the orthogonal projection of $-\bar{\mathbf{c}}_0$ onto the nullspace of $\bar{\mathbf{A}}_0$:

$$\bar{\mathbf{d}}^{(0)} = -\bar{\mathbf{P}}_0 \bar{\mathbf{c}}_0.$$

Then, compute $\bar{\mathbf{x}}^{(1)}$ using

$$\bar{\mathbf{x}}^{(1)} = \bar{\mathbf{x}}^{(0)} - \alpha_0 \bar{\mathbf{P}}_0 \bar{\mathbf{c}}_0,$$

where $\bar{\mathbf{x}}^{(0)} = \mathbf{D}_0^{-1} \mathbf{x}^{(0)}$. To obtain a point in the original coordinates, we perform the transformation

$$\mathbf{x}^{(1)} = \mathbf{D}_0 \bar{\mathbf{x}}^{(1)}.$$

The procedure above takes a point $\mathbf{x}^{(0)}$ and generates a new point $\mathbf{x}^{(1)}$. This procedure can be represented as

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)},$$

where

$$\mathbf{d}^{(0)} = -\mathbf{D}_0 \bar{\mathbf{P}} \mathbf{D}_0 \mathbf{c}.$$

We repeat the procedure iteratively to generate a sequence of points $\{\mathbf{x}^{(k)}\}$, where

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$$

with

$$\begin{aligned}\mathbf{D}_k &= \text{diag}[x_1^{(k)}, \dots, x_n^{(k)}], \\ \bar{\mathbf{A}}_k &= \mathbf{A} \mathbf{D}_k, \\ \bar{\mathbf{P}}_k &= \mathbf{I}_n - \bar{\mathbf{A}}_k^\top (\bar{\mathbf{A}}_k \bar{\mathbf{A}}_k^\top)^{-1} \bar{\mathbf{A}}_k, \\ \mathbf{d}^{(k)} &= -\mathbf{D}_k \bar{\mathbf{P}}_k \mathbf{D}_k \mathbf{c}.\end{aligned}$$

At each stage of the algorithm, we have to ensure that the point $\mathbf{x}^{(k)}$ is strictly interior. Note that the condition $\mathbf{A}\mathbf{x}^{(k)} = \mathbf{b}$ is satisfied automatically at each stage because of the way we select $\mathbf{d}^{(k)}$. However, we also need to guarantee that $x_i^{(k)} > 0$ for $i = 1, \dots, n$. This can be done through appropriate choice of the step size α_k , discussed next.

The main criterion for choosing α_k is to make it as large as possible, but not so large that some components of $\mathbf{x}^{(k+1)}$ become nonpositive. That is, we select α_k so that $x_i^{(k+1)} = x_i^{(k)} + \alpha_k d_i^{(k)} > 0$ for $i = 1, \dots, n$. To proceed, first define

$$r_k = \min_{\{i: d_i^{(k)} < 0\}} -\frac{x_i^{(k)}}{d_i^{(k)}}.$$

The number r_k represents the largest value of the step size α_k such that all the components of $\mathbf{x}^{(k+1)}$ are nonnegative. To ensure that $\mathbf{x}^{(k+1)}$ is strictly interior, we use a step size of the form $\alpha_k = \alpha r_k$, where $\alpha \in (0, 1)$. Typical values of α for this method are $\alpha = 0.9$ or 0.99 (see [96, p. 572]).

Unlike the simplex method, the affine scaling method will not reach the optimal solution in a finite number of steps. Therefore, we need a stopping criterion. For this, we can use any of the stopping criteria discussed in Section 8.2. For example, we can stop if

$$\frac{|\mathbf{c}\mathbf{x}^{(k+1)} - \mathbf{c}\mathbf{x}^{(k)}|}{\max\{1, |\mathbf{c}\mathbf{x}^{(k)}|\}} < \varepsilon,$$

where $\varepsilon > 0$ is a prespecified threshold (see also [96, p. 572] for a similar stopping criterion, as well as an alternative criterion involving duality).

Two-Phase Method

To implement the affine scaling method described above, we need an initial feasible starting point that is strictly interior. We now describe a method to find such a starting point. After the starting point is found, we can then proceed to search for an optimal solution to the problem. This approach involves two phases: In phase I we find an initial strictly interior feasible point, and in phase II we use the result of phase I to initialize the affine scaling algorithm to find an optimal solution. This procedure is analogous to the two-phase simplex algorithm described in Section 16.6.

We now describe phase I of the two-phase affine scaling method. Let \mathbf{u} be an arbitrary vector with positive components, and let

$$\mathbf{v} = \mathbf{b} - \mathbf{A}\mathbf{u}.$$

If $\mathbf{v} = \mathbf{0}$, then \mathbf{u} is a strictly interior feasible point. We can then set $\mathbf{x}^{(0)} = \mathbf{u}$ and proceed to phase II, where we apply the affine scaling method as described before. On the other hand, if $\mathbf{v} \neq \mathbf{0}$, we construct the following associated *artificial problem*:

$$\begin{aligned} & \text{minimize} && y \\ & \text{subject to} && [\mathbf{A}, \mathbf{v}] \begin{bmatrix} \mathbf{x} \\ y \end{bmatrix} = \mathbf{b} \\ & && \begin{bmatrix} \mathbf{x} \\ y \end{bmatrix} \geq \mathbf{0}. \end{aligned}$$

The artificial problem above has an obvious strictly interior feasible point:

$$\begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix}.$$

Using this point as the initial point, we can apply the affine scaling algorithm to the artificial problem. Because the objective function in the artificial problem is bounded below by 0, the affine scaling method will terminate with some optimal solution.

Proposition 18.1 *The original LP problem has a feasible solution if and only if the associated artificial problem has an optimal feasible solution with objective function value zero.* \square

Proof. \Rightarrow : If the original problem has a feasible solution \mathbf{x} , then the vector $[\mathbf{x}^\top, 0]^\top$ is a feasible solution to the artificial problem. Clearly, this solution has an objective function value of zero. This solution is therefore optimal for the artificial problem, since there can be no feasible solution with negative objective function value.

\Leftarrow : Suppose that the artificial problem has an optimal feasible solution with objective function value zero. Then, this solution must have the form $[\mathbf{x}^\top, 0]^\top$, where $\mathbf{x} \geq \mathbf{0}$. Hence, we have $\mathbf{A}\mathbf{x} = \mathbf{b}$, and \mathbf{x} is a feasible solution to the original problem. ■

Suppose that the original LP problem has a feasible solution. By Proposition 18.1, if we apply the affine scaling method to the artificial problem (with initial point $[\mathbf{u}^\top, 1]^\top$), the algorithm will terminate with objective function value zero. The optimal solution will be of the form $[\mathbf{x}^\top, 0]^\top$. We argue that \mathbf{x} will in general be a strictly interior feasible point. It is easy to see that $\mathbf{x} \geq \mathbf{0}$. To convince ourselves that each component of \mathbf{x} will be positive in general, note that the subset of optimal feasible solutions of the artificial problem in which one or more among the first n components are zero is a very small or thin subset of the set of all optimal feasible solutions. By *small* or *thin* we mean in the sense that a two-dimensional plane in \mathbb{R}^3 is small or thin. In particular, the volume of the two-dimensional plane in \mathbb{R}^3 is zero. Thus, it is very unlikely that the affine scaling algorithm will terminate with an optimal feasible solution in which one or more among the first n components are zero.

Having completed phase I as described above, we then use the first n components of the terminal optimal feasible solution for the artificial problem as our initial point for the affine scaling method applied to the original LP problem. This second application of the affine scaling algorithm constitutes phase II.

In theory, phase I generates a feasible point to initiate phase II. However, because of the finite precision of typical computer implementations, the solution obtained from phase I may not, in fact, be feasible. Moreover, even if the initial point in phase II is feasible, in practice the iterates may lose feasibility, owing to finite precision computations. Special procedures for dealing with such problems are available. For a discussion of numerical implementation of affine scaling algorithms, see [42, Section 7.1.2].

18.4 Karmarkar's Method

Basic Ideas

Like the affine scaling method, Karmarkar's method for solving LP problems differs fundamentally from the classical simplex method in various respects. First, Karmarkar's method is an interior-point method. Another difference between Karmarkar's method and the simplex method is that the latter stops when it finds an optimal solution. On the other hand, Karmarkar's method stops when it finds a solution that has an objective function value that is less than or equal to a prespecified fraction of the original guess. A third difference between the two methods is that the simplex method starts with LP problems in standard form, whereas Karmarkar's method starts with LP problems in a special canonical form, which we call *Karmarkar's canonical*

form. We discuss this canonical form in the next subsection. While more recent interior-point methods are recognized to be superior to Karmarkar's original algorithm in efficiency and robustness, a study of Karmarkar's method provides an informative introduction to the study of more advanced interior-point methods.

Karmarkar's Canonical Form

To apply Karmarkar's algorithm to a given LP problem, we must first transform the given problem into a particular form, which we refer to as Karmarkar's canonical form. Karmarkar's canonical form is written as

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{0} \\ & && \sum_{i=1}^n x_i = 1 \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{x} = [x_1, \dots, x_n]^\top$. As in our discussion of Khachiyan's method, we assume without loss of generality that the entries of \mathbf{A} and \mathbf{c} are integers.

We now introduce some notation that allows convenient manipulation of the canonical form. First, let $\mathbf{e} = [1, \dots, 1]^\top$ be the vector in \mathbb{R}^n with each component equal to 1. Let Ω denote the nullspace of \mathbf{A} , that is, the subspace

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{0}\}.$$

Define the *simplex* in \mathbb{R}^n by

$$\Delta = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{e}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}.$$

We denote the *center* of the simplex Δ by

$$\mathbf{a}_0 = \frac{\mathbf{e}}{n} = \left[\frac{1}{n}, \dots, \frac{1}{n} \right]^\top.$$

Clearly, $\mathbf{a}_0 \in \Delta$. With the notation above, Karmarkar's canonical form can be rewritten as

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{x} \in \Omega \cap \Delta. \end{aligned}$$

Note that the constraint set (or feasible set) $\Omega \cap \Delta$ can be represented as

$$\begin{aligned} \Omega \cap \Delta &= \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{0}, \mathbf{e}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\} \\ &= \left\{ \mathbf{x} \in \mathbb{R}^n : \begin{bmatrix} \mathbf{A} \\ \mathbf{e}^\top \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \mathbf{x} \geq \mathbf{0} \right\}. \end{aligned}$$

Example 18.1 Consider the following LP problem, taken from [125]:

$$\begin{aligned} & \text{minimize} && 5x_1 + 4x_2 + 8x_3 \\ & \text{subject to} && x_1 + x_2 + x_3 = 1 \\ & && x_1, x_2, x_3 \geq 0. \end{aligned}$$

Clearly, this problem is already in Karmarkar's canonical form, with $\mathbf{c}^\top = [5, 4, 8]$, and $\mathbf{A} = \mathbf{O}$. The feasible set for this example is illustrated in Figure 18.2. ■

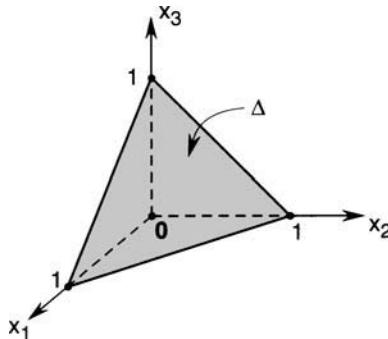


Figure 18.2 Feasible set for Example 18.1.

Example 18.2 Consider the following LP problem, taken from [110]:

$$\begin{aligned} & \text{minimize} && 3x_1 + 3x_2 - x_3 \\ & \text{subject to} && 2x_1 - 3x_2 + x_3 = 0 \\ & && x_1 + x_2 + x_3 = 1 \\ & && x_1, x_2, x_3 \geq 0. \end{aligned}$$

This problem is in Karmarkar's canonical form, with $\mathbf{c}^\top = [3, 3, -1]$ and $\mathbf{A} = [2, -3, 1]$. The feasible set for this example is illustrated in Figure 18.3 (adapted from [110]). ■

We show later that any LP problem can be converted into an equivalent problem in Karmarkar's canonical form.

Karmarkar's Restricted Problem

Karmarkar's algorithm solves LP problems in Karmarkar's canonical form, with the following assumptions:

- A. The center \mathbf{a}_0 of the simplex Δ is a feasible point: $\mathbf{a}_0 \in \Omega$.

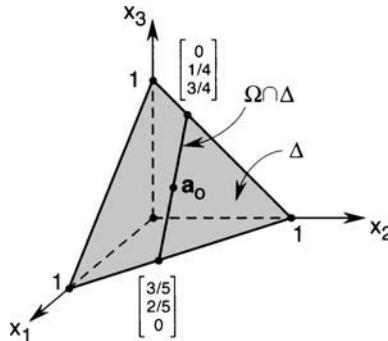


Figure 18.3 The feasible set for Example 18.2.

B. The minimum value of the objective function over the feasible set is zero.

C. The $(m + 1) \times n$ matrix

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{e}^\top \end{bmatrix}$$

has rank $m + 1$.

D. We are given a termination parameter $q > 0$, such that if we obtain a feasible point \mathbf{x} satisfying

$$\frac{\mathbf{c}^\top \mathbf{x}}{\mathbf{c}^\top \mathbf{a}_0} \leq 2^{-q},$$

then we consider the problem solved.

Any LP problem that is in Karmarkar's canonical form and that also satisfies the four assumptions above is called a *Karmarkar's restricted problem*. In the following we discuss the assumptions and their interpretations.

We begin by looking at assumption A. We point out that this assumption is not restrictive, since any LP problem that has an optimal feasible solution can be converted into a problem in Karmarkar's canonical form that satisfies assumption A. We discuss this in the next subsection.

We next turn our attention to assumption B. Any LP problem in Karmarkar's canonical form can be converted into one that satisfies assumption B, provided that we know beforehand the minimum value of its objective function over the feasible set. Specifically, suppose that we are given an LP problem where the minimum value of the objective function is M . As in [110], consider the function $f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x} - M$. Then, using the property that $\mathbf{e}^\top \mathbf{x} = 1$ on the feasible set, we have that for any feasible \mathbf{x} ,

$$f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x} - M = \mathbf{c}^\top \mathbf{x} - M \mathbf{e}^\top \mathbf{x} = (\mathbf{c}^\top - M \mathbf{e}^\top) \mathbf{x} = \tilde{\mathbf{c}}^\top \mathbf{x},$$

where $\tilde{\mathbf{c}}^\top = \mathbf{c}^\top - M\mathbf{e}^\top$. Notice that the objective function above has a minimum value of zero and is a linear function of \mathbf{x} . We can replace the original objective function with the new objective function above, without altering the solution.

Example 18.3 Recall the LP problem in Example 18.1:

$$\begin{aligned} & \text{minimize} && 5x_1 + 4x_2 + 8x_3 \\ & \text{subject to} && x_1 + x_2 + x_3 = 1 \\ & && x_1, x_2, x_3 \geq 0. \end{aligned}$$

The problem satisfies assumption A (and assumption C) but not assumption B, since the minimum value of the objective function over the feasible set is 4. To convert the above into a problem that satisfies assumption B, we replace $\mathbf{c}^\top = [5, 4, 8]$ by $\tilde{\mathbf{c}}^\top = [1, 0, 4]$. ■

Example 18.4 The reader can easily verify that the LP problem in Example 18.2 satisfies assumptions A, B, and C. ■

Assumption C is a technical assumption that is required in the implementation of the algorithm. Its significance will be clear when we discuss the update equation in Karmarkar's algorithm.

Assumption D is the basis for the stopping criterion of Karmarkar's algorithm. In particular, we stop when we have found a feasible point satisfying $\mathbf{c}^\top \mathbf{x}/\mathbf{c}^\top \mathbf{a}_0 \leq 2^{-q}$. Such a stopping criterion is inherent in any algorithm that uses finite-precision arithmetic. Observe that the stopping criterion above depends on the value of $\mathbf{c}^\top \mathbf{a}_0$. It will turn out that Karmarkar's algorithm uses \mathbf{a}_0 as the starting point. Therefore, we can see that the accuracy of the final solution in the algorithm is influenced by the starting point.

From General Form to Karmarkar's Canonical Form

We now show how any LP problem can be converted into an equivalent problem in Karmarkar's canonical form. By *equivalent* we mean that the solution to one can be used to determine the solution to the other, and vice versa. To this end, recall that any LP problem can be transformed into an equivalent problem in standard form. Therefore, it suffices to show that any LP problem in standard form can be transformed into an equivalent problem in Karmarkar's canonical form. In fact, the transformation given below (taken from [71]) will also guarantee that assumption A of the preceding subsection is satisfied.

To proceed, consider a given LP problem in standard form:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n \\ & \text{subject to} && \mathbf{Ax} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

We first present a simple way to convert this problem into Karmarkar's canonical form, ignoring the requirement to satisfy assumption A. For this, define a new variable $\mathbf{z} \in \mathbb{R}^{n+1}$ by

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}.$$

Also define $\mathbf{c}' = [\mathbf{c}^\top, 0]^\top$ and $\mathbf{A}' = [\mathbf{A}, -\mathbf{b}]$. Using this notation, we can now rewrite the LP problem above as

$$\begin{aligned} & \text{minimize} && \mathbf{c}'^\top \mathbf{z}, \quad \mathbf{z} \in \mathbb{R}^{n+1} \\ & \text{subject to} && \mathbf{A}' \mathbf{z} = \mathbf{0} \\ & && \mathbf{z} \geq \mathbf{0}. \end{aligned}$$

We need one more step to transform the problem into one that includes the constraint that the decision variables sum to 1. For this, let $\mathbf{y} = [y_1, \dots, y_n, y_{n+1}]^\top \in \mathbb{R}^{n+1}$, where

$$\begin{aligned} y_i &= \frac{x_i}{x_1 + \dots + x_n + 1}, \quad i = 1, \dots, n \\ y_{n+1} &= \frac{1}{x_1 + \dots + x_n + 1}. \end{aligned}$$

This transformation from \mathbf{x} to \mathbf{y} is called a *projective transformation*. It can be shown that (see later)

$$\begin{aligned} \mathbf{c}^\top \mathbf{x} = 0 &\Leftrightarrow \mathbf{c}'^\top \mathbf{y} = 0, \\ \mathbf{A}\mathbf{x} = \mathbf{b} &\Leftrightarrow \mathbf{A}'\mathbf{y} = \mathbf{0}, \\ \mathbf{x} \geq \mathbf{0} &\Leftrightarrow \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

Therefore, we have transformed the given LP problem in standard form into the following problem, which is in Karmarkar's canonical form:

$$\begin{aligned} & \text{minimize} && \mathbf{c}'^\top \mathbf{y}, \quad \mathbf{y} \in \mathbb{R}^{n+1} \\ & \text{subject to} && \mathbf{A}' \mathbf{y} = \mathbf{0} \\ & && \mathbf{e}^\top \mathbf{y} = 1 \\ & && \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

The transformation technique above can be modified slightly to ensure that assumption A holds. We follow the treatment of [71]. We first assume that we are given a point $\mathbf{a} = [a_1, \dots, a_n]$ that is a *strictly interior* feasible point; that is, $\mathbf{A}\mathbf{a} = \mathbf{b}$ and $\mathbf{a} > \mathbf{0}$. We show later how this assumption can be enforced. Let P_+ denote the *positive orthant* of \mathbb{R}^n , given by $P_+ = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq \mathbf{0}\}$. Let $\Delta = \{\mathbf{x} \in \mathbb{R}^{n+1} : \mathbf{e}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}$ be the simplex in \mathbb{R}^{n+1} . Define the map $\mathbf{T} : P_+ \rightarrow \Delta$ by

$$\mathbf{T}(\mathbf{x}) = [T_1(\mathbf{x}), \dots, T_{n+1}(\mathbf{x})]^\top$$

with

$$\begin{aligned} T_i(\mathbf{x}) &= \frac{x_i/a_i}{x_1/a_1 + \cdots + x_n/a_n + 1}, \quad i = 1, \dots, n \\ T_{n+1}(\mathbf{x}) &= \frac{1}{x_1/a_1 + \cdots + x_n/a_n + 1}. \end{aligned}$$

We call the map \mathbf{T} a *projective transformation* of the positive orthant P_+ into the simplex Δ (for an introduction to projective transformations, see [68]). The transformation \mathbf{T} has several interesting properties (see Exercises 18.4, 18.5, and 18.6). In particular, we can find a vector $\mathbf{c}' \in \mathbb{R}^{n+1}$ and a matrix $\mathbf{A}' \in \mathbb{R}^{m \times (n+1)}$ such that for each $\mathbf{x} \in \mathbb{R}^n$,

$$\mathbf{c}'^\top \mathbf{x} = 0 \Leftrightarrow \mathbf{c}'^\top \mathbf{T}(\mathbf{x}) = 0$$

and

$$\mathbf{A}\mathbf{x} = \mathbf{b} \Leftrightarrow \mathbf{A}'\mathbf{T}(\mathbf{x}) = \mathbf{0}$$

(see Exercises 18.5 and 18.6 for the forms of \mathbf{A}' and \mathbf{c}'). Note that for each $\mathbf{x} \in \mathbb{R}^n$, we have $\mathbf{e}^\top \mathbf{T}(\mathbf{x}) = 1$, which means that $\mathbf{T}(\mathbf{x}) \in \Delta$. Furthermore, note that for each $\mathbf{x} \in \mathbb{R}^n$,

$$\mathbf{x} \geq 0 \Leftrightarrow \mathbf{T}(\mathbf{x}) \geq 0.$$

Taking this into account, consider the following LP problem (where \mathbf{y} is the decision variable):

$$\begin{aligned} &\text{minimize} && \mathbf{c}'^\top \mathbf{y} \\ &\text{subject to} && \mathbf{A}'\mathbf{y} = \mathbf{0} \\ &&& \mathbf{e}^\top \mathbf{y} = 1 \\ &&& \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

Note that this LP problem is in Karmarkar's canonical form. Furthermore, in light of the definitions of \mathbf{c}' and \mathbf{A}' , the above LP problem is equivalent to the original LP problem in standard form. Hence, we have converted the LP problem in standard form into an equivalent problem in Karmarkar's canonical form. In addition, because \mathbf{a} is a strictly interior feasible point, and $\mathbf{a}_0 = \mathbf{T}(\mathbf{a})$ is the center of the simplex Δ (see Exercise 18.4), the point \mathbf{a}_0 is a feasible point of the transformed problem. Hence, assumption A of the preceding subsection is satisfied for the problem above.

We started this discussion with the assumption that we are given a point \mathbf{a} that is a strictly interior feasible point of the original LP problem in standard form. To see how this assumption can be made to hold, we now show that we can transform any given LP problem into an equivalent problem in standard form where such a point \mathbf{a} is explicitly given. To this end, consider a given

LP problem of the form

$$\begin{aligned} & \text{minimize } \mathbf{c}^\top \mathbf{x} \\ & \text{subject to } \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \quad \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Note that any LP problem can be converted into an equivalent problem of the above form. To see this, recall that any LP problem can be transformed into an equivalent problem in standard form. But any problem in standard form can be represented as above, since the constraint $\mathbf{Ax} = \mathbf{b}$ can be written as $\mathbf{Ax} \geq \mathbf{b}$, $-\mathbf{Ax} \geq -\mathbf{b}$. We next write the dual to the problem above:

$$\begin{aligned} & \text{maximize } \boldsymbol{\lambda}^\top \mathbf{b} \\ & \text{subject to } \boldsymbol{\lambda}^\top \mathbf{A} \leq \mathbf{c}^\top \\ & \quad \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

As we did in our discussion of Khachiyan's algorithm, we now combine the primal and dual problems to get

$$\begin{aligned} & \mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \boldsymbol{\lambda} = 0, \\ & \mathbf{Ax} \geq \mathbf{b}, \\ & \mathbf{A}^\top \boldsymbol{\lambda} \leq \mathbf{c}, \\ & \mathbf{x} \geq \mathbf{0}, \\ & \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

As we pointed out in the earlier section on Khachiyan's algorithm, the original LP problem is solved if and only if we can find a pair $(\mathbf{x}, \boldsymbol{\lambda})$ that satisfies the set of relations above. This follows from Theorem 17.1. We now introduce slack and surplus variables \mathbf{u} and \mathbf{v} to get the following equivalent set of relations:

$$\begin{aligned} & \mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \boldsymbol{\lambda} = 0, \\ & \mathbf{Ax} - \mathbf{v} = \mathbf{b}, \\ & \mathbf{A}^\top \boldsymbol{\lambda} + \mathbf{u} = \mathbf{c}, \\ & \mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, \mathbf{v} \geq \mathbf{0}. \end{aligned}$$

Let $\mathbf{x}_0 \in \mathbb{R}^n$, $\boldsymbol{\lambda}_0 \in \mathbb{R}^m$, $\mathbf{u}_0 \in \mathbb{R}^n$, and $\mathbf{v}_0 \in \mathbb{R}^m$ be points that satisfy $\mathbf{x}_0 > \mathbf{0}$, $\boldsymbol{\lambda}_0 > \mathbf{0}$, $\mathbf{u}_0 > \mathbf{0}$, and $\mathbf{v}_0 > \mathbf{0}$. For example, we could choose $\mathbf{x}_0 = [1, \dots, 1]^\top$, and likewise with $\boldsymbol{\lambda}_0$, \mathbf{u}_0 , and \mathbf{v}_0 . Consider the LP problem

$$\begin{aligned} & \text{minimize } z \\ & \text{subject to } \mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \boldsymbol{\lambda} + (-\mathbf{c}^\top \mathbf{x}_0 + \mathbf{b}^\top \boldsymbol{\lambda}_0)z = 0 \\ & \quad \mathbf{Ax} - \mathbf{v} + (\mathbf{b} - \mathbf{Ax}_0 + \mathbf{v}_0)z = \mathbf{b} \\ & \quad \mathbf{A}^\top \boldsymbol{\lambda} + \mathbf{u} + (\mathbf{c} - \mathbf{A}^\top \boldsymbol{\lambda}_0)z = \mathbf{c} \\ & \quad \mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, \mathbf{v}, z \geq \mathbf{0}. \end{aligned}$$

We refer to the above as the *Karmarkar's artificial problem*, which can be represented in matrix notation as

$$\begin{aligned} & \text{minimize} && \tilde{\mathbf{c}}^\top \tilde{\mathbf{x}} \\ & \text{subject to} && \tilde{\mathbf{A}} \tilde{\mathbf{x}} = \tilde{\mathbf{b}} \\ & && \tilde{\mathbf{x}} \geq \mathbf{0}, \end{aligned}$$

where

$$\tilde{\mathbf{x}} = [\mathbf{x}^\top, \boldsymbol{\lambda}^\top, \mathbf{u}^\top, \mathbf{v}^\top, z]^\top,$$

$$\tilde{\mathbf{c}} = [\mathbf{0}_{2m+2n}^\top, 1]^\top,$$

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{c}^\top & -\mathbf{b}^\top & \mathbf{0}_n^\top & \mathbf{0}_m^\top & (-\mathbf{c}^\top \mathbf{x}_0 + \mathbf{b}^\top \boldsymbol{\lambda}_0) \\ \mathbf{A} & \mathbf{O}_{m \times m} & \mathbf{O}_{m \times n} & -\mathbf{I}_m & (\mathbf{b} - \mathbf{A}\mathbf{x}_0 + \mathbf{v}_0) \\ \mathbf{O}_{n \times n} & \mathbf{A}^\top & \mathbf{I}_n & \mathbf{O}_{n \times m} & (\mathbf{c} - \mathbf{A}^\top \boldsymbol{\lambda}_0) \end{bmatrix}, \quad \tilde{\mathbf{b}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix}$$

(the subscripts above refer to the dimensions/sizes of the corresponding matrices/vectors). Observe that the following point is a strictly interior feasible point for the problem above:

$$\begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \\ \mathbf{u} \\ \mathbf{v} \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0 \\ \boldsymbol{\lambda}_0 \\ \mathbf{u}_0 \\ \mathbf{v}_0 \\ 1 \end{bmatrix}.$$

Furthermore, the minimum value of the objective function for Karmarkar's artificial problem is zero if and only if the previous set of relations has a solution, that is, there exists \mathbf{x} , $\boldsymbol{\lambda}$, \mathbf{u} , and \mathbf{v} satisfying

$$\begin{aligned} \mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \boldsymbol{\lambda} &= 0, \\ \mathbf{A}\mathbf{x} - \mathbf{v} &= \mathbf{b}, \\ \mathbf{A}^\top \boldsymbol{\lambda} + \mathbf{u} &= \mathbf{c}, \\ \mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, \mathbf{v} &\geq \mathbf{0}. \end{aligned}$$

Therefore, Karmarkar's artificial LP problem is equivalent to the original LP problem:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Note that the main difference between the original LP problem and Karmarkar's artificial problem is that we have an explicit strictly interior feasible point for Karmarkar's artificial problem, and hence we have satisfied the assumption that we imposed at the beginning of this subsection.

The Algorithm

We are now ready to describe Karmarkar's algorithm. Keep in mind that the LP problem we are solving is a Karmarkar's restricted problem, that is, a problem in Karmarkar's canonical form and satisfies assumptions A, B, C, and D. For convenience, we restate the problem:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n \\ & \text{subject to} && \mathbf{x} \in \Omega \cap \Delta, \end{aligned}$$

where $\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{0}\}$ and $\Delta = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{e}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}$. Karmarkar's algorithm is an iterative algorithm that, given an initial point $\mathbf{x}^{(0)}$ and parameter q , generates a sequence $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}$. Karmarkar's algorithm is described by the following steps:

1. **Initialize:** Set $k := 0$; $\mathbf{x}^{(0)} = \mathbf{a}_0 = \mathbf{e}/n$.
2. **Update:** Set $\mathbf{x}^{(k+1)} = \Psi(\mathbf{x}^{(k)})$, where Ψ is an update map described below.
3. **Check the stopping criterion:** If the condition $\mathbf{c}^\top \mathbf{x}^{(k)}/\mathbf{c}^\top \mathbf{x}^{(0)} \leq 2^{-q}$ is satisfied, then stop.
4. **Iterate:** Set $k := k + 1$; go to step 2.

We describe the update map Ψ as follows. First, consider the first step in the algorithm: $\mathbf{x}^{(0)} = \mathbf{a}_0$. To compute $\mathbf{x}^{(1)}$, we use the familiar update equation

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha \mathbf{d}^{(0)},$$

where α is a step size and $\mathbf{d}^{(0)}$ is an update direction. The step size α is chosen to be a value in $(0, 1)$. Karmarkar recommends a value of $1/4$ in his original paper [71]. The update direction $\mathbf{d}^{(0)}$ is chosen as follows. First, note that the gradient of the objective function is \mathbf{c} . Therefore, the direction of maximum rate of decrease of the objective function is $-\mathbf{c}$. However, in general, we cannot simply update along this direction, since $\mathbf{x}^{(1)}$ is required to lie in the constraint set

$$\begin{aligned} \Omega \cap \Delta &= \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{0}, \mathbf{e}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\} \\ &= \left\{ \mathbf{x} \in \mathbb{R}^n : \begin{bmatrix} \mathbf{A} \\ \mathbf{e}^\top \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \mathbf{x} \geq \mathbf{0} \right\} \\ &= \left\{ \mathbf{x} \in \mathbb{R}^n : \mathbf{B}_0 \mathbf{x} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \mathbf{x} \geq \mathbf{0} \right\}, \end{aligned}$$

where $\mathbf{B}_0 \in \mathbb{R}^{(m+1) \times n}$ is given by

$$\mathbf{B}_0 = \begin{bmatrix} \mathbf{A} \\ \mathbf{e}^\top \end{bmatrix}.$$

Note that since $\mathbf{x}^{(0)} \in \Omega \cap \Delta$, then for $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha \mathbf{d}^{(0)}$ also to lie in $\Omega \cap \Delta$, the vector $\mathbf{d}^{(0)}$ must be an element of the nullspace of \mathbf{B}_0 . Hence, we choose $\mathbf{d}^{(0)}$ to be in the direction of the orthogonal projection of $-\mathbf{c}$ onto the nullspace of \mathbf{B}_0 . This projection is accomplished by the matrix \mathbf{P}_0 given by

$$\mathbf{P}_0 = \mathbf{I}_n - \mathbf{B}_0^\top (\mathbf{B}_0 \mathbf{B}_0^\top)^{-1} \mathbf{B}_0.$$

Note that $\mathbf{B}_0 \mathbf{B}_0^\top$ is nonsingular by assumption C. Specifically, we choose $\mathbf{d}^{(0)}$ to be the vector $\mathbf{d}^{(0)} = -r \hat{\mathbf{c}}^{(0)}$, where

$$\hat{\mathbf{c}}^{(0)} = \frac{\mathbf{P}_0 \mathbf{c}}{\|\mathbf{P}_0 \mathbf{c}\|}$$

and $r = 1/\sqrt{n(n-1)}$. The scalar r is incorporated into the update vector $\mathbf{d}^{(0)}$ for the following reason. First, observe that r is the radius of the largest sphere inscribed in the simplex Δ (see Exercise 18.7). Therefore, the vector $\mathbf{d}^{(0)} = r \hat{\mathbf{c}}^{(0)}$ points in the direction of the projection $\hat{\mathbf{c}}^{(0)}$ of \mathbf{c} onto the nullspace of \mathbf{B}_0 and $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha \mathbf{d}^{(0)}$ is guaranteed to lie in the constraint set $\Omega \cap \Delta$. In fact, $\mathbf{x}^{(1)}$ lies in the set $\Omega \cap \Delta \cap \{\mathbf{x} : \|\mathbf{x} - \mathbf{a}_0\| \leq r\}$. Finally, we note that $\mathbf{x}^{(1)}$ is a strictly interior point of Δ .

The general update step $\mathbf{x}^{(k+1)} = \Psi(\mathbf{x}^{(k)})$ is performed as follows. We first give a brief description of the basic idea, which is similar to the update from $\mathbf{x}^{(0)}$ to $\mathbf{x}^{(1)}$ described above. However, note that $\mathbf{x}^{(k)}$ is, in general, not at the center of the simplex. Therefore, let us first transform this point to the center. To do this, let \mathbf{D}_k be a diagonal matrix whose diagonal entries are the components of the vector $\mathbf{x}^{(k)}$:

$$\mathbf{D}_k = \begin{bmatrix} x_1^{(k)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x_n^{(k)} \end{bmatrix}.$$

It turns out that because $\mathbf{x}^{(0)}$ is a strictly interior point of Δ , $\mathbf{x}^{(k)}$ is a strictly interior point of Δ for all k (see Exercise 18.10). Therefore, \mathbf{D}_k is nonsingular and

$$\mathbf{D}_k^{-1} = \begin{bmatrix} 1/x_1^{(k)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/x_n^{(k)} \end{bmatrix}.$$

Consider the mapping $\mathbf{U}_k : \Delta \rightarrow \Delta$ given by $\mathbf{U}_k(\mathbf{x}) = \mathbf{D}_k^{-1} \mathbf{x} / \mathbf{e}^\top \mathbf{D}_k^{-1} \mathbf{x}$. Note that $\mathbf{U}_k(\mathbf{x}^{(k)}) = \mathbf{e}/n = \mathbf{a}_0$. We use \mathbf{U}_k to change the variable from

\mathbf{x} to $\bar{\mathbf{x}} = \mathbf{U}_k(\mathbf{x})$. We do this so that $\mathbf{x}^{(k)}$ is mapped into the center of the simplex, as indicated above. Note that \mathbf{U}_k is an invertible mapping, with $\mathbf{x} = \mathbf{U}_k^{-1}(\bar{\mathbf{x}}) = \mathbf{D}_k\bar{\mathbf{x}}/\mathbf{e}^\top \mathbf{D}_k\bar{\mathbf{x}}$. Letting $\bar{\mathbf{x}}^{(k)} = \mathbf{U}_k(\mathbf{x}^{(k)}) = \mathbf{a}_0$, we can now apply the procedure that we described before for getting $\mathbf{x}^{(1)}$ from $\mathbf{x}^{(0)} = \mathbf{a}_0$. Specifically, we update $\bar{\mathbf{x}}^{(k)}$ to obtain $\bar{\mathbf{x}}^{(k+1)}$ using the update formula $\bar{\mathbf{x}}^{(k+1)} = \bar{\mathbf{x}}^{(k)} + \alpha \mathbf{d}^{(k)}$. To compute $\mathbf{d}^{(k)}$, we need to state the original LP problem in the new variable $\bar{\mathbf{x}}$:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{D}_k \bar{\mathbf{x}} \\ & \text{subject to} && \mathbf{A}\mathbf{D}_k \bar{\mathbf{x}} = \mathbf{0} \\ & && \bar{\mathbf{x}} \in \Delta. \end{aligned}$$

The reader can easily verify that the LP problem above in the new variable $\bar{\mathbf{x}}$ is equivalent to the original LP problem in the sense that \mathbf{x}^* is an optimal solution to the original problem if and only if $\mathbf{U}_k(\mathbf{x}^*)$ is an optimal solution to the transformed problem. To see this, simply note that $\bar{\mathbf{x}} = \mathbf{U}_k(\mathbf{x}) = \mathbf{D}_k^{-1}\mathbf{x}/\mathbf{e}^\top \mathbf{D}_k^{-1}\mathbf{x}$, and rewrite the objective function and constraints accordingly (see Exercise 18.8). As before, let

$$\mathbf{B}_k = \begin{bmatrix} \mathbf{A}\mathbf{D}_k \\ \mathbf{e}^\top \end{bmatrix}.$$

We choose $\mathbf{d}^{(k)} = -r\hat{\mathbf{c}}^{(k)}$, where $\hat{\mathbf{c}}^{(k)}$ is the normalized projection of $-(\mathbf{c}^\top \mathbf{D}_k)^\top = -\mathbf{D}_k \mathbf{c}$ onto the nullspace of \mathbf{B}_k , and $r = 1/\sqrt{n(n-1)}$ as before. To determine $\hat{\mathbf{c}}^{(k)}$, we define the projector matrix \mathbf{P}_k by

$$\mathbf{P}_k = \mathbf{I}_n - \mathbf{B}_k^\top (\mathbf{B}_k \mathbf{B}_k^\top)^{-1} \mathbf{B}_k.$$

Note that $\mathbf{B}_k \mathbf{B}_k^\top$ is nonsingular (see Exercise 18.9). The vector $\hat{\mathbf{c}}^{(k)}$ is therefore given by

$$\hat{\mathbf{c}}^{(k)} = \frac{\mathbf{P}_k \mathbf{D}_k \mathbf{c}}{\|\mathbf{P}_k \mathbf{D}_k \mathbf{c}\|}.$$

The direction vector $\mathbf{d}^{(k)}$ is then

$$\mathbf{d}^{(k)} = -r\hat{\mathbf{c}}^{(k)} = -r \frac{\mathbf{P}_k \mathbf{D}_k \mathbf{c}}{\|\mathbf{P}_k \mathbf{D}_k \mathbf{c}\|}.$$

The updated vector $\bar{\mathbf{x}}^{(k+1)} = \bar{\mathbf{x}}^{(k)} + \alpha \mathbf{d}^{(k)}$ is guaranteed to lie in the transformed feasible set $\{\bar{\mathbf{x}} : \mathbf{A}\mathbf{D}_k \bar{\mathbf{x}} = \mathbf{0}\} \cap \Delta$. The final step is to apply the inverse transformation \mathbf{U}_k^{-1} to obtain $\mathbf{x}^{(k+1)}$:

$$\mathbf{x}^{(k+1)} = \mathbf{U}_k^{-1}(\bar{\mathbf{x}}^{(k+1)}) = \frac{\mathbf{D}_k \bar{\mathbf{x}}^{(k+1)}}{\mathbf{e}^\top \mathbf{D}_k \bar{\mathbf{x}}^{(k+1)}}.$$

Note that $\mathbf{x}^{(k+1)}$ lies in the set $\Omega \cap \Delta$. Indeed, we have already seen that \mathbf{U}_k and \mathbf{U}_k^{-1} map Δ into Δ . To see that $\mathbf{A}\mathbf{x}^{(k+1)} = \mathbf{0}$, we simply premultiply the foregoing expression by \mathbf{A} and use the fact that $\mathbf{A}\mathbf{D}_k \bar{\mathbf{x}}^{(k+1)} = \mathbf{0}$.

We now summarize the update $\mathbf{x}^{(k+1)} = \Psi(\mathbf{x}^{(k)})$:

1. Compute the matrices:

$$\mathbf{D}_k = \begin{bmatrix} x_1^{(k)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x_n^{(k)} \end{bmatrix},$$

$$\mathbf{B}_k = \begin{bmatrix} \mathbf{A}\mathbf{D}_k \\ \mathbf{e}^\top \end{bmatrix}.$$

2. Compute the orthogonal projector onto the nullspace of \mathbf{B}_k :

$$\mathbf{P}_k = \mathbf{I}_n - \mathbf{B}_k^\top (\mathbf{B}_k \mathbf{B}_k^\top)^{-1} \mathbf{B}_k.$$

3. Compute the normalized orthogonal projection of \mathbf{c} onto the nullspace of \mathbf{B}_k :

$$\hat{\mathbf{c}}^{(k)} = \frac{\mathbf{P}_k \mathbf{D}_k \mathbf{c}}{\|\mathbf{P}_k \mathbf{D}_k \mathbf{c}\|}.$$

4. Compute the direction vector:

$$\mathbf{d}^{(k)} = -r\hat{\mathbf{c}}^{(k)},$$

where $r = 1/\sqrt{n(n-1)}$.

5. Compute $\bar{\mathbf{x}}^{(k+1)}$ using

$$\bar{\mathbf{x}}^{(k+1)} = \mathbf{a}_0 + \alpha \mathbf{d}^{(k)},$$

where α is the prespecified step size, $\alpha \in (0, 1)$.

6. Compute $\mathbf{x}^{(k+1)}$ by applying the inverse transformation \mathbf{U}_k^{-1} :

$$\mathbf{x}^{(k+1)} = \mathbf{U}_k^{-1}(\bar{\mathbf{x}}^{(k+1)}) = \frac{\mathbf{D}_k \bar{\mathbf{x}}^{(k+1)}}{\mathbf{e}^\top \mathbf{D}_k \bar{\mathbf{x}}^{(k+1)}}.$$

The matrix \mathbf{P}_k in step 2 is needed solely for computing $\mathbf{P}_k \mathbf{D}_k \mathbf{c}$ in step 3. In fact, the two steps can be combined in an efficient way without having to compute \mathbf{P}_k explicitly, as follows. We first solve a set of linear equations $\mathbf{B}_k \mathbf{B}_k^\top \mathbf{y} = \mathbf{B}_k \mathbf{D}_k \mathbf{c}$ (for the variable \mathbf{y}), and then compute $\mathbf{P}_k \mathbf{D}_k \mathbf{c}$ using the expression $\mathbf{P}_k \mathbf{D}_k \mathbf{c} = \mathbf{D}_k \mathbf{c} - \mathbf{B}_k^\top \mathbf{y}$.

For more details on Karmarkar's algorithm, see [42], [55], [71], and [124]. For an informal introduction to the algorithm, see [110]. For further reading on other nonsimplex methods in linear programming, see [42], [55], [96], and [119]. A continuous gradient system for solving linear programming problems is discussed in [26]. An interesting three-article series on developments of

the linear programming area before and after 1984 appeared in *SIAM News*, Vol. 22, No. 2, March 1989. The first article in this journal issue contains an account by Wright on recent progress and a history of linear programming from the early 1800s. The second article, by Anstreicher, focuses on interior-point algorithms developed since 1984. Finally in the third article in the series, Monma surveys computational implementations of interior-point methods.

EXERCISES

18.1 Write a simple MATLAB function to implement the affine scaling algorithm. The inputs are \mathbf{c} , \mathbf{A} , \mathbf{b} , and $\mathbf{x}^{(0)}$, where $\mathbf{x}^{(0)}$ is a strictly feasible initial point. Test the function on the problem in Example 16.2; use $\mathbf{x}^{(0)} = [2, 3, 2, 3, 3]^\top$.

18.2 Write a MATLAB routine that implements the two-phase affine scaling method. It may be useful to use the MATLAB function of Exercise 18.1. Test the routine on the problem in Example 16.5.

18.3 For a given linear programming problem of the form

$$\begin{aligned} &\text{minimize} \quad \mathbf{c}^\top \mathbf{x} \\ &\text{subject to} \quad \mathbf{Ax} \geq \mathbf{b} \\ &\quad \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

the associated Karmarkar's artificial problem can be solved directly using the affine scaling method. Write a simple MATLAB program to solve problems of the form above by using the affine scaling algorithm applied to the associated Karmarkar's artificial problem. It may be useful to use the MATLAB function of Exercise 18.1. Test your program on the problem in Example 15.15.

18.4 Let $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{a} > \mathbf{0}$. Let $\mathbf{T} = [T_1, \dots, T_{n+1}]$ be the projective transformation of the positive orthant P_+ of \mathbb{R}^n into the simplex Δ in \mathbb{R}^{n+1} , given by

$$T_i(\mathbf{x}) = \begin{cases} \frac{x_i/a_i}{x_1/a_1 + \dots + x_n/a_n + 1} & \text{if } 1 \leq i \leq n \\ \frac{1}{x_1/a_1 + \dots + x_n/a_n + 1} & \text{if } i = n+1. \end{cases}$$

Prove the following properties of \mathbf{T} (see [71]):

1. \mathbf{T} is a one-to-one mapping; that is, $\mathbf{T}(\mathbf{x}) = \mathbf{T}(\mathbf{y})$ implies that $\mathbf{x} = \mathbf{y}$.
2. \mathbf{T} maps P_+ onto $\Delta \setminus \{\mathbf{x} : x_{n+1} = 0\} \triangleq \{\mathbf{x} \in \Delta : x_{n+1} > 0\}$; that is, for each $\mathbf{y} \in \{\mathbf{x} \in \Delta : x_{n+1} > 0\}$, there exists $\mathbf{x} \in P_+$ such that $\mathbf{y} = \mathbf{T}(\mathbf{x})$.
3. The inverse transformation of \mathbf{T} exists on $\{\mathbf{x} \in \Delta : x_{n+1} > 0\}$ and is given by $\mathbf{T}^{-1} = [T_1^{-1}, \dots, T_n^{-1}]^\top$, with $T_i^{-1}(\mathbf{y}) = a_i y_i / y_{n+1}$.

4. \mathbf{T} maps \mathbf{a} to the center of the simplex Δ , that is, $\mathbf{T}(\mathbf{a}) = \mathbf{e}/(n+1) = [1/(n+1), \dots, 1/(n+1)] \in \mathbb{R}^{n+1}$.
5. Suppose that \mathbf{x} satisfies $\mathbf{A}\mathbf{x} = \mathbf{b}$, and $\mathbf{y} = \mathbf{T}(\mathbf{x})$. Let $\mathbf{x}' = [y_1 a_1, \dots, y_n a_n]^\top$. Then, $\mathbf{A}\mathbf{x}' = \mathbf{b}y_{n+1}$.

18.5 Let \mathbf{T} be the projective transformation in Exercise 18.4 and $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a given matrix. Prove that there exists a matrix $\mathbf{A}' \in \mathbb{R}^{m \times (n+1)}$ such that $\mathbf{A}\mathbf{x} = \mathbf{b}$ if and only if $\mathbf{A}'\mathbf{T}(\mathbf{x}) = \mathbf{0}$.

Hint: Let the i th column of \mathbf{A}' be given by a_i times the i th column of \mathbf{A} , $i = 1, \dots, n$, and the $(n+1)$ th column of \mathbf{A}' be given by $-\mathbf{b}$.

18.6 Let \mathbf{T} be the projective transformation in Exercise 18.4 and $\mathbf{c} \in \mathbb{R}^n$ be a given vector. Prove that there exists a vector $\mathbf{c}' \in \mathbb{R}^{n+1}$ such that $\mathbf{c}^\top \mathbf{x} = 0$ if and only if $\mathbf{c}'^\top \mathbf{T}(\mathbf{x}) = 0$.

Hint: Use property 3 in Exercise 18.4, with the $\mathbf{c}' = [c'_1, \dots, c'_{n+1}]^\top$ given by $c'_i = a_i c_i$, $i = 1, \dots, n$, and $c'_{n+1} = 0$.

18.7 Let $\Delta = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{e}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}$ be the simplex in \mathbb{R}^n , $n > 1$, and let $\mathbf{a}_0 = \mathbf{e}/n$ be its center. A *sphere* of radius r centered at \mathbf{a}_0 is the set $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{a}_0\| \leq r\}$. The sphere is said to be *inscribed in* Δ if $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{a}_0\| = r, \mathbf{e}^\top \mathbf{x} = 1\} \subset \Delta$. Show that the largest such sphere has radius $r = 1/\sqrt{n(n-1)}$.

18.8 Consider the following Karmarkar's restricted problem:

$$\begin{aligned} &\text{minimize} && \mathbf{c}^\top \mathbf{x} \\ &\text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{0} \\ &&& \mathbf{x} \in \Delta. \end{aligned}$$

Let $\mathbf{x}_0 \in \Delta$ be a strictly interior point of Δ , and \mathbf{D} be a diagonal matrix whose diagonal entries are the components of \mathbf{x}_0 . Define the map $\mathbf{U} : \Delta \rightarrow \Delta$ by $\mathbf{U}(\mathbf{x}) = \mathbf{D}^{-1}\mathbf{x}/\mathbf{e}^\top \mathbf{D}^{-1}\mathbf{x}$. Let $\bar{\mathbf{x}} = \mathbf{U}(\mathbf{x})$ represent a change of variable. Show that the following transformed LP problem in the variable $\bar{\mathbf{x}}$,

$$\begin{aligned} &\text{minimize} && \mathbf{c}^\top \mathbf{D}\bar{\mathbf{x}} \\ &\text{subject to} && \mathbf{A}\mathbf{D}\bar{\mathbf{x}} = \mathbf{0} \\ &&& \bar{\mathbf{x}} \in \Delta, \end{aligned}$$

is equivalent to the original LP problem above in the sense that \mathbf{x}^* is an optimal solution to the original problem if and only if $\bar{\mathbf{x}}^* = \mathbf{U}(\mathbf{x}^*)$ is an optimal solution to the transformed problem.

18.9 Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m < n$, and $\Omega = \{\mathbf{x} : \mathbf{Ax} = \mathbf{0}\}$. Suppose that \mathbf{A} satisfies

$$\text{rank} \begin{bmatrix} \mathbf{A} \\ \mathbf{e}^\top \end{bmatrix} = m + 1.$$

Let $\mathbf{x}_0 \in \Delta \cap \Omega$ be a strictly interior point of $\Delta \subset \mathbb{R}^n$ and \mathbf{D} be a diagonal matrix whose diagonal entries are the components of \mathbf{x}_0 . Consider the matrix \mathbf{B} defined by

$$\mathbf{B} = \begin{bmatrix} \mathbf{AD} \\ \mathbf{e}^\top \end{bmatrix}.$$

Show that $\text{rank } \mathbf{B} = m + 1$, and hence \mathbf{BB}^\top is nonsingular.

18.10 Show that in Karmarkar's algorithm, $\mathbf{x}^{(k)}$ is a strictly interior point of Δ .

CHAPTER 19

INTEGER LINEAR PROGRAMMING

19.1 Introduction

This chapter is devoted to linear programs with the additional constraint that the solution components be integers. Such problems are called *integer linear programming (ILP)* (or simply *integer programming*) problems, and arise naturally in many practical situations. For example, in Example 15.1, the decision variables represent production levels, which we allowed to take real values. If production levels correspond to actual numbers of products, then it is natural to impose the constraint that they be integer valued. If we expect solutions that are very large in magnitude, then ignoring the integer constraint might have little practical consequence. However, in cases where the solution is a relatively small integer (on the order of 10, say), then ignoring the integer constraint could lead to dramatically erroneous solutions.

Throughout this section, we use the notation \mathbb{Z} for the set of integers, \mathbb{Z}^n the set of vectors with n integer components, and $\mathbb{Z}^{m \times n}$ the set of $m \times n$ matrices with integer entries. Using this notation, we can express an ILP

problem in following form:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \\ & && \mathbf{x} \in \mathbb{Z}^n. \end{aligned}$$

19.2 Unimodular Matrices

There is a class of ILP problems that can be solved using standard linear programming methods. To proceed, we need some definitions and background results. The reader should recall the definition of a *minor* from Section 2.2.

Definition 19.1 An $m \times n$ integer matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$, $m \leq n$, is *unimodular* if all its nonzero m th-order minors are ± 1 (i.e., either 1 or -1). ■

Unimodular matrices play a special role in the context of linear equations and integer basic solutions. Consider the linear equation $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A} \in \mathbb{Z}^{m \times n}$, $m \leq n$. Let \mathbf{B} be a corresponding basis matrix (an $m \times m$ matrix consisting of m linearly independent columns of \mathbf{A}). Then, the unimodularity of \mathbf{A} is equivalent to $|\det \mathbf{B}| = 1$ for any such \mathbf{B} . The following lemma connects unimodularity with integer basic solutions.

Lemma 19.1 Consider the linear equation $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A} \in \mathbb{Z}^{m \times n}$, $m \leq n$, is unimodular and $\mathbf{b} \in \mathbb{Z}^m$. Then, all basic solutions have integer components. □

Proof. As usual, suppose that the first m columns of \mathbf{A} constitute a basis, and that \mathbf{B} is the invertible $m \times m$ matrix composed of these columns. Then the corresponding basic solution is

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}.$$

Because all the elements of \mathbf{A} are integers, \mathbf{B} is an integer matrix. Moreover, because \mathbf{A} is unimodular, $|\det \mathbf{B}| = 1$. This implies that the inverse \mathbf{B}^{-1} is also an integer matrix (see [62, p. 21]). Therefore, \mathbf{x}^* is an integer vector. ■

Corollary 19.1 Consider the LP constraint

$$\begin{aligned} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where \mathbf{A} is unimodular, $\mathbf{A} \in \mathbb{Z}^{m \times n}$, $m \leq n$, and $\mathbf{b} \in \mathbb{Z}^m$. Then, all basic feasible solutions have integer components. □

Unimodularity allows us to solve ILP problems using the simplex method. Specifically, consider the ILP problem

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \\ & && \mathbf{x} \in \mathbb{Z}^n \end{aligned}$$

where $\mathbf{A} \in \mathbb{Z}^{m \times n}$, $m \leq n$, is unimodular and $\mathbf{b} \in \mathbb{Z}^m$. Then, the corollary above tells us that if we consider the associated LP problem

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

the optimal basic feasible solution is an integer vector. This means that we can apply the simplex method to the LP problem above to obtain a solution to the original ILP problem.

Example 19.1 Consider the following ILP problem:

$$\begin{aligned} & \text{maximize} && 2x_1 + 5x_2 \\ & \text{subject to} && x_1 + x_3 = 4 \\ & && x_2 + x_4 = 6 \\ & && x_1 + x_2 + x_5 = 8 \\ & && x_1, x_2, x_3, x_4, x_5 \geq 0 \\ & && x_1, x_2, x_3, x_4, x_5 \in \mathbb{Z} \end{aligned}$$

We can write this problem in matrix form with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4 \\ 6 \\ 8 \end{bmatrix}.$$

Notice that $\mathbf{b} \in \mathbb{Z}^3$. Moreover, it is easy to check that \mathbf{A} is unimodular. Hence, the ILP problem above can be solved by solving the LP problem

$$\begin{aligned} & \text{maximize} && 2x_1 + 5x_2 \\ & \text{subject to} && x_1 + x_3 = 4 \\ & && x_2 + x_4 = 6 \\ & && x_1 + x_2 + x_5 = 8 \\ & && x_1, x_2, x_3, x_4, x_5 \geq 0. \end{aligned}$$

This was done in Example 16.2 using the simplex method, yielding optimal solution $[2, 6, 2, 0, 0]^\top$, which is an integer vector. ■

In general, when the matrix \mathbf{A} is not unimodular, the simplex method applied to the associated LP problem yields a noninteger optimal solution. However, in some cases, even if \mathbf{A} is not unimodular, the simplex method still produces an integer optimal basic feasible solution. To see this, suppose that we are given $\mathbf{A} \in \mathbb{Z}^{m \times n}$, $m \leq n$, and $\mathbf{b} \in \mathbb{Z}^m$. Note that as long as each $m \times m$ basis matrix \mathbf{B} consisting of columns of \mathbf{A} corresponding to a basic *feasible* solution has the property that $|\det \mathbf{B}| = 1$, we can use the argument in the proof of Lemma 19.1 to conclude that the basic feasible solution is an integer vector. Equivalently, we can draw this conclusion if each basis submatrix \mathbf{B} of \mathbf{A} such that $|\det \mathbf{B}| \neq 1$ corresponds to a *nonfeasible* basic solution. We illustrate this in the following example.

Example 19.2 Consider the ILP problem

$$\begin{aligned} & \text{minimize} && -x_1 - 2x_2 \\ & \text{subject to} && -2x_1 + x_2 + x_3 = 2 \\ & && -x_1 + x_2 + x_4 = 3 \\ & && x_1 + x_5 = 3 \\ & && x_i \geq 0, \quad i = 1, \dots, 5 \\ & && x_i \in \mathbb{Z}, \quad i = 1, \dots, 5. \end{aligned}$$

Can this ILP problem be solved using the simplex method? We can easily verify that the matrix

$$\mathbf{A} = \begin{bmatrix} -2 & 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

is not unimodular. Indeed, it has one (and only one) basis submatrix with determinant other than ± 1 , consisting of the first, fourth, and fifth columns of \mathbf{A} . Indeed, if we write $\mathbf{B} = [\mathbf{a}_1, \mathbf{a}_4, \mathbf{a}_5]$, then $\det \mathbf{B} = -2$. However, a closer examination of this matrix and the vector $\mathbf{b} = [2, 3, 3]^\top$ reveals that the corresponding basic solution is not feasible: $\mathbf{B}^{-1}\mathbf{b} = [-1, 2, 4]^\top$ (which, coincidentally, happens to be an integer vector). Therefore, for this problem, applying the simplex method to the associated LP problem will produce an integer optimal basic feasible solution, which also solves the original ILP problem.

We begin by forming the first tableau,

$$\begin{array}{cccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{b} \\ -2 & 1 & 1 & 0 & 0 & 2 \\ -1 & 1 & 0 & 1 & 0 & 3 \\ 1 & 0 & 0 & 0 & 1 & 3 \\ \mathbf{c}^\top & -1 & -2 & 0 & 0 & 0 & 0 \end{array}$$

We have $r_2 = -2$. Therefore, we introduce \mathbf{a}_2 into the new basis. We calculate the ratios y_{i0}/y_{i2} , $y_{i2} > 0$, to determine the pivot element:

$$\frac{y_{10}}{y_{12}} = \frac{2}{1} \quad \text{and} \quad \frac{y_{20}}{y_{22}} = \frac{3}{1}.$$

We will use y_{12} as the pivot. Performing elementary row operations, we obtain the second tableau,

$$\begin{array}{cccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{b} \\ -2 & 1 & 1 & 0 & 0 & 2 \\ 1 & 0 & -1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 3 \\ \mathbf{r}^\top & -5 & 0 & 2 & 0 & 0 & 4 \end{array}$$

We now have $r_1 = -5 < 0$. Therefore, we introduce \mathbf{a}_1 into the new basis. We next calculate the ratios y_{i0}/y_{i2} , $y_{i2} > 0$, to determine the pivot element:

$$\frac{y_{20}}{y_{21}} = \frac{1}{1} \quad \text{and} \quad \frac{y_{30}}{y_{31}} = \frac{3}{1}.$$

We will use y_{21} as the pivot. Performing row elementary operations, we obtain the third tableau,

$$\begin{array}{cccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{b} \\ 0 & 1 & -1 & 2 & 0 & 4 \\ 1 & 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 2 \\ \mathbf{r}^\top & 0 & 0 & -3 & 5 & 0 & 9 \end{array}$$

We have $r_3 = -3 < 0$. Therefore, we introduce \mathbf{a}_3 into the new basis. We next calculate the ratios y_{i0}/y_{i2} , $y_{i2} > 0$, to determine the pivot element,

$$\frac{y_{30}}{y_{33}} = \frac{2}{1}.$$

We will use y_{33} as the pivot. Performing row elementary operations, we obtain the fourth tableau,

$$\begin{array}{cccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{b} \\ 0 & 1 & 0 & 1 & 1 & 6 \\ 1 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 1 & -1 & 1 & 2 \\ \mathbf{r}^\top & 0 & 0 & 0 & 2 & 3 & 15 \end{array}$$

All reduced cost coefficients are now positive, which means that the current solution is $[3, 6, 2, 0, 0]^\top$. ■

Next, we consider ILP problems of the form

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \\ & && \mathbf{x} \in \mathbb{Z}^n \end{aligned}$$

We have seen in Section 15.5 that we can transform the inequality constraint $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ into standard form by introducing slack variables. Doing so would lead to a new problem in standard form for which the constraint has the form $[\mathbf{A}, \mathbf{I}]\mathbf{y} = \mathbf{b}$ (where the vector \mathbf{y} contains \mathbf{x} and the slack variables). To deal with matrices of the form $[\mathbf{A}, \mathbf{I}]$, we need another definition.

Definition 19.2 An $m \times n$ integer matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is *totally unimodular* if all its nonzero minors are ± 1 . ■

By minors here we mean p th-order minors for $p \leq \min(m, n)$. Equivalently, a matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is totally unimodular if and only if all its square invertible submatrices have determinant ± 1 . By a *submatrix* of \mathbf{A} we mean a matrix obtained by removing some columns and rows of \mathbf{A} . It is easy to see from this definition that if an integer matrix is totally unimodular, then each entry is 0, 1, or -1 . The next proposition relates the total unimodularity of \mathbf{A} with the unimodularity of $[\mathbf{A}, \mathbf{I}]$ (see also Exercise 19.3).

Proposition 19.1 If an $m \times n$ integer matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is totally unimodular, then the matrix $[\mathbf{A}, \mathbf{I}]$ is unimodular. □

Proof. Let \mathbf{A} satisfy the assumptions of the proposition. We will show that any $m \times m$ invertible submatrix of $[\mathbf{A}, \mathbf{I}]$ has determinant ± 1 . We first note that any $m \times m$ invertible submatrix of $[\mathbf{A}, \mathbf{I}]$ that consists only of columns of \mathbf{A} has determinant ± 1 because \mathbf{A} is totally unimodular. Moreover, the $m \times m$ submatrix \mathbf{I} satisfies $\det \mathbf{I} = 1$.

Consider now an $m \times m$ invertible submatrix of $[\mathbf{A}, \mathbf{I}]$ composed of k columns of \mathbf{A} and $m - k$ columns of \mathbf{I} . Without loss of generality, suppose that this submatrix is composed of the last k columns of \mathbf{A} and the first

$m - k$ columns of \mathbf{I} ; that is, the $m \times m$ invertible submatrix is

$$\mathbf{B} = \begin{bmatrix} \mathbf{a}_{n-k+1} & \cdots & \mathbf{a}_n & \mathbf{e}_1 & \cdots & \mathbf{e}_{m-k} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{m-k,k} & \mathbf{I}_{m-k} \\ \mathbf{B}_{k,k} & \mathbf{O} \end{bmatrix},$$

where \mathbf{e}_i is the i th column of the identity matrix. This choice of columns is without loss of generality because we can exchange rows and columns to arrive at this form, and each exchange only changes the sign of the determinant. Moreover, note that $\det \mathbf{B} = \pm \det \mathbf{B}_{k,k}$ (see also Exercises 19.4 and 2.4). Thus, $\mathbf{B}_{k,k}$ is invertible because \mathbf{B} is invertible. Moreover, because $\mathbf{B}_{k,k}$ is a submatrix of \mathbf{A} and \mathbf{A} is totally unimodular, $\det \mathbf{B}_{k,k} = \pm 1$. Hence, $\det \mathbf{B} = \pm 1$ also. Thus any $m \times m$ invertible submatrix of $[\mathbf{A}, \mathbf{I}]$ has determinant ± 1 , which implies that $[\mathbf{A}, \mathbf{I}]$ is unimodular. ■

Combining the result above with Lemma 19.1, we obtain the following corollary.

Corollary 19.2 *Consider the LP constraint*

$$\begin{aligned} [\mathbf{A}, \mathbf{I}]\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is totally unimodular and $\mathbf{b} \in \mathbb{Z}^m$. Then, all basic feasible solutions have integer components. □

Total unimodularity of \mathbf{A} allows us to solve ILP problems of the following form using the simplex method:

$$\begin{aligned} &\text{minimize } \mathbf{c}^\top \mathbf{x} \\ \text{subject to } &\mathbf{Ax} \leq \mathbf{b} \\ &\mathbf{x} \geq \mathbf{0} \\ &\mathbf{x} \in \mathbb{Z}^n \end{aligned}$$

where $\mathbf{b} \in \mathbb{Z}^m$. Specifically, we first consider the associated LP problem

$$\begin{aligned} &\text{minimize } \mathbf{c}^\top \mathbf{x} \\ \text{subject to } &\mathbf{Ax} \leq \mathbf{b} \\ &\mathbf{x} \geq \mathbf{0}. \end{aligned}$$

If \mathbf{A} is totally unimodular, then the corollary above tells us that once we convert this problem into standard form by introducing a slack-variable vector \mathbf{z} ,

$$\begin{aligned} &\text{minimize } \mathbf{c}^\top \mathbf{x} \\ \text{subject to } &[\mathbf{A}, \mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} = \mathbf{b} \\ &\mathbf{x}, \mathbf{z} \geq \mathbf{0}, \end{aligned}$$

the optimal basic feasible solution is an integer vector. This means that we can apply the simplex method to the LP problem above to obtain a solution to the original ILP problem. Note that although we only needed the \mathbf{x} part of the solution to be integer, the slack-variable vector \mathbf{z} is automatically integer for any integer \mathbf{x} , because both \mathbf{A} and \mathbf{b} only contain integers (see also Exercise 19.5).

Example 19.3 Consider the following ILP problem:

$$\begin{aligned} & \text{maximize} && 2x_1 + 5x_2 \\ & \text{subject to} && x_1 \leq 4 \\ & && x_2 \leq 6 \\ & && x_1 + x_2 \leq 8 \\ & && x_1, x_2 \geq 0 \\ & && x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

This problem can be written in the matrix form above with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4 \\ 6 \\ 8 \end{bmatrix}.$$

It is easy to check that \mathbf{A} is totally unimodular. Hence, the ILP problem above can be solved by solving the LP problem

$$\begin{aligned} & \text{maximize} && 2x_1 + 5x_2 \\ & \text{subject to} && x_1 + x_3 = 4 \\ & && x_2 + x_4 = 6 \\ & && x_1 + x_2 + x_5 = 8 \\ & && x_1, x_2, x_3, x_4, x_5 \geq 0, \end{aligned}$$

as was done in Example 16.2. ■

As discussed before, even if $[\mathbf{A}, \mathbf{I}]$ is not unimodular, the simplex algorithm might still yield a solution to the original ILP. In particular, even if \mathbf{A} is not totally unimodular, the method above might still work, as illustrated in the following example.

Example 19.4 Consider the following ILP problem:

$$\begin{aligned} & \text{maximize} && x_1 + 2x_2 \\ & \text{subject to} && -2x_1 + x_2 \leq 2 \\ & && x_1 - x_2 \geq -3 \\ & && x_1 \leq 3 \\ & && x_1 \geq 0, x_2 \geq 0, x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

We first express the given problem in this equivalent form:

$$\begin{aligned} \text{minimize} \quad & -x_1 - 2x_2 \\ \text{subject to} \quad & -2x_1 + x_2 \leq 2 \\ & -x_1 + x_2 \leq 3 \\ & x_1 \leq 3 \\ & x_1 \geq 0, \quad x_2 \geq 0, \quad x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

We next represent the problem above in standard form by introducing slack variables x_3 , x_4 , and x_5 to obtain

$$\begin{aligned} \text{minimize} \quad & -x_1 - 2x_2 \\ \text{subject to} \quad & -2x_1 + x_2 + x_3 = 2 \\ & -x_1 + x_2 + x_4 = 3 \\ & x_1 + x_5 = 3 \\ & x_i \geq 0, \quad i = 1, \dots, 5. \end{aligned}$$

This problem is now of the form in Example 19.2, where the simplex method was used. Recall that the solution is $[3, 6, 2, 0, 0]^\top$. Thus, the solution to the original problem is $\mathbf{x}^* = [3, 6]^\top$.

Note that the matrix

$$\mathbf{A} = \begin{bmatrix} -2 & 1 \\ -1 & 1 \\ 1 & 0 \end{bmatrix}$$

is not totally unimodular, because it has an entry (-2) not equal to 0 , 1 , or -1 . Indeed, the matrix $[\mathbf{A}, \mathbf{I}]$ is not unimodular. However, in this case, the simplex method still produced an optimal solution to the ILP, as explained in Example 19.2. ■

19.3 The Gomory Cutting-Plane Method

In 1958, Ralph E. Gomory [54] proposed a method where noninteger optimal solutions obtained using the simplex method are successively removed from the feasible set by adding constraints that exclude these noninteger solutions from the feasible set. The additional constraints, referred to as *Gomory cuts*, do not eliminate integer feasible solutions from the feasible set. The process is repeated until the optimal solution is an integer vector.

To describe Gomory cuts, we use the *floor* operator, defined next.

Definition 19.3 The floor of a real number, denoted $\lfloor x \rfloor$, is the integer obtained by rounding x toward $-\infty$. ■

For example, $\lfloor 3.4 \rfloor = 3$ and $\lfloor -3.4 \rfloor = -4$.

Consider the ILP problem

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \\ & && \mathbf{x} \in \mathbb{Z}^n. \end{aligned}$$

We begin by applying the simplex method to obtain an optimal basic feasible solution to the LP problem

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

As usual, suppose that the first m columns form the basis for the optimal basic feasible solution. The corresponding canonical augmented matrix is

$$\begin{array}{cccccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_i & \cdots & \mathbf{a}_m & \mathbf{a}_{m+1} & \cdots & \mathbf{a}_n & \mathbf{y}_0 \\ 1 & 0 & \cdots & 0 & \cdots & 0 & y_{1,m+1} & \cdots & y_{1,n} & y_{10} \\ 0 & 1 & \cdots & 0 & \cdots & 0 & y_{2,m+1} & \cdots & y_{2,n} & y_{20} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \cdots & 0 & y_{i,m+1} & \cdots & y_{i,n} & y_{i0} \\ \vdots & \vdots & & \vdots & & \vdots & & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 1 & y_{m,m+1} & \cdots & y_{m,n} & y_{m0} \end{array}$$

Consider the i th component of the optimal basic feasible solution, y_{i0} . Suppose that y_{i0} is not an integer. Note that any feasible vector \mathbf{x} satisfies the equality constraint (taken from the i th row)

$$x_i + \sum_{j=m+1}^n y_{ij}x_j = y_{i0}.$$

We use this equation to derive an additional constraint that would eliminate the current optimal noninteger solution from the feasible set without eliminating any integer feasible solution. To see how, consider the inequality constraint

$$x_i + \sum_{j=m+1}^n \lfloor y_{ij} \rfloor x_j \leq y_{i0}.$$

Because $\lfloor y_{ij} \rfloor \leq y_{ij}$, any $\mathbf{x} \geq \mathbf{0}$ that satisfies the first equality constraint above also satisfies this inequality constraint. Thus, any feasible \mathbf{x} satisfies this inequality constraint. Moreover, for any integer feasible vector \mathbf{x} , the

left-hand side of the inequality constraint is an integer. Therefore, any integer feasible vector \mathbf{x} also satisfies

$$x_i + \sum_{j=m+1}^n \lfloor y_{ij} \rfloor x_j \leq \lfloor y_{i0} \rfloor.$$

Subtracting this inequality from the equation above, we deduce that any integer feasible vector satisfies

$$\sum_{j=m+1}^n (y_{ij} - \lfloor y_{ij} \rfloor) x_j \geq y_{i0} - \lfloor y_{i0} \rfloor.$$

Next, notice that the optimal basic feasible solution above does not satisfy this inequality, because the left-hand side for the optimal basic feasible solution is 0, but the right-hand side is a positive number. Therefore, if we impose the additional inequality constraint above to the original LP problem, the new constraint set would be such that the current optimal basic feasible solution is no longer feasible, but yet every *integer* feasible vector remains feasible. This new constraint is called a *Gomory cut*.

To transform the new LP problem into standard form, we introduce the surplus variable x_{n+1} to obtain the equality constraint

$$\sum_{j=m+1}^n (y_{ij} - \lfloor y_{ij} \rfloor) x_j - x_{n+1} = y_{i0} - \lfloor y_{i0} \rfloor.$$

For convenience, we will also call this equality constraint a *Gomory cut*. By augmenting this equation into \mathbf{A} and \mathbf{b} , or canonical versions of them (e.g., in the form of a simplex tableau), we obtain a new LP problem in standard form. We can then solve the new problem using the simplex method and examine the resulting optimal basic feasible solution. If the solution satisfies the integer constraints, then we are done—this vector gives an optimal solution to the original ILP problem by extracting the appropriate components. If the solution does not satisfy the integer constraints, we introduce another Gomory cut and repeat the process. We call this procedure the *Gomory cutting-plane method*.

Note that in applying the Gomory cutting-plane method, we only need to introduce enough cuts to satisfy the integer constraints for the original ILP problem. The additional variables introduced by slack variables or by the Gomory cuts are not constrained to be integers.

In the following two examples, we illustrate how the Gomory cutting-plane method can be implemented by incorporating Gomory cuts directly into the simplex tableau.

Example 19.5 Consider the following ILP problem¹:

$$\begin{aligned} & \text{maximize} && 3x_1 + 4x_2 \\ & \text{subject to} && \frac{2}{5}x_1 + x_2 \leq 3 \\ & && \frac{2}{5}x_1 - \frac{2}{5}x_2 \leq 1 \\ & && x_1, x_2 \geq 0 \\ & && x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

We first solve the problem graphically. The constraint set Ω for the associated LP problem (without integer constraints) can be found by calculating the extreme points:

$$\mathbf{x}^{(1)} = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top, \quad \mathbf{x}^{(2)} = \begin{bmatrix} \frac{5}{2} & 0 \end{bmatrix}^\top, \quad \mathbf{x}^{(3)} = \begin{bmatrix} 0 & 3 \end{bmatrix}^\top, \quad \mathbf{x}^{(4)} = \begin{bmatrix} \frac{55}{14} & \frac{10}{7} \end{bmatrix}^\top.$$

In Figure 19.1, we show the feasible set Ω . In Figure 19.2, we show the feasible set for the ILP problem, which allows us to solve the problem graphically. The solution is obtained by finding the straight line $f = 3x_1 + 4x_2$ with largest f that passes through a feasible point with integer components. This can be accomplished by first drawing the line $f = 3x_1 + 4x_2$ for $f = 0$ and then gradually increasing the values of f , which corresponds to sliding across the feasible region until the straight line passes through the “last” integer feasible point yielding the largest value of the objective function. From Figure 19.2, we can see that the optimal solution to the ILP problem is $[2, 2]^\top$.

We now solve the problem using the Gomory cutting-plane method. First we represent the associated LP problem in standard form:

$$\begin{aligned} & \text{maximize} && 3x_1 + 4x_2 \\ & \text{subject to} && \frac{2}{5}x_1 + x_2 + x_3 = 3 \\ & && \frac{2}{5}x_1 - \frac{2}{5}x_2 + x_4 = 1 \\ & && x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

Note that we only need the first two components of the solution to be integers. We can start the simplex method because we have an obvious basic feasible solution. The first tableau is

$$\begin{array}{ccccc|c} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{b} \\ \frac{2}{5} & 1 & 1 & 0 & 3 \\ \frac{2}{5} & -\frac{2}{5} & 0 & 1 & 1 \\ \mathbf{c}^\top & -3 & -4 & 0 & 0 \end{array}$$

¹Thanks to David Schvartzman Cohenca for his solution to this problem.

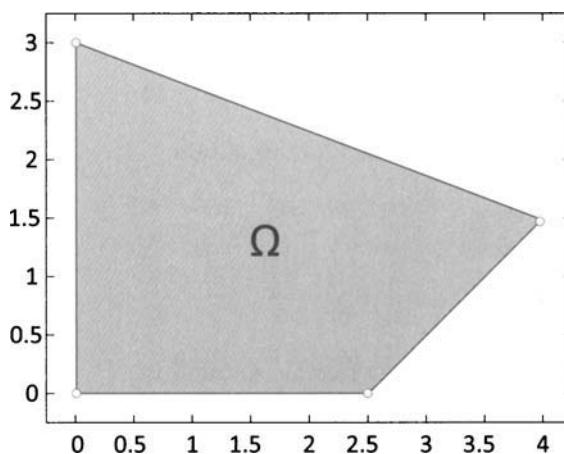


Figure 19.1 Feasible set Ω for LP problem in Example 19.5.

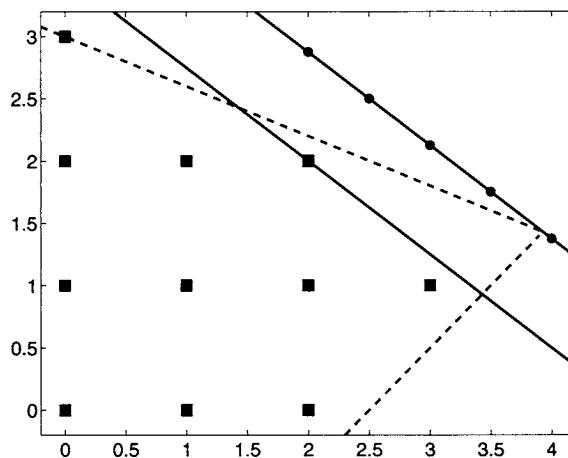


Figure 19.2 Graphical solution for ILP problem in Example 19.5.

We bring a_2 into the basis and pivot about the element (1, 2) to obtain

$$\begin{array}{ccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{b} \\ \frac{2}{5} & 1 & 1 & 0 & 3 \\ \frac{14}{25} & 0 & \frac{2}{5} & 1 & \frac{11}{5} \\ \mathbf{r}^\top & -\frac{7}{5} & 0 & 4 & 0 & 12 \end{array}$$

Next, we pivot about the element (2, 1) to obtain

$$\begin{array}{ccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{b} \\ 0 & 1 & \frac{10}{14} & -\frac{10}{14} & \frac{20}{14} \\ 1 & 0 & \frac{10}{14} & \frac{25}{14} & \frac{55}{14} \\ \mathbf{r}^\top & 0 & 0 & 5 & \frac{5}{2} & \frac{35}{2} \end{array}$$

The corresponding optimal basic feasible solution is

$$\left[\frac{55}{14} \quad \frac{10}{7} \quad 0 \quad 0 \right]^\top,$$

which does not satisfy the integer constraints.

We start by introducing the Gomory cut corresponding to the first row of the tableau. We obtain

$$\frac{10}{14}x_3 + \frac{4}{14}x_4 - x_5 = \frac{6}{14}.$$

We add this constraint to our tableau:

$$\begin{array}{cccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{b} \\ 0 & 1 & \frac{10}{14} & -\frac{10}{14} & 0 & \frac{20}{14} \\ 1 & 0 & \frac{10}{14} & \frac{25}{14} & 0 & \frac{55}{14} \\ 0 & 0 & \frac{10}{14} & \frac{4}{14} & -1 & \frac{6}{14} \\ \mathbf{r}^\top & 0 & 0 & 5 & \frac{5}{2} & 0 & \frac{35}{2} \end{array}$$

Pivoting about the element (3, 3) gives

$$\begin{array}{cccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{b} \\ 0 & 1 & 0 & -1 & 1 & 1 \\ 1 & 0 & 0 & \frac{3}{2} & 1 & \frac{7}{2} \\ 0 & 0 & 1 & \frac{2}{5} & -\frac{7}{5} & \frac{3}{5} \\ \mathbf{r}^\top & 0 & 0 & 0 & \frac{1}{2} & 7 & \frac{29}{2} \end{array}$$

The corresponding optimal basic feasible solution is $[7/2, 1, 3/5, 0, 0]^\top$, which still does not satisfy the integer constraint.

Next, we construct the Gomory cut for the second row of the tableau:

$$\frac{1}{2}x_4 - x_6 = \frac{1}{2}$$

We add this constraint to our tableau to obtain

	a_1	a_2	a_3	a_4	a_5	a_6	b
	0	1	0	-1	1	0	1
	1	0	0	$\frac{3}{2}$	1	0	$\frac{7}{2}$
	0	0	1	$\frac{2}{5}$	$-\frac{7}{5}$	0	$\frac{3}{5}$
	0	0	0	$\frac{1}{2}$	0	-1	$\frac{1}{2}$
r^T	0	0	0	$\frac{1}{2}$	7	0	$\frac{29}{2}$

Pivoting about (4, 4), we get

	a_1	a_2	a_3	a_4	a_5	a_6	b
	0	1	0	0	1	-2	2
	1	0	0	0	1	3	2
	0	0	1	0	$-\frac{7}{5}$	$\frac{4}{5}$	$\frac{1}{5}$
	0	0	0	1	0	-2	1
r^T	0	0	0	0	7	1	14

In this optimal basic feasible solution, the first two components are integers. Thus, we conclude that the solution to our ILP is $[2, 2]^T$, which agrees with the graphical solution in Figure 19.2. ■

In Example 19.5, the final solution to the LP problem after applying the Gomory cutting-plane method is not an integer vector. Only the first two components are integers, as these are the only two components in the original ILP problem. As pointed out earlier, the slack variables and variables introduced by the Gomory cuts are not constrained to be integers. However, if we are given an ILP problem with inequality constraints as in Example 19.5 but with only integer values in constraint data, then the slack variables and those introduced by the Gomory cuts are automatically integer valued (see also Exercise 19.9). We illustrate this in the following example.

Example 19.6 Consider the following ILP problem:

$$\begin{aligned} & \text{maximize} && 3x_1 + 4x_2 \\ & \text{subject to} && 3x_1 - x_2 \leq 12 \\ & && 3x_1 + 11x_2 \leq 66 \\ & && x_1, x_2 \geq 0 \\ & && x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

A graphical solution to this ILP problem is shown in Figure 19.3. As in Example 19.5, the solution is obtained by finding the straight line $f = 3x_1 + 4x_2$ with largest f that passes through a feasible point with integer components. This point is $[5, 4]^T$.

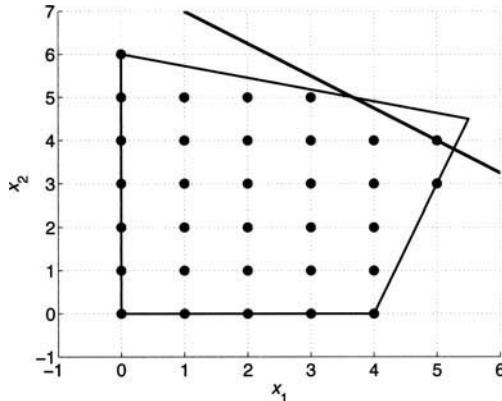


Figure 19.3 Graphical solution of the ILP problem in Example 19.6, where integer feasible solutions are marked with heavy dots.

We now solve the ILP problem above using the simplex method with Gomory cuts. We first represent the associated LP problem in standard form by introducing slack variables x_3 and x_4 . The initial tableau has the form

$$\begin{array}{ccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{b} \\ 3 & -1 & 1 & 0 & 12 \\ 3 & 11 & 0 & 1 & 66 \\ \mathbf{c}^\top & -3 & -4 & 0 & 0 \end{array}$$

In this case there is an obvious initial basic feasible solution available, which allows us to initialize the simplex method to solve the problem. After two iterations of the simplex algorithm, the final tableau is

$$\begin{array}{ccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{b} \\ 1 & 0 & \frac{11}{36} & \frac{1}{36} & \frac{11}{2} \\ 0 & 1 & -\frac{1}{12} & \frac{1}{12} & \frac{9}{2} \\ \mathbf{r}^\top & 0 & 0 & \frac{7}{12} & \frac{5}{12} & \frac{69}{2} \end{array}$$

with optimal solution

$$\mathbf{x}^* = \left[\frac{11}{2} \quad \frac{9}{2} \quad 0 \quad 0 \right]^\top.$$

Both basic components are noninteger. Let us construct a Gomory cut for the first basic component $x_1^* = 11/2$. From the first row of the tableau, the associated constraint equation is

$$x_1 + \frac{11}{36}x_3 + \frac{1}{36}x_4 = \frac{11}{2}.$$

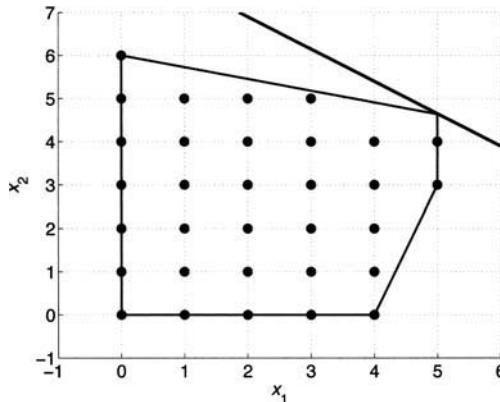


Figure 19.4 Graphical solution of the ILP in Example 19.6 after adding the constraint $x_1 \leq 5$ to the original constraints.

If we apply the floor operator to this equation as explained before, we get an inequality constraint

$$x_1 \leq 5.$$

A graphical solution of the above problem after adding this inequality constraint to the original LP problem is shown in Figure 19.4. We can see that in this new problem, the first component of the optimal solution is an integer, but not the second. This means that a single Gomory cut will not suffice.

To continue with the Gomory procedure for the problem using the simplex method, we first write down the Gomory cut

$$\frac{11}{36}x_3 + \frac{1}{36}x_4 - x_5 = \frac{1}{2}.$$

We now obtain a new tableau by augmenting the previous tableau with the above constraint:

a_1	a_2	a_3	a_4	a_5	b
1	0	$\frac{11}{36}$	$\frac{1}{36}$	0	$\frac{11}{2}$
0	1	$-\frac{1}{12}$	$\frac{1}{12}$	0	$\frac{9}{2}$
0	0	$\frac{11}{36}$	$\frac{1}{36}$	-1	$\frac{1}{2}$
r^T	0	0	$\frac{7}{12}$	$\frac{5}{12}$	0
					$\frac{69}{2}$

At this point, there is no obvious basic feasible solution. However, we can easily use the two-phase method. This yields

a_1	a_2	a_3	a_4	a_5	b
1	0	0	0	1	5
0	1	0	$\frac{1}{11}$	$-\frac{3}{11}$	$\frac{51}{11}$
0	0	1	$\frac{1}{11}$	$-\frac{36}{11}$	$\frac{18}{11}$
r^T	0	0	0	$\frac{4}{11}$	$\frac{21}{11}$
					$\frac{369}{11}$

which has all nonnegative reduced cost coefficients. Hence, we obtain the optimal basic feasible solution

$$\mathbf{x}^* = \begin{bmatrix} 5 & \frac{51}{11} & \frac{18}{11} & 0 & 0 \end{bmatrix}^\top.$$

As expected, the second component does not satisfy the integer constraint.

Next, we write down the Gomory cut for the basic component $x_2^* = 51/11$ using the numbers in the second row of the tableau:

$$\frac{1}{11}x_4 + \frac{8}{11}x_5 - x_6 = \frac{7}{11}.$$

Updating the tableau gives

a_1	a_2	a_3	a_4	a_5	a_6	b
1	0	0	0	1	0	5
0	1	0	$\frac{1}{11}$	$-\frac{3}{11}$	0	$\frac{51}{11}$
0	0	1	$\frac{1}{11}$	$-\frac{36}{11}$	0	$\frac{18}{11}$
0	0	0	$\frac{1}{11}$	$\frac{8}{11}$	-1	$\frac{7}{11}$
r^\top	0	0	0	$\frac{4}{11}$	$\frac{21}{11}$	0
						$\frac{369}{11}$

Again, there is no obvious basic feasible solution. Applying the two-phase method gives

a_1	a_2	a_3	a_4	a_5	a_6	b
1	0	0	$-\frac{1}{8}$	0	$\frac{11}{8}$	$\frac{33}{8}$
0	1	0	$\frac{1}{8}$	0	$-\frac{3}{8}$	$\frac{39}{8}$
0	0	1	$\frac{1}{2}$	0	$-\frac{9}{2}$	$\frac{9}{2}$
0	0	0	$\frac{1}{8}$	1	$-\frac{11}{8}$	$\frac{7}{8}$
r^\top	0	0	0	$\frac{1}{8}$	0	$\frac{21}{8}$
						$\frac{255}{8}$

The corresponding optimal basic feasible solution still does not satisfy the integer constraints; neither the first nor the second components are integer.

Next, we introduce the Gomory cut using the numbers in the second row of the previous tableau to obtain

a_1	a_2	a_3	a_4	a_5	a_6	a_7	b
1	0	0	$-\frac{1}{8}$	0	$\frac{11}{8}$	0	$\frac{33}{8}$
0	1	0	$\frac{1}{8}$	0	$-\frac{3}{8}$	0	$\frac{39}{8}$
0	0	1	$\frac{1}{2}$	0	$-\frac{9}{2}$	0	$\frac{9}{2}$
0	0	0	$\frac{1}{8}$	1	$-\frac{11}{8}$	0	$\frac{7}{8}$
0	0	0	$\frac{1}{8}$	0	$\frac{5}{8}$	-1	$\frac{7}{8}$
r^\top	0	0	0	$\frac{1}{8}$	0	$\frac{21}{8}$	0
							$\frac{255}{8}$

Applying the two-phase method again gives

$$\begin{array}{ccccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{a}_6 & \mathbf{a}_7 & \mathbf{b} \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 & -\frac{1}{2} & 0 & \frac{1}{2} & 4 \\ 0 & 0 & 1 & 0 & -\frac{7}{2} & 0 & \frac{1}{2} & 1 \\ 0 & 0 & 0 & 1 & \frac{5}{2} & 0 & -\frac{11}{2} & 7 \\ 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ \mathbf{r}^\top & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 31 \end{array}$$

(Note that this basic feasible solution is degenerate—the corresponding basis is not unique.) The corresponding optimal basic feasible solution is

$$\begin{bmatrix} 5 & 4 & 1 & 7 & 0 & 0 & 0 \end{bmatrix}^\top,$$

which satisfies the integer constraints. From this, we see that the integer optimal solution to the original ILP problem is $[5, 4]^\top$, which agrees with our graphical solution in Figure 19.3.

In this example, we note that the final solution to LP problem after introducing slack variables and using the Gomory cutting-plane method is an integer vector. The reason for this, in contrast with Example 19.5, is that the original ILP inequality constraint data has only integers. ■

A linear programming problem in which not all of the components are required to be integers is called a *mixed integer linear programming (MILP)* problem. Gomory cuts are also relevant to solving MILP problems. In fact, Example 19.5 is an instance of an MILP problem, because the slack variables in the standard form of the problem are not constrained to be integers. Moreover, the cutting-plane idea also has been applied to nonsimplex methods and nonlinear programming algorithms.

For other methods for solving ILPs, see [119].

EXERCISES

- 19.1** Show that if \mathbf{A} is totally unimodular, then so is any submatrix of it.
- 19.2** Show that if \mathbf{A} is totally unimodular, then so is \mathbf{A}^\top .
- 19.3** Show that \mathbf{A} is totally unimodular if and only if $[\mathbf{A}, \mathbf{I}]$ is totally unimodular. This result is stronger than Proposition 19.1.
- 19.4** Consider the matrix \mathbf{B} in the proof of Proposition 19.1:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{m-k,k} & \mathbf{I}_{m-k} \\ \mathbf{B}_{k,k} & \mathbf{O} \end{bmatrix}.$$

Show that $\det \mathbf{B} = \pm \det \mathbf{B}_{k,k}$.

19.5 Consider the constraint

$$\begin{aligned}\mathbf{A}\mathbf{x} &\leq \mathbf{b}, \\ \mathbf{x} &\in \mathbb{Z}^n,\end{aligned}$$

where \mathbf{A} and \mathbf{b} contain only integers. Suppose that we introduce the slack-variable vector \mathbf{z} to obtain the equivalent constraint

$$\begin{aligned}[\mathbf{A}, \mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} &= \mathbf{b} \\ \mathbf{x} &\in \mathbb{Z}^n \\ \mathbf{z} &\geq \mathbf{0}.\end{aligned}$$

Show that if \mathbf{z} satisfies this constraint (with some \mathbf{x}), then \mathbf{z} is an integer vector.

19.6 Write a MATLAB program to generate Figures 19.1 and 19.2.

19.7 Consider the constraint in standard form $\mathbf{Ax} = \mathbf{b}$. Suppose that we augment this with a Gomory cut to obtain

$$\bar{\mathbf{A}} \begin{bmatrix} \mathbf{x} \\ x_{n+1} \end{bmatrix} = \bar{\mathbf{b}}.$$

Let x_{n+1} satisfy this constraint with an integer vector \mathbf{x} . Show that x_{n+1} is an integer.

19.8 Consider the ILP problem in standard form

$$\begin{aligned}&\text{minimize } \mathbf{c}^\top \mathbf{x} \\ &\text{subject to } \mathbf{Ax} = \mathbf{b} \\ &\quad \mathbf{x} \geq \mathbf{0} \\ &\quad \mathbf{x} \in \mathbb{Z}^n.\end{aligned}$$

Show that if we use the Gomory cutting-plane method with the simplex algorithm, then the final optimal basic feasible solution, including the variables introduced by the Gomory method, is an integer vector. (Use Exercise 19.7.)

19.9 Consider the ILP problem

$$\begin{aligned}&\text{minimize } \mathbf{c}^\top \mathbf{x} \\ &\text{subject to } \mathbf{Ax} \leq \mathbf{b} \\ &\quad \mathbf{x} \geq \mathbf{0} \\ &\quad \mathbf{x} \in \mathbb{Z}^n.\end{aligned}$$

Suppose that we introduce slack variables to convert the problem into standard form, and we use the Gomory cutting-plane method with the simplex algorithm to solve the resulting problem. Show that the final optimal basic feasible solution, including the slack variables and the variables introduced by the Gomory method, is an integer vector. (Use Exercises 19.5 and 19.8.)

- 19.10** Use a graphical method to find an integer solution to the dual of the ILP problem in Example 19.5.

PART IV

NONLINEAR CONSTRAINED OPTIMIZATION

CHAPTER 20

PROBLEMS WITH EQUALITY CONSTRAINTS

20.1 Introduction

In this part we discuss methods for solving a class of nonlinear constrained optimization problems that can be formulated as

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && h_i(\mathbf{x}) = 0, \quad i = 1, \dots, m \\ & && g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, p, \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$, and $m \leq n$. In vector notation, the problem above can be represented in the following *standard form*:

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & && \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \end{aligned}$$

where $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^p$. As usual, we adopt the following terminology.

Definition 20.1 Any point satisfying the constraints is called a *feasible point*. The set of all feasible points,

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\},$$

is called a *feasible set*. ■

Optimization problems of the above form are not new to us. Indeed, linear programming problems of the form

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

which we studied in Part III, are of this type.

As we remarked in Part II, there is no loss of generality by considering only minimization problems. For if we are confronted with a maximization problem, it can easily be transformed into the minimization problem by observing that

$$\text{maximize } f(\mathbf{x}) = \text{minimize } -f(\mathbf{x}).$$

We illustrate the problems we study in this part by considering the following simple numerical example.

Example 20.1 Consider the following optimization problem:

$$\begin{aligned} & \text{minimize} && (x_1 - 1)^2 + x_2 - 2 \\ & \text{subject to} && x_2 - x_1 = 1, \\ & && x_1 + x_2 \leq 2. \end{aligned}$$

This problem is already in the standard form given earlier, with $f(x_1, x_2) = (x_1 - 1)^2 + x_2 - 2$, $\mathbf{h}(x_1, x_2) = x_2 - x_1 - 1$, and $\mathbf{g}(x_1, x_2) = x_1 + x_2 - 2$. This problem turns out to be simple enough to be solved graphically (see Figure 20.1). In the figure the set of points that satisfy the constraints (the feasible set) is marked by the heavy solid line. The inverted parabolae represent level sets of the objective function f —the lower the level set, the smaller the objective function value. Therefore, the solution can be obtained by finding the lowest-level set that intersects the feasible set. In this case, the minimizer lies on the level set with $f = -1/4$. The minimizer of the objective function is $\mathbf{x}^* = [1/2, 3/2]^\top$. ■

In the remainder of this chapter we discuss constrained optimization problems with only equality constraints. The general constrained optimization problem is discussed in the chapters to follow.

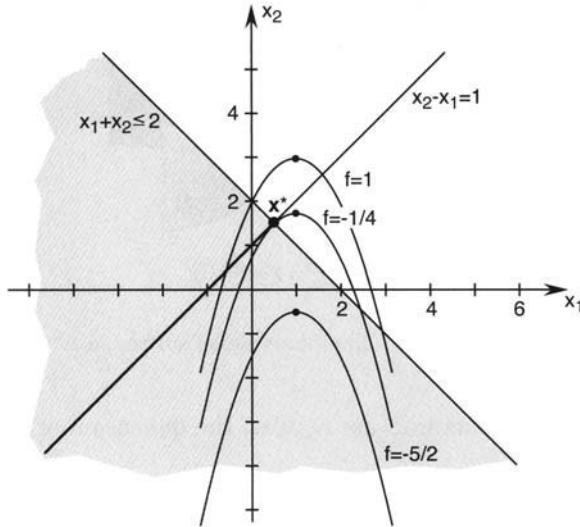


Figure 20.1 Graphical solution to the problem in Example 20.1.

20.2 Problem Formulation

The class of optimization problems we analyze in this chapter is

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0}, \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathbf{h} = [h_1, \dots, h_m]^\top$, and $m \leq n$. We assume that the function \mathbf{h} is continuously differentiable, that is, $\mathbf{h} \in \mathcal{C}^1$.

We introduce the following definition.

Definition 20.2 A point \mathbf{x}^* satisfying the constraints $h_1(\mathbf{x}^*) = 0, \dots, h_m(\mathbf{x}^*) = 0$ is said to be a *regular point* of the constraints if the gradient vectors $\nabla h_1(\mathbf{x}^*), \dots, \nabla h_m(\mathbf{x}^*)$ are linearly independent. ■

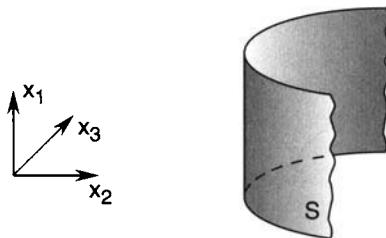
Let $D\mathbf{h}(\mathbf{x}^*)$ be the Jacobian matrix of $\mathbf{h} = [h_1, \dots, h_m]^\top$ at \mathbf{x}^* , given by

$$D\mathbf{h}(\mathbf{x}^*) = \begin{bmatrix} D h_1(\mathbf{x}^*) \\ \vdots \\ D h_m(\mathbf{x}^*) \end{bmatrix} = \begin{bmatrix} \nabla h_1(\mathbf{x}^*)^\top \\ \vdots \\ \nabla h_m(\mathbf{x}^*)^\top \end{bmatrix}.$$

Then, \mathbf{x}^* is regular if and only if $\text{rank } D\mathbf{h}(\mathbf{x}^*) = m$ (i.e., the Jacobian matrix is of full rank).

The set of equality constraints $h_1(\mathbf{x}) = 0, \dots, h_m(\mathbf{x}) = 0$, $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, describes a surface

$$S = \{\mathbf{x} \in \mathbb{R}^n : h_1(\mathbf{x}) = 0, \dots, h_m(\mathbf{x}) = 0\}.$$



$$S = \{[x_1, x_2, x_3]^\top : x_2 - x_3^2 = 0\}$$

Figure 20.2 Two-dimensional surface in \mathbb{R}^3 .

Assuming that the points in S are regular, the dimension of the surface S is $n - m$.

Example 20.2 Let $n = 3$ and $m = 1$ (i.e., we are operating in \mathbb{R}^3). Assuming that all points in S are regular, the set S is a two-dimensional surface. For example, let

$$h_1(\mathbf{x}) = x_2 - x_3^2 = 0.$$

Note that $\nabla h_1(\mathbf{x}) = [0, 1, -2x_3]^\top$, and hence for any $\mathbf{x} \in \mathbb{R}^3$, $\nabla h_1(\mathbf{x}) \neq \mathbf{0}$. In this case,

$$\dim S = \dim \{\mathbf{x} : h_1(\mathbf{x}) = 0\} = n - m = 2.$$

See Figure 20.2 for a graphical illustration. ■

Example 20.3 Let $n = 3$ and $m = 2$. Assuming regularity, the feasible set S is a one-dimensional object (i.e., a curve in \mathbb{R}^3). For example, let

$$\begin{aligned} h_1(\mathbf{x}) &= x_1, \\ h_2(\mathbf{x}) &= x_2 - x_3^2. \end{aligned}$$

In this case, $\nabla h_1(\mathbf{x}) = [1, 0, 0]^\top$ and $\nabla h_2(\mathbf{x}) = [0, 1, -2x_3]^\top$. Hence, the vectors $\nabla h_1(\mathbf{x})$ and $\nabla h_2(\mathbf{x})$ are linearly independent in \mathbb{R}^3 . Thus,

$$\dim S = \dim \{\mathbf{x} : h_1(\mathbf{x}) = 0, h_2(\mathbf{x}) = 0\} = n - m = 1.$$

See Figure 20.3 for a graphical illustration. ■

20.3 Tangent and Normal Spaces

In this section we discuss the notion of a tangent space and normal space at a point on a surface. We begin by defining a *curve* on a surface S .

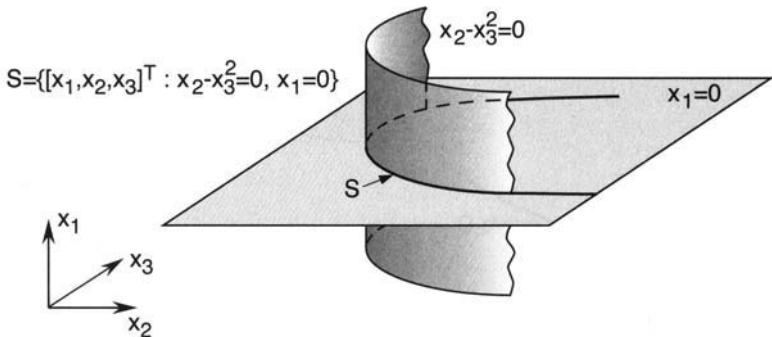


Figure 20.3 One-dimensional surface in \mathbb{R}^3 .

Definition 20.3 A curve C on a surface S is a set of points $\{\mathbf{x}(t) \in S : t \in (a, b)\}$, continuously parameterized by $t \in (a, b)$; that is, $\mathbf{x} : (a, b) \rightarrow S$ is a continuous function. ■

A graphical illustration of the definition of a curve is given in Figure 20.4. The definition of a curve implies that all the points on the curve satisfy the equation describing the surface. The curve C passes through a point \mathbf{x}^* if there exists $t^* \in (a, b)$ such that $\mathbf{x}(t^*) = \mathbf{x}^*$.

Intuitively, we can think of a curve $C = \{\mathbf{x}(t) : t \in (a, b)\}$ as the path traversed by a point \mathbf{x} traveling on the surface S . The position of the point at time t is given by $\mathbf{x}(t)$.

Definition 20.4 The curve $C = \{\mathbf{x}(t) : t \in (a, b)\}$ is differentiable if

$$\dot{\mathbf{x}}(t) = \frac{d\mathbf{x}}{dt}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix}$$

exists for all $t \in (a, b)$.

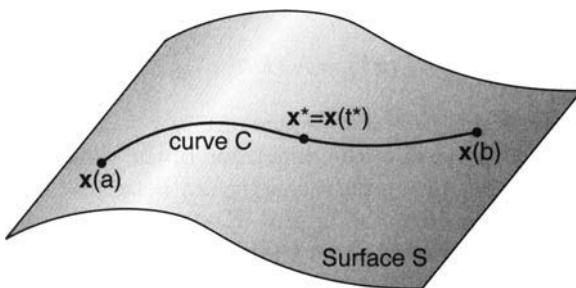


Figure 20.4 Curve on a surface.

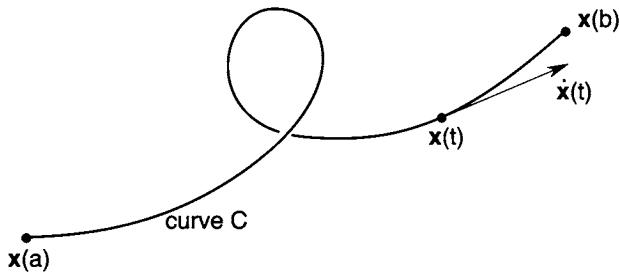


Figure 20.5 Geometric interpretation of the differentiability of a curve.

The curve $C = \{\mathbf{x}(t) : t \in (a, b)\}$ is *twice differentiable* if

$$\ddot{\mathbf{x}}(t) = \frac{d^2\mathbf{x}}{dt^2}(t) = \begin{bmatrix} \ddot{x}_1(t) \\ \vdots \\ \ddot{x}_n(t) \end{bmatrix}$$

exists for all $t \in (a, b)$. ■

Note that both $\dot{\mathbf{x}}(t)$ and $\ddot{\mathbf{x}}(t)$ are n -dimensional vectors. We can think of $\dot{\mathbf{x}}(t)$ and $\ddot{\mathbf{x}}(t)$ as the velocity and acceleration, respectively, of a point traversing the curve C with position $\mathbf{x}(t)$ at time t . The vector $\dot{\mathbf{x}}(t)$ points in the direction of the instantaneous motion of $\mathbf{x}(t)$. Therefore, the vector $\dot{\mathbf{x}}(t^*)$ is *tangent* to the curve C at \mathbf{x}^* (see Figure 20.5).

We are now ready to introduce the notions of a tangent space. For this recall the set

$$S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}\},$$

where $\mathbf{h} \in \mathcal{C}^1$. We think of S as a surface in \mathbb{R}^n .

Definition 20.5 The *tangent space* at a point \mathbf{x}^* on the surface $S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$ is the set $T(\mathbf{x}^*) = \{\mathbf{y} : D\mathbf{h}(\mathbf{x}^*)\mathbf{y} = \mathbf{0}\}$. ■

Note that the tangent space $T(\mathbf{x}^*)$ is the nullspace of the matrix $D\mathbf{h}(\mathbf{x}^*)$:

$$T(\mathbf{x}^*) = \mathcal{N}(D\mathbf{h}(\mathbf{x}^*)).$$

The tangent space is therefore a subspace of \mathbb{R}^n .

Assuming that \mathbf{x}^* is regular, the dimension of the tangent space is $n - m$, where m is the number of equality constraints $h_i(\mathbf{x}^*) = 0$. Note that the tangent space passes through the origin. However, it is often convenient to picture the tangent space as a plane that passes through the point \mathbf{x}^* . For this, we define the *tangent plane* at \mathbf{x}^* to be the set

$$TP(\mathbf{x}^*) = T(\mathbf{x}^*) + \mathbf{x}^* = \{\mathbf{x} + \mathbf{x}^* : \mathbf{x} \in T(\mathbf{x}^*)\}.$$

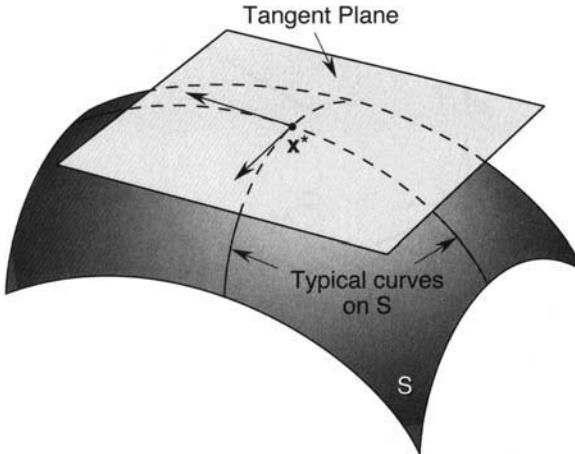


Figure 20.6 Tangent plane to the surface S at the point x^* .

Figure 20.6 illustrates the notion of a tangent plane, and Figure 20.7, the relationship between the tangent plane and the tangent space.

Example 20.4 Let

$$S = \{\mathbf{x} \in \mathbb{R}^3 : h_1(\mathbf{x}) = x_1 = 0, h_2(\mathbf{x}) = x_1 - x_2 = 0\}.$$

Then, S is the x_3 -axis in \mathbb{R}^3 (see Figure 20.8). We have

$$D\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \nabla h_1(\mathbf{x})^\top \\ \nabla h_2(\mathbf{x})^\top \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \end{bmatrix}.$$

Because ∇h_1 and ∇h_2 are linearly independent when evaluated at any $\mathbf{x} \in S$, all the points of S are regular. The tangent space at an arbitrary point of S is

$$\begin{aligned} T(\mathbf{x}) &= \{\mathbf{y} : \nabla h_1(\mathbf{x})^\top \mathbf{y} = 0, \nabla h_2(\mathbf{x})^\top \mathbf{y} = 0\} \\ &= \left\{ \mathbf{y} : \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \mathbf{0} \right\} \\ &= \{[0, 0, \alpha]^\top : \alpha \in \mathbb{R}\} \\ &= \text{the } x_3\text{-axis in } \mathbb{R}^3. \end{aligned}$$

In this example, the tangent space $T(\mathbf{x})$ at any point $\mathbf{x} \in S$ is a one-dimensional subspace of \mathbb{R}^3 . ■

Intuitively, we would expect the definition of the tangent space at a point on a surface to be the collection of all “tangent vectors” to the surface at that

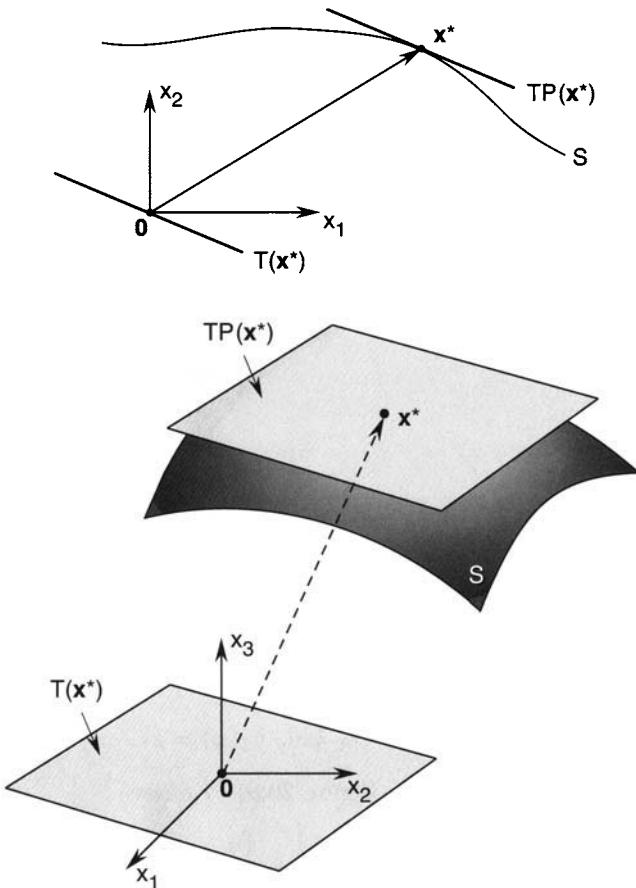


Figure 20.7 Tangent spaces and planes in \mathbb{R}^2 and \mathbb{R}^3 .

point. We have seen that the derivative of a curve on a surface at a point is a tangent vector to the curve, and hence to the surface. The intuition above agrees with our definition whenever x^* is regular, as stated in the theorem below.

Theorem 20.1 Suppose that $x^* \in S$ is a regular point and $T(x^*)$ is the tangent space at x^* . Then, $y \in T(x^*)$ if and only if there exists a differentiable curve in S passing through x^* with derivative y at x^* . \square

Proof. \Leftarrow : Suppose that there exists a curve $\{x(t) : t \in (a, b)\}$ in S such that $x(t^*) = x^*$ and $\dot{x}(t^*) = y$ for some $t^* \in (a, b)$. Then,

$$h(x(t)) = 0$$

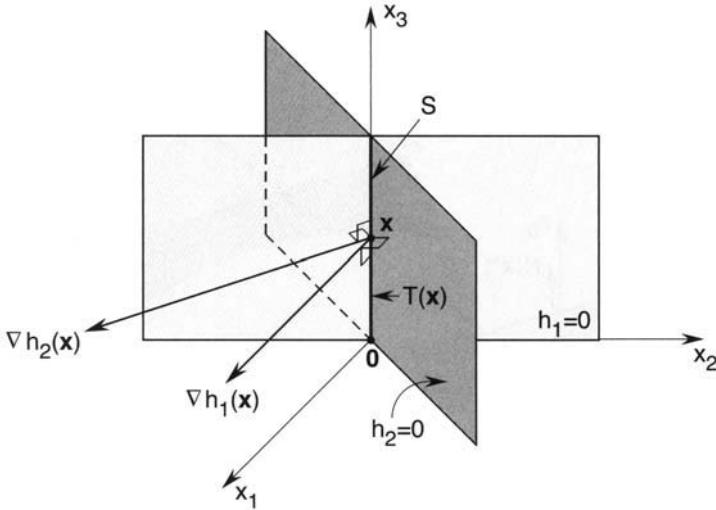


Figure 20.8 The surface $S = \{\mathbf{x} \in \mathbb{R}^3 : x_1 = 0, x_1 - x_2 = 0\}$.

for all $t \in (a, b)$. If we differentiate the function $\mathbf{h}(\mathbf{x}(t))$ with respect to t using the chain rule, we obtain

$$\frac{d}{dt} \mathbf{h}(\mathbf{x}(t)) = D\mathbf{h}(\mathbf{x}(t))\dot{\mathbf{x}}(t) = \mathbf{0}$$

for all $t \in (a, b)$. Therefore, at t^* we get

$$D\mathbf{h}(\mathbf{x}^*)\mathbf{y} = \mathbf{0},$$

and hence $\mathbf{y} \in T(\mathbf{x}^*)$.

\Rightarrow : To prove this, we need to use the implicit function theorem. We refer the reader to [88, p. 325]. ■

We now introduce the notion of a normal space.

Definition 20.6 The *normal space* $N(\mathbf{x}^*)$ at a point \mathbf{x}^* on the surface $S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$ is the set $N(\mathbf{x}^*) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = D\mathbf{h}(\mathbf{x}^*)^\top \mathbf{z}, \mathbf{z} \in \mathbb{R}^m\}$. ■

We can express the normal space $N(\mathbf{x}^*)$ as

$$N(\mathbf{x}^*) = \mathcal{R}(D\mathbf{h}(\mathbf{x}^*)^\top),$$

that is, the range of the matrix $D\mathbf{h}(\mathbf{x}^*)^\top$. Note that the normal space $N(\mathbf{x}^*)$ is the subspace of \mathbb{R}^n spanned by the vectors $\nabla h_1(\mathbf{x}^*), \dots, \nabla h_m(\mathbf{x}^*)$; that is,

$$\begin{aligned} N(\mathbf{x}^*) &= \text{span}[\nabla h_1(\mathbf{x}^*), \dots, \nabla h_m(\mathbf{x}^*)] \\ &= \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = z_1 \nabla h_1(\mathbf{x}^*) + \dots + z_m \nabla h_m(\mathbf{x}^*), z_1, \dots, z_m \in \mathbb{R}\}. \end{aligned}$$

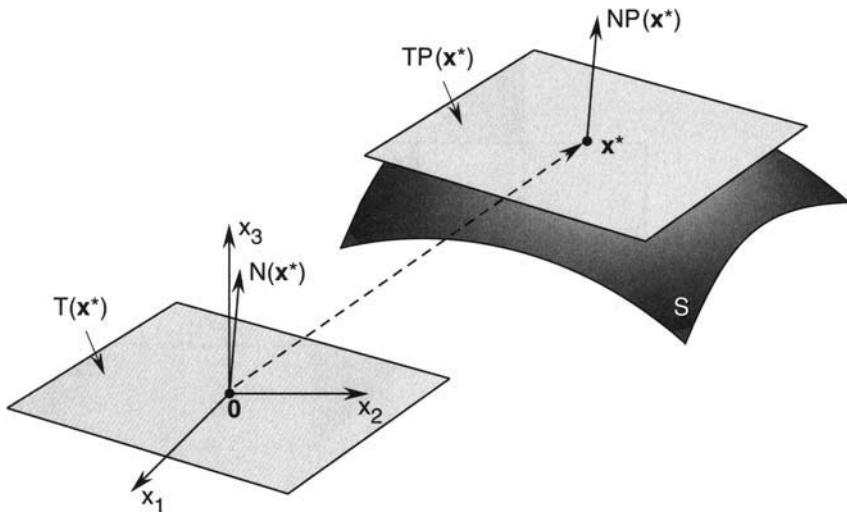


Figure 20.9 Normal space in \mathbb{R}^3 .

Note that the normal space contains the zero vector. Assuming that \mathbf{x}^* is regular, the dimension of the normal space $N(\mathbf{x}^*)$ is m . As in the case of the tangent space, it is often convenient to picture the normal space $N(\mathbf{x}^*)$ as passing through the point \mathbf{x}^* (rather than through the origin of \mathbb{R}^n). For this, we define the *normal plane* at \mathbf{x}^* as the set

$$NP(\mathbf{x}^*) = N(\mathbf{x}^*) + \mathbf{x}^* = \{\mathbf{x} + \mathbf{x}^* \in \mathbb{R}^n : \mathbf{x} \in N(\mathbf{x}^*)\}.$$

Figure 20.9 illustrates the normal space and plane in \mathbb{R}^3 (i.e., $n = 3$ and $m = 1$).

We now show that the tangent space and normal space are orthogonal complements of each other (see Section 3.3).

Lemma 20.1 *We have $T(\mathbf{x}^*) = N(\mathbf{x}^*)^\perp$ and $T(\mathbf{x}^*)^\perp = N(\mathbf{x}^*)$.* □

Proof. By definition of $T(\mathbf{x}^*)$, we may write

$$T(\mathbf{x}^*) = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{x}^\top \mathbf{y} = 0 \text{ for all } \mathbf{x} \in N(\mathbf{x}^*)\}.$$

Hence, by definition of $N(\mathbf{x}^*)$, we have $T(\mathbf{x}^*) = N(\mathbf{x}^*)^\perp$. By Exercise 3.11 we also have $T(\mathbf{x}^*)^\perp = N(\mathbf{x}^*)$. ■

By Lemma 20.1, we can write \mathbb{R}^n as the direct sum decomposition (see Section 3.3):

$$\mathbb{R}^n = N(\mathbf{x}^*) \oplus T(\mathbf{x}^*);$$

that is, given any vector $\mathbf{v} \in \mathbb{R}^n$, there are unique vectors $\mathbf{w} \in N(\mathbf{x}^*)$ and $\mathbf{y} \in T(\mathbf{x}^*)$ such that

$$\mathbf{v} = \mathbf{w} + \mathbf{y}.$$

20.4 Lagrange Condition

In this section we present a first-order necessary condition for extremum problems with constraints. The result is the well-known *Lagrange's theorem*. To better understand the idea underlying this theorem, we first consider functions of two variables and only one equality constraint. Let $h : \mathbb{R}^2 \rightarrow \mathbb{R}$ be the constraint function. Recall that at each point \mathbf{x} of the domain, the gradient vector $\nabla h(\mathbf{x})$ is orthogonal to the level set that passes through that point. Indeed, let us choose a point $\mathbf{x}^* = [x_1^*, x_2^*]^\top$ such that $h(\mathbf{x}^*) = 0$, and assume that $\nabla h(\mathbf{x}^*) \neq \mathbf{0}$. The level set through the point \mathbf{x}^* is the set $\{\mathbf{x} : h(\mathbf{x}) = 0\}$. We then parameterize this level set in a neighborhood of \mathbf{x}^* by a curve $\{\mathbf{x}(t)\}$, that is, a continuously differentiable vector function $\mathbf{x} : \mathbb{R} \rightarrow \mathbb{R}^2$ such that

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad t \in (a, b), \quad \mathbf{x}^* = \mathbf{x}(t^*), \quad \dot{\mathbf{x}}(t^*) \neq \mathbf{0}, \quad t^* \in (a, b).$$

We can now show that $\nabla h(\mathbf{x}^*)$ is orthogonal to $\dot{\mathbf{x}}(t^*)$. Indeed, because h is constant on the curve $\{\mathbf{x}(t) : t \in (a, b)\}$, we have that for all $t \in (a, b)$,

$$h(\mathbf{x}(t)) = 0.$$

Hence, for all $t \in (a, b)$,

$$\frac{d}{dt} h(\mathbf{x}(t)) = 0.$$

Applying the chain rule, we get

$$\frac{d}{dt} h(\mathbf{x}(t)) = \nabla h(\mathbf{x}(t))^\top \dot{\mathbf{x}}(t) = 0.$$

Therefore, $\nabla h(\mathbf{x}^*)$ is orthogonal to $\dot{\mathbf{x}}(t^*)$.

Now suppose that \mathbf{x}^* is a minimizer of $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ on the set $\{\mathbf{x} : h(\mathbf{x}) = 0\}$. We claim that $\nabla f(\mathbf{x}^*)$ is orthogonal to $\dot{\mathbf{x}}(t^*)$. To see this, it is enough to observe that the composite function of t given by

$$\phi(t) = f(\mathbf{x}(t))$$

achieves a minimum at t^* . Consequently, the first-order necessary condition for the unconstrained extremum problem implies that

$$\frac{d\phi}{dt}(t^*) = 0.$$

Applying the chain rule yields

$$0 = \frac{d}{dt} \phi(t^*) = \nabla f(\mathbf{x}(t^*))^\top \dot{\mathbf{x}}(t^*) = \nabla f(\mathbf{x}^*)^\top \dot{\mathbf{x}}(t^*).$$

Thus, $\nabla f(\mathbf{x}^*)$ is orthogonal to $\dot{\mathbf{x}}(t^*)$. The fact that $\dot{\mathbf{x}}(t^*)$ is tangent to the curve $\{\mathbf{x}(t)\}$ at \mathbf{x}^* means that $\nabla f(\mathbf{x}^*)$ is orthogonal to the curve at \mathbf{x}^* (see Figure 20.10).

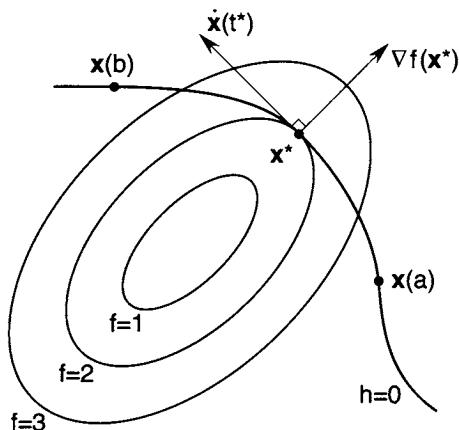


Figure 20.10 The gradient $\nabla f(\mathbf{x}^*)$ is orthogonal to the curve $\{\mathbf{x}(t)\}$ at the point \mathbf{x}^* that is a minimizer of f on the curve.

Recall that $\nabla h(\mathbf{x}^*)$ is also orthogonal to $\dot{\mathbf{x}}(t^*)$. Therefore, the vectors $\nabla h(\mathbf{x}^*)$ and $\nabla f(\mathbf{x}^*)$ are parallel; that is, $\nabla f(\mathbf{x}^*)$ is a scalar multiple of $\nabla h(\mathbf{x}^*)$. The observations above allow us now to formulate *Lagrange's theorem* for functions of two variables with one constraint.

Theorem 20.2 Lagrange's Theorem for $n = 2, m = 1$. Let the point \mathbf{x}^* be a minimizer of $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ subject to the constraint $h(\mathbf{x}) = 0$, $h : \mathbb{R}^2 \rightarrow \mathbb{R}$. Then, $\nabla f(\mathbf{x}^*)$ and $\nabla h(\mathbf{x}^*)$ are parallel. That is, if $\nabla h(\mathbf{x}^*) \neq \mathbf{0}$, then there exists a scalar λ^* such that

$$\nabla f(\mathbf{x}^*) + \lambda^* \nabla h(\mathbf{x}^*) = \mathbf{0}.$$

□

In Theorem 20.2, we refer to λ^* as the *Lagrange multiplier*. Note that the theorem also holds for maximizers. Figure 20.11 gives an illustration of Lagrange's theorem for the case where \mathbf{x}^* is a maximizer of f over the set $\{\mathbf{x} : h(\mathbf{x}) = 0\}$.

Lagrange's theorem provides a first-order necessary condition for a point to be a local minimizer. This condition, which we call the *Lagrange condition*, consists of two equations:

$$\begin{aligned}\nabla f(\mathbf{x}^*) + \lambda^* \nabla h(\mathbf{x}^*) &= \mathbf{0} \\ h(\mathbf{x}^*) &= 0.\end{aligned}$$

Note that the Lagrange condition is necessary but not sufficient. In Figure 20.12 we illustrate a variety of points where the Lagrange condition is

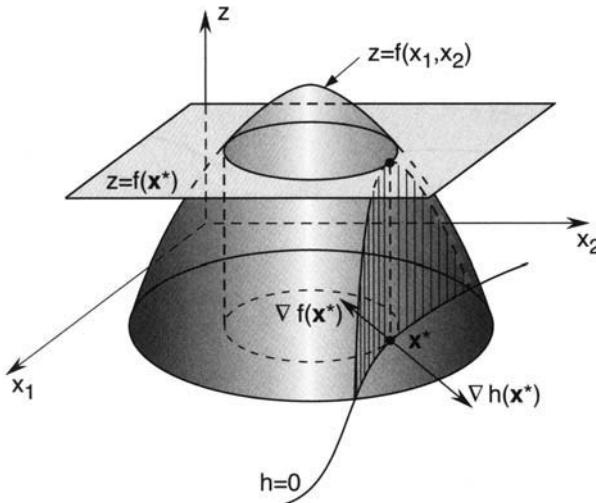


Figure 20.11 Lagrange's theorem for $n = 2, m = 1$.

satisfied, including a case where the point is not an extremizer (neither a maximizer nor a minimizer).

We now generalize Lagrange's theorem for the case when $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m \leq n$.

Theorem 20.3 Lagrange's Theorem. *Let \mathbf{x}^* be a local minimizer (or maximizer) of $f : \mathbb{R}^n \rightarrow \mathbb{R}$, subject to $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m \leq n$. Assume that \mathbf{x}^* is a regular point. Then, there exists $\lambda^* \in \mathbb{R}^m$ such that*

$$Df(\mathbf{x}^*) + \lambda^{*\top} Dh(\mathbf{x}^*) = \mathbf{0}^\top.$$

□

Proof. We need to prove that

$$\nabla f(\mathbf{x}^*) = -D\mathbf{h}(\mathbf{x}^*)^\top \lambda^*$$

for some $\lambda^* \in \mathbb{R}^m$; that is, $\nabla f(\mathbf{x}^*) \in \mathcal{R}(D\mathbf{h}(\mathbf{x}^*)^\top) = N(\mathbf{x}^*)$. But by Lemma 20.1, $N(\mathbf{x}^*) = T(\mathbf{x}^*)^\perp$. Therefore, it remains to show that $\nabla f(\mathbf{x}^*) \in T(\mathbf{x}^*)^\perp$.

We proceed as follows. Suppose that

$$\mathbf{y} \in T(\mathbf{x}^*).$$

Then, by Theorem 20.1, there exists a differentiable curve $\{\mathbf{x}(t) : t \in (a, b)\}$ such that for all $t \in (a, b)$,

$$\mathbf{h}(\mathbf{x}(t)) = \mathbf{0},$$

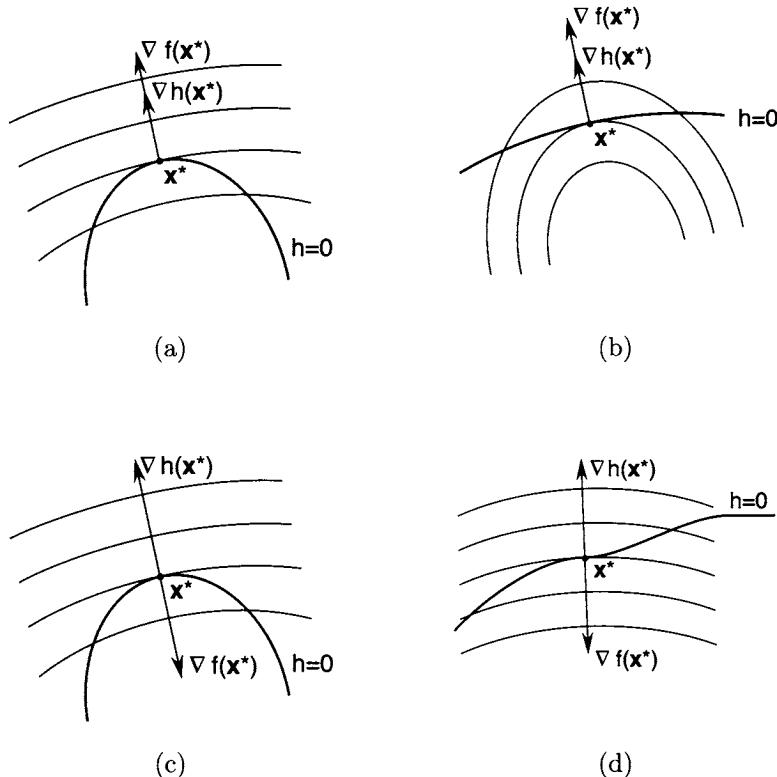


Figure 20.12 Four examples where the Lagrange condition is satisfied: (a) maximizer, (b) minimizer, (c) minimizer, (d) not an extremizer. (Adapted from [120].)

and there exists $t^* \in (a, b)$ satisfying

$$\mathbf{x}(t^*) = \mathbf{x}^*, \quad \dot{\mathbf{x}}(t^*) = \mathbf{y}.$$

Now consider the composite function $\phi(t) = f(\mathbf{x}(t))$. Note that t^* is a local minimizer of this function. By the first-order necessary condition for unconstrained local minimizers (see Theorem 6.1),

$$\frac{d\phi}{dt}(t^*) = 0.$$

Applying the chain rule yields

$$\frac{d\phi}{dt}(t^*) = Df(\mathbf{x}^*) \dot{\mathbf{x}}(t^*) = Df(\mathbf{x}^*) \mathbf{y} = \nabla f(\mathbf{x}^*)^\top \mathbf{y} = 0.$$

So all $\mathbf{y} \in T(\mathbf{x}^*)$ satisfy

$$\nabla f(\mathbf{x}^*)^\top \mathbf{y} = 0;$$

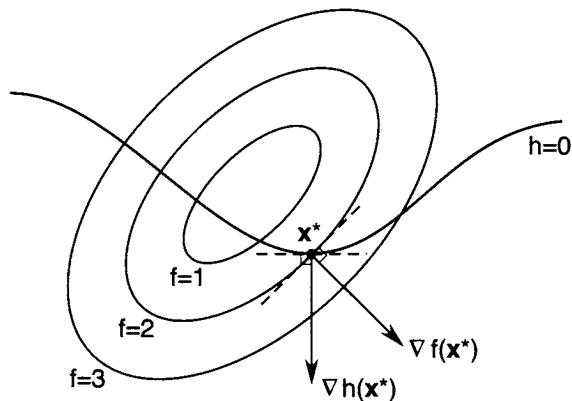


Figure 20.13 Example where the Lagrange condition does not hold.

that is,

$$\nabla f(\mathbf{x}^*) \in T(\mathbf{x}^*)^\perp.$$

This completes the proof. ■

Lagrange's theorem states that if \mathbf{x}^* is an extremizer, then the gradient of the objective function f can be expressed as a linear combination of the gradients of the constraints. We refer to the vector λ^* in Theorem 20.3 as the *Lagrange multiplier vector*, and its components as *Lagrange multipliers*.

From the proof of Lagrange's theorem, we see that a compact way to write the necessary condition is $\nabla f(\mathbf{x}^*) \in N(\mathbf{x}^*)$. If this condition fails, then \mathbf{x}^* cannot be an extremizer. This situation is illustrated in Figure 20.13.

Notice that regularity is stated as an assumption in Lagrange's theorem. This assumption plays an essential role, as illustrated in the following example.

Example 20.5 Consider the following problem:

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && h(x) = 0, \end{aligned}$$

where $f(x) = x$ and

$$h(x) = \begin{cases} x^2 & \text{if } x < 0 \\ 0 & \text{if } 0 \leq x \leq 1 \\ (x - 1)^2 & \text{if } x > 1. \end{cases}$$

The feasible set is evidently $[0, 1]$. Clearly, $x^* = 0$ is a local minimizer. However, $f'(x^*) = 1$ and $h'(x^*) = 0$. Therefore, x^* does not satisfy the

necessary condition in Lagrange's theorem. Note, however, that x^* is not a regular point, which is why Lagrange's theorem does not apply here. ■

It is convenient to introduce the *Lagrangian function* $l : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$, given by

$$l(\mathbf{x}, \boldsymbol{\lambda}) \triangleq f(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{h}(\mathbf{x}).$$

The Lagrange condition for a local minimizer \mathbf{x}^* can be represented using the Lagrangian function as

$$Dl(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}^\top$$

for some $\boldsymbol{\lambda}^*$, where the derivative operation D is with respect to the entire argument $[\mathbf{x}^\top, \boldsymbol{\lambda}^\top]^\top$. In other words, the necessary condition in Lagrange's theorem is equivalent to the first-order necessary condition for unconstrained optimization applied to the Lagrangian function.

To see the above, denote the derivative of l with respect to \mathbf{x} as $D_x l$ and the derivative of l with respect to $\boldsymbol{\lambda}$ as $D_\lambda l$. Then,

$$Dl(\mathbf{x}, \boldsymbol{\lambda}) = [D_x l(\mathbf{x}, \boldsymbol{\lambda}), D_\lambda l(\mathbf{x}, \boldsymbol{\lambda})].$$

Note that $D_x l(\mathbf{x}, \boldsymbol{\lambda}) = Df(\mathbf{x}) + \boldsymbol{\lambda}^\top Dh(\mathbf{x})$ and $D_\lambda l(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{h}(\mathbf{x})^\top$. Therefore, Lagrange's theorem for a local minimizer \mathbf{x}^* can be stated as

$$\begin{aligned} D_x l(\mathbf{x}^*, \boldsymbol{\lambda}^*) &= \mathbf{0}^\top, \\ D_\lambda l(\mathbf{x}^*, \boldsymbol{\lambda}^*) &= \mathbf{0}^\top \end{aligned}$$

for some $\boldsymbol{\lambda}^*$, which is equivalent to

$$Dl(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}^\top.$$

In other words, the Lagrange condition can be expressed as $Dl(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}^\top$.

The Lagrange condition is used to find possible extremizers. This entails solving the equations

$$\begin{aligned} D_x l(\mathbf{x}, \boldsymbol{\lambda}) &= \mathbf{0}^\top, \\ D_\lambda l(\mathbf{x}, \boldsymbol{\lambda}) &= \mathbf{0}^\top. \end{aligned}$$

The above represents $n + m$ equations in $n + m$ unknowns. Keep in mind that the Lagrange condition is necessary but not sufficient; that is, a point \mathbf{x}^* satisfying the equations above need not be an extremizer.

Example 20.6 Given a fixed area of cardboard, we wish to construct a closed cardboard box with maximum volume. We can formulate and solve this problem using the Lagrange condition. Denote the dimensions of the box with maximum volume by x_1 , x_2 , and x_3 , and let the given fixed area of cardboard be A . The problem can then be formulated as

$$\text{maximize } x_1 x_2 x_3$$

$$\text{subject to } x_1 x_2 + x_2 x_3 + x_3 x_1 = \frac{A}{2}.$$

We denote $f(\mathbf{x}) = -x_1x_2x_3$ and $h(\mathbf{x}) = x_1x_2 + x_2x_3 + x_3x_1 - A/2$. We have $\nabla f(\mathbf{x}) = -[x_2x_3, x_1x_3, x_1x_2]^\top$ and $\nabla h(\mathbf{x}) = [x_2 + x_3, x_1 + x_3, x_1 + x_2]^\top$. Note that all feasible points are regular in this case. By the Lagrange condition, the dimensions of the box with maximum volume satisfies

$$\begin{aligned} x_2x_3 - \lambda(x_2 + x_3) &= 0 \\ x_1x_3 - \lambda(x_1 + x_3) &= 0 \\ x_1x_2 - \lambda(x_1 + x_2) &= 0 \\ x_1x_2 + x_2x_3 + x_3x_1 &= \frac{A}{2}, \end{aligned}$$

where $\lambda \in \mathbb{R}$.

We now solve these equations. First, we show that that x_1 , x_2 , x_3 , and λ are all nonzero. Suppose that $x_1 = 0$. By the constraints, we have $x_2x_3 = A/2$. However, the second and third equations in the Lagrange condition yield $\lambda x_2 = \lambda x_3 = 0$, which together with the first equation implies that $x_2x_3 = 0$. This contradicts the constraints. A similar argument applies to x_2 and x_3 .

Next, suppose that $\lambda = 0$. Then, the sum of the three Lagrange equations gives $x_2x_3 + x_1x_3 + x_1x_2 = 0$, which contradicts the constraints.

We now solve for x_1 , x_2 , and x_3 in the Lagrange equations. First, multiply the first equation by x_1 and the second by x_2 , and subtract one from the other. We arrive at $x_3\lambda(x_1 - x_2) = 0$. Because neither x_3 nor λ can be zero (by part b), we conclude that $x_1 = x_2$. We similarly deduce that $x_2 = x_3$. From the constraint equation, we obtain $x_1 = x_2 = x_3 = \sqrt{A/6}$.

Notice that we have ignored the constraints that x_1 , x_2 , and x_3 are positive so that we can solve the problem using Lagrange's theorem. However, there is only one solution to the Lagrange equations, and the solution is positive. Therefore, if a solution exists for the problem with positivity constraints on the variables x_1 , x_2 , and x_3 , then this solution must necessarily be equal to the solution above obtained by ignoring the positivity constraints. ■

Next we provide an example with a quadratic objective function and a quadratic constraint.

Example 20.7 Consider the problem of extremizing the objective function

$$f(\mathbf{x}) = x_1^2 + x_2^2$$

on the ellipse

$$\{[x_1, x_2]^\top : h(\mathbf{x}) = x_1^2 + 2x_2^2 - 1 = 0\}.$$

We have

$$\begin{aligned} \nabla f(\mathbf{x}) &= [2x_1, 2x_2]^\top, \\ \nabla h(\mathbf{x}) &= [2x_1, 4x_2]^\top. \end{aligned}$$

Thus,

$$D_{\mathbf{x}} l(\mathbf{x}, \lambda) = D_{\mathbf{x}} [f(\mathbf{x}) + \lambda h(\mathbf{x})] = [2x_1 + 2\lambda x_1, 2x_2 + 4\lambda x_2]$$

and

$$D_{\lambda} l(\mathbf{x}, \lambda) = h(\mathbf{x}) = x_1^2 + 2x_2^2 - 1.$$

Setting $D_{\mathbf{x}} l(\mathbf{x}, \lambda) = \mathbf{0}^\top$ and $D_{\lambda} l(\mathbf{x}, \lambda) = 0$, we obtain three equations in three unknowns

$$\begin{aligned} 2x_1 + 2\lambda x_1 &= 0, \\ 2x_2 + 4\lambda x_2 &= 0, \\ x_1^2 + 2x_2^2 &= 1. \end{aligned}$$

All feasible points in this problem are regular. From the first of the equations above, we get either $x_1 = 0$ or $\lambda = -1$. For the case where $x_1 = 0$, the second and third equations imply that $\lambda = -1/2$ and $x_2 = \pm 1/\sqrt{2}$. For the case where $\lambda = -1$, the second and third equations imply that $x_1 = \pm 1$ and $x_2 = 0$. Thus, the points that satisfy the Lagrange condition for extrema are

$$\mathbf{x}^{(1)} = \begin{bmatrix} 0 \\ 1/\sqrt{2} \end{bmatrix}, \quad \mathbf{x}^{(2)} = \begin{bmatrix} 0 \\ -1/\sqrt{2} \end{bmatrix}, \quad \mathbf{x}^{(3)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}^{(4)} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}.$$

Because

$$f(\mathbf{x}^{(1)}) = f(\mathbf{x}^{(2)}) = \frac{1}{2}$$

and

$$f(\mathbf{x}^{(3)}) = f(\mathbf{x}^{(4)}) = 1$$

we conclude that if there are minimizers, then they are located at $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, and if there are maximizers, then they are located at $\mathbf{x}^{(3)}$ and $\mathbf{x}^{(4)}$. It turns out that, indeed, $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are minimizers and $\mathbf{x}^{(3)}$ and $\mathbf{x}^{(4)}$ are maximizers. This problem can be solved graphically, as illustrated in Figure 20.14. ■

In the example above, both the objective function f and the constraint function h are quadratic functions. In the next example we take a closer look at a class of problems where both the objective function f and the constraint h are quadratic functions of n variables.

Example 20.8 Consider the following problem:

$$\text{maximize } \frac{\mathbf{x}^\top \mathbf{Q} \mathbf{x}}{\mathbf{x}^\top \mathbf{P} \mathbf{x}},$$

where $\mathbf{Q} = \mathbf{Q}^\top \geq 0$ and $\mathbf{P} = \mathbf{P}^\top > 0$. Note that if a point $\mathbf{x} = [x_1, \dots, x_n]^\top$ is a solution to the problem, then so is any nonzero scalar multiple of it,

$$t\mathbf{x} = [tx_1, \dots, tx_n]^\top, \quad t \neq 0.$$

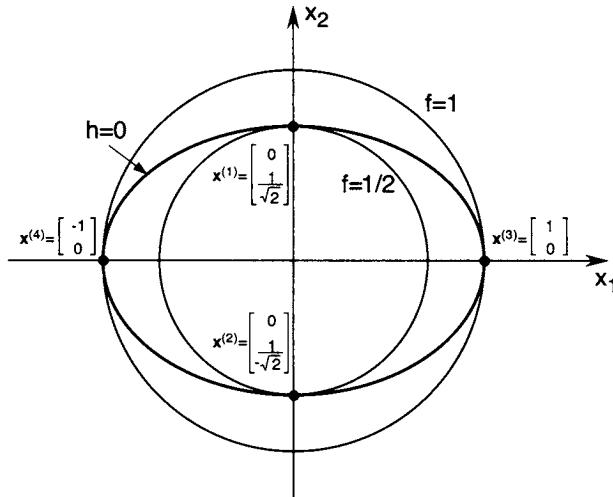


Figure 20.14 Graphical solution of the problem in Example 20.7.

Indeed,

$$\frac{(tx)^\top Q(tx)}{(tx)^\top P(tx)} = \frac{t^2 x^\top Q x}{t^2 x^\top P x} = \frac{x^\top Q x}{x^\top P x}.$$

Therefore, to avoid the multiplicity of solutions, we further impose the constraint

$$x^\top P x = 1.$$

The optimization problem becomes

$$\begin{aligned} & \text{maximize} && x^\top Q x \\ & \text{subject to} && x^\top P x = 1. \end{aligned}$$

Let us write

$$\begin{aligned} f(x) &= x^\top Q x, \\ h(x) &= 1 - x^\top P x. \end{aligned}$$

Any feasible point for this problem is regular (see Exercise 20.13). We now apply Lagrange's method. We first form the Lagrangian function

$$l(x, \lambda) = x^\top Q x + \lambda(1 - x^\top P x).$$

Applying the Lagrange condition yields

$$D_x l(x, \lambda) = 2x^\top Q - 2\lambda x^\top P = 0^\top,$$

$$D_\lambda l(x, \lambda) = 1 - x^\top P x = 0.$$

The first of the equations above can be represented as

$$\mathbf{Q}\mathbf{x} - \lambda\mathbf{P}\mathbf{x} = \mathbf{0}$$

or

$$(\lambda\mathbf{P} - \mathbf{Q})\mathbf{x} = \mathbf{0}.$$

This representation is possible because $\mathbf{P} = \mathbf{P}^\top$ and $\mathbf{Q} = \mathbf{Q}^\top$. By assumption $\mathbf{P} > 0$, hence \mathbf{P}^{-1} exists. Premultiplying $(\lambda\mathbf{P} - \mathbf{Q})\mathbf{x} = \mathbf{0}$ by \mathbf{P}^{-1} , we obtain

$$(\lambda\mathbf{I}_n - \mathbf{P}^{-1}\mathbf{Q})\mathbf{x} = \mathbf{0}$$

or, equivalently,

$$\mathbf{P}^{-1}\mathbf{Q}\mathbf{x} = \lambda\mathbf{x}.$$

Therefore, the solution, if it exists, is an eigenvector of $\mathbf{P}^{-1}\mathbf{Q}$, and the Lagrange multiplier is the corresponding eigenvalue. As usual, let \mathbf{x}^* and λ^* be the optimal solution. Because $\mathbf{x}^{*\top}\mathbf{P}\mathbf{x}^* = 1$ and $\mathbf{P}^{-1}\mathbf{Q}\mathbf{x}^* = \lambda^*\mathbf{x}^*$, we have

$$\lambda^* = \mathbf{x}^{*\top}\mathbf{Q}\mathbf{x}^*.$$

Hence, λ^* is the maximum of the objective function, and therefore is, in fact, the maximal eigenvalue of $\mathbf{P}^{-1}\mathbf{Q}$. It is also called the maximal *generalized eigenvalue*. ■

In the problems above, we are able to find points that are candidates for extremizers of the given objective function subject to equality constraints. These critical points are the only candidates because they are the only points that satisfy the Lagrange condition. To classify such critical points as minimizers, maximizers, or neither, we need a stronger condition—possibly a necessary and sufficient condition. In the next section we discuss a second-order necessary condition and a second-order sufficient condition for minimizers.

20.5 Second-Order Conditions

We assume that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are twice continuously differentiable: $f, \mathbf{h} \in \mathcal{C}^2$. Let

$$l(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{h}(\mathbf{x}) = f(\mathbf{x}) + \lambda_1 h_1(\mathbf{x}) + \cdots + \lambda_m h_m(\mathbf{x})$$

be the Lagrangian function. Let $\mathbf{L}(\mathbf{x}, \boldsymbol{\lambda})$ be the Hessian matrix of $l(\mathbf{x}, \boldsymbol{\lambda})$ with respect to \mathbf{x} :

$$\mathbf{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{F}(\mathbf{x}) + \lambda_1 \mathbf{H}_1(\mathbf{x}) + \cdots + \lambda_m \mathbf{H}_m(\mathbf{x}),$$

where $\mathbf{F}(\mathbf{x})$ is the Hessian matrix of f at \mathbf{x} and $\mathbf{H}_k(\mathbf{x})$ is the Hessian matrix of h_k at \mathbf{x} , $k = 1, \dots, m$, given by

$$\mathbf{H}_k(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 h_k}{\partial x_1^2}(\mathbf{x}) & \cdots & \frac{\partial^2 h_k}{\partial x_n \partial x_1}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial^2 h_k}{\partial x_1 \partial x_n}(\mathbf{x}) & \cdots & \frac{\partial^2 h_k}{\partial x_n^2}(\mathbf{x}) \end{bmatrix}.$$

We introduce the notation $[\boldsymbol{\lambda} \mathbf{H}(\mathbf{x})]$:

$$[\boldsymbol{\lambda} \mathbf{H}(\mathbf{x})] = \lambda_1 \mathbf{H}_1(\mathbf{x}) + \cdots + \lambda_m \mathbf{H}_m(\mathbf{x}).$$

Using the notation above, we can write

$$\mathbf{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{F}(\mathbf{x}) + [\boldsymbol{\lambda} \mathbf{H}(\mathbf{x})].$$

Theorem 20.4 Second-Order Necessary Conditions. Let \mathbf{x}^* be a local minimizer of $f : \mathbb{R}^n \rightarrow \mathbb{R}$ subject to $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m \leq n$, and $f, \mathbf{h} \in \mathcal{C}^2$. Suppose that \mathbf{x}^* is regular. Then, there exists $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ such that:

1. $Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} Dh(\mathbf{x}^*) = \mathbf{0}^\top$.
2. For all $\mathbf{y} \in T(\mathbf{x}^*)$, we have $\mathbf{y}^\top \mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) \mathbf{y} \geq 0$. □

Proof. The existence of $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ such that $Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} Dh(\mathbf{x}^*) = \mathbf{0}^\top$ follows from Lagrange's theorem. It remains to prove the second part of the result. Suppose that $\mathbf{y} \in T(\mathbf{x}^*)$; that is, \mathbf{y} belongs to the tangent space to $S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$ at \mathbf{x}^* . Because $\mathbf{h} \in \mathcal{C}^2$, following the argument of Theorem 20.1, there exists a twice-differentiable curve $\{\mathbf{x}(t) : t \in (a, b)\}$ on S such that

$$\mathbf{x}(t^*) = \mathbf{x}^*, \quad \dot{\mathbf{x}}(t^*) = \mathbf{y}$$

for some $t^* \in (a, b)$. Observe that by assumption, t^* is a local minimizer of the function $\phi(t) = f(\mathbf{x}(t))$. From the second-order necessary condition for unconstrained minimization (see Theorem 6.2), we obtain

$$\frac{d^2\phi}{dt^2}(t^*) \geq 0.$$

Using the formula

$$\frac{d}{dt}(\mathbf{y}(t)^\top \mathbf{z}(t)) = \mathbf{z}(t)^\top \frac{d\mathbf{y}}{dt}(t) + \mathbf{y}(t)^\top \frac{d\mathbf{z}}{dt}(t)$$

and applying the chain rule yields

$$\begin{aligned} \frac{d^2\phi}{dt^2}(t^*) &= \frac{d}{dt}[Df(\mathbf{x}(t^*)) \dot{\mathbf{x}}(t^*)] \\ &= \dot{\mathbf{x}}(t^*)^\top \mathbf{F}(\mathbf{x}^*) \dot{\mathbf{x}}(t^*) + Df(\mathbf{x}^*) \ddot{\mathbf{x}}(t^*) \\ &= \mathbf{y}^\top \mathbf{F}(\mathbf{x}^*) \mathbf{y} + Df(\mathbf{x}^*) \ddot{\mathbf{x}}(t^*) \geq 0. \end{aligned}$$

Because $\mathbf{h}(\mathbf{x}(t)) = \mathbf{0}$ for all $t \in (a, b)$, we have

$$\frac{d^2}{dt^2} \boldsymbol{\lambda}^{*\top} \mathbf{h}(\mathbf{x}(t)) = 0.$$

Thus, for all $t \in (a, b)$,

$$\begin{aligned}
\frac{d^2}{dt^2} \boldsymbol{\lambda}^{*\top} \mathbf{h}(\mathbf{x}(t)) &= \frac{d}{dt} \left[\boldsymbol{\lambda}^{*\top} \frac{d}{dt} \mathbf{h}(\mathbf{x}(t)) \right] \\
&= \frac{d}{dt} \left[\sum_{k=1}^m \lambda_k^* \frac{d}{dt} h_k(\mathbf{x}(t)) \right] \\
&= \frac{d}{dt} \left[\sum_{k=1}^m \lambda_k^* D h_k(\mathbf{x}(t)) \dot{\mathbf{x}}(t) \right] \\
&= \sum_{k=1}^m \lambda_k^* \frac{d}{dt} (D h_k(\mathbf{x}(t)) \dot{\mathbf{x}}(t)) \\
&= \sum_{k=1}^m \lambda_k^* [\dot{\mathbf{x}}(t)^\top \mathbf{H}_k(\mathbf{x}(t)) \dot{\mathbf{x}}(t) + D h_k(\mathbf{x}(t)) \ddot{\mathbf{x}}(t)] \\
&= \dot{\mathbf{x}}^\top(t) [\boldsymbol{\lambda}^* \mathbf{H}(\mathbf{x}(t))] \dot{\mathbf{x}}(t) + \boldsymbol{\lambda}^{*\top} D \mathbf{h}(\mathbf{x}(t)) \ddot{\mathbf{x}}(t) \\
&= 0.
\end{aligned}$$

In particular, the above is true for $t = t^*$; that is,

$$\mathbf{y}^\top [\boldsymbol{\lambda}^* \mathbf{H}(\mathbf{x}^*)] \mathbf{y} + \boldsymbol{\lambda}^{*\top} D \mathbf{h}(\mathbf{x}^*) \ddot{\mathbf{x}}(t^*) = 0.$$

Adding this equation to the inequality

$$\mathbf{y}^\top \mathbf{F}(\mathbf{x}^*) \mathbf{y} + D f(\mathbf{x}^*) \ddot{\mathbf{x}}(t^*) \geq 0$$

yields

$$\mathbf{y}^\top (\mathbf{F}(\mathbf{x}^*) + [\boldsymbol{\lambda}^* \mathbf{H}(\mathbf{x}^*)]) \mathbf{y} + (D f(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} D \mathbf{h}(\mathbf{x}^*)) \ddot{\mathbf{x}}(t^*) \geq 0.$$

But, by Lagrange's theorem, $D f(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} D \mathbf{h}(\mathbf{x}^*) = \mathbf{0}^\top$. Therefore,

$$\mathbf{y}^\top (\mathbf{F}(\mathbf{x}^*) + [\boldsymbol{\lambda}^* \mathbf{H}(\mathbf{x}^*)]) \mathbf{y} = \mathbf{y}^\top \mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) \mathbf{y} \geq 0,$$

which proves the result. ■

Observe that $\mathbf{L}(\mathbf{x}, \boldsymbol{\lambda})$ plays a similar role as the Hessian matrix $\mathbf{F}(\mathbf{x})$ of the objective function f did in the unconstrained minimization case. However, we now require that $\mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) \geq 0$ only on $T(\mathbf{x}^*)$ rather than on \mathbb{R}^n .

The conditions above are necessary, but not sufficient, for a point to be a local minimizer. We now present, without a proof, sufficient conditions for a point to be a strict local minimizer.

Theorem 20.5 Second-Order Sufficient Conditions. Suppose that $f, \mathbf{h} \in C^2$ and there exists a point $\mathbf{x}^* \in \mathbb{R}^n$ and $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ such that:

1. $D f(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} D \mathbf{h}(\mathbf{x}^*) = \mathbf{0}^\top$.

2. For all $\mathbf{y} \in T(\mathbf{x}^*)$, $\mathbf{y} \neq \mathbf{0}$, we have $\mathbf{y}^\top \mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) \mathbf{y} > 0$.

Then, \mathbf{x}^* is a strict local minimizer of f subject to $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. \square

Proof. The interested reader can consult [88, p. 334] for a proof of this result. \blacksquare

Theorem 20.5 states that if an \mathbf{x}^* satisfies the Lagrange condition, and $\mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is positive definite on $T(\mathbf{x}^*)$, then \mathbf{x}^* is a strict local minimizer. A similar result to Theorem 20.5 holds for a strict local maximizer, the only difference being that $\mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ be negative definite on $T(\mathbf{x}^*)$. We illustrate this condition in the following example.

Example 20.9 Consider the following problem:

$$\text{maximize } \frac{\mathbf{x}^\top \mathbf{Q}\mathbf{x}}{\mathbf{x}^\top \mathbf{P}\mathbf{x}},$$

where

$$\mathbf{Q} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}.$$

As pointed out earlier, we can represent this problem in the equivalent form

$$\begin{aligned} &\text{maximize } \mathbf{x}^\top \mathbf{Q}\mathbf{x} \\ &\text{subject to } \mathbf{x}^\top \mathbf{P}\mathbf{x} = 1. \end{aligned}$$

The Lagrangian function for the transformed problem is given by

$$l(\mathbf{x}, \lambda) = \mathbf{x}^\top \mathbf{Q}\mathbf{x} + \lambda(1 - \mathbf{x}^\top \mathbf{P}\mathbf{x}).$$

The Lagrange condition yields

$$(\lambda\mathbf{I} - \mathbf{P}^{-1}\mathbf{Q})\mathbf{x} = \mathbf{0},$$

where

$$\mathbf{P}^{-1}\mathbf{Q} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}.$$

There are only two values of λ that satisfy $(\lambda\mathbf{I} - \mathbf{P}^{-1}\mathbf{Q})\mathbf{x} = \mathbf{0}$, namely, the eigenvalues of $\mathbf{P}^{-1}\mathbf{Q}$: $\lambda_1 = 2$, $\lambda_2 = 1$. We recall from our previous discussion of this problem that the Lagrange multiplier corresponding to the solution is the maximum eigenvalue of $\mathbf{P}^{-1}\mathbf{Q}$, namely, $\lambda^* = \lambda_1 = 2$. The corresponding eigenvector is the maximizer—the solution to the problem. The eigenvector corresponding to the eigenvalue $\lambda^* = 2$ satisfying the constraint $\mathbf{x}^\top \mathbf{P}\mathbf{x} = 1$ is $\pm\mathbf{x}^*$, where

$$\mathbf{x}^* = \left[\frac{1}{\sqrt{2}}, 0 \right]^\top.$$

At this point, all we have established is that the pairs $(\pm \mathbf{x}^*, \lambda^*)$ satisfy the Lagrange condition. We now show that the points $\pm \mathbf{x}^*$ are, in fact, strict local maximizers. We do this for the point \mathbf{x}^* . A similar procedure applies to $-\mathbf{x}^*$. We first compute the Hessian matrix of the Lagrangian function. We have

$$\mathbf{L}(\mathbf{x}^*, \lambda^*) = 2\mathbf{Q} - 2\lambda\mathbf{P} = \begin{bmatrix} 0 & 0 \\ 0 & -2 \end{bmatrix}.$$

The tangent space $T(\mathbf{x}^*)$ to $\{\mathbf{x} : \mathbf{1} - \mathbf{x}^\top \mathbf{P} \mathbf{x} = 0\}$ is

$$\begin{aligned} T(\mathbf{x}^*) &= \{\mathbf{y} \in \mathbb{R}^2 : \mathbf{x}^{*\top} \mathbf{P} \mathbf{y} = 0\} \\ &= \{\mathbf{y} : [\sqrt{2}, 0] \mathbf{y} = 0\} \\ &= \{\mathbf{y} : \mathbf{y} = [0, a]^\top, a \in \mathbb{R}\}. \end{aligned}$$

Note that for each $\mathbf{y} \in T(\mathbf{x}^*)$, $\mathbf{y} \neq \mathbf{0}$,

$$\mathbf{y}^\top \mathbf{L}(\mathbf{x}^*, \lambda^*) \mathbf{y} = [0, a] \begin{bmatrix} 0 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} 0 \\ a \end{bmatrix} = -2a^2 < 0.$$

Hence, $\mathbf{L}(\mathbf{x}^*, \lambda^*) < 0$ on $T(\mathbf{x}^*)$, and thus $\mathbf{x}^* = [1/\sqrt{2}, 0]^\top$ is a strict local maximizer. The same is true for the point $-\mathbf{x}^*$. Note that

$$\frac{\mathbf{x}^{*\top} \mathbf{Q} \mathbf{x}^*}{\mathbf{x}^{*\top} \mathbf{P} \mathbf{x}^*} = 2,$$

which, as expected, is the value of the maximal eigenvalue of $\mathbf{P}^{-1} \mathbf{Q}$. Finally, we point out that any scalar multiple $t\mathbf{x}^*$ of \mathbf{x}^* , $t \neq 0$, is a solution to the original problem of maximizing $\mathbf{x}^\top \mathbf{Q} \mathbf{x} / \mathbf{x}^\top \mathbf{P} \mathbf{x}$. ■

20.6 Minimizing Quadratics Subject to Linear Constraints

Consider the problem

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} \\ &\text{subject to} \quad \mathbf{A} \mathbf{x} = \mathbf{b}, \end{aligned}$$

where $\mathbf{Q} > 0$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m < n$, $\text{rank } \mathbf{A} = m$. This problem is a special case of what is called a *quadratic programming problem* (the general form of a quadratic programming problem includes the constraint $\mathbf{x} \geq \mathbf{0}$). Note that the constraint set contains an infinite number of points (see Section 2.3). We now show, using Lagrange's theorem, that there is a unique solution to the optimization problem above. Following that, we provide an example illustrating the application of this solution to an optimal control problem.

To solve the problem, we first form the Lagrangian function

$$l(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \boldsymbol{\lambda}^\top (\mathbf{b} - \mathbf{A} \mathbf{x}).$$

The Lagrange condition yields

$$D_{\mathbf{x}} l(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{x}^{*\top} \mathbf{Q} - \boldsymbol{\lambda}^{*\top} \mathbf{A} = \mathbf{0}^\top.$$

Rewriting, we get

$$\mathbf{x}^* = \mathbf{Q}^{-1} \mathbf{A}^\top \boldsymbol{\lambda}^*.$$

Premultiplying both sides of the above by \mathbf{A} gives

$$\mathbf{A} \mathbf{x}^* = \mathbf{A} \mathbf{Q}^{-1} \mathbf{A}^\top \boldsymbol{\lambda}^*.$$

Using the fact that $\mathbf{A} \mathbf{x}^* = \mathbf{b}$, and noting that $\mathbf{A} \mathbf{Q}^{-1} \mathbf{A}^\top$ is invertible because $\mathbf{Q} > 0$ and $\text{rank } \mathbf{A} = m$, we can solve for $\boldsymbol{\lambda}^*$ to obtain

$$\boldsymbol{\lambda}^* = (\mathbf{A} \mathbf{Q}^{-1} \mathbf{A}^\top)^{-1} \mathbf{b}.$$

Therefore, we obtain

$$\mathbf{x}^* = \mathbf{Q}^{-1} \mathbf{A}^\top (\mathbf{A} \mathbf{Q}^{-1} \mathbf{A}^\top)^{-1} \mathbf{b}.$$

The point \mathbf{x}^* is the only candidate for a minimizer. To establish that \mathbf{x}^* is indeed a minimizer, we verify that \mathbf{x}^* satisfies the second-order sufficient conditions. For this, we first find the Hessian matrix of the Lagrangian function at $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$. We have

$$\mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{Q},$$

which is positive definite. Thus, the point \mathbf{x}^* is a strict local minimizer. We will see in Chapter 22 that \mathbf{x}^* is, in fact, a global minimizer.

The special case where $\mathbf{Q} = \mathbf{I}_n$, the $n \times n$ identity matrix, reduces to the problem considered in Section 12.3. Specifically, the problem in Section 12.3 is to minimize the norm $\|\mathbf{x}\|$ subject to $\mathbf{A} \mathbf{x} = \mathbf{b}$. The objective function here is $f(\mathbf{x}) = \|\mathbf{x}\|$, which is not differentiable at $\mathbf{x} = \mathbf{0}$. This precludes the use of Lagrange's theorem because the theorem requires differentiability of the objective function. We can overcome this difficulty by considering an equivalent optimization problem:

$$\begin{aligned} &\text{minimize} && \frac{1}{2} \|\mathbf{x}\|^2 \\ &\text{subject to} && \mathbf{A} \mathbf{x} = \mathbf{b}. \end{aligned}$$

The objective function $\|\mathbf{x}\|^2/2$ has the same minimizer as the previous objective function $\|\mathbf{x}\|$. Indeed, if \mathbf{x}^* is such that for all $\mathbf{x} \in \mathbb{R}^n$ satisfying $\mathbf{A} \mathbf{x} = \mathbf{b}$, $\|\mathbf{x}^*\| \leq \|\mathbf{x}\|$, then $\|\mathbf{x}^*\|^2/2 \leq \|\mathbf{x}\|^2/2$. The same is true for the converse. Because the problem of minimizing $\|\mathbf{x}\|^2/2$ subject to $\mathbf{A} \mathbf{x} = \mathbf{b}$ is simply the

problem considered above with $\mathbf{Q} = \mathbf{I}_n$, we easily deduce the solution to be $\mathbf{x}^* = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}\mathbf{b}$, which agrees with the solution in Section 12.3.

Example 20.10 Consider the discrete-time linear system model

$$x_k = ax_{k-1} + bu_k, \quad k \geq 1,$$

with initial condition x_0 given. We can think of $\{x_k\}$ as a discrete-time signal that is controlled by an external input signal $\{u_k\}$. In the control literature, x_k is called the *state* at time k . For a given x_0 , our goal is to choose the control signal $\{u_k\}$ so that the state remains “small” over a time interval $[1, N]$, but at the same time the control signal is “not too large.” To express the desire to keep the state $\{x_k\}$ small, we choose the control sequence to minimize

$$\frac{1}{2} \sum_{i=1}^N x_i^2.$$

On the other hand, maintaining a control signal that is not too large, we minimize

$$\frac{1}{2} \sum_{i=1}^N u_i^2.$$

The two objectives above are conflicting in the sense that they cannot, in general, be achieved simultaneously—minimizing the first may result in a large control effort, while minimizing the second may result in large states. This is clearly a problem that requires compromise. One way to approach the problem is to minimize a weighted sum of the two functions above. Specifically, we can formulate the problem as

$$\text{minimize } \frac{1}{2} \sum_{i=1}^N (qx_i^2 + ru_i^2)$$

$$\text{subject to } x_k = ax_{k-1} + bu_k, \quad k = 1, \dots, N, \quad x_0 \text{ given,}$$

where the parameters q and r reflect the relative importance of keeping the state small versus keeping the control effort not too large. This problem is an instance of the *linear quadratic regulator (LQR) problem* (see, e.g., [15], [20], [85], [86], or [99]). Combining the two conflicting objectives of keeping the state small while keeping the control effort small is an instance of the *weighted sum* approach (see Section 24.4).

To solve the problem above, we can rewrite it as a quadratic programming problem. Define

$$\begin{aligned} \mathbf{Q} &= \begin{bmatrix} q\mathbf{I}_N & \mathbf{O} \\ \mathbf{O} & r\mathbf{I}_N \end{bmatrix}, \\ \mathbf{A} &= \begin{bmatrix} 1 & \cdots & 0 & -b & \cdots & 0 \\ -a & 1 & \vdots & -b & \vdots & \\ \ddots & \ddots & \vdots & & \ddots & \\ 0 & -a & 1 & 0 & \cdots & -b \end{bmatrix}, \\ \mathbf{b} &= \begin{bmatrix} ax_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{z} = [x_1, \dots, x_N, u_1, \dots, u_N]^\top. \end{aligned}$$

With these definitions, the problem reduces to the previously considered quadratic programming problem,

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} \mathbf{z}^\top \mathbf{Q} \mathbf{z} \\ &\text{subject to} \quad \mathbf{A} \mathbf{z} = \mathbf{b}, \end{aligned}$$

where \mathbf{Q} is $2N \times 2N$, \mathbf{A} is $N \times 2N$, and $\mathbf{b} \in \mathbb{R}^N$. The solution is

$$\mathbf{z}^* = \mathbf{Q}^{-1} \mathbf{A}^\top (\mathbf{A} \mathbf{Q}^{-1} \mathbf{A}^\top)^{-1} \mathbf{b}.$$

The first N components of \mathbf{z}^* represent the optimal state signal in the interval $[1, N]$, whereas the second N components represent the optimal control signal.

In practice, computation of the matrix inverses in the formula for \mathbf{z}^* above may be too costly. There are other ways to tackle the problem by exploiting its special structure. This is the study of *optimal control* (see, e.g., [15], [20], [85], [86], or [99]). ■

The following example illustrates an application of the above discussion.

Example 20.11 Credit-Card Holder Dilemma. Suppose that we currently have a credit-card debt of \$10,000. Credit-card debts are subject to a monthly interest rate of 2%, and the account balance is increased by the interest amount every month. Each month we have the option of reducing the account balance by contributing a payment to the account. Over the next 10 months, we plan to contribute a payment every month in such a way as to minimize the overall debt level while minimizing the hardship of making monthly payments.

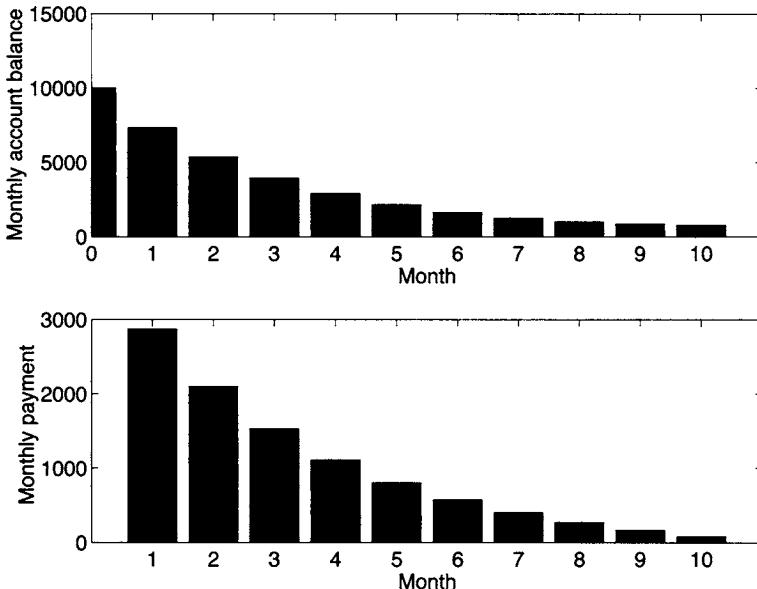


Figure 20.15 Plots for Example 20.11 with $q = 1$ and $r = 10$.

We solve our problem using the LQR framework described in Example 20.10. Let the current time be 0, x_k the account balance at the end of month k , and u_k our payment in month k . We have

$$x_k = 1.02x_{k-1} - u_k, \quad k = 1, \dots, 10;$$

that is, the account balance in a given month is equal to the account balance in the previous month plus the monthly interest on that balance minus our payment that month. Our optimization problem is then

$$\text{minimize} \quad \frac{1}{2} \sum_{i=1}^{10} (qx_i^2 + ru_i^2)$$

$$\text{subject to} \quad x_k = 1.02x_{k-1} - u_k, \quad k = 1, \dots, 10, \quad x_0 = 10,000,$$

which is an instance of the LQR problem. The parameters q and r reflect our priority in trading off between debt reduction and hardship in making payments. The more anxious we are to reduce our debt, the larger the value of q relative to r . On the other hand, the more reluctant we are to make payments, the larger the value of r relative to q .

The solution to the problem above is given by the formula derived in Example 20.10. In Figure 20.15 we plot the monthly account balances and payments over the next 10 months using $q = 1$ and $r = 10$. We can see here that our debt has been reduced to less than \$1000 after 10 months, but with a first

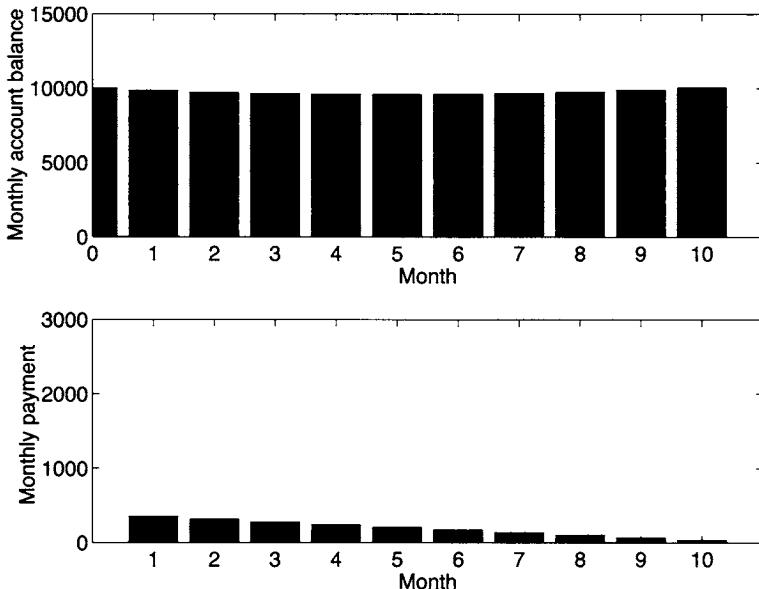


Figure 20.16 Plots for Example 20.11 with $q = 1$ and $r = 300$.

payment close to \$3000. If we feel that a payment of \$3000 is too high, then we can try to reduce this amount by increasing the value of r relative to q . However, going too far along these lines can lead to trouble. Indeed, if we use $q = 1$ and $r = 300$ (see Figure 20.16), although the monthly payments do not exceed \$400, the account balance is never reduced by much below \$10,000. In this case, the interest on the account balance eats up a significant portion of our monthly payments. In fact, our debt after 10 months will be higher than \$10,000. ■

For a treatment of optimization problems with quadratic objective functions, subject to linear or quadratic constraints, arising in communication and signal processing, see [105] and [106].

EXERCISES

20.1 Consider the following constraints on \mathbb{R}^2 :

$$h(x_1, x_2) = (x_1 - 2)^2 = 0 \quad \text{and} \quad g(x_1, x_2) = (x_2 + 1)^3 \leq 0.$$

Find the set of feasible points. Are the feasible points regular? Justify your answer.

20.2 Find local extremizers for the following optimization problems:

a. Minimize $x_1^2 + 2x_1x_2 + 3x_2^2 + 4x_1 + 5x_2 + 6x_3$

subject to $x_1 + 2x_2 = 3$

$4x_1 + 5x_3 = 6.$

b. Maximize $4x_1 + x_2^2$

subject to $x_1^2 + x_2^2 = 9.$

c. Maximize x_1x_2

subject to $x_1^2 + 4x_2^2 = 1.$

20.3 Find minimizers and maximizers of the function

$$f(\mathbf{x}) = (\mathbf{a}^\top \mathbf{x})(\mathbf{b}^\top \mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^3,$$

subject to

$$x_1 + x_2 = 0$$

$$x_2 + x_3 = 0,$$

where

$$\mathbf{a} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

20.4 Consider the problem

minimize $f(\mathbf{x})$

subject to $h(\mathbf{x}) = 0,$

where $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, $h : \mathbb{R}^2 \rightarrow \mathbb{R}$, and $\nabla f(\mathbf{x}) = [x_1, x_1 + 4]^\top$. Suppose that \mathbf{x}^* is an optimal solution and $\nabla h(\mathbf{x}^*) = [1, 4]^\top$. Find $\nabla f(\mathbf{x}^*)$.

20.5 Consider the problem

minimize $\|\mathbf{x} - \mathbf{x}_0\|^2$

subject to $\|\mathbf{x}\|^2 = 9,$

where $\mathbf{x}_0 = [1, \sqrt{3}]^\top$.

- a. Find all points satisfying the Lagrange condition for the problem.

- b. Using second-order conditions, determine whether or not each of the points in part a is a local minimizer.

20.6 We wish to construct a closed box with minimum surface area that encloses a volume of V cubic feet, where $V > 0$.

- Let a , b , and c denote the dimensions of the box with minimum surface area (with volume V). Derive the Lagrange condition that must be satisfied by a , b , and c .
- What does it mean for a point \mathbf{x}^* to be a *regular* point in this problem? Is the point $\mathbf{x}^* = [a, b, c]^\top$ a regular point?
- Find a , b , and c .
- Does the point $\mathbf{x}^* = [a, b, c]^\top$ found in part c satisfy the second-order sufficient condition?

20.7 Find local extremizers of

- $f(x_1, x_2, x_3) = x_1^2 + 3x_2^2 + x_3$ subject to $x_1^2 + x_2^2 + x_3^2 = 16$.
- $f(x_1, x_2) = x_1^2 + x_2^2$ subject to $3x_1^2 + 4x_1x_2 + 6x_2^2 = 140$.

20.8 Consider the problem

$$\begin{aligned} \text{minimize} \quad & 2x_1 + 3x_2 - 4, \quad x_1, x_2 \in \mathbb{R} \\ \text{subject to} \quad & x_1x_2 = 6. \end{aligned}$$

- Use Lagrange's theorem to find all possible local minimizers and maximizers.
- Use the second-order sufficient conditions to specify which points are strict local minimizers and which are strict local maximizers.
- Are the points in part b global minimizers or maximizers? Explain.

20.9 Find all maximizers of the function

$$f(x_1, x_2) = \frac{18x_1^2 - 8x_1x_2 + 12x_2^2}{2x_1^2 + 2x_2^2}.$$

20.10 Find all solutions to the problem

$$\begin{aligned} \text{maximize} \quad & \mathbf{x}^\top \begin{bmatrix} 3 & 4 \\ 0 & 3 \end{bmatrix} \mathbf{x} \\ \text{subject to} \quad & \|\mathbf{x}\|^2 = 1. \end{aligned}$$

20.11 Consider a matrix \mathbf{A} with the property that $\mathbf{A}^\top \mathbf{A}$ has eigenvalues ranging from 1 to 20 (i.e., the smallest eigenvalue is 1 and the largest is 20). Let \mathbf{x} be a vector such that $\|\mathbf{x}\| = 1$, and let $\mathbf{y} = \mathbf{A}\mathbf{x}$. Use Lagrange multiplier methods to find the range of values that $\|\mathbf{y}\|$ can take.

Hint: What is the largest value that $\|\mathbf{y}\|$ can take? What is the smallest value that $\|\mathbf{y}\|$ can take?

20.12 Consider a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. Define the *induced 2-norm* of \mathbf{A} , denoted $\|\mathbf{A}\|_2$, to be the number

$$\|\mathbf{A}\|_2 = \max\{\|\mathbf{A}\mathbf{x}\| : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\| = 1\},$$

where the norm $\|\cdot\|$ on the right-hand side above is the usual Euclidean norm.

Suppose that the eigenvalues of $\mathbf{A}^\top \mathbf{A}$ are $\lambda_1, \dots, \lambda_n$ (ordered from largest to smallest). Use Lagrange's theorem to express $\|\mathbf{A}\|_2$ in terms of the eigenvalues above (cf. Theorem 3.8).

20.13 Let $\mathbf{P} = \mathbf{P}^\top$ be a positive definite matrix. Show that any point \mathbf{x} satisfying $1 - \mathbf{x}^\top \mathbf{P} \mathbf{x} = 0$ is a regular point.

20.14 Consider the problem

$$\begin{aligned} & \text{maximize} && ax_1 + bx_2, \quad x_1, x_2 \in \mathbb{R} \\ & \text{subject to} && x_1^2 + x_2^2 = 2, \end{aligned}$$

where $a, b \in \mathbb{R}$. Show that if $[1, 1]^\top$ is a solution to the problem, then $a = b$.

20.15 Consider the problem

$$\begin{aligned} & \text{minimize} && x_1 x_2 - 2x_1, \quad x_1, x_2 \in \mathbb{R} \\ & \text{subject to} && x_1^2 - x_2^2 = 0. \end{aligned}$$

- a. Apply Lagrange's theorem directly to the problem to show that if a solution exists, it must be either $[1, 1]^\top$ or $[-1, 1]^\top$.
- b. Use the second-order necessary conditions to show that $[-1, 1]^\top$ cannot possibly be the solution.
- c. Use the second-order sufficient conditions to show that $[1, 1]^\top$ is a strict local minimizer.

20.16 Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m \leq n$, $\text{rank } \mathbf{A} = m$, and $\mathbf{x}_0 \in \mathbb{R}^n$. Let \mathbf{x}^* be the point on the nullspace of \mathbf{A} that is closest to \mathbf{x}_0 (in the sense of Euclidean norm).

- a. Show that \mathbf{x}^* is orthogonal to $\mathbf{x}^* - \mathbf{x}_0$.

- b. Find a formula for \mathbf{x}^* in terms of \mathbf{A} and \mathbf{x}_0 .

20.17 Consider the problem

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|^2 \\ & \text{subject to} && \mathbf{Cx} = \mathbf{d}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m > n$, $\mathbf{C} \in \mathbb{R}^{p \times n}$, $p < n$, and both \mathbf{A} and \mathbf{C} are of full rank. We wish to find an expression for the solution (in terms of \mathbf{A} , \mathbf{b} , \mathbf{C} , and \mathbf{d}).

- a. Apply Lagrange's theorem to solve this problem.
- b. As an alternative, rewrite the given optimization problem in the form of a quadratic programming problem and apply the formula in Section 20.6 to obtain the solution.

20.18 Consider the problem of minimizing a general quadratic function subject to a linear constraint:

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \mathbf{x}^\top \mathbf{Qx} - \mathbf{c}^\top \mathbf{x} + d \\ & \text{subject to} && \mathbf{Ax} = \mathbf{b}, \end{aligned}$$

where $\mathbf{Q} = \mathbf{Q}^\top > 0$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m < n$, $\text{rank } \mathbf{A} = m$, and d is a constant. Derive a closed-form solution to the problem.

20.19 Let \mathbf{L} be an $n \times n$ real symmetric matrix, and let \mathcal{M} be a subspace of \mathbb{R}^n with dimension $m < n$. Let $\{\mathbf{b}_1, \dots, \mathbf{b}_m\} \subset \mathbb{R}^n$ be a basis for \mathcal{M} , and let \mathbf{B} be the $n \times m$ matrix with \mathbf{b}_i as the i th column. Let $\mathbf{L}_{\mathcal{M}}$ be the $m \times m$ matrix defined by $\mathbf{L}_{\mathcal{M}} = \mathbf{B}^\top \mathbf{LB}$. Show that \mathbf{L} is positive semidefinite (definite) on \mathcal{M} if and only if $\mathbf{L}_{\mathcal{M}}$ is positive semidefinite (definite).

Note: This result is useful for checking that the Hessian of the Lagrangian function at a point is positive definite on the tangent space at that point.

20.20 Consider the sequence $\{x_k\}$, $x_k \in \mathbb{R}$, generated by the recursion

$$x_{k+1} = ax_k + bu_k, \quad k \geq 0 \quad (a, b \in \mathbb{R}, a, b \neq 0),$$

where u_0, u_1, u_2, \dots is a sequence of “control inputs,” and the initial condition $x_0 \neq 0$ is given. The recursion above is also called a *discrete-time linear system*. We wish to find values of control inputs u_0 and u_1 such that $x_2 = 0$, and the average input energy $(u_0^2 + u_1^2)/2$ is minimized. Denote the optimal inputs by u_0^* and u_1^* .

- a. Find expressions for u_0^* and u_1^* in terms of a , b , and x_0 .
- b. Use the second-order sufficient conditions to show that the point $\mathbf{u}^* = [u_0^*, u_1^*]^\top$ in part a is a strict local minimizer.

20.21 Consider the discrete-time linear system $x_k = 2x_{k-1} + u_k$, $k \geq 1$, with $x_0 = 1$. Find the values of the control inputs u_1 and u_2 to minimize

$$x_2^2 + \frac{1}{2}u_1^2 + \frac{1}{3}u_2^2.$$

20.22 Consider the discrete-time linear system $x_{k+1} = x_k + 2u_k$, $0 \leq k \leq 2$, with $x_0 = 3$. Use the Lagrange multiplier approach to calculate the optimal control sequence $\{u_0, u_1, u_2\}$ that transfers the initial state x_0 to $x_3 = 9$ while minimizing

$$\frac{1}{2} \sum_{k=0}^2 u_k^2.$$

CHAPTER 21

PROBLEMS WITH INEQUALITY CONSTRAINTS

21.1 Karush-Kuhn-Tucker Condition

In Chapter 20 we analyzed constrained optimization problems involving only equality constraints. In this chapter we discuss extremum problems that also involve inequality constraints. The treatment in this chapter parallels that of Chapter 20. In particular, as we shall see, problems with inequality constraints can also be treated using Lagrange multipliers.

We consider the following problem:

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0}, \\ & && \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \end{aligned}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m \leq n$, and $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^p$. For the general problem above, we adopt the following definitions.

Definition 21.1 An inequality constraint $g_j(\mathbf{x}) \leq 0$ is said to be *active* at \mathbf{x}^* if $g_j(\mathbf{x}^*) = 0$. It is *inactive* at \mathbf{x}^* if $g_j(\mathbf{x}^*) < 0$. ■

By convention, we consider an equality constraint $h_i(\mathbf{x}) = 0$ to be always active.

Definition 21.2 Let \mathbf{x}^* satisfy $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$, $\mathbf{g}(\mathbf{x}^*) \leq \mathbf{0}$, and let $J(\mathbf{x}^*)$ be the index set of active inequality constraints:

$$J(\mathbf{x}^*) \triangleq \{j : g_j(\mathbf{x}^*) = 0\}.$$

Then, we say that \mathbf{x}^* is a *regular point* if the vectors

$$\nabla h_i(\mathbf{x}^*), \quad \nabla g_j(\mathbf{x}^*), \quad 1 \leq i \leq m, \quad j \in J(\mathbf{x}^*)$$

are linearly independent. ■

We now prove a first-order necessary condition for a point to be a local minimizer. We call this condition the *Karush-Kuhn-Tucker (KKT) condition*. In the literature, this condition is sometimes also called the Kuhn-Tucker condition.

Theorem 21.1 Karush-Kuhn-Tucker (KKT) Theorem. Let $f, \mathbf{h}, \mathbf{g} \in C^1$. Let \mathbf{x}^* be a regular point and a local minimizer for the problem of minimizing f subject to $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. Then, there exist $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ and $\boldsymbol{\mu}^* \in \mathbb{R}^p$ such that:

1. $\boldsymbol{\mu}^* \geq \mathbf{0}$.
2. $Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} Dh(\mathbf{x}^*) + \boldsymbol{\mu}^{*\top} Dg(\mathbf{x}^*) = \mathbf{0}^\top$.
3. $\boldsymbol{\mu}^{*\top} \mathbf{g}(\mathbf{x}^*) = 0$.

□

In Theorem 21.1, we refer to $\boldsymbol{\lambda}^*$ as the Lagrange multiplier vector and $\boldsymbol{\mu}^*$ as the Karush-Kuhn-Tucker (KKT) multiplier vector. We refer to their components as Lagrange multipliers and Karush-Kuhn-Tucker (KKT) multipliers, respectively.

Before proving this theorem, let us first discuss its meaning. Observe that $\mu_j^* \geq 0$ (by condition 1) and $g_j(\mathbf{x}^*) \leq 0$. Therefore, the condition

$$\boldsymbol{\mu}^{*\top} \mathbf{g}(\mathbf{x}^*) = \mu_1^* g_1(\mathbf{x}^*) + \cdots + \mu_p^* g_p(\mathbf{x}^*) = 0$$

implies that if $g_j(\mathbf{x}^*) < 0$, then $\mu_j^* = 0$; that is, for all $j \notin J(\mathbf{x}^*)$, we have $\mu_j^* = 0$. In other words, the KKT multipliers μ_j^* corresponding to inactive constraints are zero. The other KKT multipliers, μ_i^* , $i \in J(\mathbf{x}^*)$, are nonnegative; they may or may not be equal to zero.

Example 21.1 A graphical illustration of the KKT theorem is given in Figure 21.1. In this two-dimensional example, we have only inequality constraints

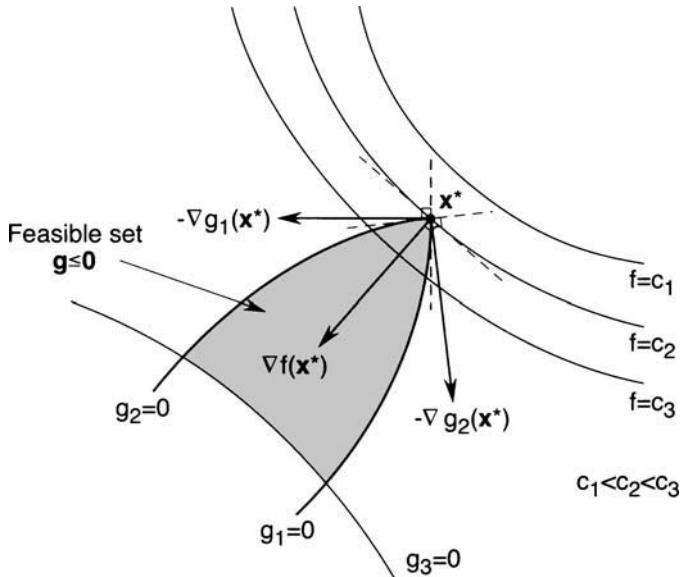


Figure 21.1 Illustration of the Karush-Kuhn-Tucker (KKT) theorem.

$g_j(\mathbf{x}) \leq 0, j = 1, 2, 3$. Note that the point \mathbf{x}^* in the figure is indeed a minimizer. The constraint $g_3(\mathbf{x}) \leq 0$ is inactive: $g_3(\mathbf{x}^*) < 0$; hence $\mu_3^* = 0$. By the KKT theorem, we have

$$\nabla f(\mathbf{x}^*) + \mu_1^* \nabla g_1(\mathbf{x}^*) + \mu_2^* \nabla g_2(\mathbf{x}^*) = \mathbf{0},$$

or, equivalently,

$$\nabla f(\mathbf{x}^*) = -\mu_1^* \nabla g_1(\mathbf{x}^*) - \mu_2^* \nabla g_2(\mathbf{x}^*),$$

where $\mu_1^* > 0$ and $\mu_2^* > 0$. It is easy to interpret the KKT condition graphically for this example. Specifically, we can see from Figure 21.1 that $\nabla f(\mathbf{x}^*)$ must be a linear combination of the vectors $-\nabla g_1(\mathbf{x}^*)$ and $-\nabla g_2(\mathbf{x}^*)$ with positive coefficients. This is reflected exactly in the equation above, where the coefficients μ_1^* and μ_2^* are the KKT multipliers. ■

We apply the KKT condition in the same way that we apply any necessary condition. Specifically, we search for points satisfying the KKT condition and treat these points as candidate minimizers. To summarize, the KKT condition consists of five parts (three equations and two inequalities):

1. $\boldsymbol{\mu}^* \geq \mathbf{0}$.
2. $Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} Dh(\mathbf{x}^*) + \boldsymbol{\mu}^{*\top} Dg(\mathbf{x}^*) = \mathbf{0}^\top$.
3. $\boldsymbol{\mu}^{*\top} g(\mathbf{x}^*) = 0$.

$$4. \quad \mathbf{h}(\mathbf{x}^*) = \mathbf{0}.$$

$$5. \quad \mathbf{g}(\mathbf{x}^*) \leq \mathbf{0}.$$

We now prove the KKT theorem.

Proof of the Karush-Kuhn-Tucker Theorem. Let \mathbf{x}^* be a regular local minimizer of f on the set $\{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$. Then, \mathbf{x}^* is also a regular local minimizer of f on the set $\{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}, g_j(\mathbf{x}) = 0, j \in J(\mathbf{x}^*)\}$ (see Exercise 21.16). Note that the latter constraint set involves only equality constraints. Therefore, from Lagrange's theorem, it follows that there exist vectors $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ and $\boldsymbol{\mu}^* \in \mathbb{R}^p$ such that

$$Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} D\mathbf{h}(\mathbf{x}^*) + \boldsymbol{\mu}^{*\top} D\mathbf{g}(\mathbf{x}^*) = \mathbf{0}^\top,$$

where for all $j \notin J(\mathbf{x}^*)$, we have $\mu_j^* = 0$. To complete the proof it remains to show that for all $j \in J(\mathbf{x}^*)$, we have $\mu_j^* \geq 0$ (and hence for all $j = 1, \dots, p$, we have $\mu_j^* \geq 0$, i.e., $\boldsymbol{\mu}^* \geq \mathbf{0}$). We use a proof by contradiction. So suppose that there exists $j \in J(\mathbf{x}^*)$ such that $\mu_j^* < 0$. Let \hat{S} and $\hat{T}(\mathbf{x}^*)$ be the surface and tangent space defined by all other active constraints at \mathbf{x}^* . Specifically,

$$\hat{S} = \{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}, g_i(\mathbf{x}) = 0, i \in J(\mathbf{x}^*), i \neq j\}$$

and

$$\hat{T}(\mathbf{x}^*) = \{\mathbf{y} : D\mathbf{h}(\mathbf{x}^*)\mathbf{y} = \mathbf{0}, Dg_i(\mathbf{x}^*)\mathbf{y} = 0, i \in J(\mathbf{x}^*), i \neq j\}.$$

We claim that by the regularity of \mathbf{x}^* , there exists $\mathbf{y} \in \hat{T}(\mathbf{x}^*)$ such that

$$Dg_j(\mathbf{x}^*)\mathbf{y} \neq 0.$$

To see this, suppose that for all $\mathbf{y} \in \hat{T}(\mathbf{x}^*)$, $\nabla g_j(\mathbf{x}^*)^\top \mathbf{y} = Dg_j(\mathbf{x}^*)\mathbf{y} = 0$. This implies that $\nabla g_j(\mathbf{x}^*) \in \hat{T}(\mathbf{x}^*)^\perp$. By Lemma 20.1, this, in turn, implies that

$$\nabla g_j(\mathbf{x}^*) \in \text{span}[\nabla h_k(\mathbf{x}^*), k = 1, \dots, m, \nabla g_i(\mathbf{x}^*), i \in J(\mathbf{x}^*), i \neq j].$$

But this contradicts the fact that \mathbf{x}^* is a regular point, which proves our claim. Without loss of generality, we assume that we have \mathbf{y} such that $Dg_j(\mathbf{x}^*)\mathbf{y} < 0$.

Consider the Lagrange condition, rewritten as

$$Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} D\mathbf{h}(\mathbf{x}^*) + \mu_j^* Dg_j(\mathbf{x}^*) + \sum_{i \neq j} \mu_i^* Dg_i(\mathbf{x}^*) = \mathbf{0}^\top.$$

If we postmultiply the above by \mathbf{y} and use the fact that $\mathbf{y} \in \hat{T}(\mathbf{x}^*)$, we get

$$Df(\mathbf{x}^*)\mathbf{y} = -\mu_j^* Dg_j(\mathbf{x}^*)\mathbf{y}.$$

Because $Dg_j(\mathbf{x}^*)\mathbf{y} < 0$ and we have assumed that $\mu_j^* < 0$, we have

$$Df(\mathbf{x}^*)\mathbf{y} < 0.$$

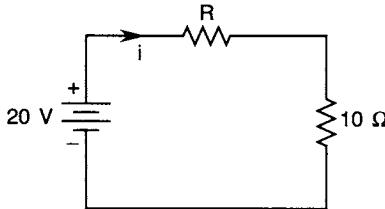


Figure 21.2 Circuit in Example 21.2.

Because $\mathbf{y} \in \hat{T}(\mathbf{x}^*)$, by Theorem 20.1 we can find a differentiable curve $\{\mathbf{x}(t) : t \in (a, b)\}$ on \hat{S} such that there exists $t^* \in (a, b)$ with $\mathbf{x}(t^*) = \mathbf{x}^*$ and $\dot{\mathbf{x}}(t^*) = \mathbf{y}$. Now,

$$\frac{d}{dt} f(\mathbf{x}(t^*)) = Df(\mathbf{x}^*)\mathbf{y} < 0.$$

The above means that there is a $\delta > 0$ such that for all $t \in (t^*, t^* + \delta]$, we have

$$f(\mathbf{x}(t)) < f(\mathbf{x}(t^*)) = f(\mathbf{x}^*).$$

On the other hand,

$$\frac{d}{dt} g_j(\mathbf{x}(t^*)) = Dg_j(\mathbf{x}^*)\mathbf{y} < 0,$$

and for some $\varepsilon > 0$ and all $t \in [t^*, t^* + \varepsilon]$, we have that $g_j(\mathbf{x}(t)) \leq 0$. Therefore, for all $t \in (t^*, t^* + \min\{\delta, \varepsilon\}]$, we have that $g_j(\mathbf{x}(t)) \leq 0$ and $f(\mathbf{x}(t)) < f(\mathbf{x}^*)$. Because the points $\mathbf{x}(t)$, $t \in (t^*, t^* + \min\{\delta, \varepsilon\}]$, are in \hat{S} , they are feasible points with lower objective function values than \mathbf{x}^* . This contradicts the assumption that \mathbf{x}^* is a local minimizer, which completes the proof. ■

Example 21.2 Consider the circuit in Figure 21.2. Formulate and solve the KKT condition for the following problems.

- Find the value of the resistor $R \geq 0$ such that the power absorbed by this resistor is maximized.
- Find the value of the resistor $R \geq 0$ such that the power delivered to the $10\text{-}\Omega$ resistor is maximized.

Solution:

- The power absorbed by the resistor R is $p = i^2 R$, where $i = \frac{20}{10+R}$. The optimization problem can be represented as

$$\begin{aligned} \text{minimize} \quad & -\frac{400R}{(10+R)^2} \\ \text{subject to} \quad & -R \leq 0. \end{aligned}$$

The derivative of the objective function is

$$-\frac{400(10+R)^2 - 800R(10+R)}{(10+R)^4} = -\frac{400(10-R)}{(10+R)^3}.$$

Thus, the KKT condition is

$$-\frac{400(10-R)}{(10+R)^3} - \mu = 0,$$

$$\mu \geq 0,$$

$$\mu R = 0,$$

$$-R \leq 0.$$

We consider two cases. In the first case, suppose that $\mu > 0$. Then, $R = 0$. But this contradicts the first condition above. Now suppose that $\mu = 0$. Then, by the first condition, we have $R = 10$. Therefore, the only solution to the KKT condition is $R = 10$, $\mu = 0$.

- b. The power absorbed by the $10\text{-}\Omega$ resistor is $p = i^2 10$, where $i = 20/(10+R)$. The optimization problem can be represented as

$$\begin{aligned} \text{minimize} \quad & -\frac{4000}{(10+R)^2} \\ \text{subject to} \quad & -R \leq 0. \end{aligned}$$

The derivative of the objective function is

$$\frac{8000}{(10+R)^3}.$$

Thus, the KKT condition is

$$\begin{aligned} \frac{8000}{(10+R)^3} - \mu &= 0, \\ \mu &\geq 0, \\ \mu R &= 0, \\ -R &\leq 0. \end{aligned}$$

As before, we consider two cases. In the first case, suppose that $\mu > 0$. Then, $R = 0$, which is feasible. For the second case, suppose that $\mu = 0$. But this contradicts the first condition. Therefore, the only solution to the KKT condition is $R = 0$, $\mu = 8$. ■

In the case when the objective function is to be maximized, that is, when the optimization problem has the form

$$\begin{aligned} \text{maximize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \end{aligned}$$

the KKT condition can be written as

1. $\mu^* \geq \mathbf{0}$.
2. $-Df(\mathbf{x}^*) + \lambda^{*\top} Dh(\mathbf{x}^*) + \mu^{*\top} Dg(\mathbf{x}^*) = \mathbf{0}^\top$.
3. $\mu^{*\top} g(\mathbf{x}^*) = 0$.
4. $h(\mathbf{x}^*) = \mathbf{0}$.
5. $g(\mathbf{x}^*) \leq \mathbf{0}$.

The above is easily derived by converting the maximization problem above into a minimization problem, by multiplying the objective function by -1 . It can be further rewritten as

1. $\mu^* \leq \mathbf{0}$.
2. $Df(\mathbf{x}^*) + \lambda^{*\top} Dh(\mathbf{x}^*) + \mu^{*\top} Dg(\mathbf{x}^*) = \mathbf{0}^\top$.
3. $\mu^{*\top} g(\mathbf{x}^*) = 0$.
4. $h(\mathbf{x}^*) = \mathbf{0}$.
5. $g(\mathbf{x}^*) \leq \mathbf{0}$.

The form shown above is obtained from the preceding one by changing the signs of μ^* and λ^* and multiplying condition 2 by -1 .

We can similarly derive the KKT condition for the case when the inequality constraint is of the form $g(\mathbf{x}) \geq \mathbf{0}$. Specifically, consider the problem

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && h(\mathbf{x}) = \mathbf{0} \\ & && g(\mathbf{x}) \geq \mathbf{0}. \end{aligned}$$

We multiply the inequality constraint function by -1 to obtain $-g(\mathbf{x}) \leq \mathbf{0}$. Thus, the KKT condition for this case is

1. $\mu^* \geq \mathbf{0}$.
2. $Df(\mathbf{x}^*) + \lambda^{*\top} Dh(\mathbf{x}^*) - \mu^{*\top} Dg(\mathbf{x}^*) = \mathbf{0}^\top$.
3. $\mu^{*\top} g(\mathbf{x}^*) = 0$.
4. $h(\mathbf{x}^*) = \mathbf{0}$.
5. $g(\mathbf{x}^*) \geq \mathbf{0}$.

Changing the sign of μ^* as before, we obtain

1. $\mu^* \leq \mathbf{0}$.

2. $Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} Dh(\mathbf{x}^*) + \boldsymbol{\mu}^{*\top} Dg(\mathbf{x}^*) = \mathbf{0}^\top$.
3. $\boldsymbol{\mu}^{*\top} g(\mathbf{x}^*) = 0$.
4. $h(\mathbf{x}^*) = \mathbf{0}$.
5. $g(\mathbf{x}^*) \geq \mathbf{0}$.

For the problem

$$\begin{aligned} & \text{maximize} && f(\mathbf{x}) \\ & \text{subject to} && h(\mathbf{x}) = \mathbf{0} \\ & && g(\mathbf{x}) \geq \mathbf{0}, \end{aligned}$$

the KKT condition is exactly the same as in Theorem 21.1, except for the reversal of the inequality constraint.

Example 21.3 In Figure 21.3, the two points \mathbf{x}_1 and \mathbf{x}_2 are feasible points; that is, $g(\mathbf{x}_1) \geq \mathbf{0}$ and $g(\mathbf{x}_2) \geq \mathbf{0}$, and they satisfy the KKT condition.

The point \mathbf{x}_1 is a maximizer. The KKT condition for this point (with KKT multiplier μ_1) is

1. $\mu_1 \geq 0$.
2. $\nabla f(\mathbf{x}_1) + \mu_1 \nabla g(\mathbf{x}_1) = \mathbf{0}$.
3. $\mu_1 g(\mathbf{x}_1) = 0$.
4. $g(\mathbf{x}_1) \geq \mathbf{0}$.

The point \mathbf{x}_2 is a minimizer of f . The KKT condition for this point (with KKT multiplier μ_2) is

1. $\mu_2 \leq 0$.
2. $\nabla f(\mathbf{x}_2) + \mu_2 \nabla g(\mathbf{x}_2) = \mathbf{0}$.
3. $\mu_2 g(\mathbf{x}_2) = 0$.
4. $g(\mathbf{x}_2) \geq \mathbf{0}$.

■

Example 21.4 Consider the problem

$$\begin{aligned} & \text{minimize} && f(x_1, x_2) \\ & \text{subject to} && x_1, x_2 \geq 0, \end{aligned}$$

where

$$f(x_1, x_2) = x_1^2 + x_2^2 + x_1 x_2 - 3x_1.$$

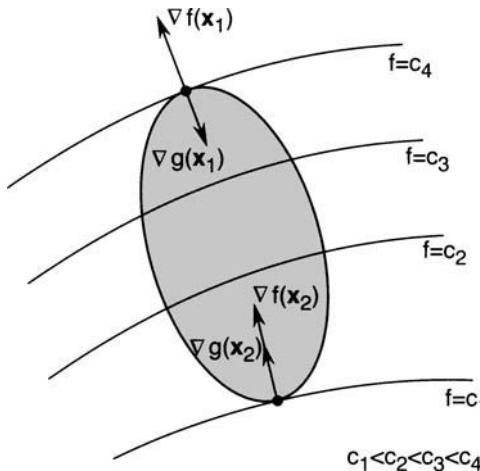


Figure 21.3 Points satisfying the KKT condition (x_1 is a maximizer and x_2 is a minimizer).

The KKT condition for this problem is

1. $\mu = [\mu_1, \mu_2]^\top \leq \mathbf{0}$.
2. $Df(\mathbf{x}) + \mu^\top = \mathbf{0}^\top$.
3. $\mu^\top \mathbf{x} = 0$.
4. $\mathbf{x} \geq \mathbf{0}$.

We have

$$Df(\mathbf{x}) = [2x_1 + x_2 - 3, x_1 + 2x_2].$$

This gives

$$\begin{aligned} 2x_1 + x_2 + \mu_1 &= 3, \\ x_1 + 2x_2 + \mu_2 &= 0, \\ \mu_1 x_1 + \mu_2 x_2 &= 0. \end{aligned}$$

We now have four variables, three equations, and the inequality constraints on each variable. To find a solution (\mathbf{x}^*, μ^*) , we first try

$$\mu_1^* = 0, \quad x_2^* = 0,$$

which gives

$$x_1^* = \frac{3}{2}, \quad \mu_2^* = -\frac{3}{2}.$$

The above satisfies all the KKT and feasibility conditions. In a similar fashion, we can try

$$\mu_2^* = 0, \quad x_1^* = 0,$$

which gives

$$x_2^* = 0, \quad \mu_1^* = 3.$$

This point clearly violates the nonpositivity constraint on μ_1^* .

The feasible point above satisfying the KKT condition is only a candidate for a minimizer. However, there is no guarantee that the point is indeed a minimizer, because the KKT condition is, in general, only necessary. A sufficient condition for a point to be a minimizer is given in the next section. ■

Example 21.4 is a special case of a more general problem of the form

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

The KKT condition for this problem has the form

$$\begin{aligned} & \boldsymbol{\mu} \leq \mathbf{0}, \\ & \nabla f(\mathbf{x}) + \boldsymbol{\mu} = \mathbf{0}, \\ & \boldsymbol{\mu}^\top \mathbf{x} = 0, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

From the above, we can eliminate $\boldsymbol{\mu}$ to obtain

$$\begin{aligned} & \nabla f(\mathbf{x}) \geq \mathbf{0}, \\ & \mathbf{x}^\top \nabla f(\mathbf{x}) = 0, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Some possible points in \mathbb{R}^2 that satisfy these conditions are depicted in Figure 21.4.

For further results related to the KKT condition, we refer the reader to [90, Chapter 7].

21.2 Second-Order Conditions

As in the case of extremum problems with equality constraints, we can also give second-order necessary and sufficient conditions for extremum problems involving inequality constraints. For this, we need to define the following matrix:

$$\mathbf{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{F}(\mathbf{x}) + [\boldsymbol{\lambda} \mathbf{H}(\mathbf{x})] + [\boldsymbol{\mu} \mathbf{G}(\mathbf{x})],$$

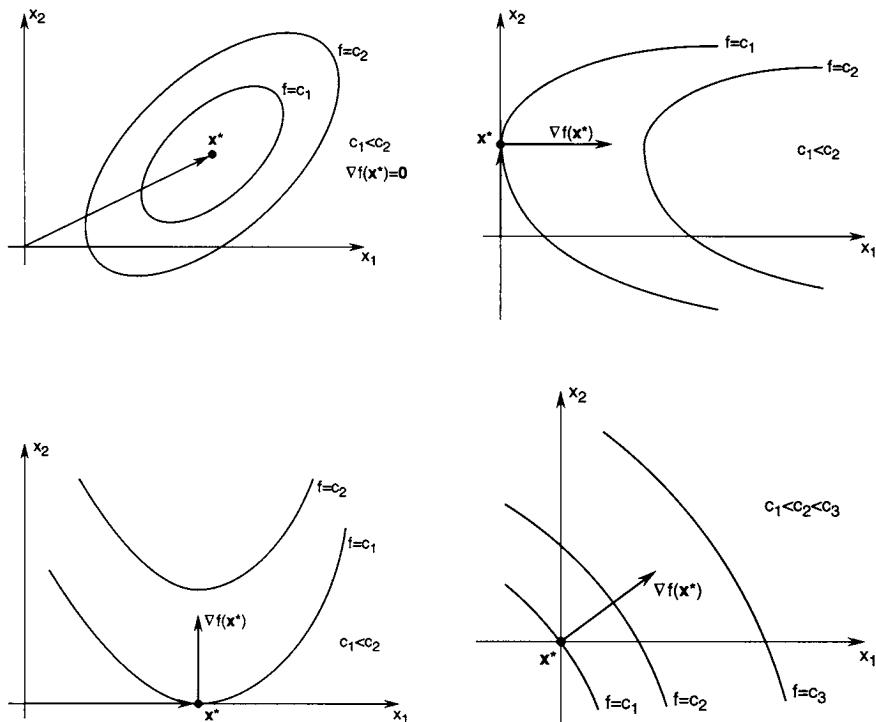


Figure 21.4 Some possible points satisfying the KKT condition for problems with positive constraints. (Adapted from [13].)

where $\mathbf{F}(\mathbf{x})$ is the Hessian matrix of f at \mathbf{x} , and the notation $[\lambda \mathbf{H}(\mathbf{x})]$ represents

$$[\lambda \mathbf{H}(\mathbf{x})] = \lambda_1 \mathbf{H}_1(\mathbf{x}) + \cdots + \lambda_m \mathbf{H}_m(\mathbf{x}),$$

as before. Similarly, the notation $[\mu \mathbf{G}(\mathbf{x})]$ represents

$$[\mu \mathbf{G}(\mathbf{x})] = \mu_1 \mathbf{G}_1(\mathbf{x}) + \cdots + \mu_p \mathbf{G}_p(\mathbf{x}),$$

where $\mathbf{G}_k(\mathbf{x})$ is the Hessian of g_k at \mathbf{x} , given by

$$\mathbf{G}_k(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 g_k}{\partial x_1^2}(\mathbf{x}) & \cdots & \frac{\partial^2 g_k}{\partial x_n \partial x_1}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial^2 g_k}{\partial x_1 \partial x_n}(\mathbf{x}) & \cdots & \frac{\partial^2 g_k}{\partial x_n^2}(\mathbf{x}) \end{bmatrix}.$$

In the following theorem, we use

$$T(\mathbf{x}^*) = \{\mathbf{y} \in \mathbb{R}^n : D\mathbf{h}(\mathbf{x}^*)\mathbf{y} = \mathbf{0}, Dg_j(\mathbf{x}^*)\mathbf{y} = 0, j \in J(\mathbf{x}^*)\},$$

that is, the tangent space to the surface defined by active constraints.

Theorem 21.2 Second-Order Necessary Conditions. *Let \mathbf{x}^* be a local minimizer of $f : \mathbb{R}^n \rightarrow \mathbb{R}$ subject to $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$, $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m \leq n$, $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^p$, and $f, \mathbf{h}, \mathbf{g} \in C^2$. Suppose that \mathbf{x}^* is regular. Then, there exist $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ and $\boldsymbol{\mu}^* \in \mathbb{R}^p$ such that:*

1. $\boldsymbol{\mu}^* \geq \mathbf{0}$, $Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} D\mathbf{h}(\mathbf{x}^*) + \boldsymbol{\mu}^{*\top} D\mathbf{g}(\mathbf{x}^*) = \mathbf{0}^\top$, $\boldsymbol{\mu}^{*\top} \mathbf{g}(\mathbf{x}^*) = 0$.

2. For all $\mathbf{y} \in T(\mathbf{x}^*)$ we have $\mathbf{y}^\top \mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{y} \geq 0$. □

Proof. Part 1 is simply a result of the KKT theorem. To prove part 2, we note that because the point \mathbf{x}^* is a local minimizer over $\{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$, it is also a local minimizer over $\{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}, g_j(\mathbf{x}) = 0, j \in J(\mathbf{x}^*)\}$; that is, the point \mathbf{x}^* is a local minimizer with active constraints taken as equality constraints (see Exercise 21.16). Hence, the second-order necessary conditions for equality constraints (Theorem 20.4) are applicable here, which completes the proof. ■

We now state the second-order sufficient conditions for extremum problems involving inequality constraints. In the formulation of the result, we use the following set:

$$\tilde{T}(\mathbf{x}^*, \boldsymbol{\mu}^*) = \{\mathbf{y} : D\mathbf{h}(\mathbf{x}^*) \mathbf{y} = \mathbf{0}, Dg_i(\mathbf{x}^*) \mathbf{y} = 0, i \in \tilde{J}(\mathbf{x}^*, \boldsymbol{\mu}^*)\},$$

where $\tilde{J}(\mathbf{x}^*, \boldsymbol{\mu}^*) = \{i : g_i(\mathbf{x}^*) = 0, \mu_i^* > 0\}$. Note that $\tilde{J}(\mathbf{x}^*, \boldsymbol{\mu}^*)$ is a subset of $J(\mathbf{x}^*)$: $\tilde{J}(\mathbf{x}^*, \boldsymbol{\mu}^*) \subset J(\mathbf{x}^*)$. This, in turn, implies that $T(\mathbf{x}^*)$ is a subset of $\tilde{T}(\mathbf{x}^*, \boldsymbol{\mu}^*)$: $T(\mathbf{x}^*) \subset \tilde{T}(\mathbf{x}^*, \boldsymbol{\mu}^*)$.

Theorem 21.3 Second-Order Sufficient Conditions. *Suppose that $f, \mathbf{g}, \mathbf{h} \in C^2$ and there exist a feasible point $\mathbf{x}^* \in \mathbb{R}^n$ and vectors $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ and $\boldsymbol{\mu}^* \in \mathbb{R}^p$ such that:*

1. $\boldsymbol{\mu}^* \geq \mathbf{0}$, $Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} D\mathbf{h}(\mathbf{x}^*) + \boldsymbol{\mu}^{*\top} D\mathbf{g}(\mathbf{x}^*) = \mathbf{0}^\top$, $\boldsymbol{\mu}^{*\top} \mathbf{g}(\mathbf{x}^*) = 0$.

2. For all $\mathbf{y} \in \tilde{T}(\mathbf{x}^*, \boldsymbol{\mu}^*)$, $\mathbf{y} \neq \mathbf{0}$, we have $\mathbf{y}^\top \mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{y} > 0$.

Then, \mathbf{x}^* is a strict local minimizer of f subject to $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. □

Proof. For a proof of this theorem, we refer the reader to [88, p. 345]. ■

A result similar to Theorem 21.3 holds for a strict local maximizer, the only difference being that we need $\boldsymbol{\mu}^* \leq \mathbf{0}$ and that $\mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ be negative definite on $\tilde{T}(\mathbf{x}^*, \boldsymbol{\mu}^*)$.

Example 21.5 Consider the following problem:

$$\begin{aligned} &\text{minimize} && x_1 x_2 \\ &\text{subject to} && x_1 + x_2 \geq 2 \\ & && x_2 \geq x_1. \end{aligned}$$

- a. Write down the KKT condition for this problem.
- b. Find all points (and KKT multipliers) satisfying the KKT condition. In each case, determine if the point is regular.
- c. Find all points in part b that also satisfy the SONC.
- d. Find all points in part c that also satisfy the SOSC.
- e. Find all points in part c that are local minimizers.

Solution:

- a. Write $f(\mathbf{x}) = x_1 x_2$, $g_1(\mathbf{x}) = 2 - x_1 - x_2$, and $g_2(\mathbf{x}) = x_1 - x_2$. The KKT condition is

$$\begin{aligned}x_2 - \mu_1 + \mu_2 &= 0, \\x_1 - \mu_1 - \mu_2 &= 0, \\\mu_1(2 - x_1 - x_2) + \mu_2(x_1 - x_2) &= 0, \\\mu_1, \mu_2 &\geq 0, \\2 - x_1 - x_2 &\leq 0, \\x_1 - x_2 &\leq 0.\end{aligned}$$

- b. It is easy to check that $\mu_1 \neq 0$ and $\mu_2 \not> 0$. This leaves us with only one solution to the KKT condition: $x_1^* = x_2^* = 1$, $\mu_1^* = 1$, $\mu_2^* = 0$. For this point, we have $Dg_1(\mathbf{x}^*) = [-1, -1]$ and $Dg_2(\mathbf{x}^*) = [1, -1]$. Hence, \mathbf{x}^* is regular.
- c. Both constraints are active. Hence, because \mathbf{x}^* is regular, $T(\mathbf{x}^*) = \{\mathbf{0}\}$. This implies that the SONC is satisfied.

- d. Now,

$$\mathbf{L}(\mathbf{x}^*, \boldsymbol{\mu}^*) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Moreover, $\tilde{T}(\mathbf{x}^*, \boldsymbol{\mu}^*) = \{\mathbf{y} : [-1, -1]\mathbf{y} = 0\} = \{\mathbf{y} : y_1 = -y_2\}$. Pick $\mathbf{y} = [1, -1]^\top \in \tilde{T}(\mathbf{x}^*, \boldsymbol{\mu}^*)$. We have $\mathbf{y}^\top \mathbf{L}(\mathbf{x}^*, \boldsymbol{\mu}^*) \mathbf{y} = -2 < 0$, which means that the SOSC fails.

- e. In fact, the point \mathbf{x}^* is not a local minimizer. To see this, draw a picture of the constraint set and level sets of the objective function. Moving in the feasible direction $[1, 1]^\top$, the objective function increases; but moving in the feasible direction $[-1, 1]^\top$, the objective function decreases.

We now solve analytically the problem in Example 20.1 that we solved graphically earlier. ■

Example 21.6 We wish to minimize $f(\mathbf{x}) = (x_1 - 1)^2 + x_2 - 2$ subject to

$$\begin{aligned} h(\mathbf{x}) &= x_2 - x_1 - 1 = 0, \\ g(\mathbf{x}) &= x_1 + x_2 - 2 \leq 0. \end{aligned}$$

For all $\mathbf{x} \in \mathbb{R}^2$, we have

$$Dh(\mathbf{x}) = [-1, 1], \quad Dg(\mathbf{x}) = [1, 1].$$

Thus, $\nabla h(\mathbf{x})$ and $\nabla g(\mathbf{x})$ are linearly independent and hence all feasible points are regular. We first write the KKT condition. Because $Df(\mathbf{x}) = [2x_1 - 2, 1]$, we have

$$\begin{aligned} Df(\mathbf{x}) + \lambda Dh(\mathbf{x}) + \mu Dg(\mathbf{x}) &= [2x_1 - 2 - \lambda + \mu, 1 + \lambda + \mu] = \mathbf{0}^\top, \\ \mu(x_1 + x_2 - 2) &= 0, \\ \mu &\geq 0, \\ x_2 - x_1 - 1 &= 0, \\ x_1 + x_2 - 2 &\leq 0. \end{aligned}$$

To find points that satisfy the conditions above, we first try $\mu > 0$, which implies that $x_1 + x_2 - 2 = 0$. Thus, we are faced with a system of four linear equations

$$\begin{aligned} 2x_1 - 2 - \lambda + \mu &= 0, \\ 1 + \lambda + \mu &= 0, \\ x_2 - x_1 - 1 &= 0, \\ x_1 + x_2 - 2 &= 0. \end{aligned}$$

Solving the system of equations above, we obtain

$$x_1 = \frac{1}{2}, \quad x_2 = \frac{3}{2}, \quad \lambda = -1, \quad \mu = 0.$$

However, the above is not a legitimate solution to the KKT condition, because we obtained $\mu = 0$, which contradicts the assumption that $\mu > 0$.

In the second try, we assume that $\mu = 0$. Thus, we have to solve the system of equations

$$\begin{aligned} 2x_1 - 2 - \lambda &= 0, \\ 1 + \lambda &= 0, \\ x_2 - x_1 - 1 &= 0, \end{aligned}$$

and the solutions must satisfy

$$g(x_1, x_2) = x_1 + x_2 - 2 \leq 0.$$

Solving the equations above, we obtain

$$x_1 = \frac{1}{2}, \quad x_2 = \frac{3}{2}, \quad \lambda = -1.$$

Note that $\mathbf{x}^* = [1/2, 3/2]^\top$ satisfies the constraint $g(\mathbf{x}^*) \leq 0$. The point \mathbf{x}^* satisfying the KKT necessary condition is therefore the candidate for being a minimizer.

We now verify if $\mathbf{x}^* = [1/2, 3/2]^\top$, $\lambda^* = -1$, $\mu^* = 0$, satisfy the second-order sufficient conditions. For this, we form the matrix

$$\begin{aligned} \mathbf{L}(\mathbf{x}^*, \lambda^*, \mu^*) &= \mathbf{F}(\mathbf{x}^*) + \lambda^* \mathbf{H}(\mathbf{x}^*) + \mu^* \mathbf{G}(\mathbf{x}^*) \\ &= \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} + (-1) \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + (0) \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

We then find the subspace

$$\tilde{T}(\mathbf{x}^*, \mu^*) = \{\mathbf{y} : Dh(\mathbf{x}^*)\mathbf{y} = 0\}.$$

Note that because $\mu^* = 0$, the active constraint $g(\mathbf{x}^*) = 0$ does not enter the computation of $\tilde{T}(\mathbf{x}^*, \mu^*)$. Note also that in this case, $T(\mathbf{x}^*) = \{\mathbf{0}\}$. We have

$$\tilde{T}(\mathbf{x}^*, \mu^*) = \{\mathbf{y} : [-1, 1]\mathbf{y} = 0\} = \{[a, a]^\top : a \in \mathbb{R}\}.$$

We then check for positive definiteness of $\mathbf{L}(\mathbf{x}^*, \lambda^*, \mu^*)$ on $\tilde{T}(\mathbf{x}^*, \mu^*)$. We have

$$\mathbf{y}^\top \mathbf{L}(\mathbf{x}^*, \lambda^*, \mu^*) \mathbf{y} = [a, a] \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} = 2a^2.$$

Thus, $\mathbf{L}(\mathbf{x}^*, \lambda^*, \mu^*)$ is positive definite on $\tilde{T}(\mathbf{x}^*, \mu^*)$. Observe that $\mathbf{L}(\mathbf{x}^*, \lambda^*, \mu^*)$ is, in fact, only positive semidefinite on \mathbb{R}^2 .

By the second-order sufficient conditions, we conclude that $\mathbf{x}^* = [1/2, 3/2]^\top$ is a strict local minimizer. ■

EXERCISES

21.1 Consider the optimization problem

$$\begin{aligned} \text{minimize} \quad & x_1^2 + 4x_2^2 \\ \text{subject to} \quad & x_1^2 + 2x_2^2 \geq 4. \end{aligned}$$

- a. Find all the points that satisfy the KKT conditions.

- b. Apply the SOSC to determine the nature of the critical points from the previous part.

21.2 Find local extremizers for:

- a. $x_1^2 + x_2^2 - 2x_1 - 10x_2 + 26$ subject to $\frac{1}{5}x_2 - x_1^2 \leq 0$, $5x_1 + \frac{1}{2}x_2 \leq 5$.
- b. $x_1^2 + x_2^2$ subject to $x_1 \geq 0$, $x_2 \geq 0$, $x_1 + x_2 \geq 5$.
- c. $x_1^2 + 6x_1x_2 - 4x_1 - 2x_2$ subject to $x_1^2 + 2x_2 \leq 1$, $2x_1 - 2x_2 \leq 1$.

21.3 Find local minimizers for $x_1^2 + x_2^2$ subject to $x_1^2 + 2x_1x_2 + x_2^2 = 1$, $x_1^2 - x_2 \leq 0$.

21.4 Write down the Karush-Kuhn-Tucker condition for the optimization problem in Exercise 15.8.

21.5 Consider the problem

$$\begin{aligned} & \text{minimize} && x_2 - (x_1 - 2)^3 + 3 \\ & \text{subject to} && x_2 \geq 1, \end{aligned}$$

where x_1 and x_2 are real variables. Answer each of the following questions, making sure that you give complete reasoning for your answers.

- a. Write down the KKT condition for the problem, and find all points that satisfy the condition. Check whether or not each point is regular.
- b. Determine whether or not the point(s) in part a satisfy the second-order necessary condition.
- c. Determine whether or not the point(s) in part b satisfy the second-order sufficient condition.

21.6 Consider the problem

$$\begin{aligned} & \text{minimize} && x_2 \\ & \text{subject to} && x_2 \geq -(x_1 - 1)^2 + 3. \end{aligned}$$

- a. Find all points satisfying the KKT condition for the problem.
- b. For each point \mathbf{x}^* in part a, find $T(\mathbf{x}^*)$, $N(\mathbf{x}^*)$, and $\tilde{T}(\mathbf{x}^*)$.
- c. Find the subset of points from part a that satisfy the second-order necessary condition.

21.7 Consider the problem of optimizing (either minimizing or maximizing) $(x_1 - 2)^2 + (x_2 - 1)^2$ subject to

$$\begin{aligned}x_2 - x_1^2 &\geq 0 \\2 - x_1 - x_2 &\geq 0 \\x_1 &\geq 0.\end{aligned}$$

The point $\mathbf{x}^* = \mathbf{0}$ satisfies the KKT conditions.

- a. Does \mathbf{x}^* satisfy the FONC for minimization or maximization? What are the KKT multipliers?
- b. Does \mathbf{x}^* satisfy the SOSC? Carefully justify your answer.

21.8 Consider the optimization problem

$$\begin{aligned}\text{minimize } & f(\mathbf{x}) \\ \text{subject to } & \mathbf{x} \in \Omega,\end{aligned}$$

where $f(\mathbf{x}) = x_1 x_2^2$, where $\mathbf{x} = [x_1, x_2]^\top$, and $\Omega = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = x_2, x_1 \geq 0\}$.

- a. Find all points satisfying the KKT condition.
- b. Do each of the points found in part a satisfy the second-order necessary condition?
- c. Do each of the points found in part a satisfy the second-order sufficient condition?

21.9 Consider the problem

$$\begin{aligned}\text{minimize } & \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|^2 \\ \text{subject to } & x_1 + \cdots + x_n = 1 \\ & x_1, \dots, x_n \geq 0.\end{aligned}$$

- a. Write down the KKT condition for the problem.
- b. Define what it means for a feasible point \mathbf{x}^* to be *regular* in this particular problem. Are there any feasible points in this problem that are not regular? If yes, find them. If not, explain why not.

21.10 Let $g : \mathbb{R}^n \rightarrow \mathbb{R}$ and $\mathbf{x}_0 \in \mathbb{R}^n$ be given, where $g(\mathbf{x}_0) > 0$. Consider the problem

$$\begin{aligned} &\text{minimize} && \frac{1}{2} \|\mathbf{x} - \mathbf{x}_0\|^2 \\ &\text{subject to} && g(\mathbf{x}) \leq 0. \end{aligned}$$

Suppose that \mathbf{x}^* is a solution to the problem and $g \in \mathcal{C}^1$. Use the KKT theorem to decide which of the following equations/inequalities hold:

- i. $g(\mathbf{x}^*) < 0$.
- ii. $g(\mathbf{x}^*) = 0$.
- iii. $(\mathbf{x}^* - \mathbf{x}_0)^\top \nabla g(\mathbf{x}^*) < 0$.
- iv. $(\mathbf{x}^* - \mathbf{x}_0)^\top \nabla g(\mathbf{x}^*) = 0$.
- v. $(\mathbf{x}^* - \mathbf{x}_0)^\top \nabla g(\mathbf{x}^*) > 0$.

21.11 Consider a square room with corners located at $[0, 0]^\top$, $[0, 2]^\top$, $[2, 0]^\top$, and $[2, 2]^\top$ (in \mathbb{R}^2). We wish to find the point in the room that is closest to the point $[3, 4]^\top$.

- a. Guess which point in the room is the closest point in the room to the point $[3, 4]^\top$.
- b. Use the second-order sufficient conditions to prove that the point you have guessed is a strict local minimizer.

Hint: Minimizing the distance is the same as minimizing the square distance.

21.12 Consider the *quadratic programming problem*

$$\begin{aligned} &\text{minimize} && \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} \\ &\text{subject to} && \mathbf{A} \mathbf{x} \leq \mathbf{b}, \end{aligned}$$

where $\mathbf{Q} = \mathbf{Q}^\top > 0$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, and $\mathbf{b} \geq \mathbf{0}$. Find all points satisfying the KKT condition.

21.13 Consider the linear programming problem

$$\begin{aligned} &\text{minimize} && ax_1 + bx_2 \\ &\text{subject to} && cx_1 + dx_2 = e \\ & && x_1, x_2 \geq 0, \end{aligned}$$

where $a, b, c, d, e \in \mathbb{R}$ are all nonzero constants. Suppose that \mathbf{x}^* is an optimal basic feasible solution to the problem.

- a. Write down the Karush-Kuhn-Tucker condition involving \mathbf{x}^* (specifying clearly the number of Lagrange and KKT multipliers).
- b. Is \mathbf{x}^* regular? Explain.
- c. Find the tangent space $T(\mathbf{x}^*)$ (defined by the active constraints) for this problem.
- d. Assume that the relative cost coefficients of all nonbasic variables are strictly positive. Does \mathbf{x}^* satisfy the second-order sufficient condition? Explain.

21.14 Consider the problem

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{0}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m < n$, is of full rank. Use the KKT theorem to show that if there exists a solution, then the optimal objective function value is 0.

21.15 Consider a linear programming problem in standard form (see Chapter 15).

- a. Write down the Karush-Kuhn-Tucker condition for the problem.
- b. Use part a to show that if there exists an optimal feasible solution to the linear program, then there exists a feasible solution to the corresponding dual problem that achieves an objective function value that is the same as the optimal value of the primal (compare this with Theorem 17.1).
- c. Use parts a and b to prove that if \mathbf{x}^* is an optimal feasible solutions of the primal, then there exists a feasible solution $\boldsymbol{\lambda}^*$ to the dual such that $(\mathbf{c}^\top - \boldsymbol{\lambda}^{*\top} \mathbf{A})\mathbf{x}^* = 0$ (compare this with Theorem 17.3).

21.16 Consider the constraint set $S = \{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$. Let $\mathbf{x}^* \in S$ be a regular local minimizer of f over S and $J(\mathbf{x}^*)$ the index set of active inequality constraints. Show that \mathbf{x}^* is also a regular local minimizer of f over the set $S' = \{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}, g_j(\mathbf{x}) = 0, j \in J(\mathbf{x}^*)\}$.

21.17 Solve the following optimization problem using the second-order sufficient conditions:

$$\begin{aligned} & \text{minimize} && x_1^2 + x_2^2 \\ & \text{subject to} && x_1^2 - x_2 - 4 \leq 0 \\ & && x_2 - x_1 - 2 \leq 0. \end{aligned}$$

See Figure 22.1 for a graphical illustration of the problem.

21.18 Solve the following optimization problem using the second-order sufficient conditions:

$$\begin{aligned} & \text{minimize} && x_1^2 + x_2^2 \\ & \text{subject to} && x_1 - x_2^2 - 4 \geq 0 \\ & && x_1 - 10 \leq 0. \end{aligned}$$

See Figure 22.2 for a graphical illustration of the problem.

21.19 Consider the problem

$$\begin{aligned} & \text{minimize} && x_1^2 + x_2^2 \\ & \text{subject to} && 4 - x_1 - x_2^2 \leq 0 \\ & && 3x_2 - x_1 \leq 0 \\ & && -3x_2 - x_1 \leq 0. \end{aligned}$$

Figure 22.3 gives a graphical illustration of the problem. Deduce from the figure that the problem has two strict local minimizers, and use the second-order sufficient conditions to verify the graphical solutions.

21.20 Consider the following optimization problem with an inequality constraint:

$$\begin{aligned} & \text{minimize} && 3x_1 \\ & \text{subject to} && x_1 + x_2^2 \geq 2. \end{aligned}$$

- a. Does the point $\mathbf{x}^* = [2, 0]^\top$ satisfy the KKT (first-order necessary) condition?
- b. Does the point $\mathbf{x}^* = [2, 0]^\top$ satisfy the second-order necessary condition (for problems with inequality constraints)?
- c. Is the point $\mathbf{x}^* = [2, 0]^\top$ a local minimizer?

(See Exercise 6.15 for a similar problem treated using set-constrained methods.)

21.21 Consider the problem

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|\mathbf{x}\|^2 \\ & \text{subject to} && \mathbf{a}^\top \mathbf{x} = b \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{a} \geq \mathbf{0}$, and $b \in \mathbb{R}$, $b > 0$. Show that if a solution to the problem exists, then it is unique, and find an expression for it in terms of \mathbf{a} and b .

21.22 Consider the problem

$$\begin{aligned} &\text{minimize} && (x_1 - a)^2 + (x_2 - b)^2, \quad x_1, x_2 \in \mathbb{R} \\ &\text{subject to} && x_1^2 + x_2^2 \leq 1, \end{aligned}$$

where $a, b \in \mathbb{R}$ are given constants satisfying $a^2 + b^2 \geq 1$.

- a. Let $\mathbf{x}^* = [x_1^*, x_2^*]^\top$ be a solution to the problem. Use the first-order necessary conditions for unconstrained optimization to show that $(x_1^*)^2 + (x_2^*)^2 = 1$.
- b. Use the KKT theorem to show that the solution $\mathbf{x}^* = [x_1^*, x_2^*]^\top$ is unique and has the form $x_1^* = \alpha a$, $x_2^* = \alpha b$, where $\alpha \in \mathbb{R}$ is a positive constant.
- c. Find an expression for α (from part b) in terms of a and b .

21.23 Consider the problem

$$\begin{aligned} &\text{minimize} && x_1^2 + (x_2 + 1)^2, \quad x_1, x_2 \in \mathbb{R} \\ &\text{subject to} && x_2 \geq \exp(x_1) \end{aligned}$$

[$\exp(x) = e^x$ is the exponential of x]. Let $\mathbf{x}^* = [x_1^*, x_2^*]^\top$ be the solution to the problem.

- a. Write down the KKT condition that must be satisfied by \mathbf{x}^* .
- b. Prove that $x_2^* = \exp(x_1^*)$.
- c. Prove that $-2 < x_1^* < 0$.

21.24 Consider the problem

$$\begin{aligned} &\text{minimize} && \mathbf{c}^\top \mathbf{x} + 8 \\ &\text{subject to} && \frac{1}{2} \|\mathbf{x}\|^2 \leq 1, \end{aligned}$$

where $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{c} \neq \mathbf{0}$. Suppose that $\mathbf{x}^* = \alpha \mathbf{e}$ is a solution to the problem, where $\alpha \in \mathbb{R}$ and $\mathbf{e} = [1, \dots, 1]^\top$, and the corresponding objective value is 4.

- a. Show that $\|\mathbf{x}^*\|^2 = 2$.
- b. Find α and \mathbf{c} (they may depend on n).

21.25 Consider the problem with equality constraint

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0}. \end{aligned}$$

We can convert the above into the equivalent optimization problem

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \frac{1}{2} \|\mathbf{h}(\mathbf{x})\|^2 \leq 0. \end{aligned}$$

Write down the KKT condition for the equivalent problem (with inequality constraint) and explain why the KKT theorem cannot be applied in this case.

CHAPTER 22

CONVEX OPTIMIZATION PROBLEMS

22.1 Introduction

The optimization problems posed at the beginning of this part are, in general, very difficult to solve. The source of these difficulties may be in the objective function or the constraints. Even if the objective function is simple and “well-behaved,” the nature of the constraints may make the problem difficult to solve. We illustrate some of these difficulties in the following examples.

Example 22.1 Consider the optimization problem

$$\begin{aligned} & \text{minimize} && x_1^2 + x_2^2 \\ & \text{subject to} && x_2 - x_1 - 2 \leq 0 \\ & && x_1^2 - x_2 - 4 \leq 0. \end{aligned}$$

The problem is depicted in Figure 22.1, where, as we can see, the constrained minimizer is the same as the unconstrained minimizer. At the minimizer, all the constraints are inactive. If we had only known this fact, we could have approached this problem as an unconstrained optimization problem using techniques from Part II. ■

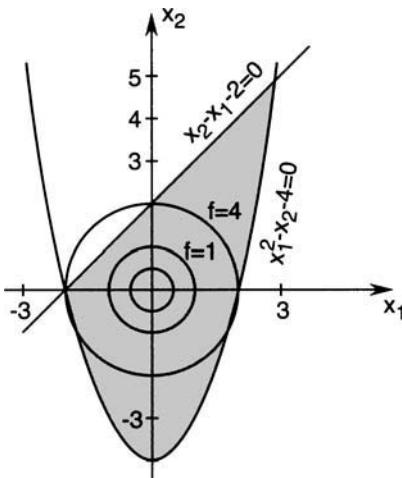


Figure 22.1 Situation where the constrained and the unconstrained minimizers are the same.

Example 22.2 Consider the optimization problem

$$\begin{aligned} & \text{minimize} && x_1^2 + x_2^2 \\ & \text{subject to} && x_1 - 10 \leq 0 \\ & && x_1 - x_2 - 4 \geq 0. \end{aligned}$$

The problem is depicted in Figure 22.2. At the solution, only one constraint is active. If we had only known about this we could have handled this problem as a constrained optimization problem using the Lagrange multiplier method. ■

Example 22.3 Consider the optimization problem

$$\begin{aligned} & \text{minimize} && x_1^2 + x_2^2 \\ & \text{subject to} && 4 - x_1 - x_2^2 \leq 0 \\ & && 3x_2 - x_1 \leq 0 \\ & && -3x_2 - x_1 \leq 0. \end{aligned}$$

The problem is depicted in Figure 22.3. This example illustrates the situation where the constraints introduce local minimizers, even though the objective function itself has only one unconstrained global minimizer. ■

Some of the difficulties illustrated in the examples above can be eliminated if we restrict our problems to convex feasible regions. Admittedly, some important real-life problems do not fit into this framework. On the other hand,

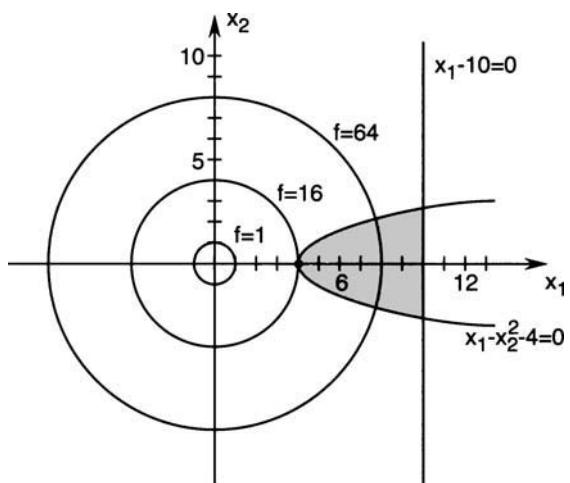


Figure 22.2 Situation where only one constraint is active.

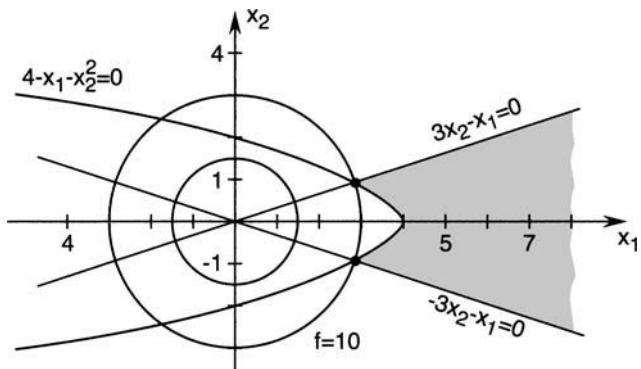


Figure 22.3 Situation where the constraints introduce local minimizers.

it is possible to give results of a *global* nature for this class of optimization problems. In the next section, we introduce the notion of a *convex function*, which plays an important role in our subsequent treatment of such problems.

22.2 Convex Functions

We begin with a definition of the graph of a real-valued function.

Definition 22.1 The *graph* of $f : \Omega \rightarrow \mathbb{R}$, $\Omega \subset \mathbb{R}^n$, is the set of points in $\Omega \times \mathbb{R} \subset \mathbb{R}^{n+1}$ given by

$$\left\{ \begin{bmatrix} \mathbf{x} \\ f(\mathbf{x}) \end{bmatrix} : \mathbf{x} \in \Omega \right\}.$$

■

We can visualize the graph of f as simply the set of points on a “plot” of $f(\mathbf{x})$ versus \mathbf{x} (see Figure 22.4). We next define the epigraph of a real-valued function.

Definition 22.2 The *epigraph* of a function $f : \Omega \rightarrow \mathbb{R}$, $\Omega \subset \mathbb{R}^n$, denoted $\text{epi}(f)$, is the set of points in $\Omega \times \mathbb{R}$ given by

$$\text{epi}(f) = \left\{ \begin{bmatrix} \mathbf{x} \\ \beta \end{bmatrix} : \mathbf{x} \in \Omega, \beta \in \mathbb{R}, \beta \geq f(\mathbf{x}) \right\}.$$

■

The epigraph $\text{epi}(f)$ of a function f is simply the set of points in $\Omega \times \mathbb{R}$ on or above the graph of f (see Figure 22.4). We can also think of $\text{epi}(f)$ as a subset of \mathbb{R}^{n+1} .

Recall that a set $\Omega \subset \mathbb{R}^n$ is convex if for every $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$ and $\alpha \in (0, 1)$, $\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2 \in \Omega$ (see Section 4.3). We now introduce the notion of a convex function.

Definition 22.3 A function $f : \Omega \rightarrow \mathbb{R}$, $\Omega \subset \mathbb{R}^n$, is *convex* on Ω if its epigraph is a convex set. ■

Theorem 22.1 If a function $f : \Omega \rightarrow \mathbb{R}$, $\Omega \subset \mathbb{R}^n$, is convex on Ω , then Ω is a convex set. □

Proof. We prove this theorem by contraposition. Suppose that Ω is not a convex set. Then, there exist two points \mathbf{y}_1 and \mathbf{y}_2 such that for some $\alpha \in (0, 1)$,

$$\mathbf{z} = \alpha\mathbf{y}_1 + (1 - \alpha)\mathbf{y}_2 \notin \Omega.$$

Let

$$\beta_1 = f(\mathbf{y}_1), \quad \beta_2 = f(\mathbf{y}_2).$$

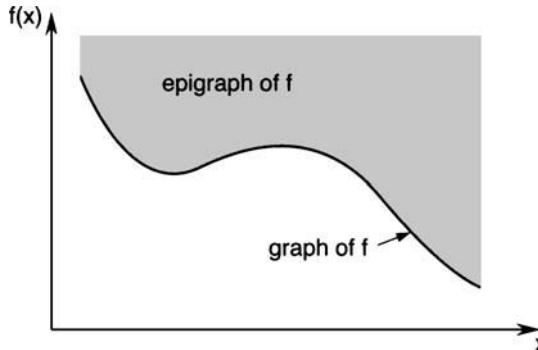


Figure 22.4 Graph and epigraph of a function $f : \mathbb{R} \rightarrow \mathbb{R}$.

Then, the pairs

$$\begin{bmatrix} \mathbf{y}_1 \\ \beta_1 \end{bmatrix}, \begin{bmatrix} \mathbf{y}_2 \\ \beta_2 \end{bmatrix}$$

belong to the graph of f , and hence also the epigraph of f . Let

$$\mathbf{w} = \alpha \begin{bmatrix} \mathbf{y}_1 \\ \beta_1 \end{bmatrix} + (1 - \alpha) \begin{bmatrix} \mathbf{y}_2 \\ \beta_2 \end{bmatrix}.$$

We have

$$\mathbf{w} = \begin{bmatrix} \mathbf{z} \\ \alpha\beta_1 + (1 - \alpha)\beta_2 \end{bmatrix}.$$

But note that $\mathbf{w} \notin \text{epi}(f)$, because $\mathbf{z} \notin \Omega$. Therefore, $\text{epi}(f)$ is not convex, and hence f is not a convex function. ■

The next theorem gives a very useful characterization of convex functions. This characterization is often used as a definition for a convex function.

Theorem 22.2 *A function $f : \Omega \rightarrow \mathbb{R}$ defined on a convex set $\Omega \subset \mathbb{R}^n$ is convex if and only if for all $\mathbf{x}, \mathbf{y} \in \Omega$ and all $\alpha \in (0, 1)$, we have*

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y}).$$

□

Proof. \Leftarrow : Assume that for all $\mathbf{x}, \mathbf{y} \in \Omega$ and $\alpha \in (0, 1)$,

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y}).$$

Let $[\mathbf{x}^\top, a]^\top$ and $[\mathbf{y}^\top, b]^\top$ be two points in $\text{epi}(f)$, where $a, b \in \mathbb{R}$. From the definition of $\text{epi}(f)$ it follows that

$$f(\mathbf{x}) \leq a, \quad f(\mathbf{y}) \leq b.$$

Therefore, using the first inequality above, we have

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \alpha a + (1 - \alpha)b.$$

Because Ω is convex, $\alpha\mathbf{x} + (1 - \alpha)\mathbf{y} \in \Omega$. Hence,

$$\begin{bmatrix} \alpha\mathbf{x} + (1 - \alpha)\mathbf{y} \\ \alpha a + (1 - \alpha)b \end{bmatrix} \in \text{epi}(f),$$

which implies that $\text{epi}(f)$ is a convex set, and hence f is a convex function.

\Rightarrow : Assume that $f : \Omega \rightarrow \mathbb{R}$ is a convex function. Let $\mathbf{x}, \mathbf{y} \in \Omega$ and

$$f(\mathbf{x}) = a, \quad f(\mathbf{y}) = b.$$

Thus,

$$\begin{bmatrix} \mathbf{x} \\ a \end{bmatrix}, \begin{bmatrix} \mathbf{y} \\ b \end{bmatrix} \in \text{epi}(f).$$

Because f is a convex function, its epigraph is a convex subset of \mathbb{R}^{n+1} . Therefore, for all $\alpha \in (0, 1)$, we have

$$\alpha \begin{bmatrix} \mathbf{x} \\ a \end{bmatrix} + (1 - \alpha) \begin{bmatrix} \mathbf{y} \\ b \end{bmatrix} = \begin{bmatrix} \alpha\mathbf{x} + (1 - \alpha)\mathbf{y} \\ \alpha a + (1 - \alpha)b \end{bmatrix} \in \text{epi}(f).$$

The above implies that for all $\alpha \in (0, 1)$,

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \alpha a + (1 - \alpha)b = \alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y}).$$

This completes the proof. ■

A geometric interpretation of Theorem 22.2 is given in Figure 22.5. The theorem states that if $f : \Omega \rightarrow \mathbb{R}$ is a convex function over a convex set Ω , then for all $\mathbf{x}, \mathbf{y} \in \Omega$, the points on the line segment in \mathbb{R}^{n+1} connecting $[\mathbf{x}^\top, f(\mathbf{x})]^\top$ and $[\mathbf{y}^\top, f(\mathbf{y})]^\top$ must lie on or above the graph of f .

Using Theorem 22.2, it is straightforward to show that any nonnegative scaling of a convex function is convex, and that the sum of convex functions is convex.

Theorem 22.3 Suppose that f , f_1 , and f_2 are convex functions. Then, for any $a \geq 0$, the function af is convex. Moreover, $f_1 + f_2$ is convex. □

Proof. Let $\mathbf{x}, \mathbf{y} \in \Omega$ and $\alpha \in (0, 1)$. Fix $a \geq 0$. For convenience, write $\bar{f} = af$. We have

$$\begin{aligned} \bar{f}(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) &= af(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \\ &\leq a(\alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y})) \text{ because } f \text{ is convex and } a \geq 0 \\ &= \alpha(af(\mathbf{x})) + (1 - \alpha)(af(\mathbf{y})) \\ &= \alpha\bar{f}(\mathbf{x}) + (1 - \alpha)\bar{f}(\mathbf{y}), \end{aligned}$$

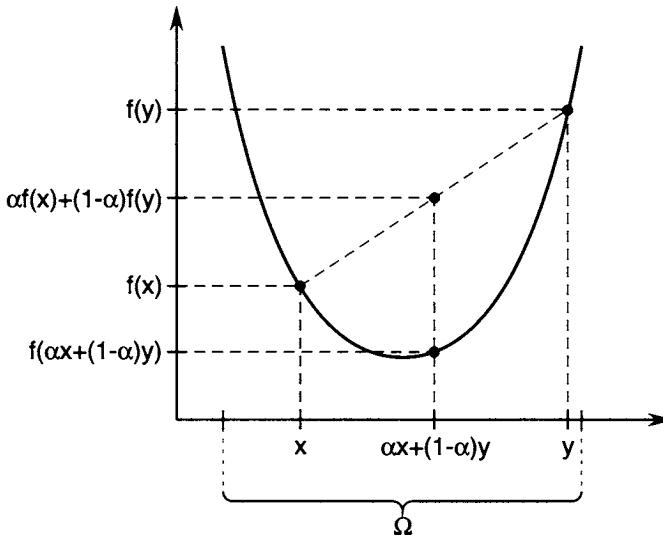


Figure 22.5 Geometric interpretation of Theorem 22.2.

which implies that \bar{f} is convex.

Next, write $f_3 = f_1 + f_2$. We have

$$\begin{aligned}
 f_3(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) &= f_1(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) + f_2(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \\
 &\leq (\alpha f_1(\mathbf{x}) + (1 - \alpha)f_1(\mathbf{y})) + (\alpha f_2(\mathbf{x}) + (1 - \alpha)f_2(\mathbf{y})) \\
 &\quad \text{by convexity of } f_1 \text{ and } f_2 \\
 &= \alpha(f_1(\mathbf{x}) + f_2(\mathbf{x})) + (1 - \alpha)(f_1(\mathbf{y}) + f_2(\mathbf{y})) \\
 &= \alpha f_3(\mathbf{x}) + (1 - \alpha)f_3(\mathbf{y}),
 \end{aligned}$$

which implies that f_3 is convex. ■

Theorem 22.3 implies that for any given collection of convex functions f_1, \dots, f_ℓ and nonnegative numbers c_1, \dots, c_ℓ , the function $c_1 f_1 + \dots + c_\ell f_\ell$ is convex. Using a method of proof similar to that used in Theorem 22.3, it is similarly straightforward to show that the function $\max\{f_1, \dots, f_\ell\}$ is convex (see Exercise 22.6).

We now define the notion of strict convexity.

Definition 22.4 A function $f : \Omega \rightarrow \mathbb{R}$ on a convex set $\Omega \subset \mathbb{R}^n$ is *strictly convex* if for all $\mathbf{x}, \mathbf{y} \in \Omega$, $\mathbf{x} \neq \mathbf{y}$, and $\alpha \in (0, 1)$, we have

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) < \alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y}).$$
■

From this definition, we see that for a strictly convex function, all points on the open line segment connecting the points $[\mathbf{x}^\top, f(\mathbf{x})]^\top$ and $[\mathbf{y}^\top, f(\mathbf{y})]^\top$ lie (strictly) above the graph of f .

Definition 22.5 A function $f : \Omega \rightarrow \mathbb{R}$ on a convex set $\Omega \subset \mathbb{R}^n$ is (strictly) *concave* if $-f$ is (strictly) convex. ■

Note that the graph of a strictly concave function always lies above the line segment connecting any two points on its graph.

To show that a function is not convex, we need only produce a pair of points $\mathbf{x}, \mathbf{y} \in \Omega$ and an $\alpha \in (0, 1)$ such that the inequality in Theorem 22.2 is violated.

Example 22.4 Let $f(\mathbf{x}) = x_1 x_2$. Is f convex over $\Omega = \{\mathbf{x} : x_1 \geq 0, x_2 \geq 0\}$?

The answer is no. Take, for example, $\mathbf{x} = [1, 2]^\top \in \Omega$ and $\mathbf{y} = [2, 1]^\top \in \Omega$. Then,

$$\alpha\mathbf{x} + (1 - \alpha)\mathbf{y} = \begin{bmatrix} 2 - \alpha \\ 1 + \alpha \end{bmatrix}.$$

Hence,

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) = (2 - \alpha)(1 + \alpha) = 2 + \alpha - \alpha^2$$

and

$$\alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y}) = 2.$$

If, for example, $\alpha = 1/2 \in (0, 1)$, then

$$f\left(\frac{1}{2}\mathbf{x} + \frac{1}{2}\mathbf{y}\right) = \frac{9}{4} > \frac{1}{2}f(\mathbf{x}) + \frac{1}{2}f(\mathbf{y}),$$

which shows that f is not convex over Ω . ■

Example 22.4 is an illustration of the following general result.

Proposition 22.1 A quadratic form $f : \Omega \rightarrow \mathbb{R}$, $\Omega \subset \mathbb{R}^n$, given by $f(\mathbf{x}) = \mathbf{x}^\top Q\mathbf{x}$, $Q \in \mathbb{R}^{n \times n}$, $Q = Q^\top$, is convex on Ω if and only if for all $\mathbf{x}, \mathbf{y} \in \Omega$, $(\mathbf{x} - \mathbf{y})^\top Q(\mathbf{x} - \mathbf{y}) \geq 0$. □

Proof. The result follows from Theorem 22.2. Indeed, the function $f(\mathbf{x}) = \mathbf{x}^\top Q\mathbf{x}$ is convex if and only if for every $\alpha \in (0, 1)$, and every $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we have

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y}),$$

or, equivalently,

$$\alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y}) - f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \geq 0.$$

Substituting for f into the left-hand side of this equation yields

$$\begin{aligned} \alpha\mathbf{x}^\top Q\mathbf{x} + (1-\alpha)\mathbf{y}^\top Q\mathbf{y} - (\alpha\mathbf{x} + (1-\alpha)\mathbf{y})^\top Q(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \\ = \alpha\mathbf{x}^\top Q\mathbf{x} + \mathbf{y}^\top Q\mathbf{y} - \alpha\mathbf{y}^\top Q\mathbf{y} - \alpha^2\mathbf{x}^\top Q\mathbf{x} \\ - (2\alpha - 2\alpha^2)\mathbf{x}^\top Q\mathbf{y} - (1 - 2\alpha + \alpha^2)\mathbf{y}^\top Q\mathbf{y} \\ = \alpha(1-\alpha)\mathbf{x}^\top Q\mathbf{x} - 2\alpha(1-\alpha)\mathbf{x}^\top Q\mathbf{y} + \alpha(1-\alpha)\mathbf{y}^\top Q\mathbf{y} \\ = \alpha(1-\alpha)(\mathbf{x} - \mathbf{y})^\top Q(\mathbf{x} - \mathbf{y}). \end{aligned}$$

Therefore, f is convex if and only if

$$\alpha(1-\alpha)(\mathbf{x} - \mathbf{y})^\top Q(\mathbf{x} - \mathbf{y}) \geq 0,$$

which proves the result. ■

Example 22.5 In Example 22.4, $f(\mathbf{x}) = x_1x_2$, which can be written as $f(\mathbf{x}) = \mathbf{x}^\top Q\mathbf{x}$, where

$$Q = \frac{1}{2} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Let $\Omega = \{\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}$, and $\mathbf{x} = [2, 2]^\top \in \Omega$, $\mathbf{y} = [1, 3]^\top \in \Omega$. We have

$$\mathbf{y} - \mathbf{x} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

and

$$(\mathbf{y} - \mathbf{x})^\top Q(\mathbf{y} - \mathbf{x}) = \frac{1}{2}[-1, 1] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = -1 < 0.$$

Hence, by Proposition 22.1, f is not convex on Ω . ■

Differentiable convex functions can be characterized using the following theorem.

Theorem 22.4 Let $f : \Omega \rightarrow \mathbb{R}$, $f \in \mathcal{C}^1$, be defined on an open convex set $\Omega \subset \mathbb{R}^n$. Then, f is convex on Ω if and only if for all $\mathbf{x}, \mathbf{y} \in \Omega$,

$$f(\mathbf{y}) \geq f(\mathbf{x}) + Df(\mathbf{x})(\mathbf{y} - \mathbf{x}).$$

□

Proof. \Rightarrow : Suppose that $f : \Omega \rightarrow \mathbb{R}$ is differentiable and convex. Then, by Theorem 22.2, for any $\mathbf{y}, \mathbf{x} \in \Omega$ and $\alpha \in (0, 1)$ we have

$$f(\alpha\mathbf{y} + (1-\alpha)\mathbf{x}) \leq \alpha f(\mathbf{y}) + (1-\alpha)f(\mathbf{x}).$$

Rearranging terms yields

$$f(\mathbf{x} + \alpha(\mathbf{y} - \mathbf{x})) - f(\mathbf{x}) \leq \alpha(f(\mathbf{y}) - f(\mathbf{x})).$$

Upon dividing both sides of this inequality by α , we get

$$\frac{f(\mathbf{x} + \alpha(\mathbf{y} - \mathbf{x})) - f(\mathbf{x})}{\alpha} \leq f(\mathbf{y}) - f(\mathbf{x}).$$

If we now take the limit as $\alpha \rightarrow 0$ and apply the definition of the directional derivative of f at \mathbf{x} in the direction $\mathbf{y} - \mathbf{x}$ (see Section 6.2), we get

$$Df(\mathbf{x})(\mathbf{y} - \mathbf{x}) \leq f(\mathbf{y}) - f(\mathbf{x})$$

or

$$f(\mathbf{y}) \geq f(\mathbf{x}) + Df(\mathbf{x})(\mathbf{y} - \mathbf{x}).$$

\Leftarrow : Assume that Ω is convex, $f : \Omega \rightarrow \mathbb{R}$ is differentiable, and for all $\mathbf{x}, \mathbf{y} \in \Omega$,

$$f(\mathbf{y}) \geq f(\mathbf{x}) + Df(\mathbf{x})(\mathbf{y} - \mathbf{x}).$$

Let $\mathbf{u}, \mathbf{v} \in \Omega$ and $\alpha \in (0, 1)$. Because Ω is convex,

$$\mathbf{w} = \alpha\mathbf{u} + (1 - \alpha)\mathbf{v} \in \Omega.$$

We also have

$$f(\mathbf{u}) \geq f(\mathbf{w}) + Df(\mathbf{w})(\mathbf{u} - \mathbf{w})$$

and

$$f(\mathbf{v}) \geq f(\mathbf{w}) + Df(\mathbf{w})(\mathbf{v} - \mathbf{w}).$$

Multiplying the first of these inequalities by α and the second by $(1 - \alpha)$ and then adding them together yields

$$\alpha f(\mathbf{u}) + (1 - \alpha)f(\mathbf{v}) \geq f(\mathbf{w}) + Df(\mathbf{w})(\alpha\mathbf{u} + (1 - \alpha)\mathbf{v} - \mathbf{w}).$$

But

$$\mathbf{w} = \alpha\mathbf{u} + (1 - \alpha)\mathbf{v}.$$

Hence,

$$\alpha f(\mathbf{u}) + (1 - \alpha)f(\mathbf{v}) \geq f(\alpha\mathbf{u} + (1 - \alpha)\mathbf{v}).$$

Hence, by Theorem 22.2, f is a convex function. ■

In Theorem 22.4, the assumption that Ω be open is not necessary, as long as $f \in \mathcal{C}^1$ on some open set that contains Ω (e.g., $f \in \mathcal{C}^1$ on \mathbb{R}^n).

A geometric interpretation of Theorem 22.4 is given in Figure 22.6. To explain the interpretation, let $\mathbf{x}_0 \in \Omega$. The function $\ell(\mathbf{x}) = f(\mathbf{x}_0) + Df(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$ is the linear approximation to f at \mathbf{x}_0 . The theorem says that the graph of f always lies above its linear approximation at any point. In other words,

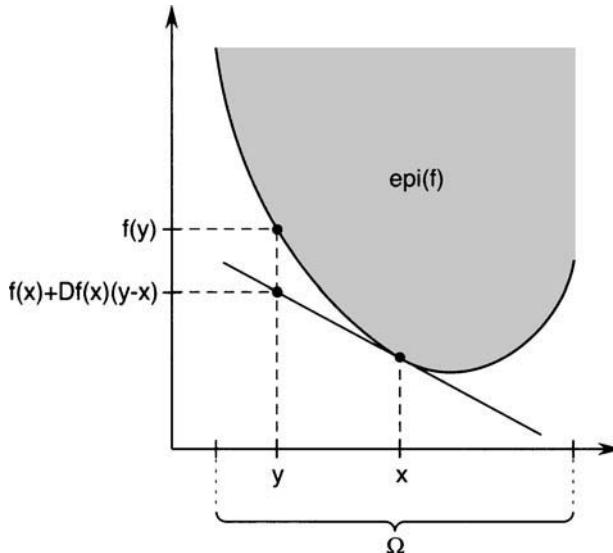


Figure 22.6 Geometric interpretation of Theorem 22.4.

the linear approximation to a convex function f at any point of its domain lies below $\text{epi}(f)$.

This geometric idea leads to a generalization of the gradient to the case where f is not differentiable. Let $f : \Omega \rightarrow \mathbb{R}$ be defined on an open convex set $\Omega \subset \mathbb{R}^n$. A vector $\mathbf{g} \in \mathbb{R}^n$ is said to be a *subgradient* of f at a point $\mathbf{x} \in \Omega$ if for all $\mathbf{y} \in \Omega$,

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{g}^\top (\mathbf{y} - \mathbf{x}).$$

As in the case of the standard gradient, if \mathbf{g} is a subgradient, then for a given $\mathbf{x}_0 \in \Omega$, the function $\ell(\mathbf{x}) = f(\mathbf{x}_0) + \mathbf{g}^\top (\mathbf{x} - \mathbf{x}_0)$ lies below $\text{epi}(f)$.

For functions that are twice continuously differentiable, the following theorem gives another possible characterization of convexity.

Theorem 22.5 *Let $f : \Omega \rightarrow \mathbb{R}$, $f \in C^2$, be defined on an open convex set $\Omega \subset \mathbb{R}^n$. Then, f is convex on Ω if and only if for each $\mathbf{x} \in \Omega$, the Hessian $\mathbf{F}(\mathbf{x})$ of f at \mathbf{x} is a positive semidefinite matrix.* \square

Proof. \Leftarrow : Let $\mathbf{x}, \mathbf{y} \in \Omega$. Because $f \in C^2$, by Taylor's theorem there exists $\alpha \in (0, 1)$ such that

$$f(\mathbf{y}) = f(\mathbf{x}) + Df(\mathbf{x})(\mathbf{y} - \mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^\top \mathbf{F}(\mathbf{x} + \alpha(\mathbf{y} - \mathbf{x}))(\mathbf{y} - \mathbf{x}).$$

Because $\mathbf{F}(\mathbf{x} + \alpha(\mathbf{y} - \mathbf{x}))$ is positive semidefinite,

$$(\mathbf{y} - \mathbf{x})^\top \mathbf{F}(\alpha\mathbf{y} + (1 - \alpha)\mathbf{x})(\mathbf{y} - \mathbf{x}) \geq 0.$$

Therefore, we have

$$f(\mathbf{y}) \geq f(\mathbf{x}) + Df(\mathbf{x})(\mathbf{y} - \mathbf{x}),$$

which implies that f is convex, by Theorem 22.4.

\Rightarrow : We use contraposition. Assume that there exists $\mathbf{x} \in \Omega$ such that $\mathbf{F}(\mathbf{x})$ is not positive semidefinite. Therefore, there exists $\mathbf{d} \in \mathbb{R}^n$ such that $\mathbf{d}^\top \mathbf{F}(\mathbf{x}) \mathbf{d} < 0$. By assumption, Ω is open; thus, the point \mathbf{x} is an interior point. By the continuity of the Hessian matrix, there exists a nonzero $s \in \mathbb{R}$ such that $\mathbf{x} + s\mathbf{d} \in \Omega$, and if we write $\mathbf{y} = \mathbf{x} + s\mathbf{d}$, then for all points \mathbf{z} on the line segment joining \mathbf{x} and \mathbf{y} , we have $\mathbf{d}^\top \mathbf{F}(\mathbf{z}) \mathbf{d} < 0$. By Taylor's theorem there exists $\alpha \in (0, 1)$ such that

$$\begin{aligned} f(\mathbf{y}) &= f(\mathbf{x}) + Df(\mathbf{x})(\mathbf{y} - \mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^\top \mathbf{F}(\mathbf{x} + \alpha(\mathbf{y} - \mathbf{x}))(\mathbf{y} - \mathbf{x}) \\ &= f(\mathbf{x}) + Df(\mathbf{x})(\mathbf{y} - \mathbf{x}) + \frac{1}{2}s^2 \mathbf{d}^\top \mathbf{F}(\mathbf{x} + \alpha s\mathbf{d}) \mathbf{d}. \end{aligned}$$

Because $\alpha \in (0, 1)$, the point $\mathbf{x} + \alpha s\mathbf{d}$ is on the line segment joining \mathbf{x} and \mathbf{y} , and therefore

$$\mathbf{d}^\top \mathbf{F}(\mathbf{x} + \alpha s\mathbf{d}) \mathbf{d} < 0.$$

Because $s \neq 0$, we have $s^2 > 0$, and hence

$$f(\mathbf{y}) < f(\mathbf{x}) + Df(\mathbf{x})(\mathbf{y} - \mathbf{x}).$$

Therefore, by Theorem 22.4, f is not a convex function. ■

Theorem 22.5 can be strengthened to include nonopen sets by modifying the condition to be $(\mathbf{y} - \mathbf{x})^\top \mathbf{F}(\mathbf{x})(\mathbf{y} - \mathbf{x}) \geq 0$ for all $\mathbf{x}, \mathbf{y} \in \Omega$ (and assuming that $f \in \mathcal{C}^2$ on some open set that contains Ω ; for example, $f \in \mathcal{C}^2$ on \mathbb{R}^n). A proof similar to that above can be used in this case.

Note that by definition of concavity, a function $f : \Omega \rightarrow \mathbb{R}$, $f \in \mathcal{C}^2$, is concave over the convex set $\Omega \subset \mathbb{R}^n$ if and only if for all $\mathbf{x} \in \Omega$, the Hessian $\mathbf{F}(\mathbf{x})$ of f is negative semidefinite.

Example 22.6 Determine whether the following functions are convex, concave, or neither:

1. $f : \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = -8x^2$.
2. $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, $f(\mathbf{x}) = 4x_1^2 + 3x_2^2 + 5x_3^2 + 6x_1x_2 + x_1x_3 - 3x_1 - 2x_2 + 15$.
3. $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, $f(\mathbf{x}) = 2x_1x_2 - x_1^2 - x_2^2$.

Solution:

1. We use Theorem 22.5. We first compute the Hessian, which in this case is just the second derivative: $(d^2f/dx^2)(x) = -16 < 0$ for all $x \in \mathbb{R}$. Hence, f is concave over \mathbb{R} .

2. The Hessian matrix of f is

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} 8 & 6 & 1 \\ 6 & 6 & 0 \\ 1 & 0 & 10 \end{bmatrix}.$$

The leading principal minors of $\mathbf{F}(\mathbf{x})$ are

$$\Delta_1 = 8 > 0,$$

$$\Delta_2 = \det \begin{bmatrix} 8 & 6 \\ 6 & 6 \end{bmatrix} = 12 > 0,$$

$$\Delta_3 = \det \mathbf{F}(\mathbf{x}) = 114 > 0.$$

Hence, $\mathbf{F}(\mathbf{x})$ is positive definite for all $\mathbf{x} \in \mathbb{R}^3$. Therefore, f is a convex function over \mathbb{R}^3 .

3. The Hessian of f is

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix},$$

which is negative semidefinite for all $\mathbf{x} \in \mathbb{R}^2$. Hence, f is concave on \mathbb{R}^2 . ■

22.3 Convex Optimization Problems

In this section we consider optimization problems where the objective function is a convex function and the constraint set is a convex set. We refer to such problems as *convex optimization problems* or *convex programming problems*. Optimization problems that can be classified as convex programming problems include linear programs and optimization problems with quadratic objective function and linear constraints. Convex programming problems are interesting for several reasons. Specifically, as we shall see, local minimizers are global for such problems. Furthermore, first-order necessary conditions become sufficient conditions for minimization.

Our first theorem below states that in convex programming problems, local minimizers are also global.

Theorem 22.6 *Let $f : \Omega \rightarrow \mathbb{R}$ be a convex function defined on a convex set $\Omega \subset \mathbb{R}^n$. Then, a point is a global minimizer of f over Ω if and only if it is a local minimizer of f .* □

Proof. \Rightarrow : This is obvious.

\Leftarrow : We prove this by contraposition. Suppose that \mathbf{x}^* is not a global minimizer of f over Ω . Then, for some $\mathbf{y} \in \Omega$, we have $f(\mathbf{y}) < f(\mathbf{x}^*)$. By assumption, the function f is convex, and hence for all $\alpha \in (0, 1)$,

$$f(\alpha\mathbf{y} + (1 - \alpha)\mathbf{x}^*) \leq \alpha f(\mathbf{y}) + (1 - \alpha)f(\mathbf{x}^*).$$

Because $f(\mathbf{y}) < f(\mathbf{x}^*)$, we have

$$\alpha f(\mathbf{y}) + (1 - \alpha)f(\mathbf{x}^*) = \alpha(f(\mathbf{y}) - f(\mathbf{x}^*)) + f(\mathbf{x}^*) < f(\mathbf{x}^*).$$

Thus, for all $\alpha \in (0, 1)$,

$$f(\alpha\mathbf{y} + (1 - \alpha)\mathbf{x}^*) < f(\mathbf{x}^*).$$

Hence, there exist points that are arbitrarily close to \mathbf{x}^* and have lower objective function value. For example, the sequence $\{\mathbf{y}_n\}$ of points given by

$$\mathbf{y}_n = \frac{1}{n}\mathbf{y} + \left(1 - \frac{1}{n}\right)\mathbf{x}^*$$

converges to \mathbf{x}^* , and $f(\mathbf{y}_n) < f(\mathbf{x}^*)$. Hence, \mathbf{x}^* is not a local minimizer, which completes the proof. ■

We now show that the set of global optimizers is convex. For this, we need the following lemma.

Lemma 22.1 *Let $g : \Omega \rightarrow \mathbb{R}$ be a convex function defined on a convex set $\Omega \subset \mathbb{R}^n$. Then, for each $c \in \mathbb{R}$, the set*

$$\Gamma_c = \{\mathbf{x} \in \Omega : g(\mathbf{x}) \leq c\}$$

is a convex set. □

Proof. Let $\mathbf{x}, \mathbf{y} \in \Gamma_c$. Then, $g(\mathbf{x}) \leq c$ and $g(\mathbf{y}) \leq c$. Because g is convex, for all $\alpha \in (0, 1)$,

$$g(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \alpha g(\mathbf{x}) + (1 - \alpha)g(\mathbf{y}) \leq c.$$

Hence, $\alpha\mathbf{x} + (1 - \alpha)\mathbf{y} \in \Gamma_c$, which implies that Γ_c is convex. ■

Corollary 22.1 *Let $f : \Omega \rightarrow \mathbb{R}$ be a convex function defined on a convex set $\Omega \subset \mathbb{R}^n$. Then, the set of all global minimizers of f over Ω is a convex set.*

□

Proof. The result follows immediately from Lemma 22.1 by setting

$$c = \min_{\mathbf{x} \in \Omega} f(\mathbf{x}).$$

We now show that if the objective function is continuously differentiable and convex, then the first-order necessary condition (see Theorem 6.1) for a point to be a minimizer is also sufficient. We use the following lemma.

Lemma 22.2 *Let $f : \Omega \rightarrow \mathbb{R}$ be a convex function defined on the convex set $\Omega \subset \mathbb{R}^n$ and $f \in \mathcal{C}^1$ on an open convex set containing Ω . Suppose that the point $\mathbf{x}^* \in \Omega$ is such that for all $\mathbf{x} \in \Omega$, $\mathbf{x} \neq \mathbf{x}^*$, we have*

$$Df(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \geq 0.$$

Then, \mathbf{x}^ is a global minimizer of f over Ω .* \square

Proof. Because the function f is convex, by Theorem 22.4, for all $\mathbf{x} \in \Omega$, we have

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) + Df(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*).$$

Hence, the condition $Df(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \geq 0$ implies that $f(\mathbf{x}) \geq f(\mathbf{x}^*)$. \blacksquare

Observe that for any $\mathbf{x} \in \Omega$, the vector $\mathbf{x} - \mathbf{x}^*$ can be interpreted as a feasible direction at \mathbf{x}^* (see Definition 6.2). Using Lemma 22.2, we have the following theorem (cf. Theorem 6.1).

Theorem 22.7 *Let $f : \Omega \rightarrow \mathbb{R}$ be a convex function defined on the convex set $\Omega \subset \mathbb{R}^n$, and $f \in \mathcal{C}^1$ on an open convex set containing Ω . Suppose that the point $\mathbf{x}^* \in \Omega$ is such that for any feasible direction \mathbf{d} at \mathbf{x}^* , we have*

$$\mathbf{d}^\top \nabla f(\mathbf{x}^*) \geq 0.$$

Then, \mathbf{x}^ is a global minimizer of f over Ω .* \square

Proof. Let $\mathbf{x} \in \Omega$, $\mathbf{x} \neq \mathbf{x}^*$. By convexity of Ω ,

$$\mathbf{x}^* + \alpha(\mathbf{x} - \mathbf{x}^*) = \alpha\mathbf{x} + (1 - \alpha)\mathbf{x}^* \in \Omega$$

for all $\alpha \in (0, 1)$. Hence, the vector $\mathbf{d} = \mathbf{x} - \mathbf{x}^*$ is a feasible direction at \mathbf{x}^* (see Definition 6.2). By assumption,

$$Df(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) = \mathbf{d}^\top \nabla f(\mathbf{x}^*) \geq 0.$$

Hence, by Lemma 22.2, \mathbf{x}^* is a global minimizer of f over Ω . \blacksquare

From Theorem 22.7, we easily deduce the following corollary (compare this with Corollary 6.1).

Corollary 22.2 *Let $f : \Omega \rightarrow \mathbb{R}$, $f \in \mathcal{C}^1$, be a convex function defined on the convex set $\Omega \subset \mathbb{R}^n$. Suppose that the point $\mathbf{x}^* \in \Omega$ is such that*

$$\nabla f(\mathbf{x}^*) = \mathbf{0}.$$

Then, \mathbf{x}^* is a global minimizer of f over Ω . □

We now consider the constrained optimization problem

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0}. \end{aligned}$$

We assume that the feasible set is convex. An example where this is the case is when

$$\mathbf{h}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}.$$

The following theorem states that provided the feasible set is convex, the Lagrange condition is sufficient for a point to be a minimizer.

Theorem 22.8 *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in \mathcal{C}^1$, be a convex function on the set of feasible points*

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}\},$$

where $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathbf{h} \in \mathcal{C}^1$, and Ω is convex. Suppose that there exist $\mathbf{x}^ \in \Omega$ and $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ such that*

$$Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} Dh(\mathbf{x}^*) = \mathbf{0}^\top.$$

Then, \mathbf{x}^ is a global minimizer of f over Ω .* □

Proof. By Theorem 22.4, for all $\mathbf{x} \in \Omega$, we have

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) + Df(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*).$$

Substituting $Df(\mathbf{x}^*) = -\boldsymbol{\lambda}^{*\top} Dh(\mathbf{x}^*)$ into the inequality above yields

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) - \boldsymbol{\lambda}^{*\top} Dh(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*).$$

Because Ω is convex, $(1 - \alpha)\mathbf{x}^* + \alpha\mathbf{x} \in \Omega$ for all $\alpha \in (0, 1)$. Thus,

$$\mathbf{h}(\mathbf{x}^* + \alpha(\mathbf{x} - \mathbf{x}^*)) = \mathbf{h}((1 - \alpha)\mathbf{x}^* + \alpha\mathbf{x}) = \mathbf{0}$$

for all $\alpha \in (0, 1)$. Premultiplying by $\boldsymbol{\lambda}^{*\top}$, subtracting $\boldsymbol{\lambda}^{*\top} \mathbf{h}(\mathbf{x}^*) = 0$, and dividing by α , we get

$$\frac{\boldsymbol{\lambda}^{*\top} \mathbf{h}(\mathbf{x}^* + \alpha(\mathbf{x} - \mathbf{x}^*)) - \boldsymbol{\lambda}^{*\top} \mathbf{h}(\mathbf{x}^*)}{\alpha} = 0$$

for all $\alpha \in (0, 1)$. If we now take the limit as $\alpha \rightarrow 0$ and apply the definition of the directional derivative of $\boldsymbol{\lambda}^{*\top} \mathbf{h}$ at \mathbf{x}^* in the direction $\mathbf{x} - \mathbf{x}^*$ (see Section 6.2), we get

$$\boldsymbol{\lambda}^{*\top} Dh(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) = 0.$$

Hence,

$$f(\mathbf{x}) \geq f(\mathbf{x}^*),$$

which implies that \mathbf{x}^* is a global minimizer of f over Ω . ■

Consider the general constrained optimization problem

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & && \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned}$$

As before, we assume that the feasible set is convex. This is the case if, for example, the two sets $\{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$ and $\{\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$ are convex, because the feasible set is the intersection of these two sets (see also Theorem 4.1). We have already seen an example where the set $\{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$ is convex. On the other hand, an example where the set $\{\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$ is convex is when the components of $\mathbf{g} = [g_1, \dots, g_p]^\top$ are all convex functions. Indeed, the set $\{\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$ is the intersection of the sets $\{\mathbf{x} : g_i(\mathbf{x}) \leq 0\}$, $i = 1, \dots, p$. Because each of these sets is convex (by Lemma 22.1), their intersection is also convex.

We now prove that the Karush-Kuhn-Tucker (KKT) condition is sufficient for a point to be a minimizer to the problem above.

Theorem 22.9 *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in \mathcal{C}^1$, be a convex function on the set of feasible points*

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\},$$

where $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $\mathbf{h}, \mathbf{g} \in \mathcal{C}^1$, and Ω is convex. Suppose that there exist $\mathbf{x}^* \in \Omega$, $\boldsymbol{\lambda}^* \in \mathbb{R}^m$, and $\boldsymbol{\mu}^* \in \mathbb{R}^p$, such that

1. $\boldsymbol{\mu}^* \geq \mathbf{0}$.
2. $Df(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} D\mathbf{h}(\mathbf{x}^*) + \boldsymbol{\mu}^{*\top} D\mathbf{g}(\mathbf{x}^*) = \mathbf{0}^\top$.
3. $\boldsymbol{\mu}^{*\top} \mathbf{g}(\mathbf{x}^*) = 0$.

Then, \mathbf{x}^* is a global minimizer of f over Ω . □

Proof. Suppose that $\mathbf{x} \in \Omega$. By convexity of f and Theorem 22.4,

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) + Df(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*).$$

Using condition 2, we get

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) - \boldsymbol{\lambda}^{*\top} D\mathbf{h}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) - \boldsymbol{\mu}^{*\top} D\mathbf{g}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*).$$

As in the proof of Theorem 22.8, we can show that $\boldsymbol{\lambda}^{*\top} D\mathbf{h}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) = 0$. We now claim that $\boldsymbol{\mu}^{*\top} D\mathbf{g}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \leq 0$. To see this, note that because Ω is convex, $(1 - \alpha)\mathbf{x}^* + \alpha\mathbf{x} \in \Omega$ for all $\alpha \in (0, 1)$. Thus,

$$\mathbf{g}(\mathbf{x}^* + \alpha(\mathbf{x} - \mathbf{x}^*)) = \mathbf{g}((1 - \alpha)\mathbf{x}^* + \alpha\mathbf{x}) \leq \mathbf{0}$$

for all $\alpha \in (0, 1)$. Premultiplying by $\mu^{*\top} \geq \mathbf{0}^\top$ (by condition 1), subtracting $\mu^{*\top} g(\mathbf{x}^*) = 0$ (by condition 3), and dividing by α , we get

$$\frac{\mu^{*\top} g(\mathbf{x}^* + \alpha(\mathbf{x} - \mathbf{x}^*)) - \mu^{*\top} g(\mathbf{x}^*)}{\alpha} \leq 0.$$

We now take the limit as $\alpha \rightarrow 0$ to obtain $\mu^{*\top} Dg(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \leq 0$.

From the above, we have

$$\begin{aligned} f(\mathbf{x}) &\geq f(\mathbf{x}^*) - \lambda^{*\top} Dh(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) - \mu^{*\top} Dg(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \\ &\geq f(\mathbf{x}^*) \end{aligned}$$

for all $\mathbf{x} \in \Omega$, which completes the proof. ■

Example 22.7 A bank account starts out with 0 dollars. At the beginning of each month, we deposit some money into the bank account. Denote by x_k the amount deposited in the k th month, $k = 1, 2, \dots$. Suppose that the monthly interest rate is $r > 0$ and the interest is paid into the account at the end of each month (and compounded). We wish to maximize the total amount of money accumulated at the end of n months, such that the total money deposited during the n months does not exceed D dollars (where $D > 0$).

To solve this problem we first show that the problem can be posed as a linear program, and is therefore a convex optimization problem. Let y_k be the total amount in the bank at the end of the k th month. Then, $y_k = (1+r)(y_{k-1} + x_k)$, $k \geq 1$, with $y_0 = 0$. Therefore, we want to maximize y_n subject to the constraint that $x_k \geq 0$, $k = 1, \dots, n$, and $x_1 + \dots + x_n \leq D$. It is easy to deduce that

$$y_n = (1+r)^n x_1 + (1+r)^{n-1} x_2 + \dots + (1+r) x_n.$$

Let $\mathbf{c}^\top = [(1+r)^n, (1+r)^{n-1}, \dots, (1+r)]$, $\mathbf{e}^\top = [1, \dots, 1]$, and $\mathbf{x} = [x_1, \dots, x_n]^\top$. Then, we can write the problem as

$$\begin{aligned} &\text{maximize } \mathbf{c}^\top \mathbf{x} \\ &\text{subject to } \mathbf{e}^\top \mathbf{x} \leq D \\ &\quad \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

which is a linear program.

It is intuitively clear that the optimal strategy is to deposit D dollars in the first month. To show that this strategy is indeed optimal, we use Theorem 22.9. Let $\mathbf{x}^* = [D, 0, \dots, 0]^\top \in \mathbb{R}^n$. Because the problem is a convex programming problem, it suffices to show that \mathbf{x}^* satisfies the KKT

condition (see Theorem 22.9). The KKT condition for this problem is

$$\begin{aligned} -\mathbf{c}^\top + \mu^{(1)} \mathbf{e}^\top - \boldsymbol{\mu}^{(2)\top} &= 0, \\ \mu^{(1)}(\mathbf{e}^\top \mathbf{x}^* - D) &= 0, \\ \boldsymbol{\mu}^{(2)\top} \mathbf{x}^* &= 0, \\ \mathbf{e}^\top \mathbf{x}^* - D &\leq 0, \\ -\mathbf{x}^* &\leq \mathbf{0}, \\ \mu^{(1)} &\geq 0, \\ \boldsymbol{\mu}^{(2)} &\geq \mathbf{0}, \\ \mathbf{e}^\top \mathbf{x} &\leq D, \\ \mathbf{x} &\geq \mathbf{0}, \end{aligned}$$

where $\mu^{(1)} \in \mathbb{R}$ and $\boldsymbol{\mu}^{(2)} \in \mathbb{R}^n$. Let $\mu^{(1)} = (1+r)^n$ and $\boldsymbol{\mu}^{(2)} = (1+r)^n \mathbf{e} - \mathbf{c}$. Then, it is clear that the KKT condition is satisfied. Therefore, \mathbf{x}^* is a global minimizer. ■

An entire book devoted to the vast topic of convexity and optimization is [7]. For extensions of the theory of convex optimization, we refer the reader to [136, Chapter 10]. The study of convex programming problems also serves as a prerequisite to *nondifferentiable optimization* (see, e.g., [38]).

22.4 Semidefinite Programming

Semidefinite programming is a subfield of convex optimization concerned with minimizing a linear objective function subject to a linear matrix inequality. The linear matrix inequality constraint defines a convex feasible set over which the linear objective function is to be minimized. Semidefinite programming can be viewed as an extension of linear programming, where the componentwise inequalities on vectors are replaced by matrix inequalities (see Exercise 22.20). For further reading on the subject of semidefinite programming, we recommend an excellent survey paper by Vandenberghe and Boyd [128].

Linear Matrix Inequalities and Their Properties

Consider $n+1$ real symmetric matrices

$$\mathbf{F}_i = \mathbf{F}_i^\top \in \mathbb{R}^{m \times m}, \quad i = 0, 1, \dots, n$$

and a vector

$$\mathbf{x} = [x_1, \dots, x_n]^\top \in \mathbb{R}^n.$$

Then,

$$\begin{aligned}\mathbf{F}(\mathbf{x}) &= \mathbf{F}_0 + x_1 \mathbf{F}_1 + \cdots + x_n \mathbf{F}_n \\ &= \mathbf{F}_0 + \sum_{i=1}^n x_i \mathbf{F}_i\end{aligned}$$

is an affine function of \mathbf{x} , because $\mathbf{F}(\mathbf{x})$ is composed of a linear term $\sum_{i=1}^n x_i \mathbf{F}_i$ and a constant term \mathbf{F}_0 .

Consider now an inequality constraint of the form

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + x_1 \mathbf{F}_1 + \cdots + x_n \mathbf{F}_n \geq 0.$$

The inequality constraint above is to be interpreted as the set of vectors \mathbf{x} such that

$$\mathbf{z}^\top \mathbf{F}(\mathbf{x}) \mathbf{z} \geq 0 \text{ for all } \mathbf{z} \in \mathbb{R}^m;$$

that is, $\mathbf{F}(\mathbf{x})$ is positive semidefinite [or, in the usual notation, $\mathbf{F}(\mathbf{x}) \geq 0$]. Recall that the terms \mathbf{F}_i represent constant matrices, \mathbf{x} is unknown, and $\mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x})^\top$ is an affine function \mathbf{x} . The expression $\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + x_1 \mathbf{F}_1 + \cdots + x_n \mathbf{F}_n \geq 0$ is referred to in the literature as a *linear matrix inequality* (LMI), although the term *affine matrix inequality* would seem to be more appropriate. It is easy to verify that the set $\{\mathbf{x} : \mathbf{F}(\mathbf{x}) \geq 0\}$ is convex (see Exercise 22.20).

We can speak similarly of LMIs of the form $\mathbf{F}(\mathbf{x}) > 0$, where the requirement is for $\mathbf{F}(\mathbf{x})$ to be positive definite (rather than just positive semidefinite). It is again easy to see that the set $\{\mathbf{x} : \mathbf{F}(\mathbf{x}) > 0\}$ is convex.

A system of LMIs

$$\mathbf{F}_1(\mathbf{x}) \geq 0, \mathbf{F}_2(\mathbf{x}) \geq 0, \dots, \mathbf{F}_k(\mathbf{x}) \geq 0$$

can be represented as one single LMI:

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \mathbf{F}_1(\mathbf{x}) & & & \\ & \mathbf{F}_2(\mathbf{x}) & & \\ & & \ddots & \\ & & & \mathbf{F}_k(\mathbf{x}) \end{bmatrix} \geq 0.$$

As an example, a linear inequality involving an $m \times n$ real constant matrix \mathbf{A} of the form

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

can be represented as m LMIs:

$$b_i - \mathbf{a}_i^\top \mathbf{x} \geq 0, \quad i = 1, 2, \dots, m,$$

where \mathbf{a}_i^\top is the i th row of the matrix \mathbf{A} . We can view each scalar inequality as an LMI. We then represent m LMIs as one LMI:

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} b_1 - \mathbf{a}_1^\top \mathbf{x} \\ & b_2 - \mathbf{a}_2^\top \mathbf{x} \\ & & \ddots \\ & & & b_m - \mathbf{a}_m^\top \mathbf{x} \end{bmatrix} \geq 0.$$

With the foregoing facts as background, we can now give an example of semidefinite programming:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{F}(\mathbf{x}) \geq 0. \end{aligned}$$

The matrix property that we discuss next is useful when converting certain LMIs or nonlinear matrix inequalities into equivalent LMIs. We start with a simple observation. Let \mathbf{P} be a nonsingular $n \times n$ matrix and let $\mathbf{x} = \mathbf{M}\mathbf{z}$, where $\mathbf{M} \in \mathbb{R}^{n \times n}$ such that $\det \mathbf{M} \neq 0$. Then, we have

$$\mathbf{x}^\top \mathbf{P} \mathbf{x} \geq 0 \text{ if and only if } \mathbf{z}^\top \mathbf{M}^\top \mathbf{P} \mathbf{M} \mathbf{z} \geq 0;$$

that is,

$$\mathbf{P} \geq 0 \text{ if and only if } \mathbf{M}^\top \mathbf{P} \mathbf{M} \geq 0.$$

Similarly,

$$\mathbf{P} > 0 \text{ if and only if } \mathbf{M}^\top \mathbf{P} \mathbf{M} > 0.$$

Suppose that we have a square matrix

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{D} \end{bmatrix}.$$

Then, by the observation above,

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{D} \end{bmatrix} \geq 0 \text{ if and only if } \begin{bmatrix} \mathbf{O} & \mathbf{I} \\ \mathbf{I} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{O} & \mathbf{I} \\ \mathbf{I} & \mathbf{O} \end{bmatrix} \geq 0,$$

where \mathbf{I} is an identity matrix of appropriate dimension. In other words,

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{D} \end{bmatrix} \geq 0 \text{ if and only if } \begin{bmatrix} \mathbf{D} & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{A} \end{bmatrix} \geq 0.$$

We now introduce the notion of the Schur complement, useful in studying LMIs. Consider a square matrix of the form

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix},$$

where \mathbf{A}_{11} and \mathbf{A}_{22} are square submatrices. Suppose that the matrix \mathbf{A}_{11} is invertible. Then, we have

$$\begin{bmatrix} \mathbf{I} & \mathbf{O} \\ -\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{O} \\ \mathbf{O} & \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \end{bmatrix}.$$

Let

$$\Delta_{11} = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12},$$

which is called the *Schur complement* of \mathbf{A}_{11} . For the case when $\mathbf{A}_{12} = \mathbf{A}_{21}^\top$, we have

$$\begin{bmatrix} \mathbf{I} & \mathbf{O} \\ -\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{21}^\top \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{21}^\top \\ \mathbf{O} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{O} \\ \mathbf{O} & \Delta_{11} \end{bmatrix},$$

where

$$\Delta_{11} = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{21}^\top.$$

Hence,

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{21}^\top \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} > 0 \text{ if and only if } \begin{bmatrix} \mathbf{A}_{11} & \mathbf{O} \\ \mathbf{O} & \Delta_{11} \end{bmatrix} > 0;$$

that is,

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{21}^\top \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} > 0 \text{ if and only if } \mathbf{A}_{11} > 0 \text{ and } \Delta_{11} > 0.$$

Given

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix},$$

we can similarly define the Schur complement of \mathbf{A}_{22} , assuming that \mathbf{A}_{22} is invertible. We have

$$\begin{bmatrix} \mathbf{I} & -\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \Delta_{22} & \mathbf{O} \\ \mathbf{O} & \mathbf{A}_{22} \end{bmatrix},$$

where $\Delta_{22} = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$ is the Schur complement of \mathbf{A}_{22} . So, for the case where $\mathbf{A}_{12} = \mathbf{A}_{21}^\top$,

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{21}^\top \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} > 0 \text{ if and only if } \mathbf{A}_{22} > 0 \text{ and } \Delta_{22} > 0.$$

Many problems of optimization, control design, and signal processing can be formulated in terms of LMIs. To determine whether or not there exists a point \mathbf{x} such that $\mathbf{F}(\mathbf{x}) > 0$ is called a *feasibility problem*. We say that the LMI is nonfeasible if no such solution exists.

Example 22.8 We now present a simple example illustrating the LMI feasibility problem. Let $\mathbf{A} \in \mathbb{R}^{m \times m}$ be a given real constant square matrix. Suppose that we wish to determine if \mathbf{A} has all its eigenvalues in the open left half-complex plane. It is well known that this condition is true if and only if there exists a real symmetric positive definite matrix \mathbf{P} such that

$$\mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} < 0,$$

or, equivalently,

$$-\mathbf{A}^\top \mathbf{P} - \mathbf{P} \mathbf{A} > 0$$

(also called the *Lyapunov inequality*; see [16]). Thus, the location of all eigenvalues of \mathbf{A} being in the open left half-complex plane is equivalent to feasibility of the following matrix inequality:

$$\begin{bmatrix} \mathbf{P} & \mathbf{O} \\ \mathbf{O} & -\mathbf{A}^\top \mathbf{P} - \mathbf{P} \mathbf{A} \end{bmatrix} > 0;$$

that is, the existence of $\mathbf{P} = \mathbf{P}^\top > 0$ such that $\mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} < 0$.

We now show that finding $\mathbf{P} = \mathbf{P}^\top > 0$ such that $\mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} < 0$ is indeed an LMI. For this, let

$$\mathbf{P} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \\ x_2 & x_{m+1} & \cdots & x_{2m-1} \\ \vdots & & & \vdots \\ x_m & x_{2m-1} & \cdots & x_n \end{bmatrix},$$

where

$$n = \frac{m(m+1)}{2}.$$

We next define the following matrices:

$$\begin{aligned}\mathbf{P}_1 &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \\ \mathbf{P}_2 &= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \\ &\vdots \\ \mathbf{P}_n &= \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.\end{aligned}$$

Note that \mathbf{P}_i has only nonzero elements corresponding to \mathbf{x}_i in \mathbf{P} . Let

$$\mathbf{F}_i = -\mathbf{A}^\top \mathbf{P}_i - \mathbf{P}_i \mathbf{A}, \quad i = 1, 2, \dots, n.$$

We can then write

$$\begin{aligned}\mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} &= x_1 (\mathbf{A}^\top \mathbf{P}_1 + \mathbf{P}_1 \mathbf{A}) + x_2 (\mathbf{A}^\top \mathbf{P}_2 + \mathbf{P}_2 \mathbf{A}) + \cdots \\ &\quad + x_n (\mathbf{A}^\top \mathbf{P}_n + \mathbf{P}_n \mathbf{A}) \\ &= -x_1 \mathbf{F}_1 - x_2 \mathbf{F}_2 - \cdots - x_n \mathbf{F}_n \\ &< 0.\end{aligned}$$

Let

$$\mathbf{F}(\mathbf{x}) = x_1 \mathbf{F}_1 + x_2 \mathbf{F}_2 + \cdots + x_n \mathbf{F}_n.$$

Then,

$$\mathbf{P} = \mathbf{P}^\top > 0 \quad \text{and} \quad \mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} < 0$$

if and only if

$$\mathbf{F}(\mathbf{x}) > 0.$$

Note that this LMI involves a strict inequality. Most numerical solvers do not handle strict inequalities. Such solvers simply treat a strict inequality ($>$) as a non-strict inequality (\geq). ■

LMI Solvers

The inequality $\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + x_1\mathbf{F}_1 + \cdots + x_n\mathbf{F}_n \geq 0$ is called the *canonical representation* of an LMI. Numerical LMI solvers do not deal directly with LMIs in canonical form because of various inefficiencies. Instead, LMI solvers use a structured representation of LMIs.

We can use MATLAB's LMI toolbox to solve LMIs efficiently. This toolbox has three types of LMI solvers, which we discuss next.

Finding a Feasible Solution Under LMI Constraints

First, we discuss MATLAB's LMI solver for solving the feasibility problem defined by a given system of LMI constraints. Using this solver, we can solve any system of LMIs of the form

$$\mathbf{N}^\top \mathcal{L}(\mathbf{X}_1, \dots, \mathbf{X}_k) \mathbf{N} \leq \mathbf{M}^\top \mathcal{R}(\mathbf{X}_1, \dots, \mathbf{X}_k) \mathbf{M},$$

where $\mathbf{X}_1, \dots, \mathbf{X}_k$ are matrix variables, \mathbf{N} is the left outer factor, \mathbf{M} is the right outer factor, $\mathcal{L}(\mathbf{X}_1, \dots, \mathbf{X}_k)$ is the left inner factor, and $\mathcal{R}(\mathbf{X}_1, \dots, \mathbf{X}_k)$ is the right inner factor. The matrices $\mathcal{L}(\cdot)$ and $\mathcal{R}(\cdot)$ are, in general, symmetric block matrices. We note that the term left-hand side refers to what is on the “smaller” side of the inequality $0 \leq \mathbf{X}$. Thus in $\mathbf{X} \geq 0$, the matrix \mathbf{X} is still on the right-hand side because it is on the “larger” side of the inequality.

We now provide a description of an approach that can be used to solve the given LMI system feasibility problem. To initialize the LMI system description, we type `setlmis([])`. Then we declare matrix variables using the command `lmivar`. The command `lmitemr` allows us to specify LMIs that constitute the LMI system under consideration. Next, we need to obtain an internal representation using the command `getlmis`. We next compute a feasible solution to the LMI system using the command `feasp`. After that, we extract matrix variable values with the command `dec2mat`. In summary, a general structure of a MATLAB program for finding a feasible solution to the set of LMIs could have the form

```
setlmis([])
lmivar
lmitemr
.
.
.
lmitemr
getlmis
feasp
dec2mat
```

We now analyze these commands in some detail so that the reader can write simple MATLAB programs for solving LMIs after completing this section.

First, to create a new matrix-valued variable, say, \mathbf{X} , in the given LMI system, we use the command

```
X = lmivar(type,structure)
```

The input **type** specifies the structure of the variable \mathbf{X} . There may be three structures of matrix variables. When **type**=1, we have a symmetric block diagonal matrix variable. The input **type**=2 refers to a full rectangular matrix variable. Finally, **type**=3 refers to other cases. The second input **structure** gives additional information on the structure of the matrix variable \mathbf{X} . For example, the matrix variable \mathbf{X} could have the form

$$\mathbf{X} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{D}_2 & \cdots & \mathbf{O} \\ \vdots & & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{D}_r \end{bmatrix},$$

where each \mathbf{D}_i is a square symmetric matrix. For the example above we would use **type**=1. The matrix variable above has r blocks. The input **structure** is then an $r \times 2$ matrix whose i th row describes the i th block, where the first component of each row gives the corresponding block size, while the second element of each row specifies the block type. For example,

```
X = lmivar(1,[3 1])
```

specifies a full symmetric 3×3 matrix variable. On the other hand,

```
X = lmivar(2,[2 3])
```

specifies a rectangular 2×3 matrix variable. Finally, a matrix variable \mathbf{S} of the form

$$\mathbf{S} = \begin{bmatrix} s_1 & 0 & | & 0 & 0 \\ 0 & s_1 & | & 0 & 0 \\ --- & --- & | & --- & --- \\ 0 & 0 & | & s_2 & s_3 \\ 0 & 0 & | & s_3 & s_4 \end{bmatrix}$$

can be declared as follows:

```
S = lmivar(1,[2 0;2 1])
```

Note above that the second component of the first row of the second input has the value of zero; that is, **structure**(1,2)=0. This describes a scalar block matrix of the form

$$\mathbf{D}_1 = s_1 \mathbf{I}_2.$$

Note that the second block is a 2×2 symmetric full block.

We next take a closer look at a command whose purpose is to specify the terms of the LMI system of interest. This command has the form

```
lmitem(termid,A,B,flag)
```

We briefly describe each of the four inputs of this command. The first input, `termid`, is a row with four elements that specify the terms of each LMI of the LMI system. We have `termid(1)=n` to specify the left-hand side of the n th LMI. We use `termid(1)=-n` to specify the right-hand side of the n th LMI. The middle two elements of the input `termid` specify the block location. Thus `termid(2,3)=[i j]` refers to the term that belongs to the (i, j) block of the LMI specified by the first component. Finally, `termid(4)=0` for the constant term, `termid(4)=X` for the variable term in the form $\mathbf{A} \mathbf{X} \mathbf{B}$, while `termid(4)=-X` for the variable term in the form $\mathbf{A} \mathbf{X}^\top \mathbf{B}$. The second and third inputs of the command `lmitem` give the values of the left and right outer factors; that is, `A` and `B` give the values of the constant outer factors in the variable terms $\mathbf{A} \mathbf{X} \mathbf{B}$ and $\mathbf{A} \mathbf{X}^\top \mathbf{B}$. Finally, the fourth input to `lmitem` serves as a compact way to specify the expression

$$\mathbf{A} \mathbf{X} \mathbf{B} + (\mathbf{A} \mathbf{X} \mathbf{B})^\top.$$

Thus, `flag='s'` can be used to denote a symmetrized expression. We now illustrate the command above on the following LMI:

$$\mathbf{P} \mathbf{A} + (\mathbf{P} \mathbf{A})^\top \leq 0.$$

We have one LMI with two terms. We could use the following description of this single LMI:

```
lmitem([1 1 1 P],1,A)
lmitem([1 1 1 -P],A',1)
```

On the other hand, we can describe this LMI compactly using the `flag` as follows:

```
lmitem([1 1 1 P],1,A,'s')
```

Now, to solve the feasibility problem we could have typed

```
[tmin,xfeas] = feas(lmis)
```

In general, for a given LMI feasibility problem of the form

$$\begin{aligned} &\text{find } \mathbf{x} \\ &\text{such that } \mathbf{L}(\mathbf{x}) \leq \mathbf{R}(\mathbf{x}), \end{aligned}$$

the command `feasp` solves the auxiliary convex problem

$$\begin{aligned} &\text{minimize } t \\ &\text{subject to } \mathbf{L}(\mathbf{x}) \leq \mathbf{R}(\mathbf{x}) + t\mathbf{I}. \end{aligned}$$

The system of LMIs is feasible if the minimal t is negative. We add that the current value of t is displayed by **feasp** at each iteration.

Finally, we convert the output of the LMI solver into matrix variables using the command

```
P = dec2mat(lmis,xfeas,P).
```

Example 22.9 Let

$$\mathbf{A}_1 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{A}_2 = \begin{bmatrix} -2 & 0 \\ 1 & -1 \end{bmatrix}.$$

We use the commands of the LMI Control Toolbox discussed above to write a program that finds \mathbf{P} such that $\mathbf{P} \geq 0.5\mathbf{I}_2$ and

$$\begin{aligned}\mathbf{A}_1^\top \mathbf{P} + \mathbf{P} \mathbf{A}_1 &\leq 0, \\ \mathbf{A}_2^\top \mathbf{P} + \mathbf{P} \mathbf{A}_2 &\leq 0.\end{aligned}$$

The program is as follows:

```
A_1 = [-1 0;0 -1];
A_2 = [-2 0;1 -1];
setlmis([])
P = lmivar(1,[2,1])
lmiterm([1 1 1 P],A_1',1,'s')
lmiterm([2 1 1 P],A_2',1,'s')
lmiterm([3 1 1 0],.5)
lmiterm([-3 1 1 P],1,1)
lmis=getlmis;
[tmin,xfeas] = feasp(lmis);
P = dec2mat(lmis,xfeas,P)
```

■

Minimizing a Linear Objective Under LMI Constraints

The next solver we discuss solves the convex optimization problem

$$\begin{aligned}&\text{minimize} \quad \mathbf{c}^\top \mathbf{x} \\ &\text{subject to} \quad \mathbf{A}(\mathbf{x}) \leq \mathbf{B}(\mathbf{x}).\end{aligned}$$

The notation $\mathbf{A}(\mathbf{x}) \leq \mathbf{B}(\mathbf{x})$ is shorthand notation for a general structured LMI system.

This solver is invoked using the function **mincx**. Thus, to solve a **mincx** problem, in addition to specifying the LMI constraints as in the **feasp** problem, we also declare the linear objective function. Then we invoke the function

`mincx`. We illustrate and contrast the `feasp` and `mincx` solvers in the following example.

Example 22.10 Consider the optimization problem

$$\begin{aligned} & \text{minimize } \mathbf{c}^\top \mathbf{x} \\ & \text{subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b}, \end{aligned}$$

where

$$\begin{aligned} \mathbf{c}^\top &= \begin{bmatrix} 4 & 5 \end{bmatrix}, \\ \mathbf{A} &= \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 2 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 8 \\ 18 \\ 14 \end{bmatrix}. \end{aligned}$$

We first solve the feasibility problem; that is, we find an \mathbf{x} such that $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, using the `feasp` solver. After that, we solve the minimization problem above using the `mincx` solver. A simple MATLAB code accomplishing these tasks is shown below.

```
% Enter problem data
A = [1 1; 1 3; 2 1];
b = [8 18 14]';
c = [-4 -5]';
setlmis([]);
X = lmivar(2,[2 1]);
lmiterm([1 1 1 X],A(1,:),1);
lmiterm([1 1 1 0],-b(1));
lmiterm([1 2 2 X],A(2,:),1);
lmiterm([1 2 2 0],-b(2));
lmiterm([1 3 3 X],A(3,:),1);
lmiterm([1 3 3 0],-b(3));
lmis = getlmis;
%-----
disp('-----feasp result-----')
[tmin,xfeas] = feasp(lmis);
x_feasp = dec2mat(lmis,xfeas,X)
disp('-----mincx result-----')
[objective,x_mincx] = mincx(lmis,c,[0.0001 1000 0 0 1])
```

The `feasp` function produces

$$\mathbf{x}_{\text{feasp}} = \begin{bmatrix} -64.3996 \\ -25.1712 \end{bmatrix}.$$

The `mincx` function produces

$$\mathbf{x}_{\text{mincx}} = \begin{bmatrix} 3.0000 \\ 5.0000 \end{bmatrix}.$$

■

In the next example, we discuss the function `defcx`, which we can use to construct the vector \mathbf{c} used by the LMI solver `mincx`.

Example 22.11 Suppose that we wish to solve the optimization problem

$$\begin{aligned} & \text{minimize} && \text{trace}(\mathbf{P}) \\ & \text{subject to} && \mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} \leq 0 \\ & && \mathbf{P} \geq 0 \end{aligned}$$

where $\text{trace}(\mathbf{P})$ is the sum of the diagonal elements of \mathbf{P} . We can use the function `mincx` to solve this problem. However, to use `mincx`, we need a vector \mathbf{c} such that

$$\mathbf{c}^\top \mathbf{x} = \text{trace}(\mathbf{P}).$$

After specifying the LMIs and obtaining their internal representation using, for example, the command `lmisys=getlmis`, we can obtain the desired \mathbf{c} with the following MATLAB code,

```
q = decnbr(lmisys);
c = zeros(q,1);
for j = 1:q
    Pj = defcx(lmisys,j,P);
    c(j) = trace(Pj);
end
```

Having obtained the vector \mathbf{c} , we can use the function `mincx` to solve the optimization problem. ■

Minimizing a Generalized Eigenvalue Under LMI Constraints

This problem can be stated as

$$\begin{aligned} & \text{minimize} && \lambda \\ & \text{subject to} && \mathbf{C}(\mathbf{x}) \leq \mathbf{D}(\mathbf{x}) \\ & && 0 \leq \mathbf{B}(\mathbf{x}) \\ & && \mathbf{A}(\mathbf{x}) \leq \lambda \mathbf{B}(\mathbf{x}). \end{aligned}$$

Here, we need to distinguish between standard LMI constraints of the form $\mathbf{C}(\mathbf{x}) \leq \mathbf{D}(\mathbf{x})$ and *linear-fractional LMIs* of the form $\mathbf{A}(\mathbf{x}) \leq \lambda \mathbf{B}(\mathbf{x})$, which

are concerned with the generalized eigenvalue λ . The generalized eigenvalue minimization problem under LMI constraints can be solved using the solver `gevp`. The basic structure of the `gevp` solver has the form

```
[lopt,xopt] = gevp{lmisys,nflc}
```

which returns `lopt`, the global minimum of the generalized eigenvalue, and `xopt`, the optimal decision vector variable. The argument `lmisys` is the system of LMIs, $C(\mathbf{x}) \leq D(\mathbf{x})$, $C(\mathbf{x}) \leq D(\mathbf{x})$, and $A(\mathbf{x}) \leq \lambda B(\mathbf{x})$ for $\lambda = 1$. As in the previous solvers, the corresponding optimal values of the matrix variables are obtained using `dec2mat`. The number of linear-fractional constraints is specified with `nflc`. There are other inputs to `gevp` but they are optional. For more information on this type of the LMI solver, we refer the reader to the LMI Lab in MATLAB's Robust Control Toolbox user's guide.

Example 22.12 Consider the problem of finding the smallest α such that

$$\begin{aligned} P &> 0 \\ A^T P + PA &\leq -\alpha P, \end{aligned}$$

where

$$A = \begin{bmatrix} -1.1853 & 0.9134 & 0.2785 \\ 0.9058 & -1.3676 & 0.5469 \\ 0.1270 & 0.0975 & -3.0000 \end{bmatrix}.$$

This problem is related to finding the decay rate of the stable linear differential equation $\dot{\mathbf{x}} = A\mathbf{x}$. Finding α that solves the optimization problem above can be accomplished using the following LMIs:

```
A = [-1.1853 0.9134 0.2785
      0.9058 -1.3676 0.5469
      0.1270 0.0975 -3.0000];
setlmis([]);
P = lmivar(1,[3 1])
lmiterm([-1 1 1 P],1,1)    % P
lmiterm([1 1 1 0],.01)      % P >= 0.01*I
lmiterm([2 1 1 P],1,A,'s') % linear fractional constraint---LHS
lmiterm([-2 1 1 P],1,1)    % linear fractional constraint---RHS
lmis = getlmis;
[gamma,P_opt] = gevp(lmis,1);
P = dec2mat(lmis,P_opt,P)
alpha = -gamma
```

The result is

$$\alpha = 0.6561 \quad \text{and} \quad P = \begin{bmatrix} 0.6996 & -0.7466 & -0.0296 \\ -0.7466 & 0.8537 & -0.2488 \\ -0.0296 & -0.2488 & 3.2307 \end{bmatrix}.$$

Notice that we used $\mathbf{P} \geq 0.01\mathbf{I}$ in place of $\mathbf{P} > 0$. ■

More examples of linear matrix inequalities in system and control theory can be found in the book by Boyd et al. [16].

A quick introduction to MATLAB's LMI toolbox is the tutorial that can be accessed with the command `lmdem` within MATLAB. In addition to the MATLAB's LMI toolbox, there is another toolbox for solving LMIs called LMITOOL, a built-in software package in Scilab toolbox, developed at INRIA in France. Scilab offers free software for numerical optimization. There is a version of LMITOOL for MATLAB that can be obtained from the website of the Scilab Consortium.

Yet Another LMI Package, YALMIP, for solving LMIs was developed in Switzerland in the Automatic Control Laboratory at ETH. YALMIP is an “intuitive and flexible modelling language for solving optimization problems in MATLAB.”

LMIs are tools of modern optimization. The following quote on numerical linear algebra from Gill, Murray, and Wright [52, p. 2] applies as well to the contents of this chapter: “At the heart of modern optimization methods are techniques associated with *linear algebra*. Numerical linear algebra applies not simply in optimization, but in all fields of scientific computation, including approximation, ordinary differential equations, and partial differential equations. *The importance of numerical linear algebra to modern scientific computing cannot be overstated*. Without fast and reliable linear algebraic building blocks, it is impossible to develop effective optimization methods; without some knowledge of the fundamental issues in linear algebra, it is impossible to understand what happens during the transition from equations in a textbook to actual computation.”

EXERCISES

22.1 Find the range of values of the parameter α for which the function

$$f(x_1, x_2, x_3) = 2x_1x_3 - x_1^2 - x_2^2 - 5x_3^2 - 2\alpha x_1x_2 - 4x_2x_3$$

is concave.

22.2 Consider the function

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} - \mathbf{x}^\top \mathbf{b},$$

where $\mathbf{Q} = \mathbf{Q}^\top > 0$ and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$. Define the function $\phi : \mathbb{R} \rightarrow \mathbb{R}$ by $\phi(\alpha) = f(\mathbf{x} + \alpha\mathbf{d})$, where $\mathbf{x}, \mathbf{d} \in \mathbb{R}^n$ are fixed vectors and $\mathbf{d} \neq \mathbf{0}$. Show that $\phi(\alpha)$ is a strictly convex quadratic function of α .

22.3 Show that $f(\mathbf{x}) = x_1 x_2$ is a convex function on $\Omega = \{[a, ma]^\top : a \in \mathbb{R}\}$, where m is any given nonnegative constant.

22.4 Suppose that the set $\Omega = \{\mathbf{x} : h(\mathbf{x}) = c\}$ is convex, where $h : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c \in \mathbb{R}$. Show that h is convex and concave over Ω .

22.5 Find all subgradients of

$$f(x) = |x|, \quad x \in \mathbb{R},$$

at $x = 0$ and at $x = 1$.

22.6 Let $\Omega \subset \mathbb{R}^n$ be a convex set, and $f_i : \Omega \rightarrow \mathbb{R}$, $i = 1, \dots, \ell$ be convex functions. Show that $\max\{f_1, \dots, f_\ell\}$ is a convex function.

Note: The notation $\max\{f_1, \dots, f_\ell\}$ denotes a function from Ω to \mathbb{R} such that for each $\mathbf{x} \in \Omega$, its value is the largest among the numbers $f_i(\mathbf{x})$, $i = 1, \dots, \ell$.

22.7 Let $\Omega \subset \mathbb{R}^n$ be an open convex set. Show that a symmetric matrix $\mathbf{Q} \in \mathbb{R}^n$ is positive semidefinite if and only if for each $\mathbf{x}, \mathbf{y} \in \Omega$, $(\mathbf{x} - \mathbf{y})^\top \mathbf{Q} (\mathbf{x} - \mathbf{y}) \geq 0$. Show that a similar result for positive definiteness holds if we replace the “ \geq ” by “ $>$ ” in the inequality above.

22.8 Consider the problem

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|^2 \\ & \text{subject to} && x_1 + \cdots + x_n = 1 \\ & && x_1, \dots, x_n \geq 0 \end{aligned}$$

(see also Exercise 21.9). Is the problem a convex optimization problem? If yes, give a complete proof. If no, explain why not, giving complete explanations.

22.9 Consider the optimization problem

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \Omega, \end{aligned}$$

where $f(\mathbf{x}) = x_1 x_2^2$, where $\mathbf{x} = [x_1, x_2]^\top$, and $\Omega = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = x_2, x_1 \geq 0\}$. (See also Exercise 21.8.) Show that the problem is a convex optimization problem.

22.10 Consider the convex optimization problem

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \Omega. \end{aligned}$$

Suppose that the points $\mathbf{y} \in \Omega$ and $\mathbf{z} \in \Omega$ are local minimizers. Determine the largest set of points $G \subset \Omega$ for which you can be sure that every point in G is a global minimizer.

22.11 Suppose that we have a convex optimization problem on \mathbb{R}^3 .

- Consider the following three feasible points: $[1, 0, 0]^\top, [0, 1, 0]^\top, [0, 0, 1]^\top$. Suppose that all three have objective function value 1. What can you say about the objective function value of the point $(1/3)[1, 1, 1]^\top$? Explain fully.
- Suppose we know that the three points in part a are global minimizers. What can you say about the point $(1/3)[1, 1, 1]^\top$? Explain fully.

22.12 Consider the optimization problem

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} \\ & \text{subject to} && \mathbf{A} \mathbf{x} = \mathbf{b}, \end{aligned}$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n}$, $\mathbf{Q} = \mathbf{Q}^\top > 0$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, and $\text{rank } \mathbf{A} = m$.

- Find all points satisfying the Lagrange condition for the problem (in terms of \mathbf{Q} , \mathbf{A} , and \mathbf{b}).
- Are the points (or point) global minimizers for this problem?

22.13 Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in \mathcal{C}^1$, be a convex function on the set of feasible points

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}_i^\top \mathbf{x} + b_i \geq 0, \quad i = 1, \dots, p\},$$

where $\mathbf{a}_1, \dots, \mathbf{a}_p \in \mathbb{R}^n$, and $b_1, \dots, b_p \in \mathbb{R}$. Suppose that there exist $\mathbf{x}^* \in S$, and $\boldsymbol{\mu}^* \in \mathbb{R}^p$, $\boldsymbol{\mu}^* \leq \mathbf{0}$, such that

$$Df(\mathbf{x}^*) + \sum_{j \in J(\mathbf{x}^*)} \mu_j^* \mathbf{a}_j^\top = \mathbf{0}^\top,$$

where $J(\mathbf{x}^*) = \{i : \mathbf{a}_i^\top \mathbf{x}^* + b_i = 0\}$. Show that \mathbf{x}^* is a global minimizer of f over Ω .

22.14 Consider the problem: minimize $\|\mathbf{x}\|^2$ ($\mathbf{x} \in \mathbb{R}^n$) subject to $\mathbf{a}^\top \mathbf{x} \geq b$, where $\mathbf{a} \in \mathbb{R}^n$ is a nonzero vector and $b \in \mathbb{R}$, $b > 0$. Suppose that \mathbf{x}^* is a solution to the problem.

- Show that the constraint set is convex.
- Use the KKT theorem to show that $\mathbf{a}^\top \mathbf{x}^* = b$.

- c. Show that \mathbf{x}^* is unique, and find an expression for \mathbf{x}^* in terms of \mathbf{a} and b .

22.15 Consider the problem

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n \\ & \text{subject to} && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

For this problem we have the following theorem (see also Exercise 17.16).

Theorem: *A solution to this problem exists if and only if $\mathbf{c} \geq \mathbf{0}$. Moreover, if a solution exists, $\mathbf{0}$ is a solution.*

- a. Show that the problem is a convex programming problem.
- b. Use the first-order necessary condition (for set constraints) to prove the theorem.
- c. Use the KKT condition to prove the above theorem.

22.16 Consider a linear programming problem in standard form.

- a. Derive the KKT condition for the problem.
- b. Explain precisely why the KKT condition is sufficient for optimality in this case.
- c. Write down the dual to the standard form primal problem (see Chapter 17).
- d. Suppose that \mathbf{x}^* and $\boldsymbol{\lambda}^*$ are feasible solutions to the primal and dual, respectively. Use the KKT condition to prove that if the complementary slackness condition $(\mathbf{c}^\top - \boldsymbol{\lambda}^{*\top} \mathbf{A})\mathbf{x}^* = 0$ holds, then \mathbf{x}^* is an optimal solution to the primal problem. Compare the above with Exercise 21.15.

22.17 Consider two real-valued discrete-time signals, $\mathbf{s}^{(1)}$ and $\mathbf{s}^{(2)}$, defined over the time interval $[1, n]$. Let $s_i^{(1)}$ and $s_i^{(2)}$ be the values at time i of the signals $\mathbf{s}^{(1)}$ and $\mathbf{s}^{(2)}$, respectively. Assume that the energies of the two signals are 1 [i.e., $(s_1^{(1)})^2 + \dots + (s_n^{(1)})^2 = 1$ and $(s_1^{(2)})^2 + \dots + (s_n^{(2)})^2 = 1$].

Let S_a be the set of all signals that are linear combinations of $\mathbf{s}^{(1)}$ and $\mathbf{s}^{(2)}$ with the property that for each signal in S_a , the value of the signal over all time is no smaller than $a \in \mathbb{R}$. For each $\mathbf{s} \in S_a$, if $\mathbf{s} = x_1 \mathbf{s}^{(1)} + x_2 \mathbf{s}^{(2)}$, we call x_1 and x_2 the *coefficients* of \mathbf{s} .

We wish to find a signal in S_a such that the sum of the squares of its coefficients is minimized.

- a. Formulate the problem as an optimization problem.

- b. Derive the Karush-Kuhn-Tucker conditions for the problem.
- c. Suppose that you have found a point satisfying the Karush-Kuhn-Tucker conditions. Does this point satisfy the second-order sufficient condition?
- d. Is this problem a convex optimization problem?

22.18 Let a *probability vector* be any vector $\mathbf{p} \in \mathbb{R}^n$ satisfying $p_i > 0$, $i = 1, \dots, n$, and $p_1 + \dots + p_n = 1$.

Let $\mathbf{p} \in \mathbb{R}^n$ and $\mathbf{q} \in \mathbb{R}^n$ be two probability vectors. Define

$$D(\mathbf{p}, \mathbf{q}) = p_1 \log\left(\frac{p_1}{q_1}\right) + \dots + p_n \log\left(\frac{p_n}{q_n}\right),$$

where “log” is the natural logarithm function.

- a. Let Ω be the set of all probability vectors (with fixed n). Show that Ω is convex.
- b. Show that for each fixed \mathbf{p} , the function f defined by $f(\mathbf{q}) = D(\mathbf{p}, \mathbf{q})$ is convex over Ω .
- c. Show the following: $D(\mathbf{p}, \mathbf{q}) \geq 0$ for any probability vectors \mathbf{p} and \mathbf{q} . Moreover, $D(\mathbf{p}, \mathbf{q}) = 0$ if and only if $\mathbf{p} = \mathbf{q}$.
- d. Describe an application of the result of part c.

22.19 Let $\Omega \subset \mathbb{R}^n$ be a nonempty closed convex set and $\mathbf{z} \in \mathbb{R}^n$ be a given point such that $\mathbf{z} \notin \Omega$. Consider the optimization problem

$$\begin{aligned} &\text{minimize} && \|\mathbf{x} - \mathbf{z}\| \\ &\text{subject to} && \mathbf{x} \in \Omega. \end{aligned}$$

Does this problem have an optimal solution? If so, is it unique? Whatever your assertion, prove it.

Hint: (i) If \mathbf{x}_1 and \mathbf{x}_2 are optimal solutions, what can you say about $\mathbf{x}_3 = (\mathbf{x}_1 + \mathbf{x}_2)/2$? (ii) The triangle inequality states that $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$, with equality holding if and only if $\mathbf{x} = \alpha\mathbf{y}$ for some $\alpha \geq 0$ (or $\mathbf{x} = 0$ or $\mathbf{y} = 0$).

22.20 This exercise is about *semidefinite programming*.

- a. Show that if $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times n}$ are symmetric and $\mathbf{A} \geq 0$, $\mathbf{B} \geq 0$, then for any $\alpha \in (0, 1)$, we have $\alpha\mathbf{A} + (1 - \alpha)\mathbf{B} \geq 0$. As usual, the notation “ ≥ 0 ” denotes positive semidefiniteness.

- b.** Consider the following semidefinite programming problem, that is, an optimization problem with linear objective function and linear matrix inequality constraints:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{F}_0 + \sum_{j=1}^n x_j \mathbf{F}_j \geq 0, \end{aligned}$$

where $\mathbf{x} = [x_1, \dots, x_n]^\top \in \mathbb{R}^n$ is the decision variable, $\mathbf{c} \in \mathbb{R}^n$, and $\mathbf{F}_0, \mathbf{F}_1, \dots, \mathbf{F}_n \in \mathbb{R}^{m \times m}$ are symmetric.

Show that this problem is a convex optimization problem.

- c.** Consider the linear programming problem

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \geq \mathbf{b}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, and the inequality $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ has the usual elementwise interpretation. Show that this linear programming problem can be converted to the problem in part b.

Hint: First consider diagonal \mathbf{F}_j .

- 22.21** Suppose that you have a cake and you need to divide it among n different children. Suppose that the i th child receives a fraction x_i of the cake. We will call the vector $\mathbf{x} = [x_1, \dots, x_n]^\top$ an *allocation*. We require that every child receives at least some share of the cake, and that the entire cake is completely used up in the allocation. We also impose the additional condition that the first child ($i = 1$) is allocated a share that is at least twice that of any other child. We say that the allocation is feasible if it meets all these requirements.

A feasible allocation \mathbf{x} is said to be *proportionally fair* if for any other allocation \mathbf{y} ,

$$\sum_{i=1}^n \frac{y_i - x_i}{x_i} \leq 0.$$

- a.** Let Ω be the set of all feasible allocations. Show that Ω is convex.
- b.** Show that a feasible allocation is proportionally fair if and only if it solves the following optimization problem:

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n \log(x_i) \\ & \text{subject to} && \mathbf{x} \in \Omega. \end{aligned}$$

22.22 Let $U_i : \mathbb{R} \rightarrow \mathbb{R}$, $U_i \in \mathcal{C}^1$, $i = 1, \dots, n$, be a set of concave increasing functions. Consider the optimization problem

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n U_i(x_i) \\ & \text{subject to} && \sum_{i=1}^n x_i \leq C, \end{aligned}$$

where $C > 0$ is a given constant.

- a. Show that the optimization problem above is a convex optimization problem.
- b. Show that $\mathbf{x}^* = [x_1^*, \dots, x_n^*]^\top$ is an optimal solution to the optimization problem if and only if there exists a scalar $\mu^* \geq 0$ such that $x_i^* = \arg \max_x (U_i(x) - \mu^* x)$. [The quantity $U_i(x)$ has the interpretation of the “utility” of x , whereas μ^* has the interpretation of a “price” per unit of x .]
- c. Show that $\sum_{i=1}^n x_i^* = C$.

22.23 Give an example of a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, a set $\Omega = \{\mathbf{x} : g(\mathbf{x}) \leq 0\}$, and a regular point $\mathbf{x}^* \in \Omega$, such that the following all hold simultaneously:

1. \mathbf{x}^* satisfies the FONC for set constraint Ω (Theorem 6.1).
2. \mathbf{x}^* satisfies the KKT condition for inequality constraint $g(\mathbf{x}) \leq 0$ (Theorem 21.1).
3. \mathbf{x}^* satisfies the SONC for set constraint Ω (Theorem 6.2).
4. \mathbf{x}^* does not satisfy the SONC for inequality constraint $g(\mathbf{x}) \leq 0$ (Theorem 21.2).

Be sure to show carefully that your choice of f , $\Omega = \{\mathbf{x} : g(\mathbf{x}) \leq 0\}$, and \mathbf{x}^* satisfies all the conditions above simultaneously.

22.24 This question is on duality theory for *nonlinear* programming problems, analogous to the theory for linear programming (Chapter 17). (A version for quadratic programming is considered in Exercise 17.24.)

Consider the following optimization problem:

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \end{aligned}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex, each component of $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is convex, and $f, \mathbf{g} \in \mathcal{C}^1$. Let us call this problem the *primal* problem.

Define the *dual* of the problem above as

$$\begin{aligned} & \text{maximize } q(\boldsymbol{\mu}) \\ & \text{subject to } \boldsymbol{\mu} \geq \mathbf{0}, \end{aligned}$$

where q is defined by

$$q(\boldsymbol{\mu}) = \min_{\mathbf{x} \in \mathbb{R}^n} l(\mathbf{x}, \boldsymbol{\mu}),$$

with $l(\mathbf{x}, \boldsymbol{\mu}) = f(\mathbf{x}) + \boldsymbol{\mu}^\top \mathbf{g}(\mathbf{x})$ the Lagrangian at $\mathbf{x}, \boldsymbol{\mu}$.

Prove the following results:

- a. If \mathbf{x}_0 and $\boldsymbol{\mu}_0$ are feasible points in the primal and dual, respectively, then $f(\mathbf{x}_0) \geq q(\boldsymbol{\mu}_0)$. This is the *weak duality lemma* for nonlinear programming, analogous to Lemma 17.1.
- b. If \mathbf{x}_0 and $\boldsymbol{\mu}_0$ are feasible points in the primal and dual, and $f(\mathbf{x}_0) = q(\boldsymbol{\mu}_0)$, then \mathbf{x}_0 and $\boldsymbol{\mu}_0$ are optimal solutions to the primal and dual, respectively.
- c. If the primal has an optimal (feasible) solution, then so does the dual, and their objective function values are equal. (You may assume regularity.) This is the *duality theorem* for nonlinear programming, analogous to Theorem 17.2.

22.25 Consider the matrix

$$\mathbf{M} = \begin{bmatrix} 1 & \gamma & -1 \\ \gamma & 1 & 2 \\ -1 & 2 & 5 \end{bmatrix},$$

where γ is a parameter.

- a. Find the Schur complement of $\mathbf{M}(1,1)$;
- b. Find the Schur complement of $\mathbf{M}(2:3, 2:3)$ (the bottom-right 2×2 submatrix of \mathbf{M} , using MATLAB notation).

22.26 Represent the Lyapunov inequality

$$\mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} < 0,$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix},$$

as a canonical LMI.

22.27 Let \mathbf{A} , \mathbf{B} , and \mathbf{R} be given matrices such that $\mathbf{R} = \mathbf{R}^\top > 0$. Suppose that we wish to find a symmetric positive definite matrix \mathbf{P} satisfying the following quadratic inequality:

$$\mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P} < 0.$$

Represent this inequality in the form of LMIs. (This inequality should not be confused with the *algebraic Riccati inequality*, which has a negative sign in front of the third term.)

22.28 Let

$$\mathbf{A} = \begin{bmatrix} -0.9501 & -0.4860 & -0.4565 \\ -0.2311 & -0.8913 & -0.0185 \\ -0.6068 & -0.7621 & -0.8214 \end{bmatrix}.$$

Write a MATLAB program that finds a matrix \mathbf{P} satisfying $0.1\mathbf{I}_3 \leq \mathbf{P} \leq \mathbf{I}_3$ and

$$\mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} \leq 0.$$

CHAPTER 23

ALGORITHMS FOR CONSTRAINED OPTIMIZATION

23.1 Introduction

In Part II we discussed algorithms for solving *unconstrained* optimization problems. In this chapter we present some simple algorithms for solving special *constrained* optimization problems. The methods here build on those of Part II.

We begin our presentation in the next section with a discussion of *projected methods*, including a treatment of projected gradient methods for problems with linear equality constraints. We then consider *Lagrangian methods*. Finally, we consider *penalty methods*. This chapter is intended as an introduction to ideas underlying methods for solving constrained optimization problems. For an in-depth coverage of the subject, we refer the reader to [11].

23.2 Projections

The optimization algorithms considered in Part II have the general form

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)},$$

where $\mathbf{d}^{(k)}$ is typically a function of $\nabla f(\mathbf{x}^{(k)})$. The value of $\mathbf{x}^{(k)}$ is not constrained to lie inside any particular set. Such an algorithm is not immediately applicable to solving constrained optimization problems in which the decision variable is required to lie within a prespecified constraint set.

Consider the optimization problem

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \Omega. \end{aligned}$$

If we use the algorithm above to solve this constrained problem, the iterates $\mathbf{x}^{(k)}$ may not satisfy the constraints. Therefore, we need to modify the algorithms to take into account the presence of the constraints. A simple modification involves the introduction of a *projection*. The idea is as follows. If $\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ is in Ω , then we set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ as usual. If, on the other hand, $\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ is not in Ω , then we “project” it back into Ω before setting $\mathbf{x}^{(k+1)}$.

To illustrate the projection method, consider the case where the constraint set $\Omega \subset \mathbb{R}^n$ is given by

$$\Omega = \{\mathbf{x} : l_i \leq x_i \leq u_i, i = 1, \dots, n\}.$$

In this case, Ω is a “box” in \mathbb{R}^n ; for this reason, this form of Ω is called a *box constraint*. Given a point $\mathbf{x} \in \mathbb{R}^n$, define $\mathbf{y} = \mathbf{\Pi}[\mathbf{x}] \in \mathbb{R}^n$ by

$$y_i = \min\{u_i, \max\{l_i, x_i\}\} = \begin{cases} u_i & \text{if } x_i > u_i \\ x_i & \text{if } l_i \leq x_i \leq u_i \\ l_i & \text{if } x_i < l_i. \end{cases}$$

The point $\mathbf{\Pi}[\mathbf{x}]$ is called the *projection* of \mathbf{x} onto Ω . Note that $\mathbf{\Pi}[\mathbf{x}]$ is actually the “closest” point in Ω to \mathbf{x} . Using the projection operator $\mathbf{\Pi}$, we can modify the previous unconstrained algorithm as follows:

$$\mathbf{x}^{(k+1)} = \mathbf{\Pi}[\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}].$$

Note that the iterates $\mathbf{x}^{(k)}$ now all lie inside Ω . We call the algorithm above a *projected algorithm*.

In the more general case, we can define the projection onto Ω :

$$\mathbf{\Pi}[\mathbf{x}] = \arg \min_{\mathbf{z} \in \Omega} \|\mathbf{z} - \mathbf{x}\|.$$

In this case, $\mathbf{\Pi}[\mathbf{x}]$ is again the “closest” point in Ω to \mathbf{x} . This projection operator is well-defined only for certain types of constraint sets: for example, closed convex sets (see Exercise 22.19). For some sets Ω , the “arg min” above is not well-defined. If the projection $\mathbf{\Pi}$ is well-defined, we can similarly apply the projected algorithm

$$\mathbf{x}^{(k+1)} = \mathbf{\Pi}[\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}].$$

In some cases, there is a formula for computing $\Pi[\mathbf{x}]$. For example, if Ω represents a box constraint as described above, then the formula given previously can be used. Another example is where Ω is a linear variety, which is discussed in the next section. In general, even if the projection Π is well-defined, computation of $\Pi[\mathbf{x}]$ for a given \mathbf{x} may not be easy. Often, the projection $\Pi[\mathbf{x}]$ may have to be computed numerically. However, the numerical computation of $\Pi[\mathbf{x}]$ itself entails solving an optimization algorithm. Indeed, the computation of $\Pi[\mathbf{x}]$ may be as difficult as the original optimization problem, as is the case in the following example:

$$\begin{aligned} & \text{minimize} && \|\mathbf{x}\|^2 \\ & \text{subject to} && \mathbf{x} \in \Omega. \end{aligned}$$

Note that the solution to the problem in this case can be written as $\Pi[\mathbf{0}]$. Therefore, if $\mathbf{0} \notin \Omega$, the computation of a projection is equivalent to solving the given optimization problem.

As an example, consider the projection method applied specifically to the gradient algorithm (see Chapter 8). Recall that the vector $-\nabla f(\mathbf{x})$ points in the direction of maximum rate of decrease of f at \mathbf{x} . This was the basis for gradient methods for unconstrained optimization, which have the form $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})$, where α_k is the step size. The choice of the step size α_k depends on the particular gradient algorithm. For example, recall that in the steepest descent algorithm, $\alpha_k = \arg \min_{\alpha \geq 0} f(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)}))$.

The projected version of the gradient algorithm has the form

$$\mathbf{x}^{(k+1)} = \Pi[\mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})].$$

We refer to the above as the *projected gradient algorithm*.

Example 23.1 Consider the problem

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} \\ & \text{subject to} && \|\mathbf{x}\|^2 = 1, \end{aligned}$$

where $\mathbf{Q} = \mathbf{Q}^\top > 0$. Suppose that we apply a *fixed-step-size projected gradient algorithm* to this problem.

- a. Derive a formula for the update equation for the algorithm (i.e., write down an explicit formula for $\mathbf{x}^{(k+1)}$ as a function of $\mathbf{x}^{(k)}$, \mathbf{Q} , and the fixed step size α). You may assume that the argument in the projection operator to obtain $\mathbf{x}^{(k)}$ is never zero.
- b. Is it possible for the algorithm not to converge to an optimal solution even if the step size $\alpha > 0$ is taken to be arbitrarily small?
- c. Show that for $0 < \alpha < 1/\lambda_{\max}$ (where λ_{\max} is the largest eigenvalue of \mathbf{Q}), the fixed-step-size projected gradient algorithm (with step size α)

converges to an optimal solution, provided that $\mathbf{x}^{(0)}$ is not orthogonal to the eigenvectors of \mathbf{Q} corresponding to the smallest eigenvalue. (Assume that the eigenvalues are distinct.)

Solution:

- a. The projection operator in this case simply maps any vector to the closest point on the unit circle. Therefore, the projection operator is given by $\Pi[\mathbf{x}] = \mathbf{x}/\|\mathbf{x}\|$, provided that $\mathbf{x} \neq \mathbf{0}$. The update equation is

$$\mathbf{x}^{(k+1)} = \beta_k(\mathbf{x}^{(k)} - \alpha\mathbf{Q}\mathbf{x}^{(k)}) = \beta_k(\mathbf{I} - \alpha\mathbf{Q})\mathbf{x}^{(k)},$$

where $\beta_k = 1/\|(\mathbf{I} - \alpha\mathbf{Q})\mathbf{x}^{(k)}\|$ (i.e., it is whatever constant scaling is needed to make $\mathbf{x}^{(k+1)}$ have unit norm).

- b. If we start with $\mathbf{x}^{(0)}$ being an eigenvector of \mathbf{Q} , then $\mathbf{x}^{(k)} = \mathbf{x}^{(0)}$ for all k . Therefore, if the corresponding eigenvalue is not the smallest, then clearly the algorithm is stuck at a point that is not optimal.

- c. We have

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \beta_k(\mathbf{I} - \alpha\mathbf{Q})\mathbf{x}^{(k)} \\ &= \beta_k(\mathbf{I} - \alpha\mathbf{Q})(y_1^{(k)}\mathbf{v}_1 + \cdots + y_n^{(k)}\mathbf{v}_n) \\ &= \beta_k(y_1^{(k)}(\mathbf{I} - \alpha\mathbf{Q})\mathbf{v}_1 + \cdots + y_n^{(k)}(\mathbf{I} - \alpha\mathbf{Q})\mathbf{v}_n).\end{aligned}$$

But $(\mathbf{I} - \alpha\mathbf{Q})\mathbf{v}_i = (1 - \alpha\lambda_i)\mathbf{v}_i$, where λ_i is the eigenvalue corresponding to \mathbf{v}_i . Hence,

$$\mathbf{x}^{(k+1)} = \beta_k(y_1^{(k)}(1 - \alpha\lambda_1)\mathbf{v}_1 + \cdots + y_n^{(k)}(1 - \alpha\lambda_n)\mathbf{v}_n),$$

which means that $y_i^{(k+1)} = \beta_k y_i^{(k)}(1 - \alpha\lambda_i)$. In other words, $y_i^{(k)} = \beta^{(k)} y_i^{(0)}(1 - \alpha\lambda_i)^k$, where $\beta^{(k)} = \prod_{i=0}^{k-1} \beta_k$. We rewrite $\mathbf{x}^{(k)}$ as

$$\begin{aligned}\mathbf{x}^{(k)} &= \sum_{i=1}^n y_i^{(k)} \mathbf{v}_i \\ &= y_1^{(k)} \left(\mathbf{v}_1 + \sum_{i=2}^n \frac{y_i^{(k)}}{y_1^{(k)}} \mathbf{v}_i \right).\end{aligned}$$

Assuming that $y_1^{(0)} \neq 0$, we obtain

$$\frac{y_i^{(k)}}{y_1^{(k)}} = \frac{y_i^{(0)}(1 - \alpha\lambda_i)^k}{y_1^{(0)}(1 - \alpha\lambda_1)^k} = \frac{y_i^{(0)}}{y_1^{(0)}} \left(\frac{1 - \alpha\lambda_i}{1 - \alpha\lambda_1} \right)^k.$$

Using the fact that $(1 - \alpha\lambda_i)/(1 - \alpha\lambda_1) < 1$ (because the $\lambda_i > \lambda_1$ for $i > 1$ and $\alpha < 1/\lambda_{\max}$), we deduce that

$$\frac{y_i^{(k)}}{y_1^{(k)}} \rightarrow 0,$$

which implies that $\mathbf{x}^{(k)} \rightarrow \mathbf{v}_1$, as required. ■

23.3 Projected Gradient Methods with Linear Constraints

In this section we consider optimization problems of the form

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b}, \end{aligned}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m < n$, $\text{rank } \mathbf{A} = m$, $\mathbf{b} \in \mathbb{R}^m$. We assume throughout that $f \in \mathcal{C}^1$. In the problem above, the constraint set is $\Omega = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}\}$. The specific structure of the constraint set allows us to compute the projection operator Π using the *orthogonal projector* (see Section 3.3). Specifically, $\Pi[\mathbf{x}]$ can be defined using the orthogonal projector matrix \mathbf{P} given by

$$\mathbf{P} = \mathbf{I}_n - \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1} \mathbf{A}$$

(see Example 12.5). Two important properties of the orthogonal projector \mathbf{P} that we use in this section are (see Theorem 3.5):

1. $\mathbf{P} = \mathbf{P}^\top$.
2. $\mathbf{P}^2 = \mathbf{P}$.

Another property of the orthogonal projector that we need in our discussion is given in the following lemma.

Lemma 23.1 *Let $\mathbf{v} \in \mathbb{R}^n$. Then, $\mathbf{P}\mathbf{v} = \mathbf{0}$ if and only if $\mathbf{v} \in \mathcal{R}(\mathbf{A}^\top)$. In other words, $\mathcal{N}(\mathbf{P}) = \mathcal{R}(\mathbf{A}^\top)$. Moreover, $\mathbf{A}\mathbf{v} = \mathbf{0}$ if and only if $\mathbf{v} \in \mathcal{R}(\mathbf{P})$; that is, $\mathcal{N}(\mathbf{A}) = \mathcal{R}(\mathbf{P})$.* \square

Proof. \Rightarrow : We have

$$\begin{aligned} \mathbf{P}\mathbf{v} &= (\mathbf{I}_n - \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1} \mathbf{A})\mathbf{v} \\ &= \mathbf{v} - \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1} \mathbf{A}\mathbf{v}. \end{aligned}$$

If $\mathbf{P}\mathbf{v} = \mathbf{0}$, then

$$\mathbf{v} = \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1} \mathbf{A}\mathbf{v}$$

and hence $\mathbf{v} \in \mathcal{R}(\mathbf{A}^\top)$.

\Leftarrow : Suppose that there exists $\mathbf{u} \in \mathbb{R}^m$ such that $\mathbf{v} = \mathbf{A}^\top \mathbf{u}$. Then,

$$\begin{aligned} \mathbf{P}\mathbf{v} &= (\mathbf{I}_n - \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1} \mathbf{A})\mathbf{A}^\top \mathbf{u} \\ &= \mathbf{A}^\top \mathbf{u} - \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1} \mathbf{A}\mathbf{A}^\top \mathbf{u} \\ &= \mathbf{0}. \end{aligned}$$

Hence, we have proved that $\mathcal{N}(\mathbf{P}) = \mathcal{R}(\mathbf{A}^\top)$.

Using an argument similar to that above, we can show that $\mathcal{N}(\mathbf{A}) = \mathcal{R}(\mathbf{P})$. ■

Recall that in unconstrained optimization, the first-order necessary condition for a point \mathbf{x}^* to be a local minimizer is $\nabla f(\mathbf{x}^*) = \mathbf{0}$ (see Section 6.2). In optimization problems with equality constraints, the Lagrange condition plays the role of the first-order necessary condition (see Section 20.4). When the constraint set takes the form $\{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}\}$, the Lagrange condition can be written as $\mathbf{P}\nabla f(\mathbf{x}^*) = \mathbf{0}$, as stated in the following proposition.

Proposition 23.1 *Let $\mathbf{x}^* \in \mathbb{R}^n$ be a feasible point. Then, $\mathbf{P}\nabla f(\mathbf{x}^*) = \mathbf{0}$ if and only if \mathbf{x}^* satisfies the Lagrange condition. \square*

Proof. By Lemma 23.1, $\mathbf{P}\nabla f(\mathbf{x}^*) = \mathbf{0}$ if and only if we have $\nabla f(\mathbf{x}^*) \in \mathcal{R}(\mathbf{A}^\top)$. This is equivalent to the condition that there exists $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ such that $\nabla f(\mathbf{x}^*) + \mathbf{A}^\top \boldsymbol{\lambda}^* = \mathbf{0}$, which together with the feasibility equation $\mathbf{A}\mathbf{x} = \mathbf{b}$, constitutes the Lagrange condition. \blacksquare

Recall that the projected gradient algorithm has the form

$$\mathbf{x}^{(k+1)} = \mathbf{\Pi}[\mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})].$$

For the case where the constraints are linear, it turns out that we can express the projection $\mathbf{\Pi}$ in terms of the matrix \mathbf{P} as follows:

$$\mathbf{\Pi}[\mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})] = \mathbf{x}^{(k)} - \alpha_k \mathbf{P}\nabla f(\mathbf{x}^{(k)}),$$

assuming that $\mathbf{x}^{(k)} \in \Omega$. Although the formula above can be derived algebraically (see Exercise 23.4), it is more insightful to derive the formula using a geometric argument, as follows. In our constrained optimization problem, the vector $-\nabla f(\mathbf{x})$ is not necessarily a feasible direction. In other words, if $\mathbf{x}^{(k)}$ is a feasible point and we apply the algorithm $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})$, then $\mathbf{x}^{(k+1)}$ need not be feasible. This problem can be overcome by replacing $-\nabla f(\mathbf{x}^{(k)})$ by a vector that points in a feasible direction. Note that the set of feasible directions is simply the nullspace $\mathcal{N}(\mathbf{A})$ of the matrix \mathbf{A} . Therefore, we should first project the vector $-\nabla f(\mathbf{x})$ onto $\mathcal{N}(\mathbf{A})$. This projection is equivalent to multiplication by the matrix \mathbf{P} . In summary, in the projection gradient algorithm, we update $\mathbf{x}^{(k)}$ according to the equation

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{P}\nabla f(\mathbf{x}^{(k)}).$$

The projected gradient algorithm has the following property.

Proposition 23.2 *In a projected gradient algorithm, if $\mathbf{x}^{(0)}$ is feasible, then each $\mathbf{x}^{(k)}$ is feasible; that is, for each $k \geq 0$, $\mathbf{A}\mathbf{x}^{(k)} = \mathbf{b}$. \square*

Proof. We proceed by induction. The result holds for $k = 0$ by assumption. Suppose now that $\mathbf{A}\mathbf{x}^{(k)} = \mathbf{b}$. We now show that $\mathbf{A}\mathbf{x}^{(k+1)} = \mathbf{b}$. To show this, first observe that $\mathbf{P}\nabla f(\mathbf{x}^{(k)}) \in \mathcal{N}(\mathbf{A})$. Therefore,

$$\begin{aligned} \mathbf{A}\mathbf{x}^{(k+1)} &= \mathbf{A}(\mathbf{x}^{(k)} - \alpha_k \mathbf{P}\nabla f(\mathbf{x}^{(k)})) \\ &= \mathbf{A}\mathbf{x}^{(k)} - \alpha_k \mathbf{A}\mathbf{P}\nabla f(\mathbf{x}^{(k)}) \\ &= \mathbf{b}, \end{aligned}$$

which completes the proof. ■

The projected gradient algorithm updates $\mathbf{x}^{(k)}$ in the direction of $-\mathbf{P}\nabla f(\mathbf{x}^{(k)})$. This vector points in the direction of maximum rate of decrease of f at $\mathbf{x}^{(k)}$ along the surface defined by $\mathbf{A}\mathbf{x} = \mathbf{b}$, as described in the following argument. Let \mathbf{x} be any feasible point and \mathbf{d} a feasible direction such that $\|\mathbf{d}\| = 1$. The rate of increase of f at \mathbf{x} in the direction \mathbf{d} is $\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle$. Next, we note that because \mathbf{d} is a feasible direction, it lies in $\mathcal{N}(\mathbf{A})$ and hence by Lemma 23.1, we have $\mathbf{d} \in \mathcal{R}(\mathbf{P}) = \mathcal{R}(\mathbf{P}^\top)$. So, there exists \mathbf{v} such that $\mathbf{d} = \mathbf{P}\mathbf{v}$. Hence,

$$\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle = \langle \nabla f(\mathbf{x}), \mathbf{P}^\top \mathbf{v} \rangle = \langle \mathbf{P}\nabla f(\mathbf{x}), \mathbf{v} \rangle.$$

By the Cauchy-Schwarz inequality,

$$\langle \mathbf{P}\nabla f(\mathbf{x}), \mathbf{v} \rangle \leq \|\mathbf{P}\nabla f(\mathbf{x})\| \|\mathbf{v}\|$$

with equality if and only if the direction of \mathbf{v} is parallel with the direction of $\mathbf{P}\nabla f(\mathbf{x})$. Therefore, the vector $-\mathbf{P}\nabla f(\mathbf{x})$ points in the direction of maximum rate of decrease of f at \mathbf{x} among all feasible directions.

Following the discussion in Chapter 8 for gradient methods in unconstrained optimization, we suggest the following gradient method for our constrained problem. Suppose that we have a starting point $\mathbf{x}^{(0)}$, which we assume is feasible; that is, $\mathbf{A}\mathbf{x}^{(0)} = \mathbf{b}$. Consider the point $\mathbf{x} = \mathbf{x}^{(0)} - \alpha \mathbf{P}\nabla f(\mathbf{x}^{(0)})$, where $\alpha \in \mathbb{R}$. As usual, the scalar α is called the step size. By the discussion above, \mathbf{x} is also a feasible point. Using a Taylor series expansion of f about $\mathbf{x}^{(0)}$ and the fact that $\mathbf{P} = \mathbf{P}^2 = \mathbf{P}^\top \mathbf{P}$, we get

$$\begin{aligned} f(\mathbf{x}^{(0)} - \alpha \mathbf{P}\nabla f(\mathbf{x}^{(0)})) &= f(\mathbf{x}^{(0)}) - \alpha \nabla f(\mathbf{x}^{(0)})^\top \mathbf{P}\nabla f(\mathbf{x}^{(0)}) + o(\alpha) \\ &= f(\mathbf{x}^{(0)}) - \alpha \|\mathbf{P}\nabla f(\mathbf{x}^{(0)})\|^2 + o(\alpha). \end{aligned}$$

Thus, if $\mathbf{P}\nabla f(\mathbf{x}^{(0)}) \neq 0$, that is, $\mathbf{x}^{(0)}$ does not satisfy the Lagrange condition, then we can choose an α sufficiently small such that $f(\mathbf{x}) < f(\mathbf{x}^{(0)})$, which means that $\mathbf{x} = \mathbf{x}^{(0)} - \alpha \mathbf{P}\nabla f(\mathbf{x}^{(0)})$ is an improvement over $\mathbf{x}^{(0)}$. This is the basis for the projected gradient algorithm $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{P}\nabla f(\mathbf{x}^{(k)})$, where the initial point $\mathbf{x}^{(0)}$ satisfies $\mathbf{A}\mathbf{x}^{(0)} = \mathbf{b}$ and α_k is some step size. As for unconstrained gradient methods, the choice of α_k determines the behavior of the algorithm. For small step sizes, the algorithm progresses slowly, while large step sizes may result in a zigzagging path. A well-known variant of the projected gradient algorithm is the *projected steepest descent algorithm*, where α_k is given by

$$\alpha_k = \arg \min_{\alpha \geq 0} f(\mathbf{x}^{(k)} - \alpha \mathbf{P}\nabla f(\mathbf{x}^{(k)})).$$

The following theorem states that the projected steepest descent algorithm is a descent algorithm, in the sense that at each step the value of the objective function decreases.

Theorem 23.1 If $\{\mathbf{x}^{(k)}\}$ is the sequence of points generated by the projected steepest descent algorithm and if $\mathbf{P}\nabla f(\mathbf{x}^{(k)}) \neq \mathbf{0}$, then $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$. \square

Proof. First, recall that

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{P}\nabla f(\mathbf{x}^{(k)}),$$

where $\alpha_k \geq 0$ is the minimizer of

$$\phi_k(\alpha) = f(\mathbf{x}^{(k)} - \alpha \mathbf{P}\nabla f(\mathbf{x}^{(k)}))$$

over all $\alpha \geq 0$. Thus, for $\alpha \geq 0$, we have

$$\phi_k(\alpha_k) \leq \phi_k(\alpha).$$

By the chain rule,

$$\begin{aligned}\phi'_k(0) &= \frac{d\phi_k}{d\alpha}(0) \\ &= -\nabla f(\mathbf{x}^{(k)} - 0 \mathbf{P}\nabla f(\mathbf{x}^{(k)}))^T \mathbf{P}\nabla f(\mathbf{x}^{(k)}) \\ &= -\nabla f(\mathbf{x}^{(k)})^T \mathbf{P}\nabla f(\mathbf{x}^{(k)}).\end{aligned}$$

Using the fact that $\mathbf{P} = \mathbf{P}^2 = \mathbf{P}^\top \mathbf{P}$, we get

$$\phi'_k(0) = -\nabla f(\mathbf{x}^{(k)})^T \mathbf{P}^\top \mathbf{P}\nabla f(\mathbf{x}^{(k)}) = -\|\mathbf{P}\nabla f(\mathbf{x}^{(k)})\|^2 < 0,$$

because $\mathbf{P}\nabla f(\mathbf{x}^{(k)}) \neq \mathbf{0}$ by assumption. Thus, there exists $\bar{\alpha} > 0$ such that $\phi_k(0) > \phi_k(\alpha)$ for all $\alpha \in (0, \bar{\alpha}]$. Hence,

$$f(\mathbf{x}^{(k+1)}) = \phi_k(\alpha_k) \leq \phi_k(\bar{\alpha}) < \phi_k(0) = f(\mathbf{x}^{(k)}),$$

which completes the proof of the theorem. \blacksquare

In Theorem 23.1 we needed the assumption that $\mathbf{P}\nabla f(\mathbf{x}^{(k)}) \neq \mathbf{0}$ to prove that the algorithm possesses the descent property. If for some k , we have $\mathbf{P}\nabla f(\mathbf{x}^{(k)}) = \mathbf{0}$, then by Proposition 23.1 the point $\mathbf{x}^{(k)}$ satisfies the Lagrange condition. This condition can be used as a stopping criterion for the algorithm. Note that in this case, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$. For the case where f is a convex function, the condition $\mathbf{P}\nabla f(\mathbf{x}^{(k)}) = \mathbf{0}$ is, in fact, equivalent to $\mathbf{x}^{(k)}$ being a global minimizer of f over the constraint set $\{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}\}$. We show this in the following proposition.

Proposition 23.3 The point $\mathbf{x}^* \in \mathbb{R}^n$ is a global minimizer of a convex function f over $\{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}\}$ if and only if $\mathbf{P}\nabla f(\mathbf{x}^*) = \mathbf{0}$. \square

Proof. We first write $\mathbf{h}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$. Then, the constraints can be written as $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, and the problem is of the form considered in earlier chapters.

Note that $D\mathbf{h}(\mathbf{x}) = \mathbf{A}$. Hence, $\mathbf{x}^* \in \mathbb{R}^n$ is a global minimizer of f if and only if the Lagrange condition holds (see Theorem 22.8). By Proposition 23.1, this is true if and only if $P\nabla f(\mathbf{x}^*) = \mathbf{0}$, and this completes the proof. ■

For an application of the projected steepest descent algorithm to minimum fuel and minimum amplitude control problems in linear discrete systems, see [78].

23.4 Lagrangian Algorithms

In this section we consider an optimization method based on the Lagrangian function (see Section 20.4). The basic idea is to use gradient algorithms to update simultaneously the decision variable and Lagrange multiplier vector. We consider first the case with equality constraints, followed by inequality constraints.

Lagrangian Algorithm for Equality Constraints

Consider the following optimization problem with equality constraints:

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{aligned}$$

where $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Recall that for this problem the Lagrangian function is given by

$$l(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{h}(\mathbf{x}).$$

Assume that $f, \mathbf{h} \in \mathcal{C}^2$; as usual, denote the Hessian of the Lagrangian by $\mathbf{L}(\mathbf{x}, \boldsymbol{\lambda})$.

The Lagrangian algorithm for this problem is given by

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} - \alpha_k (\nabla f(\mathbf{x}^{(k)}) + D\mathbf{h}(\mathbf{x}^{(k)})^\top \boldsymbol{\lambda}^{(k)}), \\ \boldsymbol{\lambda}^{(k+1)} &= \boldsymbol{\lambda}^{(k)} + \beta_k \mathbf{h}(\mathbf{x}^{(k)}). \end{aligned}$$

Notice that the update equation for $\mathbf{x}^{(k)}$ is a gradient algorithm for minimizing the Lagrangian with respect to its \mathbf{x} argument, and the update equation for $\boldsymbol{\lambda}^{(k)}$ is a gradient algorithm for maximizing the Lagrangian with respect to its $\boldsymbol{\lambda}$ argument. Because only the gradient is used, the method is also called the *first-order Lagrangian algorithm*.

The following lemma establishes that if the algorithm converges, the limit must satisfy the Lagrange condition. More specifically, the lemma states that any *fixed point* of the algorithm must satisfy the Lagrange condition. A fixed point of an update algorithm is simply a point with the property that when updated using the algorithm, the resulting point is equal to the given point. For the case of the Lagrangian algorithm, which updates both $\mathbf{x}^{(k)}$ and

$\lambda^{(k)}$ vectors, a fixed point is a *pair* of vectors. If the Lagrangian algorithm converges, the limit must be a fixed point. We omit the proof of the lemma because it follows easily by inspection.

Lemma 23.2 *For the Lagrangian algorithm for updating $\mathbf{x}^{(k)}$ and $\boldsymbol{\lambda}^{(k)}$, the pair $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is a fixed point if and only if it satisfies the Lagrange condition.*

□

Below, we use $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ to denote a pair satisfying the Lagrange condition. Assume that $L(\mathbf{x}^*, \boldsymbol{\lambda}^*) > 0$. Also assume that \mathbf{x}^* is a *regular* point. With these assumptions, we are now ready to state and prove that the algorithm is locally convergent. For simplicity, we will take α_k and β_k to be fixed constants (not depending on k), denoted α and β , respectively.

Theorem 23.2 *For the Lagrangian algorithm for updating $\mathbf{x}^{(k)}$ and $\boldsymbol{\lambda}^{(k)}$, provided that α and β are sufficiently small, there is a neighborhood of $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ such that if the pair $(\mathbf{x}^{(0)}, \boldsymbol{\lambda}^{(0)})$ is in this neighborhood, then the the algorithm converges to $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ with at least a linear order of convergence.*

□

Proof. We can rescale \mathbf{x} and $\boldsymbol{\lambda}$ by appropriate constants (so that the assumptions are preserved) and effectively change the relative values of the step sizes for the update equations. Therefore, without loss of generality, we can take $\beta = \alpha$.

We will set up our proof by introducing some convenient notation. Given a pair $(\mathbf{x}, \boldsymbol{\lambda})$, let $\mathbf{w} = [\mathbf{x}^\top, \boldsymbol{\lambda}^\top]^\top$ be the $(n+m)$ -vector constructed by concatenating \mathbf{x} and $\boldsymbol{\lambda}$. Similarly define $\mathbf{w}^{(k)} = [\mathbf{x}^{(k)\top}, \boldsymbol{\lambda}^{(k)\top}]^\top$ and $\mathbf{w}^* = [\mathbf{x}^{*\top}, \boldsymbol{\lambda}^{*\top}]^\top$. Define the map $\mathbf{U} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^{n+m}$ by

$$\mathbf{U}(\mathbf{w}) = \begin{bmatrix} \mathbf{x} - \alpha(\nabla f(\mathbf{x}) + D\mathbf{h}(\mathbf{x})^\top \boldsymbol{\lambda}) \\ \boldsymbol{\lambda} + \alpha\mathbf{h}(\mathbf{x}) \end{bmatrix}.$$

Then, the Lagrangian algorithm can be rewritten as

$$\mathbf{w}^{(k+1)} = \mathbf{U}(\mathbf{w}^{(k)}).$$

We now write $\|\mathbf{w}^{(k+1)} - \mathbf{w}^*\|$ in terms of $\|\mathbf{w}^{(k)} - \mathbf{w}^*\|$, where $\|\cdot\|$ denotes the usual Euclidean norm. By Lemma 23.2, $\mathbf{w}^* = [\mathbf{x}^{*\top}, \boldsymbol{\lambda}^{*\top}]^\top$ is a fixed point of $\mathbf{w}^{(k+1)} = \mathbf{U}(\mathbf{w}^{(k)})$. Therefore,

$$\|\mathbf{w}^{(k+1)} - \mathbf{w}^*\| = \|\mathbf{U}(\mathbf{w}^{(k)}) - \mathbf{U}(\mathbf{w}^*)\|.$$

Let $D\mathbf{U}$ be the (matrix) derivative of \mathbf{U} :

$$D\mathbf{U}(\mathbf{w}) = \mathbf{I} + \alpha \begin{bmatrix} -L(\mathbf{x}, \boldsymbol{\lambda}) & -D\mathbf{h}(\mathbf{x})^\top \\ D\mathbf{h}(\mathbf{x}) & \mathbf{O} \end{bmatrix}.$$

By the mean value theorem (see Theorem 5.9),

$$\mathbf{U}(\mathbf{w}^{(k)}) - \mathbf{U}(\mathbf{w}^*) = \mathbf{G}(\mathbf{w}^{(k)})(\mathbf{w}^{(k)} - \mathbf{w}^*),$$

where $\mathbf{G}(\mathbf{w}^{(k)})$ is a matrix whose rows are the rows of $D\mathbf{U}$ evaluated at points that lie on the line segment joining $\mathbf{w}^{(k)}$ and \mathbf{w}^* (these points may differ from row to row). Taking norms of both sides of the equation above,

$$\|\mathbf{U}(\mathbf{w}^{(k)}) - \mathbf{U}(\mathbf{w}^*)\| \leq \|\mathbf{G}(\mathbf{w}^{(k)})\| \|\mathbf{w}^{(k)} - \mathbf{w}^*\|.$$

Finally, combining the above, we have

$$\|\mathbf{w}^{(k+1)} - \mathbf{w}^*\| \leq \|\mathbf{G}(\mathbf{w}^{(k)})\| \|\mathbf{w}^{(k)} - \mathbf{w}^*\|.$$

We now claim that for sufficiently small $\alpha > 0$, $\|D\mathbf{U}(\mathbf{w}^*)\| < 1$. Our argument here follows [11, Section 4.4]. Let

$$\mathbf{M} = \begin{bmatrix} -\mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) & -D\mathbf{h}(\mathbf{x}^*)^\top \\ D\mathbf{h}(\mathbf{x}^*) & \mathbf{O} \end{bmatrix},$$

so that $D\mathbf{U}(\mathbf{w}^*) = \mathbf{I} + \alpha\mathbf{M}$. Hence, to prove the claim, it suffices to show that the eigenvalues of \mathbf{M} all lie in the open left-half complex plane.

For any complex vector \mathbf{y} , let \mathbf{y}^H represent its complex conjugate transpose (or Hermitian) and $\Re(\mathbf{y})$ its real part. Let λ be an eigenvalue of \mathbf{M} and $\mathbf{w} = [\mathbf{x}^\top, \boldsymbol{\lambda}^\top]^\top \neq \mathbf{0}$ be a corresponding eigenvector. Now, $\Re(\mathbf{w}^H \mathbf{M} \mathbf{w}) = \Re(\lambda) \|\mathbf{w}\|^2$. However, from the structure of \mathbf{M} , we can readily see that

$$\begin{aligned} \Re(\mathbf{w}^H \mathbf{M} \mathbf{w}) &= -\Re(\mathbf{x}^H \mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) \mathbf{x}) - \Re(\mathbf{x}^H D\mathbf{h}(\mathbf{x}^*)^\top \boldsymbol{\lambda}) + \Re(\boldsymbol{\lambda}^H D\mathbf{h}(\mathbf{x}^*) \mathbf{x}) \\ &= -\Re(\mathbf{x}^H \mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) \mathbf{x}). \end{aligned}$$

By the assumption that $\mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) > 0$, we know that $\Re(\mathbf{x}^H \mathbf{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) \mathbf{x}) > 0$ if $\mathbf{x} \neq \mathbf{0}$. Therefore, comparing the two equations above, we deduce that $\Re(\lambda) < 0$, as required, provided that \mathbf{x} is nonzero, as we now demonstrate.

Now, suppose that $\mathbf{x} = \mathbf{0}$. Because \mathbf{w} is an eigenvector of \mathbf{M} , we have $\mathbf{M}\mathbf{w} = \lambda\mathbf{w}$. Extracting the first n components, we have $D\mathbf{h}(\mathbf{x}^*)^\top \boldsymbol{\lambda} = \mathbf{0}$. By the regularity assumption, we deduce that $\boldsymbol{\lambda} = \mathbf{0}$. This contradicts the assumption that $\mathbf{w} \neq \mathbf{0}$. Hence we conclude that $\mathbf{x} \neq \mathbf{0}$, which completes the proof of our claim that for sufficiently small $\alpha > 0$, $\|D\mathbf{U}(\mathbf{w}^*)\| < 1$.

The result of the foregoing claim allows us to pick constants $\eta > 0$ and $\kappa < 1$ such that for all $\mathbf{w} = [\mathbf{x}^\top, \boldsymbol{\lambda}^\top]^\top$ satisfying $\|\mathbf{w} - \mathbf{w}^*\| \leq \eta$, we have $\|\mathbf{G}(\mathbf{w})\| \leq \kappa$ (this follows from the continuity of $D\mathbf{U}$ and norms).

To complete the proof, suppose that $\|\mathbf{w}^{(0)} - \mathbf{w}^*\| \leq \eta$. We will show by induction that for all $k \geq 0$, $\|\mathbf{w}^{(k)} - \mathbf{w}^*\| \leq \eta$ and $\|\mathbf{w}^{(k+1)} - \mathbf{w}^*\| \leq \kappa \|\mathbf{w}^{(k)} - \mathbf{w}^*\|$, from which we conclude that $\mathbf{w}^{(k)}$ converges to \mathbf{w}^* with at least linear order of convergence. For $k = 0$, the result follows because $\|\mathbf{w}^{(0)} - \mathbf{w}^*\| \leq \eta$ by assumption, and

$$\|\mathbf{w}^{(1)} - \mathbf{w}^*\| \leq \|\mathbf{G}(\mathbf{w}^{(0)})\| \|\mathbf{w}^{(0)} - \mathbf{w}^*\| \leq \kappa \|\mathbf{w}^{(0)} - \mathbf{w}^*\|,$$

which follows from $\|\mathbf{w}^{(0)} - \mathbf{w}^*\| \leq \eta$. So suppose that the result holds for k . This implies that $\|\mathbf{G}(\mathbf{w}^{(k)})\| \leq \kappa$. To show the $k + 1$ case, we write

$$\|\mathbf{w}^{(k+1)} - \mathbf{w}^*\| \leq \|\mathbf{G}(\mathbf{w}^{(k)})\| \|\mathbf{w}^{(k)} - \mathbf{w}^*\| \leq \kappa \|\mathbf{w}^{(k)} - \mathbf{w}^*\| \leq \eta.$$

This means that $\|\mathbf{G}(\mathbf{w}^{(k+1)})\| \leq \kappa$, from which we can write

$$\|\mathbf{w}^{(k+2)} - \mathbf{w}^*\| \leq \|\mathbf{G}(\mathbf{w}^{(k+1)})\| \|\mathbf{w}^{(k+1)} - \mathbf{w}^*\| \leq \kappa \|\mathbf{w}^{(k+1)} - \mathbf{w}^*\|.$$

This completes the proof. ■

Lagrangian Algorithm for Inequality Constraints

Consider the following optimization problem with inequality constraints:

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \end{aligned}$$

where $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^p$. Recall that for this problem the Lagrangian function is given by

$$l(\mathbf{x}, \boldsymbol{\mu}) = f(\mathbf{x}) + \boldsymbol{\mu}^\top \mathbf{g}(\mathbf{x}).$$

As before, assume that $f, \mathbf{g} \in \mathcal{C}^2$; as usual, denote the Hessian of the Lagrangian by $\mathbf{L}(\mathbf{x}, \boldsymbol{\mu})$.

The Lagrangian algorithm for this problem is given by

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} - \alpha_k (\nabla f(\mathbf{x}^{(k)}) + D\mathbf{g}(\mathbf{x}^{(k)})^\top \boldsymbol{\mu}^{(k)}), \\ \boldsymbol{\mu}^{(k+1)} &= [\boldsymbol{\mu}^{(k)} + \beta_k \mathbf{g}(\mathbf{x}^{(k)})]_+, \end{aligned}$$

where $[\cdot]_+ = \max\{\cdot, 0\}$ (applied componentwise). Notice that, as before, the update equation for $\mathbf{x}^{(k)}$ is a gradient algorithm for minimizing the Lagrangian with respect to its \mathbf{x} argument. The update equation for $\boldsymbol{\mu}^{(k)}$ is a *projected gradient algorithm* for maximizing the Lagrangian with respect to its $\boldsymbol{\mu}$ argument. The reason for the projection is that the KKT multiplier vector is required to be nonnegative to satisfy the KKT condition.

The following lemma establishes that if the algorithm converges, the limit must satisfy the KKT condition. As before, we use the notion of a fixed point to state the result formally. The proof is omitted because the result follows easily by inspection.

Lemma 23.3 *For the Lagrangian algorithm for updating $\mathbf{x}^{(k)}$ and $\boldsymbol{\mu}^{(k)}$, the pair $(\mathbf{x}^*, \boldsymbol{\mu}^*)$ is a fixed point if and only if it satisfies the KKT condition.* \square

As before, we use the notation $(\mathbf{x}^*, \boldsymbol{\mu}^*)$ to denote a pair satisfying the KKT condition. Assume that $\mathbf{L}(\mathbf{x}^*, \boldsymbol{\mu}^*) > 0$. Also assume that \mathbf{x}^* is a *regular* point. With these assumptions, we are now ready to state and prove that the algorithm is locally convergent. As before, we will take α_k and β_k to

be fixed constants (not depending on k), denoted α and β , respectively. Our analysis examines the behavior of the algorithm in two phases. In the first phase, the “nonactive” multipliers decrease to zero in finite time and remain at zero thereafter. In the second phase, the $\mathbf{x}^{(k)}$ iterates and the “active” multipliers converge jointly to their respective solutions, with at least a linear order of convergence.

Theorem 23.3 *For the Lagrangian algorithm for updating $\mathbf{x}^{(k)}$ and $\boldsymbol{\mu}^{(k)}$, provided that α and β are sufficiently small, there is a neighborhood of $(\mathbf{x}^*, \boldsymbol{\mu}^*)$ such that if the pair $(\mathbf{x}^{(0)}, \boldsymbol{\mu}^{(0)})$ is in this neighborhood, then (1) the nonactive multipliers reduce to zero in finite time and remain at zero thereafter and (2) the algorithm converges to $(\mathbf{x}^*, \boldsymbol{\mu}^*)$ with at least a linear order of convergence.*

□

Proof. As in the proof of Theorem 23.2, we can rescale \mathbf{x} and $\boldsymbol{\mu}$ by appropriate constants (so that the assumptions are preserved) and effectively change the relative values of the step sizes for the update equations. Therefore, without loss of generality, we can take $\beta = \alpha$.

We set up our proof using the same vector notation as before. Given a pair $(\mathbf{x}, \boldsymbol{\mu})$, let $\mathbf{w} = [\mathbf{x}^\top, \boldsymbol{\mu}^\top]^\top$ be the $(n + p)$ -vector constructed by concatenating \mathbf{x} and $\boldsymbol{\mu}$. Similarly define $\mathbf{w}^{(k)} = [\mathbf{x}^{(k)\top}, \boldsymbol{\mu}^{(k)\top}]^\top$ and $\mathbf{w}^* = [\mathbf{x}^{*\top}, \boldsymbol{\mu}^{*\top}]^\top$. Define the map \mathbf{U} as

$$\mathbf{U}(\mathbf{w}) = \begin{bmatrix} \mathbf{x} - \alpha(\nabla f(\mathbf{x}) + D\mathbf{g}(\mathbf{x})^\top \boldsymbol{\mu}) \\ \boldsymbol{\mu} + \alpha\mathbf{g}(\mathbf{x}) \end{bmatrix}.$$

Also, define the map $\mathbf{\Pi}$ by

$$\mathbf{\Pi}[\mathbf{w}] = \begin{bmatrix} \mathbf{x} \\ [\boldsymbol{\mu}]_+ \end{bmatrix}.$$

Then, the update equations can be rewritten as

$$\mathbf{w}^{(k+1)} = \mathbf{\Pi}[\mathbf{U}(\mathbf{w}^{(k)})].$$

Because $\mathbf{\Pi}$ is a projection onto the convex set $\{\mathbf{w} = [\mathbf{x}^\top, \boldsymbol{\mu}^\top]^\top : \boldsymbol{\mu} \geq \mathbf{0}\}$, it is a nonexpansive map (see [12, Proposition 3.2]), which means that $\|\mathbf{\Pi}[\mathbf{v}] - \mathbf{\Pi}[\mathbf{w}]\| \leq \|\mathbf{v} - \mathbf{w}\|$.

We now write $\|\mathbf{w}^{(k+1)} - \mathbf{w}^*\|$ in terms of $\|\mathbf{w}^{(k)} - \mathbf{w}^*\|$, where $\|\cdot\|$ denotes the usual Euclidean norm. By Lemma 23.3, $\mathbf{w}^* = [\mathbf{x}^{*\top}, \boldsymbol{\mu}^{*\top}]^\top$ is a fixed point of $\mathbf{w}^{(k+1)} = \mathbf{U}(\mathbf{w}^{(k)})$. Therefore,

$$\begin{aligned} \|\mathbf{w}^{(k+1)} - \mathbf{w}^*\| &= \|\mathbf{\Pi}[\mathbf{U}(\mathbf{w}^{(k)})] - \mathbf{\Pi}[\mathbf{U}(\mathbf{w}^*)]\| \\ &\leq \|\mathbf{U}(\mathbf{w}^{(k)}) - \mathbf{U}(\mathbf{w}^*)\| \end{aligned}$$

by the nonexpansiveness of Π . Let $D\mathbf{U}$ be the (matrix) derivative of \mathbf{U} :

$$D\mathbf{U}(\mathbf{w}) = \mathbf{I} + \alpha \begin{bmatrix} -\mathbf{L}(\mathbf{x}, \boldsymbol{\mu}) & -D\mathbf{g}(\mathbf{x})^\top \\ D\mathbf{g}(\mathbf{x}) & \mathbf{O} \end{bmatrix}.$$

By the mean value theorem,

$$\mathbf{U}(\mathbf{w}^{(k)}) - \mathbf{U}(\mathbf{w}^*) = \mathbf{G}(\mathbf{w}^{(k)})(\mathbf{w}^{(k)} - \mathbf{w}^*),$$

where $\mathbf{G}(\mathbf{w}^{(k)})$ is a matrix whose rows are the rows of $D\mathbf{U}$ evaluated at points that lie on the line segment joining $\mathbf{w}^{(k)}$ and \mathbf{w}^* (these points may differ from row to row). Taking norms of both sides of the equation above yields

$$\|\mathbf{U}(\mathbf{w}^{(k)}) - \mathbf{U}(\mathbf{w}^*)\| \leq \|\mathbf{G}(\mathbf{w}^{(k)})\| \|\mathbf{w}^{(k)} - \mathbf{w}^*\|.$$

Finally, combining the above, we obtain

$$\|\mathbf{w}^{(k+1)} - \mathbf{w}^*\| \leq \|\mathbf{G}(\mathbf{w}^{(k)})\| \|\mathbf{w}^{(k)} - \mathbf{w}^*\|.$$

Let \mathbf{g}_A represent those rows of \mathbf{g} corresponding to active constraints (at \mathbf{x}^*) and $\mathbf{g}_{\bar{A}}$ represent the remaining rows of \mathbf{g} . [Recall that by regularity, $D\mathbf{g}_A(\mathbf{x}^*)$ has full rank.] Given a vector $\boldsymbol{\mu}$, we divide it into two subvectors $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_{\bar{A}}$, according to active and nonactive components, respectively. (Note that $\boldsymbol{\mu}_{\bar{A}}^* = \mathbf{0}$, the zero vector.) Next, write $\mathbf{w}_A = [\mathbf{x}^\top, \boldsymbol{\mu}_A^\top]^\top$ and

$$\mathbf{U}_A(\mathbf{w}_A) = \begin{bmatrix} \mathbf{x} - \alpha(\nabla f(\mathbf{x}) + D\mathbf{g}_A(\mathbf{x})^\top \boldsymbol{\mu}_A) \\ \boldsymbol{\mu}_A + \alpha\mathbf{g}_A(\mathbf{x}) \end{bmatrix},$$

so that

$$D\mathbf{U}_A(\mathbf{w}_A) = \mathbf{I} + \alpha \begin{bmatrix} -\mathbf{L}(\mathbf{x}, \boldsymbol{\mu}_A) & -D\mathbf{g}_A(\mathbf{x})^\top \\ D\mathbf{g}_A(\mathbf{x}) & \mathbf{O} \end{bmatrix}.$$

Finally, let \mathbf{G}_A be such that $\mathbf{U}_A(\mathbf{w}_A^{(k)}) - \mathbf{U}_A(\mathbf{w}_A^*) = \mathbf{G}_A(\mathbf{w}_A^{(k)})(\mathbf{w}_A^{(k)} - \mathbf{w}_A^*)$ (by the mean value theorem as before).

We organize the remainder of our proof into four claims.

Claim 1: For sufficiently small $\alpha > 0$, $\|D\mathbf{U}_A(\mathbf{w}_A^*)\| < 1$.

The argument here parallels that of the proof of Theorem 23.2. So for the sake of brevity we omit the details.

The result of claim 1 allows us to pick constants $\eta > 0$, $\delta > 0$, and $\kappa_A < 1$ such that for all $\mathbf{w} = [\mathbf{x}^\top, \boldsymbol{\mu}^\top]^\top$ satisfying $\|\mathbf{w} - \mathbf{w}^*\| \leq \eta$, $\|\mathbf{G}_A(\mathbf{w}_A)\| \leq \kappa_A$, and $\mathbf{g}_{\bar{A}}(\mathbf{x}) \leq -\delta\mathbf{e}$, where \mathbf{e} is the vector with all components equal to 1. The first inequality follows from claim 1 and the continuity of $D\mathbf{U}_A(\cdot)$ and $\|\cdot\|$. The second follows from the fact that the components of $\mathbf{g}_{\bar{A}}(\mathbf{x}^*)$ are negative.

Let $\kappa = \max\{\|\mathbf{G}(\mathbf{w})\| : \|\mathbf{w} - \mathbf{w}^*\| \leq \eta\}$, which we assume to be at least 1; otherwise, set $\kappa = 1$. Now pick $\varepsilon > 0$ such that $\varepsilon\kappa^{\varepsilon/(\alpha\delta)} \leq \eta$. We can do this because the left side of this inequality goes to 0 as $\varepsilon \rightarrow 0$. Assume for

convenience that $k_0 = \varepsilon/(\alpha\delta)$ is an integer; otherwise, replace all instances of $\varepsilon/(\alpha\delta)$ by the smallest integer that exceeds it (i.e., round it up to the closest integer).

For the remainder of this proof, let $\mathbf{w}^{(0)}$ satisfy $\|\mathbf{w}^{(0)} - \mathbf{w}^*\| \leq \varepsilon$.

Claim 2: For $k = 0, \dots, k_0$, $\|\mathbf{w}^{(k)} - \mathbf{w}^*\| \leq \eta$.

To prove the claim, we show by induction that $\|\mathbf{w}^{(k)} - \mathbf{w}^*\| \leq \varepsilon\kappa^k$ (which is bounded above by η provided that $k \leq k_0$). For $k = 0$, by assumption $\|\mathbf{w}^{(0)} - \mathbf{w}^*\| \leq \varepsilon = \varepsilon\kappa^0$, as required. For the inductive step, suppose that $\|\mathbf{w}^{(k)} - \mathbf{w}^*\| \leq \varepsilon\kappa^k$ for $k < k_0$. Now, using $\|\mathbf{w}^{(k+1)} - \mathbf{w}^*\| \leq \|G(\mathbf{w}^{(k)})\| \|\mathbf{w}^{(k)} - \mathbf{w}^*\|$ and the fact that $\|\mathbf{w}^{(k)} - \mathbf{w}^*\| \leq \eta$,

$$\|\mathbf{w}^{(k+1)} - \mathbf{w}^*\| \leq \|G(\mathbf{w}^{(k)})\| \|\mathbf{w}^{(k)} - \mathbf{w}^*\| \leq \kappa(\varepsilon\kappa^k) = \varepsilon\kappa^{k+1},$$

and the result now follows by induction.

Claim 3: For $k = 0, \dots, k_0$, $\mu_{\bar{A}}^{(k)}$ is monotonically nonincreasing in k , and $\mu_{\bar{A}}^{(k_0)} = \mathbf{0}$ (which is equal to $\mu_{\bar{A}}^*$).

By claim 2, $\mathbf{g}_{\bar{A}}(\mathbf{x}^{(k)}) \leq -\delta\mathbf{e}$ for all $k = 0, \dots, k_0$. Hence, for $k < k_0$,

$$\begin{aligned}\mu_{\bar{A}}^{(k+1)} &= [\mu_{\bar{A}}^{(k)} + \alpha\mathbf{g}_{\bar{A}}(\mathbf{x}^{(k)})]_+ \\ &\leq [\mu_{\bar{A}}^{(k)} - \alpha\delta\mathbf{e}]_+ \\ &\leq \mu_{\bar{A}}^{(k)},\end{aligned}$$

which establishes nonincreasing monotonicity.

To show that $\mu_{\bar{A}}^{(k_0)} = \mathbf{0}$, suppose that for some nonactive component l , we have $\mu_l^{(k_0)} > 0$. By the monotonicity above, $\mu_l^{(k)} > 0$ for $k = 0, \dots, k_0$. Hence,

$$\begin{aligned}\mu_l^{(k_0)} &= \mu_l^{(k_0-1)} + \alpha g_l(\mathbf{x}^{(k_0-1)}) \\ &= \mu_l^{(0)} + \sum_{k=0}^{k_0-1} \alpha g_l(\mathbf{x}^{(k)}).\end{aligned}$$

But by claim 2, $g_l(\mathbf{x}^{(k)}) \leq -\delta$ for all $k = 0, \dots, k_0 - 1$. Hence, $\mu_l^{(k_0)} \leq \varepsilon - k_0\alpha\delta \leq 0$, which is a contradiction.

Finally, we will state and prove claim 4, which completes the proof of the theorem.

Claim 4: For $k \geq k_0$, we have $\mu_{\bar{A}}^{(k)} = \mathbf{0} = \mu_{\bar{A}}^*$, $\|\mathbf{w}_A^{(k+1)} - \mathbf{w}_A^*\| \leq \kappa_A \|\mathbf{w}_A^{(k)} - \mathbf{w}_A^*\|$, and $\|\mathbf{w}^{(k)} - \mathbf{w}^*\| \leq \eta$.

We use induction. For $k = k_0$, we have $\|\mathbf{w}(k_0) - \mathbf{w}^*\| \leq \eta$ by claim 2, $\mu_{\bar{A}}^{(k_0)} = \mathbf{0}$ by claim 3. Hence,

$$\mathbf{w}_A^{(k_0+1)} = \mathbf{\Pi}[\mathbf{U}_A(\mathbf{w}_A^{(k_0)}) + \alpha D\mathbf{g}_{\bar{A}}(\mathbf{x}^{(k_0)})^\top \mu_{\bar{A}}^{(k_0)}] = \mathbf{\Pi}[\mathbf{U}_A(\mathbf{w}_A^{(k_0)})].$$

Because $\mu_A^* = \mathbf{0}$, it is, similarly, also true that $w_A^* = \Pi[U_A(w_A^*)]$. Thus,

$$\begin{aligned}\|w_A^{(k_0+1)} - w_A^*\| &= \|\Pi[U_A(w_A^{(k_0)})] - \Pi[U_A(w_A^*)]\| \\ &\leq \|U_A(w_A^{(k_0)}) - U_A(w_A^*)\| \\ &\leq \|G_A(w_A^{(k_0)})\| \|w_A^{(k_0)} - w_A^*\|,\end{aligned}$$

where $\|G_A(w_A^{(k_0)})\| \leq \kappa_A$ because $\|w^{(k_0)} - w^*\| \leq \eta$. Hence, $\|w_A^{(k_0+1)} - w_A^*\| \leq \kappa_A \|w_A^{(k_0)} - w_A^*\|$, as required.

For the inductive step, suppose that the result holds for $k \geq k_0$. Now, $g_{\bar{A}}(\mathbf{x}^{(k)}) \leq -\delta e$ and

$$\mu_{\bar{A}}^{(k+1)} = [\mu_{\bar{A}}^{(k)} + \alpha g_{\bar{A}}(\mathbf{x}^{(k)})]_+ \leq [\mathbf{0} - \alpha \delta e]_+ = \mathbf{0},$$

which implies that $\mu_{\bar{A}}^{(k+1)} = \mathbf{0}$. It follows that

$$\begin{aligned}w_A^{(k+2)} &= \Pi[U_A(w_A^{(k+1)}) + \alpha D g_{\bar{A}}(\mathbf{x}^{(k+1)})^\top \mu_{\bar{A}}^{(k+1)}] \\ &= \Pi[U_A(w_A^{(k+1)})],\end{aligned}$$

and now using the same argument as in the case of $k = k_0$ above we get $\|w_A^{(k+2)} - w_A^*\| \leq \kappa_A \|w_A^{(k+1)} - w_A^*\|$. Finally,

$$\|w^{(k+1)} - w^*\| = \|w_A^{(k+1)} - w_A^*\| \leq \kappa_A \|w_A^{(k)} - w_A^*\| \leq \eta.$$

Because $\kappa_A < 1$, claim 4 implies that $w^{(k)}$ converges to w^* , with at least a linear order of convergence. ■

An application of Lagrangian algorithms to a problem in decentralized rate control for sensor networks appears in [24], [25], and [93]. The proof above is based on [25].

23.5 Penalty Methods

Consider a general constrained optimization problem

$$\begin{aligned}&\text{minimize } f(\mathbf{x}) \\ &\text{subject to } \mathbf{x} \in \Omega\end{aligned}$$

We now discuss a method for solving this problem using techniques from unconstrained optimization. Specifically, we approximate the constrained optimization problem above by the unconstrained optimization problem

$$\text{minimize } f(\mathbf{x}) + \gamma P(\mathbf{x}),$$

where $\gamma \in \mathbb{R}$ is a positive constant and $P : \mathbb{R}^n \rightarrow \mathbb{R}$ is a given function. We then solve the associated unconstrained optimization problem and use the solution as an approximation to the minimizer of the original problem. The constant γ is called the *penalty parameter*, and the function P is called the *penalty function*. Formally, we define a penalty function as follows.

Definition 23.1 A function $P : \mathbb{R}^n \rightarrow \mathbb{R}$ is called a *penalty function* for the constrained optimization problem above if it satisfies the following three conditions:

1. P is continuous.
2. $P(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$.
3. $P(\mathbf{x}) = 0$ if and only if \mathbf{x} is feasible (i.e., $\mathbf{x} \in \Omega$). ■

Clearly, for the unconstrained problem above to be a good approximation to the original problem, the penalty function P must be chosen appropriately. The role of the penalty function is to “penalize” points that are outside the feasible set.

To illustrate how we choose penalty functions, consider a constrained optimization problem of the form

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, p, \end{aligned}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, p$. Considering only inequality constraints is not restrictive, because an equality constraint of the form $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ is equivalent to the inequality constraint $\|\mathbf{h}(\mathbf{x})\|^2 \leq 0$ (however, see Exercise 21.25 for a caveat). For the constrained problem above, it is natural that the penalty function be defined in terms of the constraint functions g_1, \dots, g_p . A possible choice for P is

$$P(\mathbf{x}) = \sum_{i=1}^p g_i^+(\mathbf{x}),$$

where

$$g_i^+(\mathbf{x}) = \max\{0, g_i(\mathbf{x})\} = \begin{cases} 0 & \text{if } g_i(\mathbf{x}) \leq 0 \\ g_i(\mathbf{x}) & \text{if } g_i(\mathbf{x}) > 0. \end{cases}$$

We refer to this penalty function as the *absolute value penalty function*, because it is equal to $\sum |g_i(\mathbf{x})|$, where the summation is taken over all constraints that are violated at \mathbf{x} . We illustrate this penalty function in the following example.

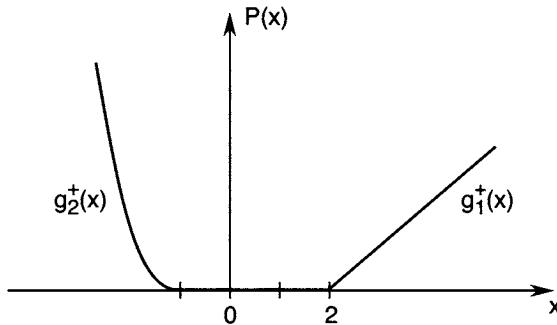


Figure 23.1 g^+ for Example 23.2.

Example 23.2 Let $g_1, g_2 : \mathbb{R} \rightarrow \mathbb{R}$ be defined by $g_1(x) = x - 2$, $g_2(x) = -(x + 1)^3$. The feasible set defined by $\{x \in \mathbb{R} : g_1(x) \leq 0, g_2(x) \leq 0\}$ is simply the interval $[-1, 2]$. In this example, we have

$$g_1^+(x) = \max\{0, g_1(x)\} = \begin{cases} 0 & \text{if } x \leq 2 \\ x - 2 & \text{otherwise,} \end{cases}$$

$$g_2^+(x) = \max\{0, g_2(x)\} = \begin{cases} 0 & \text{if } x \geq -1 \\ -(x + 1)^3 & \text{otherwise,} \end{cases}$$

and

$$P(x) = g_1^+(x) + g_2^+(x) = \begin{cases} x - 2 & \text{if } x > 2 \\ 0 & \text{if } -1 \leq x \leq 2 \\ -(x + 1)^3 & \text{if } x < -1. \end{cases}$$

Figure 23.1 provides a graphical illustration of g^+ for this example. ■

The absolute value penalty function may not be differentiable at points \mathbf{x} where $g_i(\mathbf{x}) = 0$, as is the case at the point $x = 2$ in Example 23.2 (notice, though, that in Example 23.2, P is differentiable at $x = -1$). Therefore, in such cases we cannot use techniques for optimization that involve derivatives. A form of the penalty function that is guaranteed to be differentiable is the *Courant-Beltrami penalty function*, given by

$$P(\mathbf{x}) = \sum_{i=1}^p (g_i^+(\mathbf{x}))^2.$$

In the following discussion we do not assume any particular form of the penalty function P . We only assume that P satisfies conditions 1 to 3 given in Definition 23.1.

The penalty function method for solving constrained optimization problems involves constructing and solving an associated unconstrained optimization

problem and using the solution to the unconstrained problem as the solution to the original constrained problem. Of course, the solution to the unconstrained problem (the approximated solution) may not be exactly equal to the solution to the constrained problem (the true solution). Whether or not the solution to the unconstrained problem is a good approximation to the true solution depends on the penalty parameter γ and the penalty function P . We would expect that the larger the value of the penalty parameter γ , the closer the approximated solution will be to the true solution, because points that violate the constraints are penalized more heavily. Ideally, in the limit as $\gamma \rightarrow \infty$, the penalty method should yield the true solution to the constrained problem. In the remainder of this section, we analyze this property of the penalty function method.

Example 23.3 Consider the problem

$$\begin{aligned} & \text{minimize } \mathbf{x}^\top \mathbf{Q} \mathbf{x} \\ & \text{subject to } \|\mathbf{x}\|^2 = 1, \end{aligned}$$

where $\mathbf{Q} = \mathbf{Q}^\top > 0$.

- a. Using the penalty function $P(\mathbf{x}) = (\|\mathbf{x}\|^2 - 1)^2$ and penalty parameter γ , write down an unconstrained optimization problem whose solution \mathbf{x}_γ approximates the solution to this problem.
- b. Show that for any γ , \mathbf{x}_γ is an eigenvector of \mathbf{Q} .
- c. Show that $\|\mathbf{x}_\gamma\|^2 - 1 = O(1/\gamma)$ as $\gamma \rightarrow \infty$.

Solution:

- a. The unconstrained problem based on the given penalty function is

$$\text{minimize } \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \gamma(\|\mathbf{x}\|^2 - 1)^2.$$

- b. By the FONC, \mathbf{x}_γ satisfies

$$2\mathbf{Q}\mathbf{x}_\gamma + 4\gamma(\|\mathbf{x}_\gamma\|^2 - 1)\mathbf{x}_\gamma = 0.$$

Rearranging, we obtain

$$\mathbf{Q}\mathbf{x}_\gamma = 2\gamma(1 - \|\mathbf{x}_\gamma\|^2)\mathbf{x}_\gamma = \lambda_\gamma \mathbf{x}_\gamma,$$

where λ_γ is a scalar. Hence, \mathbf{x}_γ is an eigenvector of \mathbf{Q} . (This agrees with Example 20.8.)

- c. Now, $\lambda_\gamma = 2\gamma(1 - \|\mathbf{x}_\gamma\|^2) \leq \lambda_{\max}$, where λ_{\max} is the largest eigenvalue of \mathbf{Q} . Hence, $\|\mathbf{x}_\gamma\|^2 - 1 = -\lambda_{\max}/(2\gamma) = O(1/\gamma)$ as $\gamma \rightarrow \infty$. ■

We now analyze the penalty method in a more general setting. In our analysis, we adopt the following notation. Denote by \mathbf{x}^* a solution (global minimizer) to the problem. Let P be a penalty function for the problem. For each $k = 1, 2, \dots$, let $\gamma_k \in \mathbb{R}$ be a given positive constant. Define an associated function $q(\gamma_k, \cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ by

$$q(\gamma_k, \mathbf{x}) = f(\mathbf{x}) + \gamma_k P(\mathbf{x}).$$

For each k , we can write the following associated unconstrained optimization problem:

$$\text{minimize } q(\gamma_k, \mathbf{x}).$$

Denote by $\mathbf{x}^{(k)}$ a minimizer of $q(\gamma_k, \mathbf{x})$. The following technical lemma describes certain useful relationships between the constrained problem and the associated unconstrained problems.

Lemma 23.4 *Suppose that $\{\gamma_k\}$ is a nondecreasing sequence; that is, for each k , we have $\gamma_k \leq \gamma_{k+1}$. Then, for each k we have*

1. $q(\gamma_{k+1}, \mathbf{x}^{(k+1)}) \geq q(\gamma_k, \mathbf{x}^{(k)})$.
2. $P(\mathbf{x}^{(k+1)}) \leq P(\mathbf{x}^{(k)})$.
3. $f(\mathbf{x}^{(k+1)}) \geq f(\mathbf{x}^{(k)})$.
4. $f(\mathbf{x}^*) \geq q(\gamma_k, \mathbf{x}^{(k)}) \geq f(\mathbf{x}^{(k)})$.

□

Proof. We first prove part 1. From the definition of q and the fact that $\{\gamma_k\}$ is an increasing sequence, we have

$$q(\gamma_{k+1}, \mathbf{x}^{(k+1)}) = f(\mathbf{x}^{(k+1)}) + \gamma_{k+1} P(\mathbf{x}^{(k+1)}) \geq f(\mathbf{x}^{(k+1)}) + \gamma_k P(\mathbf{x}^{(k+1)}).$$

Now, because $\mathbf{x}^{(k)}$ is a minimizer of $q(\gamma_k, \mathbf{x})$,

$$q(\gamma_k, \mathbf{x}^{(k)}) = f(\mathbf{x}^{(k)}) + \gamma_k P(\mathbf{x}^{(k)}) \leq f(\mathbf{x}^{(k+1)}) + \gamma_k P(\mathbf{x}^{(k+1)}).$$

Combining the above, we get part 1.

We next prove part 2. Because $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k+1)}$ minimize $q(\gamma_k, \mathbf{x})$ and $q(\gamma_{k+1}, \mathbf{x})$, respectively, we can write

$$q(\gamma_k, \mathbf{x}^{(k)}) = f(\mathbf{x}^{(k)}) + \gamma_k P(\mathbf{x}^{(k)}) \leq f(\mathbf{x}^{(k+1)}) + \gamma_k P(\mathbf{x}^{(k+1)}),$$

$$q(\gamma_{k+1}, \mathbf{x}^{(k+1)}) = f(\mathbf{x}^{(k+1)}) + \gamma_{k+1} P(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)}) + \gamma_{k+1} P(\mathbf{x}^{(k)}).$$

Adding the inequalities above yields

$$\gamma_k P(\mathbf{x}^{(k)}) + \gamma_{k+1} P(\mathbf{x}^{(k+1)}) \leq \gamma_{k+1} P(\mathbf{x}^{(k)}) + \gamma_k P(\mathbf{x}^{(k+1)}).$$

Rearranging, we get

$$(\gamma_{k+1} - \gamma_k) P(\mathbf{x}^{(k+1)}) \leq (\gamma_{k+1} - \gamma_k) P(\mathbf{x}^{(k)}).$$

We know by assumption that $\gamma_{k+1} \geq \gamma_k$. If $\gamma_{k+1} > \gamma_k$, then we get $P(\mathbf{x}^{(k+1)}) \leq P(\mathbf{x}^{(k)})$. If, on the other hand, $\gamma_{k+1} = \gamma_k$, then clearly $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$ and so $P(\mathbf{x}^{(k+1)}) = P(\mathbf{x}^{(k)})$. Therefore, in either case, we arrive at part 2.

We now prove part 3. Because $\mathbf{x}^{(k)}$ is a minimizer of $q(\gamma_k, \mathbf{x})$, we obtain

$$q(\gamma_k, \mathbf{x}^{(k)}) = f(\mathbf{x}^{(k)}) + \gamma_k P(\mathbf{x}^{(k)}) \leq f(\mathbf{x}^{(k+1)}) + \gamma_k P(\mathbf{x}^{(k+1)}).$$

Therefore,

$$f(\mathbf{x}^{(k+1)}) \geq f(\mathbf{x}^{(k)}) + \gamma_k (P(\mathbf{x}^{(k)}) - P(\mathbf{x}^{(k+1)})).$$

From part 2 we have $P(\mathbf{x}^{(k)}) - P(\mathbf{x}^{(k+1)}) \geq 0$, and $\gamma_k > 0$ by assumption; therefore, we get

$$f(\mathbf{x}^{(k+1)}) \geq f(\mathbf{x}^{(k)}).$$

Finally, we now prove part 4. Because $\mathbf{x}^{(k)}$ is a minimizer of $q(\gamma_k, \mathbf{x})$, we get

$$f(\mathbf{x}^*) + \gamma_k P(\mathbf{x}^*) \geq q(\gamma_k, \mathbf{x}^{(k)}) = f(\mathbf{x}^{(k)}) + \gamma_k P(\mathbf{x}^{(k)}).$$

Because \mathbf{x}^* is a minimizer for the constrained optimization problem, we have $P(\mathbf{x}^*) = 0$. Therefore,

$$f(\mathbf{x}^*) \geq f(\mathbf{x}^{(k)}) + \gamma_k P(\mathbf{x}^{(k)}).$$

Because $P(\mathbf{x}^{(k)}) \geq 0$ and $\gamma_k \geq 0$,

$$f(\mathbf{x}^*) \geq q(\gamma_k, \mathbf{x}^{(k)}) \geq f(\mathbf{x}^{(k)}),$$

which completes the proof. ■

With the above lemma, we are now ready to prove the following theorem.

Theorem 23.4 Suppose that the objective function f is continuous and $\gamma_k \rightarrow \infty$ as $k \rightarrow \infty$. Then, the limit of any convergent subsequence of the sequence $\{\mathbf{x}^{(k)}\}$ is a solution to the constrained optimization problem. □

Proof. Suppose that $\{\mathbf{x}^{(m_k)}\}$ is a convergent subsequence of the sequence $\{\mathbf{x}^{(k)}\}$. (See Section 5.1 for a discussion of sequences and subsequences.) Let $\hat{\mathbf{x}}$ be the limit of $\{\mathbf{x}^{(m_k)}\}$. By Lemma 23.4, the sequence $\{q(\gamma_k, \mathbf{x}^{(k)})\}$ is non-decreasing and bounded above by $f(\mathbf{x}^*)$. Therefore, the sequence $\{q(\gamma_k, \mathbf{x}^{(k)})\}$ has a limit $q^* = \lim_{k \rightarrow \infty} q(\gamma_k, \mathbf{x}^{(k)})$ such that $q^* \leq f(\mathbf{x}^*)$ (see Theorem 5.3). Because the function f is continuous and $f(\mathbf{x}^{(m_k)}) \leq f(\mathbf{x}^*)$ by Lemma 23.4, we have

$$\lim_{k \rightarrow \infty} f(\mathbf{x}^{(m_k)}) = f\left(\lim_{k \rightarrow \infty} \mathbf{x}^{(m_k)}\right) = f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*).$$

Because the sequences $\{f(\mathbf{x}^{(m_k)})\}$ and $\{q(\gamma_{m_k}, \mathbf{x}^{(m_k)})\}$ both converge, the sequence $\{\gamma_{m_k} P(\mathbf{x}^{(m_k)})\} = \{q(\gamma_{m_k}, \mathbf{x}^{(m_k)}) - f(\mathbf{x}^{(m_k)})\}$ also converges, with

$$\lim_{k \rightarrow \infty} \gamma_{m_k} P(\mathbf{x}^{(m_k)}) = q^* - f(\hat{\mathbf{x}}).$$

By Lemma 23.4, the sequence $\{P(\mathbf{x}^{(k)})\}$ is nonincreasing and bounded from below by 0. Therefore, $\{P(\mathbf{x}^{(k)})\}$ converges (again see Theorem 5.3), and hence so does $\{P(\mathbf{x}^{(m_k)})\}$. Because $\gamma_{m_k} \rightarrow \infty$ we conclude that

$$\lim_{k \rightarrow \infty} P(\mathbf{x}^{(m_k)}) = 0.$$

By continuity of P , we have

$$0 = \lim_{k \rightarrow \infty} P(\mathbf{x}^{(m_k)}) = P\left(\lim_{k \rightarrow \infty} \mathbf{x}^{(m_k)}\right) = P(\hat{\mathbf{x}}),$$

and hence $\hat{\mathbf{x}}$ is a feasible point. Because $f(\mathbf{x}^*) \geq f(\hat{\mathbf{x}})$ from above, we conclude that $\hat{\mathbf{x}}$ must be a solution to the constrained optimization problem. ■

If we perform an infinite number of minimization runs, with the penalty parameter $\gamma_k \rightarrow \infty$, then Theorem 23.4 ensures that the limit of any convergent subsequence is a minimizer \mathbf{x}^* to the original constrained optimization problem. There is clearly a practical limitation in applying this theorem. It is certainly desirable to find a minimizer to the original constrained optimization problem using a *single* minimization run for the unconstrained problem that approximates the original problem using a penalty function. In other words, we desire an exact solution to the original constrained problem by solving the associated unconstrained problem [minimize $f(\mathbf{x}) + \gamma P(\mathbf{x})$] with a finite $\gamma > 0$. It turns out that indeed this can be accomplished, in which case we say that the penalty function is *exact*. However, it is necessary that exact penalty functions be nondifferentiable, as shown in [10], and illustrated in the following example.

Example 23.4 Consider the problem

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in [0, 1], \end{aligned}$$

where $f(x) = 5 - 3x$. Clearly, the solution is $x^* = 1$.

Suppose that we use the penalty method to solve the problem, with a penalty function P that is differentiable at $x^* = 1$. Then, $P'(x^*) = 0$, because $P(x) = 0$ for all $x \in [0, 1]$. Hence, if we let $g = f + \gamma P$, then $g'(x^*) = f'(x^*) + \gamma P'(x^*) \neq 0$ for all finite $\gamma > 0$. Hence, $x^* = 1$ does not satisfy the first-order necessary condition to be a local minimizer of g . Thus, P is not an exact penalty function. ■

Here, we prove a result on the necessity of nondifferentiability of exact penalty functions for a special class of problems.

Proposition 23.4 *Consider the problem*

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && \mathbf{x} \in \Omega, \end{aligned}$$

with $\Omega \subset \mathbb{R}^n$ convex. Suppose that the minimizer \mathbf{x}^* lies on the boundary of Ω and there exists a feasible direction \mathbf{d} at \mathbf{x}^* such that $\mathbf{d}^\top \nabla f(\mathbf{x}^*) > 0$. If P is an exact penalty function, then P is not differentiable at \mathbf{x}^* . \square

Proof. We use contraposition. Suppose that P is differentiable at \mathbf{x}^* . Then, $\mathbf{d}^\top \nabla P(\mathbf{x}^*) = 0$, because $P(\mathbf{x}) = 0$ for all $\mathbf{x} \in \Omega$. Hence, if we let $g = f + \gamma P$, then $\mathbf{d}^\top \nabla g(\mathbf{x}^*) > 0$ for all finite $\gamma > 0$, which implies that $\nabla g(\mathbf{x}^*) \neq \mathbf{0}$. Hence, \mathbf{x}^* is not a local minimizer of g , and thus P is not an exact penalty function. \blacksquare

Note that the result of Proposition 23.4 does not hold if we remove the assumption that $\mathbf{d}^\top \nabla f(\mathbf{x}^*) > 0$. Indeed, consider a convex problem where $\nabla f(\mathbf{x}^*) = \mathbf{0}$. Choose P to be differentiable. Clearly, in this case we have $\nabla g(\mathbf{x}^*) = \nabla f(\mathbf{x}^*) + \gamma \nabla P(\mathbf{x}^*) = \mathbf{0}$. The function P is therefore an exact penalty function, although differentiable.

For further reading on the subject of optimization of nondifferentiable functions, see, for example, [38]. References [11] and [96] provide further discussions on the penalty method, including nondifferentiable exact penalty functions. These references also discuss exact penalty methods involving differentiable functions; these methods go beyond the elementary type of penalty method introduced in this chapter.

EXERCISES

23.1 Consider the constrained optimization problem

$$\begin{aligned} &\text{maximize} && f(\mathbf{x}) \\ &\text{subject to} && \|\mathbf{x}\| = 1, \end{aligned}$$

where $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x}$ and $\mathbf{Q} = \mathbf{Q}^\top$. We wish to apply a fixed-step-size projected gradient algorithm to this problem:

$$\mathbf{x}^{(k+1)} = \mathbf{\Pi}[\mathbf{x}^{(k)} + \alpha \nabla f(\mathbf{x}^{(k)})],$$

where $\alpha > 0$ and $\mathbf{\Pi}$ is the usual projection operator defined by $\mathbf{\Pi}[\mathbf{x}] = \arg \min_{\mathbf{z} \in \Omega} \|\mathbf{z} - \mathbf{x}\|$ and Ω is the constraint set.

- a. Find a simple formula for $\mathbf{\Pi}[\mathbf{x}]$ in this problem (an explicit expression in terms of \mathbf{x}), assuming that $\mathbf{x} \neq \mathbf{0}$.
- b. For the remainder of the question, suppose that

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}.$$

Find the solution(s) to this optimization problem.

- c. Let $y^{(k)} = x_1^{(k)}/x_2^{(k)}$. Derive an expression for $y^{(k+1)}$ in terms of $y^{(k)}$ and α .
- d. Assuming that $x_2^{(0)} \neq 0$, use parts b and c to show that for any $\alpha > 0$, $\mathbf{x}^{(k)}$ converges to a solution to the optimization problem (i.e., the algorithm works).
- e. In part d, what if $x_2^{(0)} = 0$?

23.2 Consider the problem

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && \mathbf{x} \in \Omega, \end{aligned}$$

where $f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}$ and $\mathbf{c} \in \mathbb{R}^n$ is a given nonzero vector. (Linear programming is a special case of this problem.) We wish to apply a fixed-step-size projected gradient algorithm

$$\mathbf{x}^{(k+1)} = \Pi[\mathbf{x}^{(k)} - \nabla f(\mathbf{x}^{(k)})],$$

where, as usual, Π is the projection operator onto Ω (assume that for any \mathbf{y} , $\Pi[\mathbf{y}] = \arg \min_{\mathbf{x} \in \Omega} \|\mathbf{y} - \mathbf{x}\|^2$ is unique).

- a. Suppose that for some k , $\mathbf{x}^{(k)}$ is a global minimizer of the problem. Is it necessarily the case that $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$? Explain fully.
- b. Suppose that for some k , $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$. Is it necessarily the case that $\mathbf{x}^{(k)}$ is a local minimizer of the problem? Explain fully.

23.3 Consider the optimization problem

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && \mathbf{x} \in \Omega, \end{aligned}$$

where $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, $f \in C^1$, and $\Omega = [-1, 1]^2 = \{\mathbf{x} : -1 \leq x_i \leq 1, i = 1, 2\}$. Consider the projected steepest descent algorithm applied to this problem:

$$\mathbf{x}^{(k+1)} = \Pi[\mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})],$$

where Π represents the projection operator with respect to Ω and $\alpha_k = \arg \min_{\alpha \geq 0} f(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)}))$. Our goal is to prove the following statement:

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$ if and only if $\mathbf{x}^{(k)}$ satisfies the first-order necessary condition.

We will do this in two parts.

- a. Prove the statement above for the case where $\mathbf{x}^{(k)}$ is an interior point of Ω .

- b. Prove the statement for the case where $\mathbf{x}^{(k)}$ is a boundary point of Ω .

Hint: Consider two further subcases: (i) $\mathbf{x}^{(k)}$ is a corner point, and (ii) $\mathbf{x}^{(k)}$ is not a corner point. For subcase (i) it suffices to take $\mathbf{x}^{(k)} = [1, 1]^\top$. For subcase (ii) it suffices to take $\mathbf{x}^{(k)} \in \{\mathbf{x} : x_1 = 1, -1 < x_2 < 1\}$.

23.4 Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m < n$, $\text{rank } \mathbf{A} = m$, and $\mathbf{b} \in \mathbb{R}^m$. Define $\Omega = \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}\}$ and let $\mathbf{x}_0 \in \Omega$. Show that for any $\mathbf{y} \in \mathbb{R}^n$,

$$\mathbf{\Pi}[\mathbf{x}_0 + \mathbf{y}] = \mathbf{x}_0 + \mathbf{P}\mathbf{y},$$

where $\mathbf{P} = \mathbf{I} - \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}\mathbf{A}$.

Hint: Use Exercise 6.7 and Example 12.5.

23.5 Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be given by $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} - \mathbf{x}^\top \mathbf{c}$, where $\mathbf{Q} = \mathbf{Q}^\top > 0$. We wish to minimize f over $\{\mathbf{x} : \mathbf{Ax} = \mathbf{b}\}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m < n$, and $\text{rank } \mathbf{A} = m$. Show that the projected steepest descent algorithm for this case takes the form

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \frac{\mathbf{g}^{(k)\top} \mathbf{P} \mathbf{g}^{(k)}}{\mathbf{g}^{(k)\top} \mathbf{P} \mathbf{Q} \mathbf{P} \mathbf{g}^{(k)}} \mathbf{P} \mathbf{g}^{(k)},$$

where

$$\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)}) = \mathbf{Q}\mathbf{x}^{(k)} - \mathbf{c},$$

and $\mathbf{P} = \mathbf{I}_n - \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}\mathbf{A}$.

23.6 Consider the problem

$$\begin{aligned} &\text{minimize} && \frac{1}{2} \|\mathbf{x}\|^2 \\ &\text{subject to} && \mathbf{Ax} = \mathbf{b}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m < n$, and $\text{rank } \mathbf{A} = m$. Show that if $\mathbf{x}^{(0)} \in \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}\}$, then the projected steepest descent algorithm converges to the solution in one step.

23.7 Show that in the projected steepest descent algorithm, we have that for each k :

- a. $\mathbf{g}^{(k+1)\top} \mathbf{P} \mathbf{g}^{(k)} = 0$.

- b. The vector $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ is orthogonal to the vector $\mathbf{x}^{(k+2)} - \mathbf{x}^{(k+1)}$.

23.8 Consider the optimization problem

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \Omega, \end{aligned}$$

where $\Omega \subset \mathbb{R}^n$. Suppose that we apply the *penalty method* to this problem, which involves solving an associated unconstrained optimization problem with penalty function P and penalty parameter $\gamma > 0$.

- a. Write down the unconstrained problem associated with penalty function P and penalty parameter γ .
- b. Let \mathbf{x}^* be a global minimizer of the given constrained problem, and let \mathbf{x}^γ be a global minimizer of the associated unconstrained optimization problem (in part a) with penalty parameter γ . Show that if $\mathbf{x}^\gamma \notin \Omega$, then $f(\mathbf{x}^\gamma) < f(\mathbf{x}^*)$.

23.9 Use the penalty method to solve the following problem:

$$\begin{aligned} & \text{minimize} && x_1^2 + 2x_2^2 \\ & \text{subject to} && x_1 + x_2 = 3. \end{aligned}$$

Hint: Use the penalty function $P(x) = (x_1 + x_2 - 3)^2$. The solution you find must be exact, not approximate.

23.10 Consider the simple optimization problem

$$\begin{aligned} & \text{minimize} && x \\ & \text{subject to} && x \geq a, \end{aligned}$$

where $a \in \mathbb{R}$. Suppose that we use the penalty method to solve this problem, with penalty function

$$P(x) = (\max\{a - x, 0\})^2$$

(the *Courant-Beltrami penalty function*). Given a number $\varepsilon > 0$, find the smallest value of the penalty parameter γ such that the solution obtained using the penalty method is no further than ε from the true solution to the given problem. (Think of ε as the desired accuracy.)

23.11 Consider the problem

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|\mathbf{x}\|^2 \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $m \leq n$, and $\text{rank } \mathbf{A} = m$. Let \mathbf{x}^* be the solution. Suppose that we solve the problem using the penalty method, with the penalty function

$$P(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|^2.$$

Let \mathbf{x}_γ^* be the solution to the associated unconstrained problem with the penalty parameter $\gamma > 0$; that is, \mathbf{x}_γ^* is the solution to

$$\text{minimize } \frac{1}{2} \|\mathbf{x}\|^2 + \gamma \|\mathbf{Ax} - \mathbf{b}\|^2.$$

- a. Suppose that

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \mathbf{b} = [1].$$

Verify that \mathbf{x}_γ^* converges to the solution \mathbf{x}^* of the original constrained problem as $\gamma \rightarrow \infty$.

- b. Prove that $\mathbf{x}_\gamma^* \rightarrow \mathbf{x}^*$ as $\gamma \rightarrow \infty$ holds in general.

Hint: Use the following result: There exist orthogonal matrices $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V}^\top \in \mathbb{R}^{n \times n}$ such that

$$\mathbf{A} = \mathbf{U} [\mathbf{S}, \mathbf{O}] \mathbf{V}^\top,$$

where

$$\mathbf{S} = \text{diag} \left(\sqrt{\lambda_1(\mathbf{A}\mathbf{A}^\top)}, \dots, \sqrt{\lambda_m(\mathbf{A}\mathbf{A}^\top)} \right)$$

is a diagonal matrix with diagonal elements that are the square roots of the eigenvalues of $\mathbf{A}\mathbf{A}^\top$.

The result above is called the *singular value decomposition* (see, e.g., [62, p. 411]).

CHAPTER 24

MULTIOBJECTIVE OPTIMIZATION

24.1 Introduction

When an optimization problem involves only one objective function, it is a single-objective optimization. Most engineering problems require the designer to optimize a number of conflicting objectives. The objectives are in conflict with each other if an improvement in one objective leads to deterioration in another. Multiobjective problems in which there is competition between objectives may have no single, unique optimal solution. Multiobjective optimization problems are also referred to as multicriteria or vector optimization problems. We can formulate a multiobjective optimization problem as follows: Find a decision variable that satisfies the given constraints and optimizes a vector function whose components are objective functions. Formally, the mul-

tiobjective optimization problem is stated as follows:

$$\begin{aligned} \text{minimize } \mathbf{f}(\mathbf{x}) = & \begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_\ell(x_1, x_2, \dots, x_n) \end{bmatrix} \\ \text{subject to } \mathbf{x} \in \Omega, \end{aligned}$$

where $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^\ell$ and $\Omega \subset \mathbb{R}^n$. For example, the constraint set Ω can have the form

$$\Omega = \{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\},$$

where

$$\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad \mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^p, \quad m \leq n.$$

In general, we may have three different types of multiobjective optimization problems:

- Minimize all the objective functions.
- Maximize all the objective functions.
- Minimize some and maximize others.

However, as usual, any of these can be converted into an equivalent minimization problem.

24.2 Pareto Solutions

Note that multiobjective function assigns to each decision variable a multi-objective vector value in the objective function space. A graphical illustration of this statement is illustrated in Figures 24.1 and 24.2. In Figure 24.1 the decision variable is a point $\mathbf{x} \in \mathbb{R}^2$ while the vector of objective functions is given by $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$. In Figure 24.2 the decision variable is a point $\mathbf{x} \in \mathbb{R}^2$ while the vector of objective functions is $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$. In single-objective optimization problems our goal is to find a single solution, where we focus mainly on the decision variable space. On the other hand, in multiobjective problems we are usually more interested in the objective space. As pointed out by Miettinen [92, p. 11], multiobjective problems are in a sense ill-defined because there is no natural ordering in the objective space. Miettinen [92] illustrates this statement with the following simple example. One can say that $[1, 1]^\top$ is less than $[3, 3]^\top$. But how do we compare $[1, 3]^\top$ and $[3, 1]^\top$? In general, in multiobjective optimization problems our goal is to find good compromises. Roughly speaking, in a multiobjective optimization problem, a solution is optimal if there exists no other solution, within the feasible set,

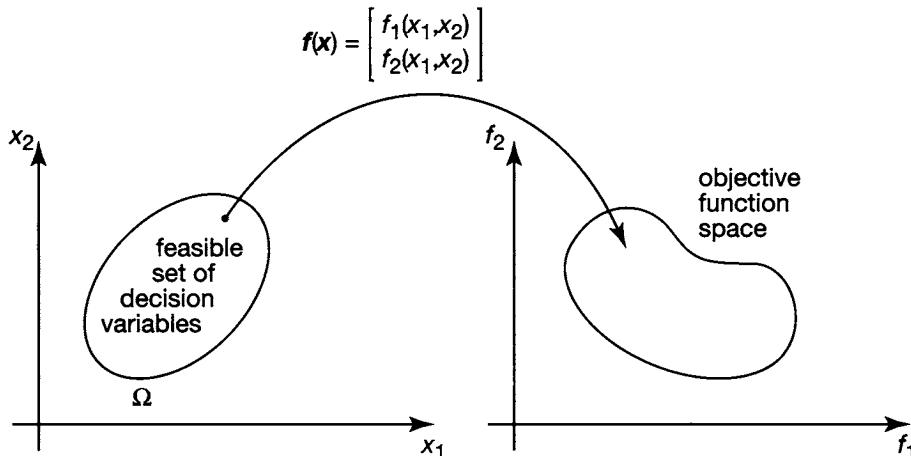


Figure 24.1 Two-dimensional illustration of a multiobjective vector function assigning to each decision variable a multiobjective vector value.

that gives improved performance with regard to all the objectives. A formal definition of an optimal point for a multiobjective optimization problem was proposed by Francis Y. Edgeworth in 1881 and generalized by Vilfredo Pareto in 1896. It is customary now to refer to an optimal point of a multiobjective optimization problem as the Pareto minimizer, whose formal definition is given next.

Definition 24.1 Let $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^\ell$ and $\mathbf{x} \in \Omega$ be given. For the optimization problem

$$\begin{aligned} & \text{minimize } \mathbf{f}(\mathbf{x}) \\ & \text{subject to } \mathbf{x} \in \Omega \end{aligned}$$

a point $\mathbf{x}^* \in \Omega$ is called a *Pareto minimizer* if there exists no $\mathbf{x} \in \Omega$ such that for $i = 1, 2, \dots, \ell$,

$$f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$$

and for at least one i ,

$$f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$$

■

In other words, the point \mathbf{x}^* is a Pareto minimizer, or a nondominated solution, if there exists no other feasible decision variable \mathbf{x} that would decrease some objectives without causing simultaneous increase in at least one other variable.

The set of Pareto minimizers (optimizers) is called the *Pareto front*, as illustrated in Figure 24.3. Most multiobjective optimization algorithms use

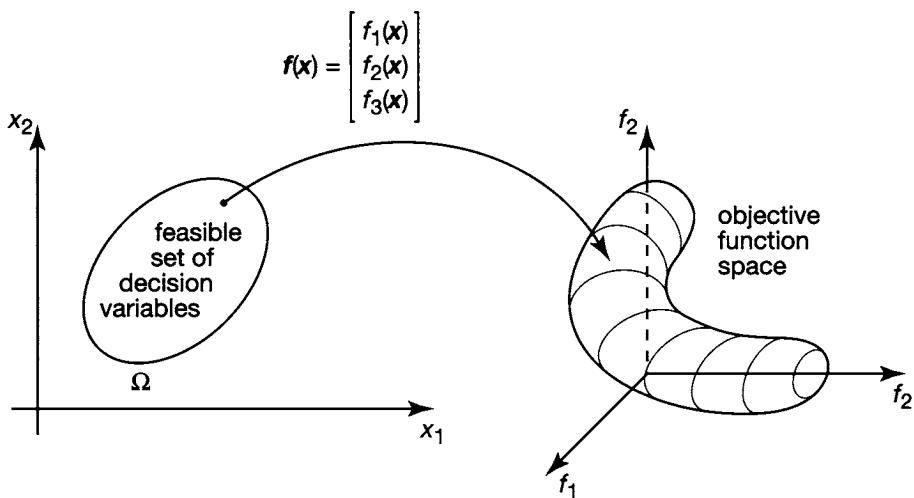


Figure 24.2 Three-dimensional illustration of a multiobjective vector function assigning to each decision variable a multiobjective vector function value.

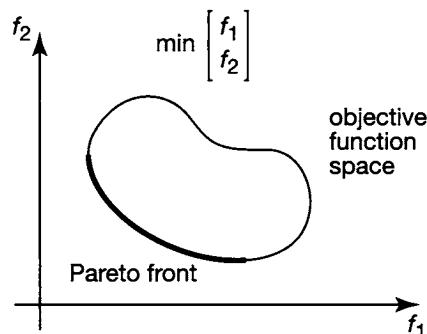


Figure 24.3 The Pareto front is marked with a heavy line.

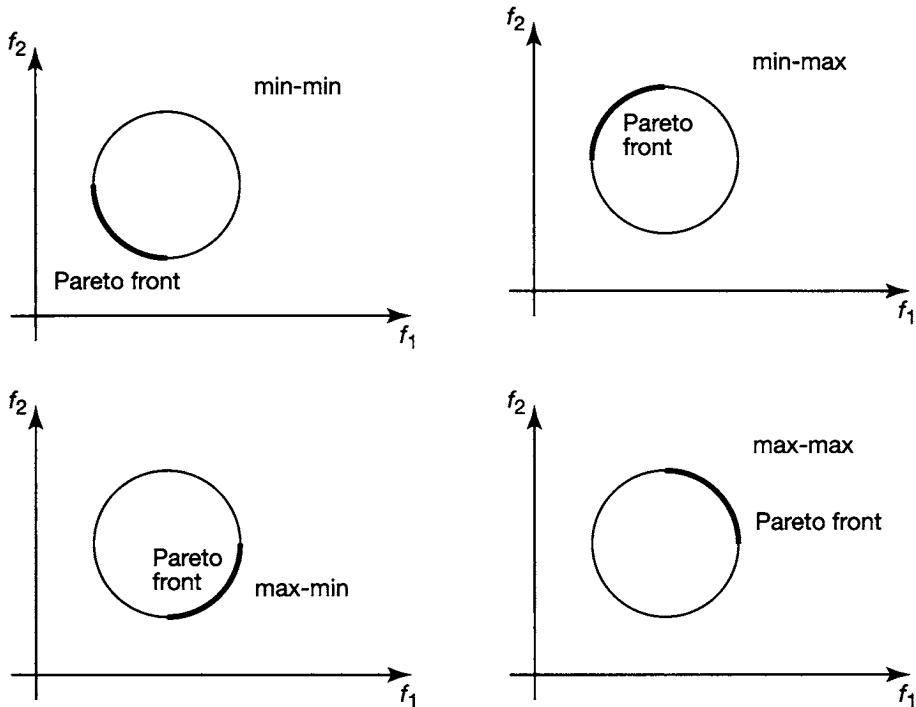


Figure 24.4 Pareto fronts for four possible cases of two-objective optimization.

the concept of domination. A solution is said to be nondominated if it is Pareto optimal.

In Figure 24.4 we show different combinations of two-objective optimization and the corresponding Pareto fronts. In particular, in the upper left, we show the Pareto front for the case when we are minimizing both components of the objective function vector, which we represent by “min-min.” Similarly, “min-max” represents the case when we are minimizing the first objective function and maximizing the second; and so forth.

24.3 Computing the Pareto Front

When computing the Pareto front, two solutions are compared and the dominated solution is eliminated from the set of candidates of Pareto optimizers. Thus, the Pareto front consists of nondominated solutions.

To proceed, we need some notation. Let

$$\mathbf{x}^{*r} = [x_1^{*r}, x_2^{*r}, \dots, x_n^{*r}]^\top$$

be the r th candidate Pareto optimal solution, $r = 1, 2, \dots, R$, where R is the number of current candidate Pareto solutions. Let

$$\mathbf{f}(\mathbf{x}^{*r}) = [f_1(\mathbf{x}^{*r}), f_2(\mathbf{x}^{*r}), \dots, f_\ell(\mathbf{x}^{*r})]^\top$$

be the corresponding value of the objective function vector. For any new solution candidate \mathbf{x}^j , we evaluate the objective function vector $\mathbf{f}(\mathbf{x}^j)$. We then compare the new solution candidate with the existing Pareto solutions. We need to consider three cases:

- \mathbf{x}^j dominates at least one candidate solution.
- \mathbf{x}^j does not dominate any existing candidate solutions.
- \mathbf{x}^j is dominated by a candidate solution.

If \mathbf{x}^j dominates at least one candidate solution, we delete the dominated solutions from the set and add the new solution \mathbf{x}^j to the set of candidates. In the second case, when the new candidate solution \mathbf{x}^j does not dominate any of the existing candidate Pareto solutions, add this new Pareto solution to the set of candidate Pareto solutions. Finally, in the third case, when the new candidate solution is dominated by at least one of the existing candidate Pareto solutions, we do not change the set of the existing candidate Pareto solutions.

Example 24.1 Consider the two-objective minimization problem whose data are as follows:

$\mathbf{x}^{(i)\top}$	$\mathbf{f}(\mathbf{x}^{(i)})^\top$
[5, 6]	[30, 45]
[4, 5]	[22, 29]
[3, 7]	[19, 53]
[6, 8]	[41, 75]
[1, 4]	[13, 45]
[6, 7]	[42, 55]
[2, 5]	[37, 46]
[3, 6]	[28, 37]
[2, 7]	[12, 51]
[4, 7]	[41, 67]

Suppose that we wish to find nondominated pairs for this problem. Recall that a point \mathbf{x}^* is a nondominated point if for all i and all \mathbf{x} ,

$$f_i(\mathbf{x}^*) \leq f_i(\mathbf{x}),$$

and at least for one component j of the objective vector, we have

$$f_j(\mathbf{x}^*) < f_j(\mathbf{x}).$$

To find the Pareto front, we start with the first pair as a candidate Pareto optimal solution and then compare the other pairs against this first pair, replacing the first pair as necessary. We then continue with the other pairs, building up a set of candidate Pareto solutions and modifying this set when appropriate. The result of the search gives the following Pareto optimal set:

$\mathbf{x}^{(i)\top}$	$\mathbf{f}(\mathbf{x}^{(i)})^\top$
[4, 5]	[22, 29]
[1, 4]	[13, 45]
[2, 7]	[12, 51]

■

We now discuss an algorithm for generating the Pareto front that implements the foregoing ideas. This algorithm is a minor modification of the algorithm of Osyczka [98, pp. 100–101]. We use the following notation. Let J be the number of candidate solutions to be checked for optimality, while R is the number of current candidate Pareto solutions. Recall that ℓ is the number of objective functions, the dimension of the objective function vector, and n is the dimension of the decision space, that is, the number of components of \mathbf{x} . The algorithm consists of eight steps.

Algorithm for Generating a Pareto Front

1. Generate an initial solution \mathbf{x}^1 and evaluate $\mathbf{f}^{*1} = \mathbf{f}(\mathbf{x}^1)$. This first solution generated is taken as a candidate Pareto solution. Set initial indices $R := 1$ and $j := 1$.
2. Set $j := j + 1$. If $j \leq J$, then generate solution \mathbf{x}^j and go to step 3. Otherwise, stop, because all the candidate solutions have already been considered.
3. Set $r := 1$ and $q := 0$ (q represents the number of eliminated solutions from the existing set of Pareto solutions).
4. If for all $i = 1, 2, \dots, \ell$,
$$f_i(\mathbf{x}^j) < f_i(\mathbf{x}^{*r}),$$

then set $q := q + 1$, $\mathbf{f}^{*R} := \mathbf{f}(\mathbf{x}^j)$, remember the solution that should be eliminated, and go to step 6.

5. If for all $i = 1, 2, \dots, \ell$,
$$f_i(\mathbf{x}^j) \geq f_i(\mathbf{x}^{*r}),$$

then go to step 2.

6. Set $r := r + 1$. If $r \leq R$, go to step 4.

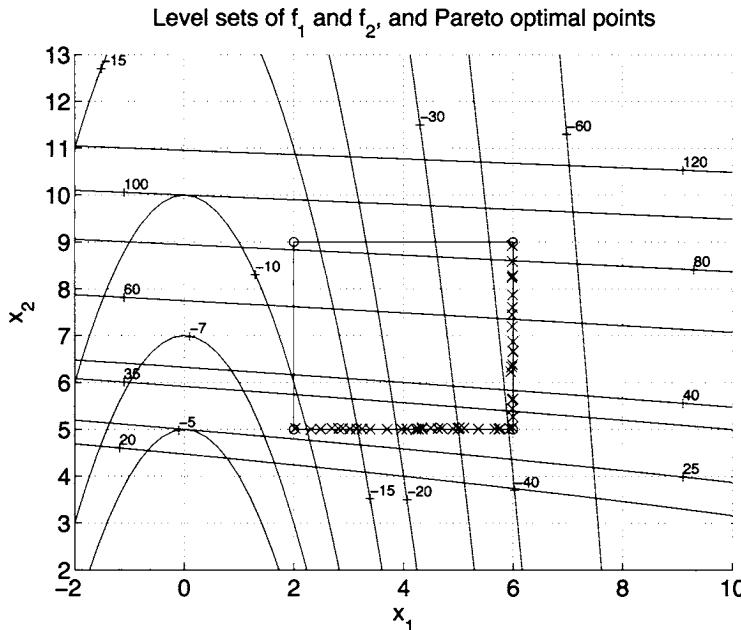


Figure 24.5 Pareto optimal points in the decision space along with the level sets of the objective functions f_1 and f_2 .

7. If $q \neq 0$, remove from the candidate Pareto set the solutions that are eliminated in step 4, add solution \mathbf{x}^j as a new candidate Pareto solution, and go to step 2.
8. Set $R := R + 1$, $\mathbf{x}^{*R} := \mathbf{x}^j$, $\mathbf{f}^{*R} := \mathbf{f}(\mathbf{x}^j)$, and go to step 2.

Example 24.2 We apply the algorithm above to generate the Pareto front for the multiobjective optimization problem

$$\begin{aligned} & \text{minimize} && \begin{bmatrix} -(x_1^2 + x_2) \\ x_1 + x_2^2 \end{bmatrix} \\ & \text{subject to} && 2 \leq x_1 \leq 6 \\ & && 5 \leq x_2 \leq 9. \end{aligned}$$

We performed 100 iterations. At each iteration we randomly generated 50 feasible points. Then we applied the algorithm above to extract from this set of feasible points candidate Pareto optimal solutions. In Figure 24.5 we show Pareto optimal points obtained after 100 iterations of the algorithm. We also show level sets of the objective functions in the (x_1, x_2) -space. In Figure 24.6 we show the Pareto front in the objective function space after 100

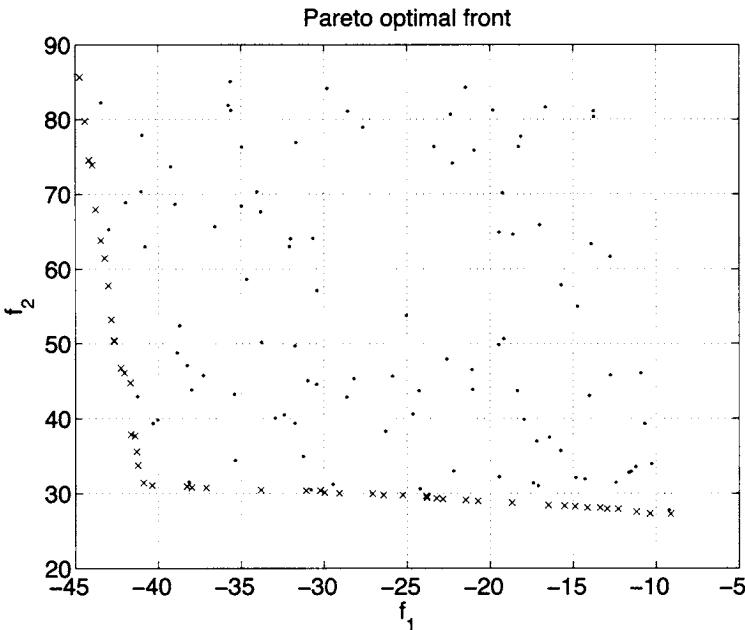


Figure 24.6 Pareto front for the problem of Example 24.2. Also marked are the objective vector values for the remaining candidate points generated in the last iteration.

iterations of the algorithm. The Pareto optimal points are marked with x's. The points marked with '-'s are the candidate solutions generated randomly at the beginning of the last iteration of the algorithm. ■

We have described a simple approach to computing the Pareto front. Alternative methods include those that apply genetic algorithms to solving multiobjective optimization problems, as discussed in Deb [37], Coello Coello et al. [31], and Osyczka [98].

24.4 From Multiobjective to Single-Objective Optimization

In some cases it is possible to deal with a multiobjective optimization problem by converting the problem into a single-objective optimization problem, so that standard optimization methods can be brought to bear. Here, we discuss four techniques to convert a multiobjective problem to a single-objective problem. We assume throughout that an objective function vector $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_\ell(\mathbf{x})]^\top$ is given.

The first method is to form a single objective function by taking a linear combination, with positive coefficients, of the components of the objective

function vector. Equivalently, we form a convex combination of the components of the objective function vector. In other words, we use

$$f(\mathbf{x}) = \mathbf{c}^\top \mathbf{f}(\mathbf{x})$$

as the (single) objective function, where \mathbf{c} is a vector of positive components. This method is also called the *weighted-sum method*, where the coefficients of the linear combination (i.e., the components of \mathbf{c}) are called *weights*. These weights reflect the relative importance of the individual components in the objective vector. Of course, it might be difficult to determine suitable weight values.

A second method is to form a single objective function by taking the maximum of the components of the objective vector:

$$f(\mathbf{x}) = \max\{f_1(\mathbf{x}), \dots, f_\ell(\mathbf{x})\}.$$

In other words, we convert the multiobjective minimization problem into one of minimizing the maximum of the components. For this reason, it is also called the *minimax method*. Note that this method applies to situations where the components of the objective vector are comparable or compatible, in the sense that they are in the same units (e.g., they are all lengths measured in meters, or masses in kilograms). A limitation of this method is that the resulting single objective function might not be differentiable, thereby precluding the use of optimization methods that rely on differentiability (e.g., gradient algorithms). However, as we show in the following, a minimax problem with linear objective vector components and linear constraints can be reduced to a linear programming problem.

Example 24.3 Given vectors $\mathbf{v}_1, \dots, \mathbf{v}_p \in \mathbb{R}^n$ and scalars u_1, \dots, u_p , consider the minimax problem

$$\begin{aligned} & \text{minimize} && \max\{\mathbf{v}_1^\top \mathbf{x} + u_1, \dots, \mathbf{v}_p^\top \mathbf{x} + u_p\} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Call this problem P1.

- a. Consider the optimization problem

$$\begin{aligned} & \text{minimize} && y \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & && y \geq \mathbf{v}_i^\top \mathbf{x} + u_i, \quad i = 1, \dots, p, \end{aligned}$$

where the decision variable is the vector $[\mathbf{x}^\top, y]^\top$. Call this problem P2. Show that \mathbf{x}^* solves P1 if and only if $[\mathbf{x}^*^\top, y^*]^\top$ with $y^* = \max\{\mathbf{v}_1^\top \mathbf{x}^* + u_1, \dots, \mathbf{v}_p^\top \mathbf{x}^* + u_p\}$ solves P2.

Hint: $y \geq \max\{a, b, c\}$ if and only if $y \geq a$, $y \geq b$, and $y \geq c$.

- b. Use part a to derive a linear programming problem

$$\begin{aligned} & \text{minimize } \hat{\mathbf{c}}^\top \mathbf{z} \\ & \text{subject to } \hat{\mathbf{A}}\mathbf{z} \leq \hat{\mathbf{b}} \end{aligned}$$

that is equivalent to P1 (by “equivalent” we mean that the solution to one gives us the solution to the other). Explain how a solution to the linear programming problem above gives a solution to P1.

Solution:

- a. First suppose that \mathbf{x}^* is optimal in P1. Let $y^* = \max\{\mathbf{v}_1^\top \mathbf{x}^* + u_1, \dots, \mathbf{v}_p^\top \mathbf{x}^* + u_p\}$. Then, $[\mathbf{x}^{*\top}, y^*]^\top$ is feasible in P2. Let $[\mathbf{x}^\top, y]^\top$ be any feasible point in P2. Then (by the hint)

$$y \geq \max\{\mathbf{v}_1^\top \mathbf{x} + u_1, \dots, \mathbf{v}_p^\top \mathbf{x} + u_p\}.$$

Moreover, \mathbf{x} is feasible in P1, and hence

$$\begin{aligned} y &\geq \max\{\mathbf{v}_1^\top \mathbf{x} + u_1, \dots, \mathbf{v}_p^\top \mathbf{x} + u_p\} \\ &\geq \max\{\mathbf{v}_1^\top \mathbf{x}^* + u_1, \dots, \mathbf{v}_p^\top \mathbf{x}^* + u_p\} \\ &= y^*. \end{aligned}$$

Hence, $[\mathbf{x}^{*\top}, y^*]^\top$ is optimal in the linear programming problem.

To prove the converse, suppose that \mathbf{x}^* is not optimal in P1. Then, there is some \mathbf{x}' that is feasible in P1 such that

$$\begin{aligned} y' &= \max\{\mathbf{v}_1^\top \mathbf{x}' + u_1, \dots, \mathbf{v}_p^\top \mathbf{x}' + u_p\} \\ &< \max\{\mathbf{v}_1^\top \mathbf{x}^* + u_1, \dots, \mathbf{v}_p^\top \mathbf{x}^* + u_p\} \\ &= y^*. \end{aligned}$$

But $[\mathbf{x}'^\top, y']^\top$ is evidently feasible in P2, and has objective function value (y') that is lower than that of $[\mathbf{x}^{*\top}, y^*]^\top$. Hence, $[\mathbf{x}^{*\top}, y^*]^\top$ is not optimal in P2.

- b. Define

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ y \end{bmatrix}, \quad \hat{\mathbf{c}} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \quad \hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{v}_1^\top & -1 \\ \vdots & \vdots \\ \mathbf{v}_p^\top & -1 \end{bmatrix}, \quad \hat{\mathbf{b}} = \begin{bmatrix} \mathbf{b} \\ -u_1 \\ \vdots \\ -u_p \end{bmatrix}.$$

Then the equivalent problem can be written as

$$\begin{aligned} & \text{minimize } \hat{\mathbf{c}}^\top \mathbf{z} \\ & \text{subject to } \hat{\mathbf{A}}\mathbf{z} \leq \hat{\mathbf{b}}. \end{aligned}$$

By part a, if we obtain a solution to this linear programming problem, then the first n components form a solution to the original minimax problem. ■

A third method to convert a multiobjective problem to a single-objective problem, assuming that the components of the objective vector are nonnegative, is to form a single objective function by taking the p -norm of the objective vector:

$$f(\mathbf{x}) = \|\mathbf{f}(\mathbf{x})\|_p.$$

The minimax method can be viewed as a special case of this method, with $p = \infty$. The weighted-sum method with uniform weights can be viewed as this method with $p = 1$. To make the objective function differentiable in the case where p is finite (so that we can apply gradient methods, for example), we replace it by its p th power:

$$f(\mathbf{x}) = \|\mathbf{f}(\mathbf{x})\|_p^p = (f_1(\mathbf{x}))^p + \cdots + (f_\ell(\mathbf{x}))^p.$$

A fourth method is to minimize one of the components of the objective vector subject to constraints on the other components. For example, given \mathbf{f} , we solve

$$\begin{aligned} & \text{minimize} && f_1(\mathbf{x}) \\ & \text{subject to} && f_2(\mathbf{x}) \leq b_2, \\ & && \vdots \\ & && f_\ell(\mathbf{x}) \leq b_\ell, \end{aligned}$$

where b_2, \dots, b_ℓ are given constants that reflect satisfactory values for the objectives f_2, \dots, f_ℓ , respectively. Of course, this approach is suitable only in situations where these satisfactory values can be determined.

24.5 Uncertain Linear Programming Problems

In this section we show how multiobjective optimization methods can be used to solve linear programming problems with uncertain coefficients, including uncertain constraints and uncertain objective functions.

Uncertain Constraints

We first consider a generalization of linear programming to problems with uncertain constraints. Our exposition is based on a discussion of fuzzy linear programming by Wang [131, Chapter 30]. We consider the following general

linear programming problem:

$$\begin{aligned} & \text{minimize } \mathbf{c}^\top \mathbf{x} \\ & \text{subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \quad \mathbf{x} \geq 0. \end{aligned}$$

We can represent the constraints in the form

$$(\mathbf{A}\mathbf{x})_i \leq b_i, \quad i = 1, 2, \dots, m.$$

Suppose that the constraints' bounds are uncertain in the sense that they can vary within given tolerance values and can be represented as

$$(\mathbf{A}\mathbf{x})_i \leq b_i + \theta t_i, \quad i = 1, 2, \dots, m,$$

where $\theta \in [0, 1]$ and $t_i > 0$, $i = 1, 2, \dots, m$.

We now discuss a method to solve the problem above. First, solve the following two linear programming problems:

$$\begin{aligned} & \text{minimize } \mathbf{c}^\top \mathbf{x} \\ & \text{subject to } (\mathbf{A}\mathbf{x})_i \leq b_i, \quad i = 1, 2, \dots, m \\ & \quad \mathbf{x} \geq 0 \end{aligned}$$

and

$$\begin{aligned} & \text{minimize } \mathbf{c}^\top \mathbf{x} \\ & \text{subject to } (\mathbf{A}\mathbf{x})_i \leq b_i + t_i, \quad i = 1, 2, \dots, m \\ & \quad \mathbf{x} \geq 0. \end{aligned}$$

Denote the solution to the two programs as $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(0)}$, respectively, and let $z_1 = \mathbf{c}^\top \mathbf{x}^{(1)}$ and $z_0 = \mathbf{c}^\top \mathbf{x}^{(0)}$. Using these definitions, we construct a function that characterizes the "degree of the penalty" associated with the uncertain constraints in the linear programming problem

$$\mu_0(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{c}^\top \mathbf{x} < z_0 \\ \frac{\mathbf{c}^\top \mathbf{x} - z_0}{z_1 - z_0} & \text{if } z_0 \leq \mathbf{c}^\top \mathbf{x} \leq z_1 \\ 1 & \text{if } \mathbf{c}^\top \mathbf{x} > z_1. \end{cases}$$

A plot of this function is given in Figure 24.7. Note that when $\mathbf{c}^\top \mathbf{x} \leq z_0$, then $\mu_0(\mathbf{x}) = 0$, which represents minimum degree of penalty. On the other hand, when $\mathbf{c}^\top \mathbf{x} \geq z_1$, then $\mu_0(\mathbf{x}) = 1$, and we have a maximum degree of penalty. When $z_0 \leq \mathbf{c}^\top \mathbf{x} \leq z_1$, the degree of penalty varies from 0 to 1.

Next, we introduce a function that describes the degree of penalty for violating the i th constraint:

$$\mu_i(\mathbf{x}) = \begin{cases} 0 & \text{if } (\mathbf{A}\mathbf{x})_i - b_i < 0 \\ \frac{(\mathbf{A}\mathbf{x})_i - b_i}{t_i} & \text{if } 0 \leq (\mathbf{A}\mathbf{x})_i - b_i \leq t_i \\ 1 & \text{if } (\mathbf{A}\mathbf{x})_i - b_i > t_i. \end{cases}$$

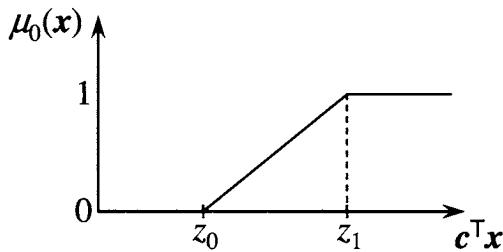


Figure 24.7 Plot of the function $\mu_0(\mathbf{x})$.

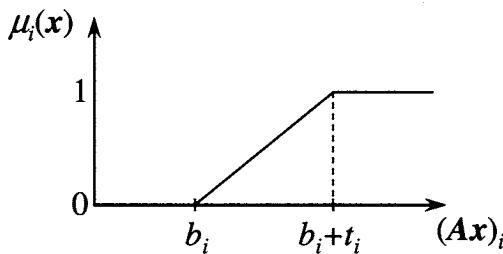


Figure 24.8 Plot of the function $\mu_i(\mathbf{x})$.

A plot of this function is shown in Figure 24.8.

Using the definitions above we can reformulate the original linear programming problem as a multiobjective optimization problem, with the goal of minimizing the functions that penalize constraint violations:

$$\begin{aligned} & \text{minimize} && \begin{bmatrix} \mu_0(\mathbf{x}) \\ \mu_1(\mathbf{x}) \\ \vdots \\ \mu_m(\mathbf{x}) \end{bmatrix} \\ & \text{subject to} && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

We can employ the minimax method to solve the multiobjective optimization problem as a single-objective problem

$$\begin{aligned} & \text{minimize} && \max \{\mu_0(\mathbf{x}), \mu_1(\mathbf{x}), \dots, \mu_m(\mathbf{x})\} \\ & \text{subject to} && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

As shown in Example 24.3, the problem above can be stated equivalently as

$$\begin{aligned} & \text{minimize } \theta \\ & \text{subject to } \mu_0(\mathbf{x}) \leq \theta \\ & \quad \mu_i(\mathbf{x}) \leq \theta, \quad i = 1, 2, \dots, m \\ & \quad \theta \in [0, 1], \quad \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Using now the definitions of μ_0 and μ_i , $i = 1, \dots, m$, we restate the optimization problem above as

$$\begin{aligned} & \text{minimize } \theta \\ & \text{subject to } \mathbf{c}^\top \mathbf{x} \leq z_0 + \theta(z_1 - z_0) \\ & \quad (\mathbf{A}\mathbf{x})_i \leq b_i + \theta t_i, \quad i = 1, 2, \dots, m \\ & \quad \theta \in [0, 1], \quad \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Example 24.4 Consider the following linear programming problem:

$$\begin{aligned} & \text{minimize } -\frac{1}{2}x_1 - x_2 \\ & \text{subject to } x_1 + x_2 \leq 5 \\ & \quad x_2 \leq 3 \\ & \quad x_1 \geq 0, \quad x_2 \geq 0, \end{aligned}$$

where the tolerances are $t_1 = 2$ and $t_2 = 1$.

- Solve the two linear programming problems to obtain $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(0)}$ using the data above. Then find z_1 and z_0 .
- Construct the equivalent optimization problem (involving θ) using the data above.
- Express the optimization problem as a linear programming problem in standard form.

Solution:

- We can solve these problems graphically to obtain

$$\mathbf{x}^{(1)} = [2, 3]^\top \quad \text{and} \quad \mathbf{x}^{(0)} = [3, 4]^\top.$$

Hence,

$$z_1 = \mathbf{c}^\top \mathbf{x}^{(1)} = -4 \quad \text{and} \quad z_0 = \mathbf{c}^\top \mathbf{x}^{(0)} = -5\frac{1}{2}.$$

b. The optimization problem has the form

$$\begin{aligned} & \text{minimize } \theta \\ & \text{subject to } \mu_0(\mathbf{x}) \leq \theta \\ & \quad \mu_1(\mathbf{x}) \leq \theta \\ & \quad \mu_2(\mathbf{x}) \leq \theta \\ & \quad \theta \in [0, 1], \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where

$$\begin{aligned} \mu_0(\mathbf{x}) &= \begin{cases} 0 & \text{if } -\frac{1}{2}x_1 - x_2 < -5\frac{1}{2} \\ \frac{-\frac{1}{2}x_1 - x_2 + 5\frac{1}{2}}{3/2} & \text{if } -5\frac{1}{2} \leq -\frac{1}{2}x_1 - x_2 \leq -4 \\ 1 & \text{if } -\frac{1}{2}x_1 - x_2 > -4, \end{cases} \\ \mu_1(\mathbf{x}) &= \begin{cases} 0 & \text{if } x_1 + x_2 - 5 < 0 \\ \frac{x_1 + x_2 - 5}{2} & \text{if } 0 \leq x_1 + x_2 - 5 \leq 2 \\ 1 & \text{if } x_1 + x_2 - 5 > 2, \end{cases} \\ \mu_2(\mathbf{x}) &= \begin{cases} 0 & \text{if } x_2 - 3 < 0 \\ x_2 - 3 & \text{if } 0 \leq x_2 - 3 \leq 1 \\ 1 & \text{if } x_2 - 3 > 1. \end{cases} \end{aligned}$$

c. We have

$$\begin{aligned} & \text{minimize } \theta \\ & \text{subject to } \mathbf{c}^\top \mathbf{x} \leq z_0 + \theta(z_1 - z_0) \\ & \quad (\mathbf{Ax})_i \leq b_i + (1 - \theta)t_i, \quad i = 1, 2 \\ & \quad \theta \in [0, 1], \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Using our data, we obtain

$$\begin{aligned} & \text{minimize } \theta \\ & \text{subject to } \frac{1}{2}x_1 + x_2 \geq 5\frac{1}{2} - \frac{3}{2}\theta \\ & \quad x_1 + x_2 \leq 5 + 2\theta \\ & \quad x_2 \leq 3 + \theta \\ & \quad \theta \in [0, 1], \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Write $x_3 = \theta$. Then, the above problem can be represented as

$$\begin{aligned} & \text{minimize } x_3 \\ & \text{subject to } x_1 + 2x_2 + 3x_3 \geq 11 \\ & \quad x_1 + x_2 - 2x_3 \leq 5 \\ & \quad x_2 - x_3 \leq 3 \\ & \quad x_3 \leq 1 \\ & \quad x_i \geq 0, \quad i = 1, 2, 3. \end{aligned}$$

The above linear program expressed in the form of a linear programming problem in standard form is

$$\begin{aligned} & \text{minimize } x_3 \\ & \text{subject to } x_1 + 2x_2 + 3x_3 - x_4 = 11 \\ & \quad x_1 + x_2 - 2x_3 + x_5 = 5 \\ & \quad x_2 - x_3 + x_6 = 3 \\ & \quad x_3 + x_7 = 1 \\ & \quad x_i \geq 0, \quad i = 1, 2, \dots, 7. \end{aligned}$$

■

Uncertain Objective Function Coefficients

We now consider a linear programming problem with uncertain objective function coefficients. We assume that uncertainties of the objective coefficients are modeled by the following triangular function:

$$\mu(x; a, b, c) = \begin{cases} 0 & \text{if } x < a \\ (x-a)/(b-a) & \text{if } a \leq x < b \\ (c-x)/(c-b) & \text{if } b \leq x \leq c \\ 0 & \text{if } x > c. \end{cases}$$

A plot of this function for $a = 1$, $b = 2$, and $c = 6$ is shown in Figure 24.9. In other words, the uncertain objective coefficients will be represented by the triangular functions of the form given above. Following Wang [131, p. 386], we use the notation $\tilde{c}_i = (c_i^-, c_i^0, c_i^+)$ to denote the uncertain coefficient c_i represented by the triangular function $\mu(x; c_i^-, c_i^0, c_i^+)$. Then the linear programming problem

$$\begin{aligned} & \text{minimize } \mathbf{c}^\top \mathbf{x} \\ & \text{subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \quad \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

becomes

$$\begin{aligned} & \text{minimize} && \begin{bmatrix} \mathbf{c}^-\mathbf{x} \\ \mathbf{c}^0\mathbf{x} \\ \mathbf{c}^+\mathbf{x} \end{bmatrix} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where

$$\mathbf{c}^- = [c_1^- \quad \cdots \quad c_n^-], \quad \mathbf{c}^0 = [c_1^0 \quad \cdots \quad c_n^0], \quad \mathbf{c}^+ = [c_1^+ \quad \cdots \quad c_n^+].$$

This is a multiobjective optimization problem. Wang [131] suggests that instead of minimizing the three values $\mathbf{c}^-\mathbf{x}$, $\mathbf{c}^0\mathbf{x}$, and $\mathbf{c}^+\mathbf{x}$ simultaneously, the center, $\mathbf{c}^0\mathbf{x}$, be minimized; the left leg, $(\mathbf{c}^0 - \mathbf{c}^-)\mathbf{x}$, be maximized; and the right leg, $(\mathbf{c}^+ - \mathbf{c}^0)\mathbf{x}$, be minimized. This results in pushing the triangular functions to the left in the minimization process. Thus, the multiobjective optimization problem above can be changed to the following multiobjective optimization problem:

$$\begin{aligned} & \text{minimize} && \begin{bmatrix} -(\mathbf{c}^0 - \mathbf{c}^-)\mathbf{x} \\ \mathbf{c}^0\mathbf{x} \\ (\mathbf{c}^+ - \mathbf{c}^0)\mathbf{x} \end{bmatrix} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Uncertain Constraint Coefficients

We may be faced with solving a linear programming problem with uncertain constraint coefficients. In this case the coefficients of the constraint matrix \mathbf{A} would be represented by triangular functions of the form given in the preceding section. That is, the coefficient a_{ij} of the constraint matrix \mathbf{A} would be modeled by the function $\tilde{a}_{ij} = \mu(x; a_{ij}^-, a_{ij}^0, a_{ij}^+)$. Then, the linear programming problem with uncertain constraint coefficients would take the form

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \begin{bmatrix} \mathbf{A}^-\mathbf{x} \\ \mathbf{A}^0\mathbf{x} \\ \mathbf{A}^+\mathbf{x} \end{bmatrix} \leq \begin{bmatrix} \mathbf{b} \\ \mathbf{b} \\ \mathbf{b} \end{bmatrix} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{A}^- = [a_{ij}^-]$, $\mathbf{A}^0 = [a_{ij}^0]$, and $\mathbf{A}^+ = [a_{ij}^+]$.

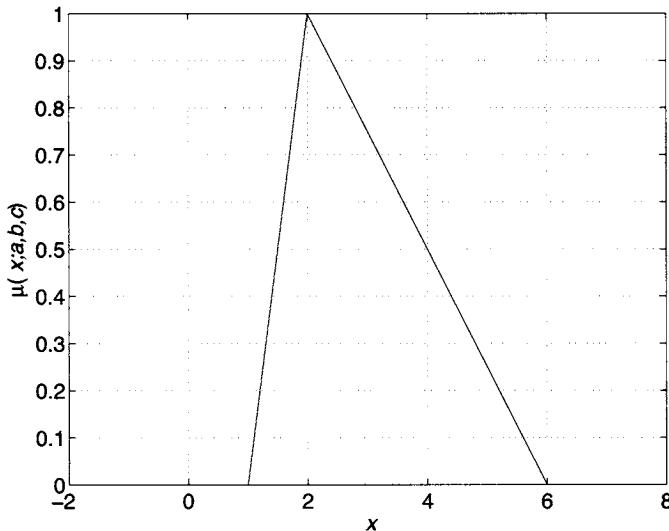


Figure 24.9 Plot of the triangular function $\mu(x; a, b, c)$ for $a = 1$, $b = 2$, and $c = 6$.

General Uncertainties

Finally, we may be faced with solving an uncertain linear programming problem that is a combination of the basic uncertain linear programming problems discussed above. For example, suppose that we are asked to solve the following quite general uncertain linear programming problem:

$$\begin{aligned} & \text{minimize} && \tilde{\mathbf{c}}^\top \mathbf{x} \\ & \text{subject to} && \tilde{\mathbf{A}}\mathbf{x} \leq \tilde{\mathbf{b}} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where the tilde symbols refer to the uncertain data; that is, we have

$$\tilde{\mathbf{c}} = (\mathbf{c}^-, \mathbf{c}^0, \mathbf{c}^+), \quad \tilde{\mathbf{A}} = (\mathbf{A}^-, \mathbf{A}^0, \mathbf{A}^+), \quad \tilde{\mathbf{b}} = (\mathbf{b}^-, \mathbf{b}^0, \mathbf{b}^+).$$

We can represent the uncertain linear programming problem above as a multiobjective optimization problem of the form

$$\begin{aligned} & \text{minimize} && \begin{bmatrix} -(\mathbf{c}^0 - \mathbf{c}^-)\mathbf{x} \\ \mathbf{c}^0\mathbf{x} \\ (\mathbf{c}^+ - \mathbf{c}^0)\mathbf{x} \end{bmatrix} \\ & \text{subject to} && \begin{bmatrix} \mathbf{A}^-\mathbf{x} \\ \mathbf{A}^0\mathbf{x} \\ \mathbf{A}^+\mathbf{x} \end{bmatrix} \leq \begin{bmatrix} \mathbf{b}^- \\ \mathbf{b}^0 \\ \mathbf{b}^+ \end{bmatrix} \\ & && \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

EXERCISES

24.1 Write a MATLAB program that implements the algorithm for generating a Pareto front, and test it on the problem in Example 24.1.

24.2 Consider the multiobjective problem

$$\begin{aligned} & \text{minimize } \mathbf{f}(\mathbf{x}) \\ & \text{subject to } \mathbf{x} \in \Omega, \end{aligned}$$

where $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^\ell$.

- a. Suppose that we solve the single-objective problem

$$\begin{aligned} & \text{minimize } \mathbf{c}^\top \mathbf{f}(\mathbf{x}) \\ & \text{subject to } \mathbf{x} \in \Omega, \end{aligned}$$

where $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{c} > \mathbf{0}$ (i.e., we use the weighted-sum approach). Show that if \mathbf{x}^* is a global minimizer for the single-objective problem above, then \mathbf{x}^* is a Pareto minimizer for the given multiobjective problem. Then show that it is not necessarily the case that if \mathbf{x}^* is a Pareto minimizer for the multiobjective problem, then there exists a $\mathbf{c} > \mathbf{0}$ such that \mathbf{x}^* is a global minimizer for the single-objective (weighted-sum) problem.

- b. Assuming that for all $\mathbf{x} \in \Omega$, $\mathbf{f}(\mathbf{x}) \geq \mathbf{0}$, suppose that we solve the single-objective problem

$$\begin{aligned} & \text{minimize } (f_1(\mathbf{x}))^p + \cdots + (f_\ell(\mathbf{x}))^p \\ & \text{subject to } \mathbf{x} \in \Omega, \end{aligned}$$

where $p \in \mathbb{R}$, $p > 0$ (i.e., we use the minimum-norm approach). Show that if \mathbf{x}^* is a global minimizer for the single-objective problem above, then \mathbf{x}^* is a Pareto minimizer for the given multiobjective problem. Then show that it is not necessarily the case that if \mathbf{x}^* is a Pareto minimizer for the multiobjective problem, then there exists a $p > 0$ such that \mathbf{x}^* is a global minimizer for the single-objective (minimum-norm) problem.

- c. Suppose that we solve the single-objective problem

$$\begin{aligned} & \text{minimize } \max\{f_1(\mathbf{x}), \dots, f_\ell(\mathbf{x})\} \\ & \text{subject to } \mathbf{x} \in \Omega \end{aligned}$$

(i.e., we use the minimax approach). Show that it is not necessarily the case that if \mathbf{x}^* is a Pareto minimizer for the given multiobjective problem, then \mathbf{x}^* is a global minimizer for the single-objective (minimax) problem. Then show that it also is not necessarily the case that if \mathbf{x}^* is a global

minimizer for the single-objective problem, then \mathbf{x}^* is a Pareto minimizer for the multiobjective problem.

24.3 Let $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^\ell$ be given. Consider the following multiobjective problem with equality constraints:

$$\begin{aligned} & \text{minimize} && \mathbf{f}(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \Omega. \end{aligned}$$

Suppose that $\mathbf{f} \in \mathcal{C}^1$, all the components of \mathbf{f} are convex, and Ω is convex. Suppose that there exists \mathbf{x}^* and $\mathbf{c}^* > \mathbf{0}$ such that for any feasible direction \mathbf{d} at \mathbf{x}^* , we have

$$\mathbf{c}^{*\top} D\mathbf{f}(\mathbf{x}^*) \mathbf{d} \geq 0.$$

Show that \mathbf{x}^* is a Pareto minimizer.

24.4 Let $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^\ell$ and $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be given. Consider the following multiobjective problem with equality constraints:

$$\begin{aligned} & \text{minimize} && \mathbf{f}(\mathbf{x}) \\ & \text{subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0}. \end{aligned}$$

Suppose that $\mathbf{f}, \mathbf{h} \in \mathcal{C}^1$, all the components of \mathbf{f} are convex, and the constraint set is convex. Show that if there exists \mathbf{x}^* , $\mathbf{c}^* > \mathbf{0}$, and $\boldsymbol{\lambda}^*$ such that

$$\begin{aligned} \mathbf{c}^{*\top} D\mathbf{f}(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} Dh(\mathbf{x}^*) &= \mathbf{0}^\top \\ \mathbf{h}(\mathbf{x}^*) &= \mathbf{0}, \end{aligned}$$

then \mathbf{x}^* is a Pareto minimizer. We can think of the above as a Lagrange condition for the constrained multiobjective function.

24.5 Let $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^\ell$ and $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ be given. Consider the following multiobjective problem with inequality constraints:

$$\begin{aligned} & \text{minimize} && \mathbf{f}(\mathbf{x}) \\ & \text{subject to} && \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned}$$

Suppose that $\mathbf{f}, \mathbf{g} \in \mathcal{C}^1$, all the components of \mathbf{f} are convex, and the constraint set is convex. Show that if there exists \mathbf{x}^* , $\mathbf{c}^* > \mathbf{0}$, and $\boldsymbol{\mu}^*$ such that

$$\begin{aligned} \boldsymbol{\mu}^* &\geq \mathbf{0}, \\ \mathbf{c}^{*\top} D\mathbf{f}(\mathbf{x}^*) + \boldsymbol{\mu}^{*\top} Dg(\mathbf{x}^*) &= \mathbf{0}^\top, \\ \boldsymbol{\mu}^{*\top} g(\mathbf{x}^*) &= 0, \\ g(\mathbf{x}^*) &\leq \mathbf{0}, \end{aligned}$$

then \mathbf{x}^* is a Pareto minimizer. We can think of the above as a KKT condition for the constrained multiobjective function.

24.6 Let $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^\ell$, $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, and $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ be given. Consider the general constrained multiobjective problem

$$\begin{aligned} & \text{minimize} && \mathbf{f}(\mathbf{x}) \\ & \text{subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & && \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned}$$

Suppose that $\mathbf{f}, \mathbf{g}, \mathbf{h} \in \mathcal{C}^1$, all the components of \mathbf{f} are convex, and the constraint set is convex. Show that if there exists \mathbf{x}^* , $\mathbf{c}^* > \mathbf{0}$, $\boldsymbol{\lambda}^*$, and $\boldsymbol{\mu}^*$ such that

$$\begin{aligned} & \boldsymbol{\mu}^* \geq \mathbf{0}, \\ & \mathbf{c}^{*\top} D\mathbf{f}(\mathbf{x}^*) + \boldsymbol{\lambda}^{*\top} D\mathbf{h}(\mathbf{x}^*) + \boldsymbol{\mu}^{*\top} D\mathbf{g}(\mathbf{x}^*) = \mathbf{0}^\top, \\ & \boldsymbol{\mu}^{*\top} \mathbf{g}(\mathbf{x}^*) = 0, \\ & \mathbf{h}(\mathbf{x}^*) = \mathbf{0}, \\ & \mathbf{g}(\mathbf{x}^*) \leq \mathbf{0}, \end{aligned}$$

then \mathbf{x}^* is a Pareto minimizer.

24.7 Let $f_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ and $f_2 : \mathbb{R}^n \rightarrow \mathbb{R}$, $f_1, f_2 \in \mathcal{C}^1$. Consider the minimax problem

$$\text{minimize } \max\{f_1(\mathbf{x}), f_2(\mathbf{x})\}.$$

Show that if \mathbf{x}^* is a local minimizer, then there exist $\mu_1^*, \mu_2^* \in \mathbb{R}$ such that

$$\mu_1^*, \mu_2^* \geq 0, \quad \mu_1^* \nabla f_1(\mathbf{x}^*) + \mu_2^* \nabla f_2(\mathbf{x}^*) = \mathbf{0}, \quad \mu_1^* + \mu_2^* = 1,$$

and $\mu_i^* = 0$ if $f_i(\mathbf{x}^*) < \max\{f_1(\mathbf{x}^*), f_2(\mathbf{x}^*)\}$.

Hint: Consider the following problem: minimize z subject to $z \geq f_i(\mathbf{x})$, $i = 1, 2$.

REFERENCES

1. J. S. Arora, *Introduction to Optimum Design*. New York: McGraw-Hill Book Co., 1989.
2. R. G. Bartle, *The Elements of Real Analysis*, 2nd ed. New York: Wiley, 1976.
3. M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 2nd ed. New York: Wiley, 1993.
4. A. Bhaya and E. Kaszkurewicz, *Control Perspectives on Numerical Algorithms and Matrix Problems*. Philadelphia: Society for Industrial and Applied Mathematics, 2006.
5. B. Beliczynski, A. Dzielinski, M. Iwanowski, and B. Ribeiro, Eds., *Adaptive and Natural Computing Algorithms*, vol. 4431 of *Lecture Notes in Computer Science*. Berlin: Springer, 2007.
6. A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*. New York: Wiley-Interscience, 1974.
7. L. D. Berkovitz, *Convexity and Optimization in \mathbb{R}^n* . Hoboken, NJ: Wiley, 2002.
8. C. C. Berresford, A. M. Rockett, and J. C. Stevenson, “Khachiyan’s algorithm, Part 1: A new solution to linear programming problems,” *Byte*, vol. 5, no. 8, pp. 198–208, Aug. 1980.
9. C. C. Berresford, A. M. Rockett, and J. C. Stevenson, “Khachiyan’s algorithm, Part 2: Problems with the algorithm,” *Byte*, vol. 5, no. 9, pp. 242–255, Sept. 1980.

10. D. P. Bertsekas, "Necessary and sufficient conditions for a penalty method to be exact," *Mathematical Programming*, vol. 9, no. 1, pp. 87–99, Aug. 1975.
11. D. P. Bertsekas, *Nonlinear Programming: 2nd ed.* Belmont, MA: Athena Scientific, 1999.
12. D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.
13. K. G. Binmore, *Calculus*. Cambridge, England: Cambridge University Press, 1986.
14. R. G. Bland, D. Goldfarb, and M. J. Todd, "The ellipsoid method: A survey," *Operations Research*, vol. 29, pp. 1039–1091, 1981.
15. V. G. Boltyanskii, *Mathematical Methods of Optimal Control*. New York: Holt, Rinehart and Winston, 1971.
16. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994.
17. R. P. Brent, *Algorithms for Minimization without Derivatives*. Englewood Cliffs, NJ: Prentice Hall, 1973.
18. L. Brickman, *Mathematical Introduction to Linear Programming and Game Theory*. New York: Springer-Verlag, 1989.
19. C. G. Broyden, "Quasi-Newton methods," in *Optimization Methods in Electronics and Communications* (K. W. Cattermole and J. J. O'Reilly, Eds.), vol. 1 of *Mathematical Topics in Telecommunications*. New York: Wiley, 1984, pp. 105–110,
20. A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*, rev. print. Washington, DC: Hemisphere Publishing Corporation, 1975.
21. B. D. Bunday, *Basic Optimization Methods*. London: Edward Arnold, 1984.
22. J. Campbell, *The Improbable Machine*. New York: Simon and Schuster, 1989.
23. S. L. Campbell and C. D. Meyer, Jr., *Generalized Inverses of Linear Transformations*. New York: Dover Publications, 1991.
24. E. K. P. Chong and B. E. Brewington, "Distributed communications resource management for tracking and surveillance networks," in *Proceedings of the Conference on Signal and Data Processing of Small Targets 2005* (SPIE Vol. 5913), part of the *SPIE Symposium on Optics & Photonics*, San Diego, California, July 31–Aug. 4, 2005, pp. 280–291.
25. E. K. P. Chong and B. E. Brewington, "Decentralized rate control for tracking and surveillance networks," *Ad Hoc Networks*, special issue on *Recent Advances in Wireless Sensor Networks*, vol. 5, no. 6, pp. 910–928, Aug. 2007.
26. E. K. P. Chong, S. Hui, and S. H. Žak, "An analysis of a class of neural networks for solving linear programming problems," *IEEE Transactions on Automatic Control*, special section on *Neural Networks in Control, Identification, and Decision Making*, vol. 44, no. 11, pp. 1995–2006, Nov. 1999.
27. E. K. P. Chong and S. H. Žak, "Single-dimensional search methods," in *Wiley Encyclopedia of Operations Research and Management Science*, 2011, ISBN: 978-0-470-40063-0.

28. A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*. Chichester, England: Wiley, 1993.
29. M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the Congress of Evolutionary Computation*, Washington, DC, July 1999, pp. 1951–1957.
30. M. Clerc and J. Kennedy, "The particle swarm: Explosion, stability and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 58–73, Feb. 2002.
31. C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Kluwer Academic/Plenum Publishers, 2002.
32. S. D. Conte and C. de Boor, *Elementary Numerical Analysis: An Algorithmic Approach*, 3rd ed. New York: McGraw-Hill Book Co., 1980.
33. M. A. Dahleh and I. J. Diaz-Bobillo, *Control of Uncertain Systems: A Linear Programming Approach*. Upper Saddle River, NJ: Prentice Hall, 1995.
34. G. B. Dantzig, *Linear Programming and Extensions*. Princeton, NJ: Princeton University Press, 1963.
35. G. B. Dantzig and M. N. Thapa, *Linear Programming*, vol. 1, *Introduction*. New York: Springer-Verlag, 1997.
36. L. Davis, Ed., *Genetic Algorithms and Simulated Annealing*, Research Notes in Artificial Intelligence. London: Pitman, 1987.
37. K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*. Chichester, England: Wiley, 2001.
38. V. F. Dem'yanov and L. V. Vasil'ev, *Nondifferentiable Optimization*. New York: Optimization Software, Inc., Publications Division, 1985.
39. J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice Hall, 1983.
40. J. Dongarra and F. Sullivan, "The top 10 algorithms," *Computing in Science and Engineering*, pp. 22–23, Jan./Feb. 2000.
41. V. N. Faddeeva, *Computational Methods of Linear Algebra*. New York: Dover Publications, 1959.
42. S.-C. Fang and S. Puthenpura, *Linear Optimization and Extensions: Theory and Algorithms*. Englewood Cliffs, NJ: Prentice Hall, 1993.
43. R. Fletcher, *Practical Methods of Optimization*, 2nd ed. Chichester, England: Wiley, 1987.
44. F. R. Gantmacher, *The Theory of Matrices*, vol. 1. New York: Chelsea Publishing Co., 1959.
45. F. R. Gantmacher, *The Theory of Matrices*, 2nd ed. Moscow: Nauka, revised 1966. In Russian.
46. S. I. Gass, *An Illustrated Guide to Linear Programming*. New York: McGraw-Hill Book Co., 1970.

47. I. M. Gel'fand, *Lectures on Linear Algebra*. New York: Interscience Publishers, 1961.
48. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.
49. P. E. Gill and W. Murray, "Safeguarded steplength algorithms for optimization using descent methods," Tech. Rep. NPL NAC 37, National Physical Laboratory, Division of Numerical Analysis and Computing, Teddington, England, Aug. 1974.
50. P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, "Two step-length algorithms for numerical optimization," Tech. Rep. SOL 79-25, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, Dec. 1979.
51. P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. London: Academic Press, 1981.
52. P. E. Gill, W. Murray, and M. H. Wright, *Numerical Linear Algebra and Optimization*. Redwood City, CA: Addison-Wesley, 1991.
53. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed.. Baltimore, MD: The Johns Hopkins University Press, 1983.
54. R. E. Gomory, "Outline of an algorithm for integer solutions to linear programs," *Bulletin of the American Mathematical Society*, vol. 64, no. 5, pp. 275–278, Sep. 1958.
55. C. C. Gonzaga, "Path-following methods for linear programming," *SIAM Review*, vol. 34, no. 2, pp. 167–224, June 1992.
56. B. Hajek, "Cooling schedules for optimal annealing," *Mathematics of Operations Research*, vol. 13, no. 2, pp. 311–329, 1988.
57. J. Hannig, E. K. P. Chong, and S. R. Kulkarni, "Relative frequencies of generalized simulated annealing," *Mathematics of Operations Research*, vol. 31, no. 1, pp. 199–216, Feb. 2006.
58. R. L. Harvey, *Neural Network Principles*. Englewood Cliffs, NJ: Prentice Hall, 1994.
59. S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.
60. J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, vol. 1 of *Santa Fe Institute Studies in the Sciences of Complexity*. Redwood City, CA: Addison-Wesley, 1991.
61. J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA: MIT Press, 1992.
62. R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, England: Cambridge University Press, 1985.
63. A. S. Householder, *The Theory of Matrices in Numerical Analysis*. New York: Dover Publications, 1975.

64. S. Hui and S. H. Źak, "The Widrow-Hoff algorithm for McCulloch-Pitts type neurons," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 924–929, Nov. 1994.
65. D. R. Hush and B. G. Horne, "Progress in supervised neural networks: What's new since Lippmann," *IEEE Signal Processing Magazine*, pp. 8–39, Jan. 1993.
66. S. Isaak and M. N. Manougian, *Basic Concepts of Linear Algebra*. New York: W. W. Norton & Co., 1976.
67. J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Upper Saddle River, NJ: Prentice Hall, 1997.
68. W. E. Jenner, *Rudiments of Algebraic Geometry*. New York: Oxford University Press, 1963.
69. E. M. Johansson, F. U. Dowla, and D. M. Goodman, "Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method," *International Journal of Neural Systems*, vol. 2, no. 4, pp. 291–301, 1992.
70. S. Kaczmarz, "Approximate solution of systems of linear equations," *International Journal of Control*, vol. 57, no. 6, pp. 1269–1271, 1993. A reprint of the original paper: S. Kaczmarz, "Angenäherte Auflösung von Systemen linearer Gleichungen," *Bulletin International de l'Academie Polonaise des Sciences et des Lettres, Serie A*, pp. 355–357, 1937.
71. N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.
72. M. F. Kelly, P. A. Parker, and R. N. Scott, "The application of neural networks to myoelectric signal analysis: A preliminary study," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 3, pp. 221–230, Mar. 1990.
73. J. Kennedy and R. C. Eberhart, with Y. Shi, *Swarm Intelligence*. San Francisco: Morgan Kaufmann, 2001.
74. L. G. Khachiyan, "A polynomial algorithm in linear programming," *Soviet Mathematics Doklady*, vol. 20, no. 1, pp. 191–194, 1979.
75. S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
76. V. Klee and G. J. Minty, "How good is the simplex algorithm?" in *Inequalities-III* (O. Shisha, Ed.), New York: Academic Press, 1972, pp. 159–175.
77. D. E. Knuth, *The Art of Computer Programming*, vol. 1, *Fundamental Algorithms*, 2nd ed. Reading, MA: Addison-Wesley, 1973.
78. L. Kolev, "Iterative algorithm for the minimum fuel and minimum amplitude problems for linear discrete systems," *International Journal of Control*, vol. 21, no. 5, pp. 779–784, 1975.
79. J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.
80. T. Kozek, T. Roska, and L. O. Chua, "Genetic algorithm for CNN template learning," *IEEE Transactions on Circuits and Systems, I: Fundamental Theory and Applications*, vol. 40, no. 6, pp. 392–402, June 1993.

81. K. Kuratowski, *Introduction to Calculus*, 2nd ed., vol. 17 of *International Series of Monographs in Pure and Applied Mathematics*. Warsaw, Poland: Pergamon Press, 1969.
82. J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, “Convergence properties of the Nelder-Mead simplex method in low dimensions,” *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998.
83. S. Lang, *Calculus of Several Variables*, 3rd ed. New York: Springer-Verlag, 1987.
84. J. M. Layton, *Multivariable Control Theory*. Stevenage, England: Peter Peregrinus on behalf of the Institution of Electrical Engineers, 1976.
85. E. B. Lee and L. Markus, *Foundations of Optimal Control Theory*. Malabar, FL: Robert E. Krieger Publishing Company, 1986.
86. G. Leitmann, *The Calculus of Variations and Optimal Control: An Introduction*. New York: Plenum Press, 1981.
87. D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1969.
88. D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 3rd ed. New York, NY: Springer Science + Business Media, 2008.
89. I. J. Maddox, *Elements of Functional Analysis*, 2nd ed. Cambridge, England: Cambridge University Press, 1988.
90. O. L. Mangasarian, *Nonlinear Programming*. New York: McGraw-Hill Book Co., 1969.
91. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
92. K. M. Miettinen, *Nonlinear Multiobjective Optimization*. Norwell, MA: Kluwer Academic Publishers, 1998.
93. S. A. Miller and E. K. P. Chong, “Flow-rate control for managing communications in tracking and surveillance networks,” in *Proceedings of the Conference on Signal and Data Processing of Small Targets 2007* (SPIE Vol. 6699), part of the *SPIE Symposium on Optics & Photonics*, San Diego, California, Aug. 26–30, 2007.
94. M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996.
95. A. Mostowski and M. Stark, *Elements of Higher Algebra*. Warsaw, Poland: PWN—Polish Scientific Publishers, 1958.
96. S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*. New York: McGraw-Hill Book Co., 1996.
97. J. A. Nelder and R. Mead, “A simplex method for function minimization,” *Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.
98. A. Osyczka, *Evolutionary Algorithms for Single and Multicriteria Design Optimization*. Heidelberg, Germany: Physica-Verlag, 2002.

99. D. H. Owens, *Multivariable and Optimal Systems*. London: Academic Press, 1981.
100. T. M. Ozan, *Applied Mathematical Programming for Production and Engineering Management*. Englewood Cliffs, NJ: Prentice Hall, 1986.
101. C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice Hall, 1982.
102. P. C. Parks, "S. Kaczmarz (1895–1939)," *International Journal of Control*, vol. 57, no. 6, pp. 1263–1267, 1993.
103. R. J. Patton and G. P. Liu, "Robust control design via eigenstructure assignment, genetic algorithms and gradient-based optimisation," *IEE Proceedings on Control Theory and Applications*, vol. 141, no. 3, pp. 202–208, May 1994.
104. A. L. Peressini, F. E. Sullivan, and J. J. Uhl, Jr., *The Mathematics of Nonlinear Programming*. New York: Springer-Verlag, 1988.
105. A. Pezeshki, L. L. Scharf, M. Lundberg, and E. K. P. Chong, "Constrained quadratic minimizations for signal processing and communications," in *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, Seville, Spain, Dec. 12–15, 2005, pp. 7949–7953.
106. A. Pezeshki, L. L. Scharf, and E. K. P. Chong, "The geometry of linearly and quadratically constrained optimization problems for signal processing and communications," *Journal of the Franklin Institute*, special issue on *Modelling and Simulation in Advanced Communications*, vol. 347, no. 5, pp. 818–835, June 2010.
107. M. J. D. Powell, "Convergence properties of algorithms for nonlinear optimization," *SIAM Review*, vol. 28, no. 4, pp. 487–500, Dec. 1986.
108. S. S. Rangwala and D. A. Dornfeld, "Learning and optimization of machining operations using computing abilities of neural networks," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 2, pp. 299–314, Mar./Apr. 1989.
109. G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell, *Engineering Optimization: Methods and Applications*. New York: Wiley-Interscience, 1983.
110. A. M. Rockett and J. C. Stevenson, "Karmarkar's algorithm: A method for solving large linear programming problems," *Byte*, vol. 12, no. 10, pp. 146–160, Sept. 1987.
111. H. L. Royden, *Real Analysis*, 3rd ed. New York: Macmillan Company, 1988.
112. W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. New York: McGraw-Hill Book Co., 1976.
113. D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, *Foundations*. Cambridge, MA: MIT Press, 1986.
114. D. Russell, *Optimization Theory*. New York: W. A. Benjamin, 1970.
115. S. L. Salas and E. Hille, *Calculus: One and Several Variables*, 4th ed. New York: Wiley, 1982.

116. L. L. Scharf, L. T. McWhorter, E. K. P. Chong, J. S. Goldstein, and M. D. Zoltowski, "Algebraic equivalence of conjugate direction and multistage Wiener filters," in *Proceedings of the Eleventh Annual Workshop on Adaptive Sensor Array Processing (ASAP)*, Lexington, Massachusetts, Mar. 11–13, 2003.
117. L. L. Scharf, E. K. P. Chong, and Z. Zhang, "Algebraic equivalence of matrix conjugate direction and matrix multistage filters for estimating random vectors," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC'04)*, Atlantis Resort, Paradise Island, Bahamas, Dec. 14–17, 2004, pp. 4175–4179.
118. L. L. Scharf, E. K. P. Chong, M. D. Zoltowski, J. S. Goldstein, and I. S. Reed, "Subspace expansion and the equivalence of conjugate direction and multistage Wiener filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5013–5019, Oct. 2008.
119. A. Schrijver, *Theory of Linear and Integer Programming*. New York: Wiley, 1986.
120. R. T. Seeley, *Calculus of Several Variables: An Introduction*. Glenview, IL: Scott, Foresman and Co., 1970.
121. J. R. Silvester, "Determinants of block matrices," *The Mathematical Gazette*, vol. 48, no. 51, pp. 460–467, Nov. 2000.
122. W. Spendley, G. R. Hext, and F. R. Himsorth, "Sequential application of simplex designs in optimization and evolutionary operation," *Technometrics*, vol. 4, pp. 441–461, 1962.
123. W. A. Spivey, *Linear Programming: An Introduction*. New York: Macmillan Company, 1963.
124. R. E. Stone and C. A. Tovey, "The simplex and projective scaling algorithms as iteratively reweighted least squares methods," *SIAM Review*, vol. 33, no. 2, pp. 220–237, June 1991.
125. G. Strang, *Introduction to Applied Mathematics*. Wellesley, MA: Wellesley-Cambridge Press, 1986.
126. G. Strang, *Linear Algebra and Its Applications*. New York: Academic Press, 1980.
127. T. W. Then and E. K. P. Chong, "Genetic algorithms in noisy environments," in *Proceedings of the 9th IEEE Symposium on Intelligent Control*, pp. 225–230, Aug. 1994.
128. L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, Mar. 1996.
129. P. P. Varaiya, *Notes on Optimization*. New York: Van Nostrand Reinhold Co., 1972.
130. D. J. Velleman, *How To Prove It: A Structured Approach*. Cambridge, England: Cambridge University Press, 1994.
131. L.-X. Wang, *A Course in Fuzzy Systems and Control*. Upper Saddle River, NJ: Prentice Hall, 1999.

132. B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, Sept. 1990.
133. D. J. Wilde, *Optimum Seeking Methods*. Englewood Cliffs, NJ: Prentice Hall, 1964.
134. R. E. Williamson and H. F. Trotter, *Multivariable Mathematics, 2nd ed.* Englewood Cliffs, NJ: Prentice Hall, 1979.
135. W. I. Zangwill, *Nonlinear Programming: A Unified Approach*. Englewood Cliffs, NJ: Prentice Hall, 1969.
136. G. Zoutendijk, *Mathematical Programming Methods*. Amsterdam, The Netherlands: North-Holland, 1976.
137. J. M. Zurada, *Introduction to Artificial Neural Systems*. St. Paul, MN: West Publishing Co., 1992.

Index

- Absolute value, 19
Absolute value penalty function, 561
Activation function, 254
Active constraint, 483
Adaline, 257, 271
Adaptive linear element, 257
Additivity, 19
Affine function, 62, 63, 247, 524
Affine matrix inequality, 524
Affine scaling, 406, 407
Affine scaling method, 406–411
 artificial problem, 410
 stopping criterion, 409
 strictly interior feasible point, 406
Algebraic Riccati inequality, 544
Algorithm
 affine scaling, 406–411
 backpropagation, 253, 258–269
 BFGS, 207–211
 Broyden-Fletcher-Goldfarb-Shanno, *see* BFGS algorithm
 complexity of, 306, 402
 conjugate gradient, *see* Conjugate gradient algorithm
 convergence of, *see* Convergence
 Davidon-Fletcher-Powell, *see* DFP algorithm
 DFP, 202–207
 ellipsoid, *see* Khachiyan's method
 exponential complexity, 402
 fixed step size, 146, 151, 261
 for constrained optimization, 545–567
 genetic, 285–298
 globally monotone, 157
 gradient, 131–153
 Gram-Schmidt, 177, 189
 interior-point, 307, 403, 406, 411, 423
 iterative, 124, 159. *See also* Search methods
 Kaczmarz's, 232–236, 257
 Karmarkar's, *see* Karmarkar's method
 Khachiyan's, 306, 402–405, 418
 Lagrangian, 553–560
 naive random search, 279
 Nelder-Mead, 274–278
 particle swarm optimization, 282–285
 polynomial complexity, 402

- probabilistic search, 278
- projected, 546, 556
- projected gradient, 407, 547, 549–553
- projected steepest descent, 551
- quasi-Newton, *see* Quasi-Newton methods
- randomized search, 278
- rank one, 197–202
- rank two, 202
- RLS, 227–232, 250
- secant method, 120–122, 172
- simplex, *see* Simplex method
- simulated annealing, 278–282
- single-rank symmetric, 197
- SRS, 197
- steepest descent, 133–141
- symmetric Huang family, 215
- variable metric, 202
- Widrow-Hoff, 257, 271
- zero finding, 118, 155
- Allocation**, 541
- Alphabet** in genetic algorithm, 286
- Argmin**, 82
- Armijo** backtracking algorithm, 126
- Armijo** condition, 125
- Armijo-Goldstein** condition, 125
- Artificial neural networks**, *see* Feedforward neural networks
- Artificial problem**
 - in affine scaling method, 410
 - in Karmarkar's method, 418
 - in simplex method, 361
- Associative**, 8, 9
- Asymmetric duality**, 379
- Augmented matrix**, 325
- Backpropagation algorithm**, 253, 258–269
 - as a gradient algorithm, 261
 - forward pass, 263
 - reverse pass, 263
- Ball**, 50
- Banana (Rosenbrock's) function**, 68, 159, 173, 191, 215
- Basic columns**, 325
- Basic feasible solution**, 305, 325, 327, 331, 349
- Basic solutions**, 324–327
- Basic variables**, 325, 347
- Basis**
 - definition of, 11
 - entering, 351
 - in linear equations, 325, 347, 348
- leaving**, 351
- natural**, 11
- orthogonal**, 29
- Beltrami**, 562
- Best-so-far**, 280, 283, 290
- BFGS algorithm**, 207–211
- Big-oh notation**, 74, 149
- Bisection method**, 116
- Bland's rule**, 360, 375
- Boltzmann**, 281
- Bolzano-Weierstrass theorem**, 58, 234
- Boundary**, 51
- Boundary point**, 51
- Bounded above**, 57
- Bounded below**, 57
- Bounded sequence**, 56, 57
- Bounded set**, 51
- Box constraint**, 546
- Bracketing**, 123, 129
- Brent's method**, 122
- Broyden**, 207
- Broyden-Fletcher-Goldfarb-Shanno algorithm**, *see* BFGS algorithm
- Canonical augmented matrix**, 346–348
- Canonical form**, 346
- Canonical representation**, 347
- Canonical representation of LMI**, 529
- Canonical tableau**, 358
- Carrier of polyhedron**, 53
- Cauchy-Schwarz inequality**, 20, 132, 207, 551
- Center of gravity**, 102, 275
- Centroid**, 102, 274
- Chain rule**, 67
- Characteristic equation**, 26
- Characteristic polynomial**, 26
- Chromosome in genetic algorithm**, 286
- Circuit**, 121, 253, 254, 311, 487
- Citation style**, 6
- Clairaut's theorem**, 65
- Closed set**, 51
- Column vector**, 7
- Combinatorial optimization**, 273, 282, 297
- Commutative**, 8
- Compact set**, 51
- Compatible matrix norm**, 36
- Complementarity**, 207
- Complementary slackness**, 390, 539
- Complex inner product**, 21
- Complex vector space**, 12
- Complexity of algorithm**, 306, 402
 - exponential, 402

- polynomial, 402
- Component of vector, 7
- Composite function, 67, 85, 90, 459
- Concave function, 512, *see* Convex function
- Condition number, 148
- Conjugate direction methods, 175–188
- Conjugate gradient algorithm
 - Fletcher-Reeves formula, 187
 - Hestenes-Stiefel formula, 187
 - nonquadratic problems, 186–188
 - Polak-Ribière formula, 187
 - Powell formula, 188
 - quadratic problems, 182–186
 - stopping criterion, 188
- Consistent linear inequalities, 404
- Constrained optimization, 82, 305, 449
- Constraint
 - active, 483
 - box, 546
 - convex, 517
 - equality, 450, 504
 - functional, 82
 - inactive, 483
 - inequality, 483, 561
 - set, 82
- Constraint set, 81. *See also* Feasible set
- Continuity, 21, 60, 453, 565
- Continuous function, 21, 60, 453, 565
- Continuously differentiable function, 65, 451, 513
- Contradiction, proof, 5
- Contraposition, proof, 4
- Contrapositive, 4
- Control system, 102, 472, 474, 475, 481, 553
- Convergence
 - fixed-step-size gradient algorithm, 146
 - globally convergent, 141
 - gradient algorithms, 143
 - Kaczmarz's algorithm, 233
 - linear, 148
 - locally convergent, 141, 554, 556
 - Newton's method, 165
 - of sequence of matrices, 59
 - order of, 148, 149, 152, 153, 157, 165
 - penalty method, 565
 - quadratic (second-order), 148
 - rate of, 141, 148
 - ratio, 148
 - steepest descent algorithm, 145
 - sublinear, 148
 - superlinear, 148
- Convergent sequence, 56
- Convex combination, 48, 297, 582
- Convex constraint, 517
- Convex function, 508–517
 - definition of, 509
 - differentiable, 513
 - equivalent definition of, 509
 - minimizers of, 517
 - optimization of, 517–536
 - quadratic, 512
 - strict, 512
 - twice differentiable, 515
- Convex optimization, 517–536
- Convex programming, *see* Convex optimization
- Convex set, 48–50
 - definition of, 48, 330, 509
 - extreme point, 50, 331, 332, 335
 - in definition of convex function, 509
 - polyhedron, 52, 317
 - polytope, 52, 316
 - properties of, 49
 - supporting hyperplane, 52, 316
- Cooling schedule, 281
- Coordinates, 11
- Cost function, 81
- Courant-Beltrami penalty function, 562, 570
- Cramer's rule, 33
- Crossing site, 287
- Crossover in genetic algorithm, 287
 - crossing site, 287
 - multiple-point crossover, 287, 300
 - one-point crossover, 287
- Cubic fit, 122
- Curve, 453
- Cutting-plane method, 435
- Cycling in simplex method, 360, 375
- Dantzig, 306
- Davidon, 202
- Davidon-Fletcher-Powell algorithm, *see* DFP algorithm
- Decision variable, 81, 305, 541
- Decomposition
 - direct sum, 30, 458
 - orthogonal, 30, 226
- Decreasing sequence, 55
- Degenerate basic feasible solution, 325, 328, 360
- DeMorgan's law, 3
- Derivative, 62, 83
 - partial, 64

- Derivative descent search, 154
 Derivative matrix, 64
 Descent property, 135, 141, 144, 167, 168, 193, 552
 Determinant, 14
 DFP algorithm, 202–207
 Diagonal matrix, 27, 407, 421, 541, 571
 Diet problem, 308, 381, 388
 Differentiable curve, 453
 Differentiable function, 62, 63, 453
 Dimension, 11, 452
 Direct sum decomposition, 30, 458
 Directional derivative, 84
 Discrete Fourier series, 225
 Discrete-time linear system, 102, 248, 474, 481
 Distributive, 9
 Domination, 575
 Dual linear program, 378, 403
 Dual nonlinear program, 543
 Dual quadratic program, 399
 Duality
 asymmetric, 379
 dual nonlinear program, 543
 dual problem, 378, 403
 dual quadratic program, 399
 dual vector, 378
 duality theorem, 387, 543
 in quasi-Newton methods, 207
 Karush-Kuhn-Tucker conditions, 395, 539
 linear programming, 377, 539
 nonlinear programming, 542
 primal nonlinear program, 542
 primal problem, 378, 403
 primal quadratic program, 399
 quadratic programming, 399
 symmetric, 378, 403
 weak duality lemma, 385, 399, 543
 Duality theorem, 387, 543
 Dyadic product, 197
 Eberhart, Russell, 282
 Edge of polyhedron, 53
 Eigenvalue
 definition of, 26
 maximal, 144
 minimal, 144
 of symmetric matrix, 28, 35
 Eigenvector
 definition of, 26
 of symmetric matrix, 28
 orthogonal, 28
 relation to \mathbf{Q} -conjugacy, 189
 Electric circuit, 311, 487
 Elementary matrix
 elementary row operation, 341
 first kind, 340
 second kind, 340
 third kind, 341
 Elementary row operation, 341
 Elitism in genetic algorithm, 290
 Ellipsoid, 405
 Ellipsoid algorithm, *see* Khachiyan's method
 Encoding in genetic algorithm, 285, 290, 297
 Entry of matrix, 12
 Epigraph, 508
 Equality constraint, 450, 504
 Estimation, 121, 223, 244
 Euclidean inner product, 19
 Euclidean norm, 20
 Evolution in genetic algorithm, 287
 Exact penalty function, 566
 Exclusive OR, *see* XOR
 Expanding subspace theorem, 181
 Exponential complexity, 402
 Extreme point, 50, 331, 332, 334
 Extremizer, 82
 Face of polyhedron, 53, 316
 Farkas's transposition theorem, 396
 Feasibility problem, 398, 526, 529
 Feasible direction, 84, 519, 566
 Feasible point, 305, 450
 Feasible set, 81, 450
 Feedforward neural networks, 253–269
 activation function, 254
 Adaline, 257
 backpropagation algorithm, 253, 258–269
 function approximation, 255
 hidden layer, 254
 input layer, 254
 learning, 253
 neuron, 253
 output layer, 254
 single-neuron training, 256–258
 supervised learning, 256
 training, 253
 training set, 255
 unsupervised learning, 256
 weights, 253, 254
 Fibonacci method, 108–115
 Fibonacci sequence, 109
 First-order Lagrangian algorithm, 553
 First-order necessary condition

- equality constraint (Lagrange), 460, 461
 - in convex optimization, 518
 - inequality constraint (KKT), 484
 - interior case, 86
 - set constraint, 85
- Fitness in genetic algorithm, 286
- Fitting straight line, 101, 221, 227, 246, 247
- Fixed point, 553
- Fixed step size, 146, 151, 261
- Fletcher, 187, 202, 207
- Fletcher-Reeves formula, 187
- Floor, 435
- FONC, *see* First-order necessary condition
- Fourier series, 225
- Frobenius norm, 36
- Full-rank factorization, 236
- Function
 - affine, 62, 63, 247, 524
 - banana, 68, 159, 173, 191, 215
 - composite, 67, 85, 90, 459
 - concave, 512, *see* Convex function
 - continuous, 21, 60, 453, 565
 - continuously differentiable, 65, 451, 513
 - convex, 508–517
 - cost, 81
 - derivative matrix of, 64
 - derivative of, 62, 83
 - differentiable, 62, 63, 453
 - directional derivative of, 84
 - gradient of, 65, 71, 83, 131
 - graph of, 71, 508
 - Jacobian matrix of, 64
 - Lagrangian, 464, 468, 543, 553
 - linear, *see* Linear transformation
 - matrix-valued, 60
 - maximum rate of decrease, 132
 - maximum rate of increase, 71, 131
 - notation, 5
 - objective, 81
 - partial derivative of, 64
 - penalty, 560
 - Powell, 162
 - Rosenbrock's, 68, 159, 173, 191, 215
 - sigmoid, 266
 - twice continuously differentiable, 65, 515
 - twice differentiable, 65, 454
 - uniformly continuous, 23
 - unimodal, 104
 - utility, 542
- Function approximation, 255
- Functional constraint, 82
- Fundamental theorem of algebra, 26
- Fundamental theorem of linear algebra, 41
- Fundamental theorem of LP, 328
- Fuzzy linear programming, 585
- Gale's transposition theorem, 397
- Gauss-Newton method, 171
- Generalized eigenvalue, 468, 534
- Generalized inverse, 236, 238
- Genetic algorithm, 285–298
 - alphabet, 286
 - analysis of, 291–297
 - best-so-far chromosome, 290
 - chromosome, 286
 - crossover, 287
 - elitism, 290
 - encoding, 285, 290, 297
 - evolution, 287
 - fitness, 286
 - initial population, 285
 - length of schema, 294
 - mating pool, 286
 - mutation, 288
 - offspring, 287
 - order of schema, 294
 - parents, 287
 - population size, 286
 - real-number, 297–298
 - representation scheme, 286
 - roulette-wheel scheme, 286
 - schema, 292
 - selection, 286
 - stopping criterion, 290
 - tournament scheme, 286
- Gibbs, 281
- Global minimizer, 82, 94, 473, 517–519, 521, 537, 552
- Globally convergent, 141
- Globally monotone algorithm, 157
- Golden section, 106
- Golden section search, 104–108
- Goldfarb, 207
- Goldstein condition, 125
- Gomory cut, 435
- Gomory cutting-plane method, 435
- Gordan's transposition theorem, 397
- Gradient, 65, 71, 83, 131
- Gradient descent algorithm, *see* Algorithm, gradient
- Gradient methods, 131–153

- backpropagation algorithm, 253, 258–269
constrained optimization, see Projected gradient method
 convergence of, 141–147
 convergence rate of, 147–153
 descent property, 135, 141, 144
equality constraints, see Lagrangian algorithms
 fixed step size, 145
inequality constraints, see Lagrangian algorithms
 Lagrangian, 553–560
 order of convergence, 152
 projected, 547, 549–553
 stopping criterion, 135
- Gram matrix, 220
 Gram-Schmidt, 177, 189
 Grammian, 220
 Graph, 71, 508
 Greatest lower bound, 57
- Hačijan, *see* Khachiyan
 Hadamard product, 283
 Hajek, 282
 Half-space, 46, 316
 negative, 46
 positive, 46
 Hessian, 65, 468, 515
 Hessian matrix, 83
 Hestenes, Magnus, 187, 188
 Hestenes-Stiefel formula, 187
 Hidden layer in neural network, 254
 Hoff, 257
 Holland, John, 285
 Homogeneity, 19, 21
 Huang family, 215
 Hyperplane
 definition of, 46
 supporting, 52, 316
 tangent to graph, 71
- Identity matrix, 16
 ILP, *see* Integer linear programming
 Image of matrix, *see* Range of matrix
 Implicit function theorem, 457
 Impulse response, 248
 Inactive constraint, 483
 Inconsistent system of equations, 217
 Increasing sequence, 55
 Indefinite matrix, 35
 Induced matrix norm, 36, 480
 Induction, principle of, 5
 Inequality constraint, 483, 561
- Infimum, *see* Greatest lower bound
 Inner product
 complex, 21
 Euclidean, 19
 properties of, 19
 Innovation, 229
 Input layer in neural network, 254
 Integer linear programming, 427–444
Integer programming, see Integer linear programming
 Interior, 51
 Interior point, 51
 Interior-point method, 307, 403, 406, 411, 423
 Inverse
 continuity of, 60
 matrix, 16
 Inverse Hessian, 194
 Inverse parabolic interpolation, 122
 Invertible matrix, *see* Nonsingular matrix
 Iterative algorithm, 124, *see* Search methods, 159
- Jacobian matrix, 64
 Jordan form, 59
- Kaczmarz's algorithm, 232–236, 257
 Kantorovich, 306
 Karmarkar, 306, 403
 Karmarkar's method, 306, 403, 411–423
 artificial problem, 418
 complexity, 403
 Karmarkar's canonical form, 411–413, 415
 Karmarkar's restricted problem, 414–415
 projective transformation, 416, 424
 simplex, 412
 stopping criterion, 415, 420
 strictly interior feasible point, 407, 416
- Karush-Kuhn-Tucker condition, *see* KKT condition
 Karush-Kuhn-Tucker multiplier, *see* KKT multiplier
 Karush-Kuhn-Tucker theorem, 484
 Kennedy, James, 282
 Kernel of matrix, *see* Nullspace of matrix
 Khachiyan, 306, 402
 Khachiyan's method, 306, 402–405, 418
 KKT condition, 484, 485, 489, 521, 556, 594
 KKT multiplier, 484, 490

- KKT theorem, 484
- Klee-Minty problem, 401
- Koopmans, 306
- Krylov subspace, 188
- Kuhn-Tucker condition, *see* KKT condition
- Lagrange condition, 460, 464, 520, 550, 553, 593
- Lagrange multiplier, 460, 463
- Lagrange's theorem, 460, 461
- Lagrangian algorithms, 553–560
- Lagrangian function, 464, 468, 543, 553
- Lanczos, Cornelius, 188
- Leading principal minor, 31
- Learning in neural network, 253
- Least squares, 217–227, 238
 - nonlinear, 169
- Least upper bound, 57
- Left pseudoinverse, 238
- Level set, 68, 131, 134
- Levenberg-Marquardt algorithm, 171
- Levenberg-Marquardt modification, 168
- Limit of sequence, 55
- Line fitting, 101, 221, 227, 246, 247
- Line search, 103, 124, 133, 167, 186, 188, 194, 209
- Line segment, 45, 48
- Linear combination, 10
- Linear convergence, 148
- Linear dynamical system, *see* Discrete-time linear system
- Linear equations
 - augmented matrix, 325
 - basic solution, 325
 - basis, 325, 347, 348
 - canonical augmented matrix, 347
 - canonical form, 346
 - degenerate basic solutions, 325
 - existence of solution, 17
 - inconsistent, 217
 - Kaczmarz's algorithm, 232–236
 - least-squares solution, 217, 218, 221
 - minimum-norm solution, 231, 241, 257, 473
 - overdetermined, 217
 - particular solution, 346
 - pivot, 349, 352, 364
 - solving in general, 217–243
 - solving using row operations, 339–346
- Linear function, *see* Linear transformation
- Linear inequalities
- consistent, 404
- in linear programming, 305, 307, 316
- Linear least squares, 217–227, 238
- Linear matrix inequality, 524, 541
- Linear programming
 - affine scaling method, 406–411
 - artificial problem in affine scaling method, 410
 - artificial problem in Karmarkar's method, 418
 - artificial problem in simplex method, 361
 - artificial variables in simplex method, 361
 - as constrained problem, 450
 - asymmetric duality, 379
 - basic columns, 325
 - basic feasible solution, 305, 325, 327, 331, 349
 - basic solutions, 324–327
 - basic variables, 325, 347
 - Bland's rule, 360, 375
 - brief history of LP, 305
 - canonical augmented matrix, 347
 - canonical tableau, 358
 - complementary slackness, 390, 539
 - cycling, 360, 375
 - degenerate basic feasible solution, 325, 328, 360
 - dual problem, 378, 403
 - duality, *see* Duality
 - duality theorem, 387
 - examples of, 100, 307–314
 - extreme point, 331, 332, 334
 - feasible solution, 325
 - fundamental theorem of LP, 328
 - fuzzy, 585
 - geometric view of, 330
 - integer linear programming, 427–444
 - interior-point method, 307, 403, 406, 411, 423
 - Karmarkar's method, *see* Karmarkar's method
 - Karush-Kuhn-Tucker condition, 395, 501, 539
 - Khachiyan's method, 306, 403–405, 418
 - Klee-Minty problem, 401
 - optimal basic feasible solution, 328
 - optimal feasible solution, 328
 - primal problem, 378, 403

- reduced cost coefficient, 353, 357, 358, 391
- revised simplex method, 364–368
- sensitivity, 398
- simplex method, 306, 339–368
- slack variable, 319
- standard form, 318, 324
- surplus variable, 319
- symmetric duality, 378, 403
- tableau, 358
- two-dimensional, 314
- two-phase affine scaling method, 409
- two-phase simplex method, 360–364
- uncertain, 584–591
- weak duality lemma, 385, 399
- Linear quadratic regulator, 474
- Linear regression, *see* Line fitting
- Linear space, *see* Vector space
- Linear transformation, 25, 63
- Linear variety, 47
- Linear-fractional LMIs, 534
- Linearly dependent, 10
- Linearly independent, 9, 176, 324, 451, 484
- Little-oh notation, 74, 85
- LMI, *see* Linear matrix inequality
- LMI solvers, 529
- LMI toolbox for MATLAB, 529, 536
- LMITOOL, 536
- Local minimizer, 82, 83, 85, 90, 92, 461, 469, 471, 484, 494, 517
- Locally convergent, 141, 554, 556
- Location parameter, 267
- Lower bound, 57
- LP, *see* Linear programming
- LQR, 474
- Lyapunov inequality, 527, 543
- MacDuffee, 241
- Markov chain, 397
- Mating pool in genetic algorithm, 286
- MATLAB, xiii, 127, 128, 136, 139, 159, 191, 215, 271, 290, 291, 299, 301, 337, 376, 424, 543
- LMI toolbox, 529, 536
- Matrix
 - affine matrix inequality, 524
 - compatible norm, 36
 - condition number, 148
 - continuous, 60
 - convergence of sequence, 59
 - definition of, 12
 - derivative, 64
 - determinant, 14
 - diagonal, 27, 407, 421, 541, 571
 - eigenvalue of, *see* Eigenvalue
 - eigenvector of, *see* Eigenvector
 - elementary, *see* Elementary matrix
 - entry of, 12
 - full-rank factorization, 236
 - function, matrix-valued, 60
 - game theory, 314
 - generalized inverse, 236, 238
 - Gram, 220
 - Hadamard product, 283
 - Hessian, 65, 83, 468, 515
 - identity, 16
 - image of, *see* Range of matrix
 - indefinite, 35
 - induced norm, 36, 480
 - inverse, 16
 - invertible, *see* Nonsingular matrix
 - Jacobian, 64
 - Jordan form, 59
 - kernel of, *see* Nullspace of matrix
 - leading principal minor of, 31
 - left pseudoinverse, 238
 - linear matrix inequality, 524, 541
 - minor of, 15, 428
 - Moore-Penrose inverse, 236, 237
 - negative definite, 35
 - negative semidefinite, 35
 - nonsingular, 16, 208, 218, 229, 325, 342
 - notation, 12
 - nullspace of, 30, 41, 227, 372, 406, 454
 - orthogonal, 29, 571
 - orthogonal projector, 30, 226, 406, 408, 423, 549
 - Penrose generalized inverse, 243
 - positive definite, 35
 - positive semidefinite, 35
 - principal minor of, 31
 - pseudoinverse, 236, 238
 - range of, 30, 41
 - rank of, 13–16
 - representation of linear transformation, 25
 - right pseudoinverse, 238
 - Schur complement, 526
 - Schur product, 283
 - sequence of, 59
 - series of, 60
 - similar, 26
 - square, 14

- stochastic, 397
- submatrix of, 432
- Sylvester's criterion, 31
- symmetric, 28, 35, 139
- totally unimodular, 432
- trace, 534
- transformation, 26
- transpose of, 12
- unimodular, 428
- Matrix norm**, 35–39
- Matrix-valued function**, 60
- Max**, 21
- Maximizer**, 82
- Mean value theorem**, 76, 554, 558
- MILP**, *see* Mixed integer linear programming
- Min**, 15, 82
- Minimax**, 582, 587, 592
- Minimizer**
 - description of, 81
 - global, 82, 94, 473, 517–519, 521, 537, 552
 - local, 82, 83, 85, 90, 461, 469, 484, 494, 517
 - Pareto, 575
 - strict global, 82
 - strict local, 82, 92, 102, 471, 494
- Minimum norm**, 231, 242, 257, 584, 592
- Minor**
 - definition of, 15, 428
 - leading principal, 31
 - principal, 31
- Minty**, 401
- Mixed integer linear programming**, 444
- Monotone sequence**, 55, 57
- Moore-Penrose inverse**, 236, 237
- Morrison**, 208, 229
- Multicriteria optimization**, 573
- Multiobjective optimization**, 573, 586
- Mutation** in genetic algorithm, 288
- Naive random search**, 279
- Natural basis**, 11
- Negative definite**
 - matrix, 35
 - quadratic form, 31
- Negative half-space**, 46
- Negative semidefinite**
 - matrix, 35
 - quadratic form, 31
- Neighborhood**, 50
- Nelder-Mead algorithm**, 274–278
 - centroid, 274
 - contraction, 276
- expansion**, 275
- Neural networks**, *see* Feedforward neural networks
- Neuron**, 253
- Newton's method**
 - convergence of, 165
 - descent direction, 167
 - descent property, 167
 - for nonlinear least squares, 168–171
 - Gauss-Newton method, 171
 - general, 161–171
 - Levenberg-Marquardt modification of, 168
 - modification of, 167
 - of tangents, 119
 - one-dimensional, 116–119
 - order of convergence, 165
- Newton-Raphson method**, *see* Newton's method
- Non-strict inequality**, 528
- Nondecreasing sequence**, 55
- Nondifferentiable optimization**, 523
- Nondifferentiable penalty function**, 566
- Nonincreasing sequence**, 55
- Nonlinear least squares**, 169
- Nonsingular matrix**, 16, 208, 218, 229, 325, 342
- Norm**
 - compatible, 36
 - Euclidean, 20
 - Frobenius, 36
 - general vector norm, 21
 - induced, 36, 480
 - matrix, 35–39
 - p -norm, 21, 584
 - properties of, 21
- Normal**, 47, 70
- Normal plane**, 458
- Normal space**, 457, 458
- Notation**, 5
- Nullspace of matrix**, 30, 41, 227, 372, 406, 454
- Objective function**, 81
- Offspring** in genetic algorithm, 287
- One-dimensional search methods**, 103–126
- Open set**, 51
- Optimal basic feasible solution**, 328
- Optimal control**, 472, 475, 481, 482, 553
- Optimal feasible solution in LP**, 328
- Optimization**
 - combinatorial, 273, 282, 297
 - constrained, 82, 305, 449

- convex, 517–536
- linear, *see* Linear programming
- multicriteria, 573
- multiobjective, 573, 586
- nondifferentiable, 523
- semidefinite, 523
- unconstrained, *see* Unconstrained optimization
- vector, 573
- with equality constraints, 449, 553
- with inequality constraints, 483, 556
- with set constraint, 82, 502
- Optimization algorithm, *see* Search methods
- Order of convergence, 148, 149, 152, 153, 157, 165
- Order symbol, 73, 149
- Orthant, 416
- Orthogonal, 70
- Orthogonal basis, 29
- Orthogonal complement, 29, 226, 458
- Orthogonal decomposition, 30, 226
- Orthogonal matrix, 29, 571
- Orthogonal projection, 30, 219, 406, 408, 423
- Orthogonal projector, 30, 226, 406, 408, 423, 549
- Orthogonal vectors, 20
- Outer product, 197
- Output layer in neural network, 254
- Overdetermined system of equations, 217
- Parents in genetic algorithm, 287
- Pareto front, 575
- Pareto minimizer, 575
- Partial derivative, 64
- Particle swarm optimization, 282–285
- Particular solution, 346
- Penalty function, 560
- Penalty method, 560–567
 - absolute value penalty function, 561
 - convergence, 565
 - Courant-Beltrami penalty function, 562, 570
 - exact penalty function, 566
 - nondifferentiable penalty function, 566
 - penalty function, 560
 - penalty parameter, 560
- Penalty parameter, 560
- Penrose, *see* Moore-Penrose inverse
- Penrose generalized inverse, 243
- Perp, *see* Orthogonal complement
- Pivot, 349, 352, 364
- Polak-Ribière formula, 187
- Polyhedron
 - carrier of, 53
 - definition of, 52
 - edge of, 53
 - face of, 53, 316
 - in linear programming, 316–318
 - vertex of, 53
- Polynomial, characteristic, 26
- Polynomial complexity, 402
- Polytope
 - definition of, 52
 - in linear programming, 316
- Population in genetic algorithm, 285, 286
- Positive definite
 - matrix, 35
 - quadratic form, 31
 - relation to eigenvalues, 35
 - Sylvester's criterion, 31
- Positive half-space, 46
- Positive orthant, 416
- Positive semidefinite
 - matrix, 35
 - quadratic form, 31
 - relation to eigenvalues, 35
 - relation to principal minors, 35
- Positivity, 19, 21
- Powell, 162, 188, 202
- Powell formula, 188
- Powell function, 162
- Primal linear program, 378, 403
- Primal nonlinear program, 542
- Primal quadratic program, 399
- Primal-dual method, 377
- Principal minor, 31
- Principle of induction, 5
- Probabilistic search, 278
- Probability vector, 397, 540
- Product
 - dyadic, 197
 - inner, 19, 21
 - outer, 197
- Product rule, 68
- Projected algorithm, 546, 556
- Projected gradient method, 407, 547, 549–553
 - stopping criterion, 552
- Projected steepest descent algorithm, 551
- Projection, 297, *see* Orthogonal projection, 546
- Projective transformation, 416, 424

- Proof**
- contradiction (*reductio ad absurdum*), 5
 - contraposition, 4
 - direct method, 4
 - methods of, 3–5
 - principle of induction, 5
- Proportional fairness**, 541
- Pseudoinverse**, 236, 238
- Pythagorean theorem**, 21
- Q -conjugate**
- definition of, 176
 - linear independence, 176
 - relation to eigenvectors, 189
 - relation to orthogonality, 189
- Quadratic convergence**, 148
- Quadratic fit**, 122, 128
- Quadratic form**
- convex, 512
 - definition of, 31
 - maximizing, 467, 471
 - negative definite, 31
 - negative semidefinite, 31
 - positive definite, 31
 - positive semidefinite, 31, 35
 - Sylvester’s criterion, 31
- Quadratic programming**, 399, 472, 481, 500
- Quasi-Newton methods**, 193–211
- approximating inverse Hessian, 194
 - BFGS algorithm, 207–211
 - complementarity, 207
 - conjugate direction property, 196
 - descent property, 193
 - DFP algorithm, 202–207
 - duality, 207
 - rank one formula, 197–202
 - rank two update, 202
 - single-rank symmetric, 197
 - symmetric Huang family, 215
 - variable metric algorithm, 202
- Randomized search**, 278
- Range of matrix**, 30, 41
- Rank of matrix**, 13–16
- Rank one formula**, 197–202
- Rank two update**, 202
- Rate of convergence**, 141, 148
- Ratio of convergence**, 148
- Rayleigh’s inequality**, 38, 92, 144, 146, 152
- Real vector space**, 7
- Recursive least-squares**, *see* RLS algorithm
- Reduced cost coefficient**, 353, 357, 358, 391
- Reductio ad absurdum**, 5
- Reeves**, 187
- Regular point**, 451, 456, 461, 464, 484, 554, 556
- Relative cost coefficient**, *see* Reduced cost coefficient
- Representation scheme in genetic algorithm**, 286
- Revised simplex method**, 364–368
- Revised tableau**, 365
- Ribière**, 187
- Riccati inequality**, 544
- Right pseudoinverse**, 238
- RLS algorithm**, 227–232, 250
- Rosenbrock’s function**, 68, 159, 173, 191, 215
- Roulette-wheel scheme**, 286
- Row operations**, 339–346
- Row vector**, 7
- Scalar**, 9
- Scale parameter**, 267
- Scheme in genetic algorithm**, 292
- length of, 294
 - order of, 294
- Schmidt**, *see* Gram-Schmidt
- Schur complement**, 526
- Schur product**, 283
- Schwarz**, *see* Cauchy-Schwarz inequality
- Schwarz’s theorem**, 65
- Scilab Consortium**, 536
- Search direction**, 124, 128, 167, 168
- Search methods**
- bisection method, 116
 - conjugate direction methods, 175–188
 - conjugate gradient algorithm, 182–188
 - constrained optimization, 545–567
 - derivative descent search, 154
 - Fibonacci, 108–115
 - general algorithm, 211
 - genetic algorithm, 285–298
 - Golden section, 104–108
 - gradient methods, 131–153
 - Kaczmarz’s algorithm, 232–236, 257
 - Lagrangian, 553–560
 - line search, 103, 124, 133, 167, 186, 188, 194, 209

- naive random search, 279
- Nelder-Mead algorithm, 274–278
- neural network training, 255
- Newton's method, 116–119, 161–171
- Newton-Raphson method, *see* Newton's method
- one-dimensional, 103–126
- particle swarm optimization, 282–285
- penalty method, 560–567
- probabilistic, 278
- projected, 546, 556
- projected gradient methods, 547, 549–553
- quasi-Newton methods, 193–211
- randomized, 278
- secant method, 120–122, 172
- simulated annealing algorithm, 278–282
- steepest descent method, 133–141
- Secant method, 120–122, 172
- Second-order necessary condition
 - equality constraints, 469
 - inequality constraints, 494
 - interior case, 90
 - set constraint, 90
- Second-order sufficient condition
 - equality constraints, 470
 - inequality constraints, 494
 - interior case, 92
 - set constraint, 102
- Selection in genetic algorithm, 286
- Semidefinite programming, 523
- Sensitivity, 398
- Sequence
 - bounded, 56, 57
 - bounded above, 57
 - bounded below, 57
 - convergent, 56
 - decreasing, 55
 - Fibonacci, 109
 - greatest lower bound, 57
 - increasing, 55
 - least upper bound, 57
 - limit, 55
 - lower bound, 57
 - monotone, 55, 57
 - nondecreasing, 55, 564
 - nonincreasing, 55
 - of matrices, 59
 - of real numbers, 55
 - order of convergence, 148, 149, 152, 153, 157, 165
- subsequence of, 58
- upper bound, 57
- Set
 - boundary of, 51
 - bounded, 51
 - closed, 51
 - compact, 51
 - constraint, 81, *see* Feasible set
 - convex, *see* Convex set
 - feasible, 81, 450
 - interior of, 51
 - minus, 5
 - notation, 5
 - open, 51
 - simplex, 274, 412
 - subset of, 5
- Set constraint, 82, 502
- Shanno, 207
- Sherman-Morrison formula, 208, 229
- Sherman-Morrison-Woodbury formula, 229
- Shift parameter, 267
- Sigmoid, 266
- Signal-to-interference ratio, 88, 98
- Similar matrices, 26
- Simplex, 274, 412
- Simplex algorithm, *see* Simplex method
- Simplex method, 306, 339–368
 - algorithm, 349–356
 - artificial problem, 361
 - artificial variables, 361
 - Bland's rule, 360, 375
 - canonical augmented matrix, 346–348
 - canonical tableau, 358
 - complexity, 402
 - cycling, 360, 375
 - exponential complexity, 402
 - integer linear programming, 427–444
 - matrix form, 356–360
 - reduced cost coefficient, 353, 357, 358, 391
 - revised simplex method, 364–368
 - revised tableau, 365
 - row operations, 339–346
 - stopping criterion, 351, 372
 - tableau, 358
 - two-phase, 360–364
 - updating augmented matrix, 348–349
 - updating canonical tableau, 358
- Simulating annealing algorithm, 278–282

- Simultaneous equations, *see* Linear equations
 Single-rank symmetric algorithm, 197
 Singular value decomposition, 571
 Slack variable, 319
 SONC, *see* Second-order necessary condition
 SOSC, *see* Second-order sufficient condition
 Span, 10
 Sphere, 425
 Square matrix, 14
 SRS algorithm, 197
 Standard form linear program, 318, 324
 Statement
 biconditional, 4
 conditional, 3
 Steepest ascent, 71
 Steepest ascent method, *see* Steepest descent method
 Steepest descent
 order of convergence, 152
 Steepest descent method, 133–141
 for constrained optimization, 551
 for quadratic, 139
 projected, 551
 Step response, 248
 Step size, 132, 146, 156, 168, 233, 261, 547, 551
 Stiefel, Eduard, 187, 188
 Stochastic matrix, 397
 Stopping criterion
 affine scaling method, 409
 conjugate gradient method, 188
 genetic algorithm, 290
 gradient method, 135
 Karmarkar's method, 415, 420
 line search, 128
 projected gradient method, 552
 simplex method, 351, 372
 Strict inequality, 528
 Strictly interior feasible point, 406, 407, 416
 Strong Wolfe condition, 126
 Structured representation of LMI, 529
 Subgradient, 515, 537
 Sublinear convergence, 148
 Submatrix, 432
 Subsequence, 58
 Subset, 5
 Subspace, 10
 Superlinear convergence, 148
 Supervised learning, 256
 Supporting hyperplane, 52, 316
 Supremum, *see* Least upper bound
 Surface, 451, 454
 Surplus variable, 319
 SVD, *see* Singular value decomposition
 Sylvester's criterion, 31
 Symmetric duality, 378, 403
 Symmetric Huang family, 215
 Symmetric matrix, 28, 35, 139
 Symmetry, 19
 Tableau in linear programming, 358
 Tangent line, 70
 Tangent plane, 70, 454
 Tangent space, 454, 456, 458
 Tangent vector, 65, 70, 454, 456
 Taylor series, 72–76, 162, 165, 551. *See also* Taylor's theorem
 Taylor's formula, 72, 75. *See also* Taylor's theorem
 Taylor's theorem, 72, 85, 90, 92, 132, 515, 516
 Temperature schedule, 281
 Termination criterion, *see* Stopping criterion
 Third-order necessary condition, 94
 Third-order sufficient condition, 94
 Threshold, 267
 Totally unimodular, 432
 Tournament scheme, 286
 Trace, 534
 Training of neural network, 253
 Training set, 255
 Transformation
 affine scaling, 407
 linear, 25, 63
 matrix, 26
 matrix representation of, 25
 projective, 416, 424
 Transportation problem, 306, 310
 Transpose
 matrix, 12
 vector, 8
 Transposition theorems, 396
 Traveling salesperson problem, 282
 Triangle inequality, 21
 Truth table, 3
 Tucker, *see* KKT condition
 Twice continuously differentiable function, 65, 515
 Twice differentiable function, 65, 454
 Two-dimensional linear program, 314
 Two-phase affine scaling method, 409
 Two-phase simplex method, 360–364

- Uncertainty range, 106
- Unconstrained optimization
 - basics of, 82
 - conditions for, 83–92
- Uniform continuity, 23
- Uniformly continuous function, 23
- Unimodal, 104
- Unimodular, 428
- Unimodular, totally 432
- Unsupervised learning, 256
- Upper bound, 57
- Utility function, 542

- Variable metric algorithm, 202
- Variety, linear, 47
- Vector
 - column, 7
 - complex, 12
 - component of, 7
 - convex combination, 48, 297, 582
 - definition of, 7
 - difference, 8
 - field, 65
 - linear combination, 10
 - linearly dependent, 10
 - linearly independent, 9, 176, 324, 451, 484
 - normal, 47
 - orthogonal, 20
 - probability, 397, 540
 - row, 7
 - tangent, 65, 70, 454, 456
 - transpose of, 8
 - zero vector, 8
- Vector field, 65
- Vector optimization, 573
- Vector space
 - basis for, 11
 - complex, 12
 - definition of, 7
 - dimension of, 11
 - real, 7
 - subspace of, 10
- Vertex, 53, 274, 406

- Weak duality lemma, 385, 399, 543
- Weierstrass theorem, 36, 51
- Weighted sum, 474, 582, 592
- Weights in neural network, 253, 254
- Widrow, 257
- Widrow-Hoff algorithm, 257, 271
- Wiener filter, 188, 245
- Wolfe condition, 125
- Woodbury, 229

- XOR, 268, 271

- YALMIP, 536
- Yet Another LMI Package (YALMIP), 536

- Zero finding, 118, 155
- Zero matrix, 22, 30, 36
- Zero vector, 8

WILEY SERIES IN DISCRETE MATHEMATICS AND OPTIMIZATION

- AARTS AND KORST • Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing
- AARTS AND LENSTRA • Local Search in Combinatorial Optimization
- ALON AND SPENCER • The Probabilistic Method, Third Edition
- ANDERSON AND NASH • Linear Programming in Infinite-Dimensional Spaces: Theory and Application
- ARLINGHAUS, ARLINGHAUS, AND HARARY • Graph Theory and Geography: An Interactive View E-Book
- AZENCOTT • Simulated Annealing: Parallelization Techniques
- BARTHÉLEMY AND GUÉNOCHE • Trees and Proximity Representations
- BAZARRA, JARVIS, AND SHERALI • Linear Programming and Network Flows
- BRUEN AND FORCINITO • Cryptography, Information Theory, and Error-Correction: A Handbook for the 21st Century
- CHANDRU AND HOOKER • Optimization Methods for Logical Inference
- CHONG AND ŽAK • An Introduction to Optimization, Fourth Edition
- COFFMAN AND LUEKER • Probabilistic Analysis of Packing and Partitioning Algorithms
- COOK, CUNNINGHAM, PULLEYBLANK, AND SCHRIJVER • Combinatorial Optimization
- DASKIN • Network and Discrete Location: Modes, Algorithms and Applications
- DINITZ AND STINSON • Contemporary Design Theory: A Collection of Surveys
- DU AND KO • Theory of Computational Complexity
- ERICKSON • Introduction to Combinatorics
- GLOVER, KLINGHAM, AND PHILLIPS • Network Models in Optimization and Their Practical Problems
- GOLSHTEIN AND TRETYAKOV • Modified Lagrangians and Monotone Maps in Optimization
- GONDTRAN AND MINOUX • Graphs and Algorithms (*Translated by S. Vajdā*)
- GRAHAM, ROTHSCILD, AND SPENCER • Ramsey Theory, Second Edition
- GROSS AND TUCKER • Topological Graph Theory
- HALL • Combinatorial Theory, Second Edition
- HOOKER • Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction
- IMRICH AND KLAVŽAR • Product Graphs: Structure and Recognition
- JANSON, LUCZAK, AND RUCINSKI • Random Graphs
- JENSEN AND TOFT • Graph Coloring Problems
- KAPLAN • Maxima and Minima with Applications: Practical Optimization and Duality
- LAWLER, LENSTRA, RINNOOY KAN, AND SHMOYS, Editors • The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization
- LAYWINE AND MULLEN • Discrete Mathematics Using Latin Squares
- LEVITIN • Perturbation Theory in Mathematical Programming Applications
- MAHMOUD • Evolution of Random Search Trees
- MAHMOUD • Sorting: A Distribution Theory
- MARTELLI • Introduction to Discrete Dynamical Systems and Chaos
- MARTELLO AND TOTH • Knapsack Problems: Algorithms and Computer Implementations
- McALOON AND TRETКОFF • Optimization and Computational Logic
- MERRIS • Combinatorics, Second Edition
- MERRIS • Graph Theory
- MINC • Nonnegative Matrices
- MINOUX • Mathematical Programming: Theory and Algorithms (*Translated by S. Vajdā*)
- MIRCHANDANI AND FRANCIS, Editors • Discrete Location Theory
- NEMHAUSER AND WOLSEY • Integer and Combinatorial Optimization
- NEMIROVSKY AND YUDIN • Problem Complexity and Method Efficiency in Optimization
(Translated by E. R. Dawson)

PACH AND AGARWAL • Combinatorial Geometry
PLESS • Introduction to the Theory of Error-Correcting Codes, Third Edition
ROOS AND VIAL • Ph. Theory and Algorithms for Linear Optimization: An Interior Point Approach
SCHEINERMAN AND ULLMAN • Fractional Graph Theory: A Rational Approach to the Theory of
Graphs
SCHIFF • Cellular Automata: A Discrete View of the World
SCHRIVVER • Theory of Linear and Integer Programming
SPALL • Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control
STIEBITZ, SCHEIDE, TOFT, AND FAVRHOLDT • Graph Edge Coloring: Vizing's Theorem and
Goldberg's Conjecture
SZPANKOWSKI • Average Case Analysis of Algorithms on Sequences
TOMESCU • Problems in Combinatorics and Graph Theory (*Translated by R. A. Melter*)
TUCKER • Applied Combinatorics, Second Edition
WOLSEY • Integer Programming
YE • Interior Point Algorithms: Theory and Analysis