3

Introduction to Linear Programming

The development of linear programming has been ranked among the most important scientific advances of the mid-20th century, and we must agree with this assessment. Its impact since just 1950 has been extraordinary. Today it is a standard tool that has saved many thousands or millions of dollars for most companies or businesses of even moderate size in the various industrialized countries of the world; and its use in other sectors of society has been spreading rapidly. A major proportion of all scientific computation on computers is devoted to the use of linear programming. Dozens of textbooks have been written about linear programming, and *published* articles describing important applications now number in the hundreds.

What is the nature of this remarkable tool, and what kinds of problems does it address? You will gain insight into this topic as you work through subsequent examples. However, a verbal summary may help provide perspective. Briefly, the most common type of application involves the general problem of allocating *limited resources* among *competing activities* in a best possible (i.e., *optimal*) way. More precisely, this problem involves selecting the level of certain activities that compete for scarce resources that are necessary to perform those activities. The choice of activity levels then dictates how much of each resource will be consumed by each activity. The variety of situations to which this description applies is diverse, indeed, ranging from the allocation of production facilities to products to the allocation of national resources to domestic needs, from portfolio selection to the selection of shipping patterns, from agricultural planning to the design of radiation therapy, and so on. However, the one common ingredient in each of these situations is the necessity for allocating resources to activities by choosing the levels of those activities.

Linear programming uses a mathematical model to describe the problem of concern. The adjective *linear* means that all the mathematical functions in this model are required to be *linear functions.* The word *programming* does not refer here to computer programming; rather, it is essentially a synonym for *planning.* Thus, linear programming involves the *planning of activities* to obtain an optimal result, i.e., a result that reaches the specified goal best (according to the mathematical model) among all feasible alternatives.

Although allocating resources to activities is the most common type of application, linear programming has numerous other important applications as well. In fact, *any* problem whose mathematical model fits the very general format for the linear programming model is a linear programming problem. Furthermore, a remarkably efficient solution pro-

cedure, called the **simplex method,** is available for solving linear programming problems of even enormous size. These are some of the reasons for the tremendous impact of linear programming in recent decades.

Because of its great importance, we devote this and the next six chapters specifically to linear programming. After this chapter introduces the general features of linear programming, Chaps. 4 and 5 focus on the simplex method. Chapter 6 discusses the further analysis of linear programming problems *after* the simplex method has been initially applied. Chapter 7 presents several widely used extensions of the simplex method and introduces an *interior-point algorithm* that sometimes can be used to solve even larger linear programming problems than the simplex method can handle. Chapters 8 and 9 consider some special types of linear programming problems whose importance warrants individual study.

You also can look forward to seeing applications of linear programming to other areas of operations research (OR) in several later chapters.

We begin this chapter by developing a miniature prototype example of a linear programming problem. This example is small enough to be solved graphically in a straightforward way. The following two sections present the general *linear programming model* and its basic assumptions. Sections 3.4 and 3.5 give some additional examples of linear programming applications, including three case studies. Section 3.6 describes how linear programming models of modest size can be conveniently displayed and solved on a spreadsheet. However, some linear programming problems encountered in practice require truly *massive* models. Section 3.7 illustrates how a massive model can arise and how it can still be formulated successfully with the help of a special modeling language such as MPL (described in this section) or LINGO (described in the appendix to this chapter).



3.1 PROTOTYPE EXAMPLE

The WYNDOR GLASS CO. produces high-quality glass products, including windows and glass doors. It has three plants. Aluminum frames and hardware are made in Plant 1, wood frames are made in Plant 2, and Plant 3 produces the glass and assembles the products.

Because of declining earnings, top management has decided to revamp the company's product line. Unprofitable products are being discontinued, releasing production capacity to launch two new products having large sales potential:

Product 1: An 8-foot glass door with aluminum framing
Product 2: A 4×6 foot double-hung wood-framed window

Product 1 requires some of the production capacity in Plants 1 and 3, but none in Plant 2. Product 2 needs only Plants 2 and 3. The marketing division has concluded that the company could sell as much of either product as could be produced by these plants. However, because both products would be competing for the same production capacity in Plant 3, it is not clear which *mix* of the two products would be *most profitable.* Therefore, an OR team has been formed to study this question.

The OR team began by having discussions with upper management to identify management's objectives for the study. These discussions led to developing the following definition of the problem:

> Determine what the *production rates* should be for the two products in order to *maximize their total profit,* subject to the restrictions imposed by the limited production capacities

available in the three plants. (Each product will be produced in batches of 20, so the *production rate* is defined as the number of batches produced per week.) *Any* combination of production rates that satisfies these restrictions is permitted, including producing none of one product and as much as possible of the other.

The OR team also identified the data that needed to be gathered:

1. Number of hours of production time available per week in each plant for these new products. (Most of the time in these plants already is committed to current products, so the available capacity for the new products is quite limited.)
2. Number of hours of production time used in each plant for each batch produced of each new product.
3. Profit per batch produced of each new product. (*Profit per batch produced* was chosen as an appropriate measure after the team concluded that the incremental profit from each additional batch produced would be roughly *constant* regardless of the total number of batches produced. Because no substantial costs will be incurred to initiate the production and marketing of these new products, the total profit from each one is approximately this *profit per batch produced* times *the number of batches produced.*)

Obtaining reasonable estimates of these quantities required enlisting the help of key personnel in various units of the company. Staff in the manufacturing division provided the data in the first category above. Developing estimates for the second category of data required some analysis by the manufacturing engineers involved in designing the production processes for the new products. By analyzing cost data from these same engineers and the marketing division, along with a pricing decision from the marketing division, the accounting department developed estimates for the third category.

Table 3.1 summarizes the data gathered.

The OR team immediately recognized that this was a linear programming problem of the classic **product mix** type, and the team next undertook the formulation of the corresponding mathematical model.

Formulation as a Linear Programming Problem

To formulate the mathematical (linear programming) model for this problem, let

x_1 = number of batches of product 1 produced per week

x_2 = number of batches of product 2 produced per week

Z = total profit per week (in thousands of dollars) from producing these two products

Thus, x_1 and x_2 are the *decision variables* for the model. Using the bottom row of Table 3.1, we obtain

$$Z = 3x_1 + 5x_2.$$

The objective is to choose the values of x_1 and x_2 so as to *maximize* $Z = 3x_1 + 5x_2$, subject to the restrictions imposed on their values by the limited production capacities available in the three plants. Table 3.1 indicates that each batch of product 1 produced per week uses 1 hour of production time per week in Plant 1, whereas only 4 hours per week are available. This restriction is expressed mathematically by the inequality $x_1 \leq 4$. Similarly, Plant 2 imposes the restriction that $2x_2 \leq 12$. The number of hours of production

TABLE 3.1 Data for the Wyndor Glass Co. problem

Plant	Production Time per Batch, Hours: Product 1	Production Time per Batch, Hours: Product 2	Production Time Available per Week, Hours
1	1	0	4
2	0	2	12
3	3	2	18
Profit per batch	$3,000	$5,000	

time used per week in Plant 3 by choosing x_1 and x_2 as the new products' production rates would be $3x_1 + 2x_2$. Therefore, the mathematical statement of the Plant 3 restriction is $3x_1 + 2x_2 \leq 18$. Finally, since production rates cannot be negative, it is necessary to restrict the decision variables to be nonnegative: $x_1 \geq 0$ and $x_2 \geq 0$.

To summarize, in the mathematical language of linear programming, the problem is to choose values of x_1 and x_2 so as to

$$\text{Maximize} \qquad Z = 3x_1 + 5x_2,$$

subject to the restrictions

$$\begin{aligned} x_1 &\leq 4 \\ 2x_2 &\leq 12 \\ 3x_1 + 2x_2 &\leq 18 \end{aligned}$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0.$$

(Notice how the layout of the coefficients of x_1 and x_2 in this linear programming model essentially duplicates the information summarized in Table 3.1.)



Graphical Solution

This very small problem has only two decision variables and therefore only two dimensions, so a graphical procedure can be used to solve it. This procedure involves constructing a two-dimensional graph with x_1 and x_2 as the axes. The first step is to identify the values of (x_1, x_2) that are permitted by the restrictions. This is done by drawing each line that borders the range of permissible values for one restriction. To begin, note that the nonnegativity restrictions $x_1 \geq 0$ and $x_2 \geq 0$ require (x_1, x_2) to lie on the *positive* side of the axes (including actually *on* either axis), i.e., in the first quadrant. Next, observe that the restriction $x_1 \leq 4$ means that (x_1, x_2) cannot lie to the right of the line $x_1 = 4$. These results are shown in Fig. 3.1, where the shaded area contains the only values of (x_1, x_2) that are still allowed.

In a similar fashion, the restriction $2x_2 \leq 12$ (or, equivalently, $x_2 \leq 6$) implies that the line $2x_2 = 12$ should be added to the boundary of the permissible region. The final restriction, $3x_1 + 2x_2 \leq 18$, requires plotting the points (x_1, x_2) such that $3x_1 + 2x_2 = 18$

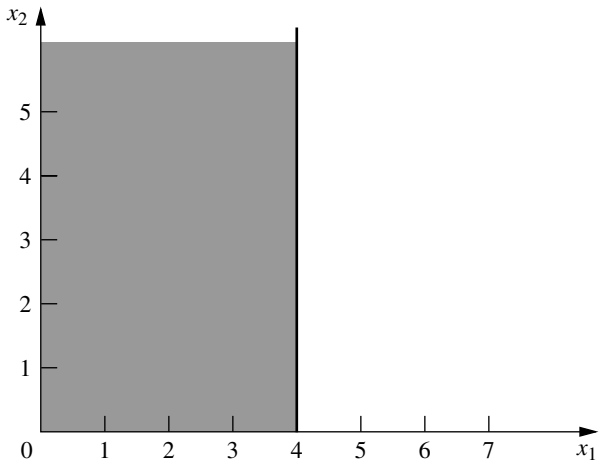
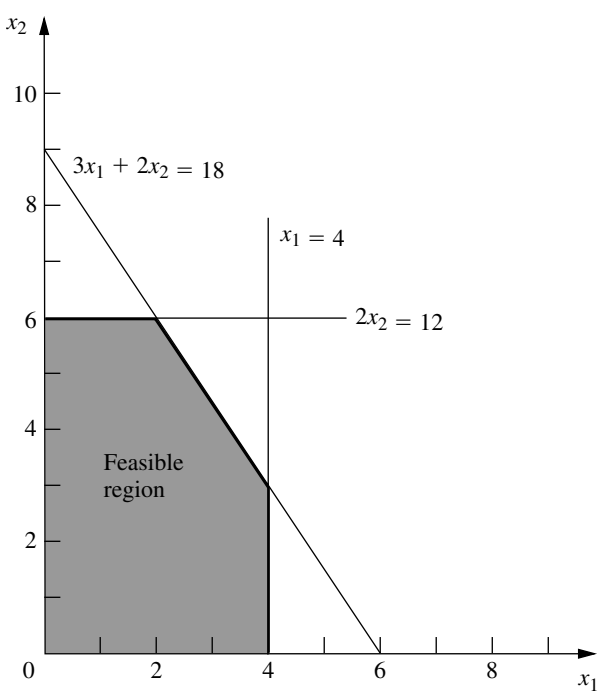


FIGURE 3.1
Shaded area shows values of (x_1, x_2) allowed by $x_1 \geq 0$, $x_2 \geq 0$, $x_1 \leq 4$.

(another line) to complete the boundary. (Note that the points such that $3x_1 + 2x_2 \leq 18$ are those that lie either underneath or on the line $3x_1 + 2x_2 = 18$, so this is the limiting line above which points do not satisfy the inequality.) The resulting region of permissible values of (x_1, x_2), called the **feasible region,** is shown in Fig. 3.2. (The demo called *Graphical Method* in your OR Tutor provides a more detailed example of constructing a feasible region.)

FIGURE 3.2
Shaded area shows the set of permissible values of (x_1, x_2), called the feasible region.



The final step is to pick out the point in this feasible region that maximizes the value of $Z = 3x_1 + 5x_2$. To discover how to perform this step efficiently, begin by trial and error. Try, for example, $Z = 10 = 3x_1 + 5x_2$ to see if there are in the permissible region any values of (x_1, x_2) that yield a value of Z as large as 10. By drawing the line $3x_1 + 5x_2 = 10$ (see Fig. 3.3), you can see that there are many points on this line that lie within the region. Having gained perspective by trying this arbitrarily chosen value of $Z = 10$, you should next try a larger arbitrary value of Z, say, $Z = 20 = 3x_1 + 5x_2$. Again, Fig. 3.3 reveals that a segment of the line $3x_1 + 5x_2 = 20$ lies within the region, so that the maximum permissible value of Z must be at least 20.

Now notice in Fig. 3.3 that the two lines just constructed are parallel. This is no coincidence, since *any* line constructed in this way has the form $Z = 3x_1 + 5x_2$ for the chosen value of Z, which implies that $5x_2 = -3x_1 + Z$ or, equivalently,

$$x_2 = -\frac{3}{5}x_1 + \frac{1}{5}Z$$

This last equation, called the **slope-intercept form** of the objective function, demonstrates that the *slope* of the line is $-\frac{3}{5}$ (since each unit increase in x_1 changes x_2 by $-\frac{3}{5}$), whereas the *intercept* of the line with the x_2 axis is $\frac{1}{5}Z$ (since $x_2 = \frac{1}{5}Z$ when $x_1 = 0$). The fact that the slope is fixed at $-\frac{3}{5}$ means that *all* lines constructed in this way are parallel.

Again, comparing the $10 = 3x_1 + 5x_2$ and $20 = 3x_1 + 5x_2$ lines in Fig. 3.3, we note that the line giving a larger value of Z ($Z = 20$) is farther up and away from the origin than the other line ($Z = 10$). This fact also is implied by the slope-intercept form of the objective function, which indicates that the intercept with the x_1 axis ($\frac{1}{5}Z$) increases when the value chosen for Z is increased.

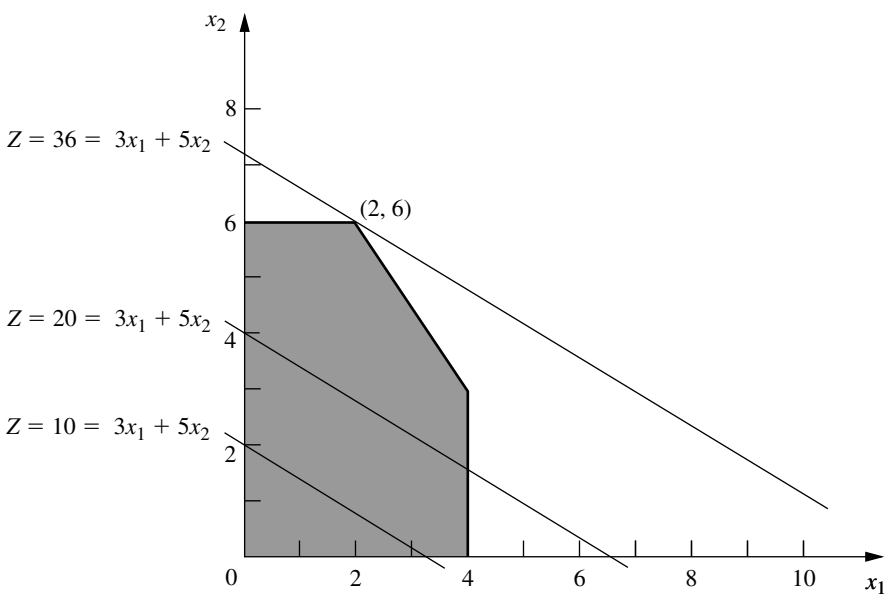


FIGURE 3.3
The value of (x_1, x_2) that maximizes $3x_1 + 5x_2$ is (2, 6).

These observations imply that our trial-and-error procedure for constructing lines in Fig. 3.3 involves nothing more than drawing a family of parallel lines containing at least one point in the feasible region and selecting the line that corresponds to the largest value of Z. Figure 3.3 shows that this line passes through the point (2, 6), indicating that the **optimal solution** is $x_1 = 2$ and $x_2 = 6$. The equation of this line is $3x_1 + 5x_2 = 3(2) + 5(6) = 36 = Z$, indicating that the optimal value of Z is $Z = 36$. The point (2, 6) lies at the intersection of the two lines $2x_2 = 12$ and $3x_1 + 2x_2 = 18$, shown in Fig. 3.2, so that this point can be calculated algebraically as the simultaneous solution of these two equations.

Having seen the trial-and-error procedure for finding the optimal point (2, 6), you now can streamline this approach for other problems. Rather than draw several parallel lines, it is sufficient to form a single line with a ruler to establish the slope. Then move the ruler with fixed slope through the feasible region in the direction of improving Z. (When the objective is to *minimize* Z, move the ruler in the direction that *decreases* Z.) Stop moving the ruler at the last instant that it still passes through a point in this region. This point is the desired *optimal solution.*

This procedure often is referred to as the **graphical method** for linear programming. It can be used to solve any linear programming problem with two decision variables. With considerable difficulty, it is possible to extend the method to three decision variables but not more than three. (The next chapter will focus on the *simplex method* for solving larger problems.)

Conclusions

The OR team used this approach to find that the optimal solution is $x_1 = 2$, $x_2 = 6$, with $Z = 36$. This solution indicates that the Wyndor Glass Co. should produce products 1 and 2 at the rate of 2 batches per week and 6 batches per week, respectively, with a resulting total profit of $36,000 per week. No other mix of the two products would be so profitable—*according to the model.*

However, we emphasized in Chap. 2 that well-conducted OR studies do not simply find *one* solution for the *initial* model formulated and then stop. All six phases described in Chap. 2 are important, including thorough testing of the model (see Sec. 2.4) and postoptimality analysis (see Sec. 2.3).

In full recognition of these practical realities, the OR team now is ready to evaluate the validity of the model more critically (to be continued in Sec. 3.3) and to perform sensitivity analysis on the effect of the estimates in Table 3.1 being different because of inaccurate estimation, changes of circumstances, etc. (to be continued in Sec. 6.7).



Continuing the Learning Process with Your OR Courseware

This is the first of many points in the book where you may find it helpful to use your *OR Courseware* in the CD-ROM that accompanies this book. A key part of this courseware is a program called OR Tutor. This program includes a complete demonstration example of the *graphical method* introduced in this section. Like the many other demonstration examples accompanying other sections of the book, this computer demonstration highlights concepts that are difficult to convey on the printed page. You may refer to Appendix 1 for documentation of the software.

When you formulate a linear programming model with more than two decision variables (so the graphical method cannot be used), the *simplex method* described in Chap. 4

enables you to still find an optimal solution immediately. Doing so also is helpful for *model validation,* since finding a *nonsensical* optimal solution signals that you have made a mistake in formulating the model.

We mentioned in Sec. 1.4 that your OR Courseware introduces you to three particularly popular commercial software packages—the Excel Solver, LINGO/LINDO, and MPL/CPLEX—for solving a variety of OR models. All three packages include the simplex method for solving linear programming models. Section 3.6 describes how to use Excel to formulate and solve linear programming models in a spreadsheet format. Descriptions of the other packages are provided in Sec. 3.7 (MPL and LINGO), Appendix 3.1 (LINGO), Sec. 4.8 (CPLEX and LINDO), and Appendix 4.1 (LINDO). In addition, your OR Courseware includes a file for each of the three packages showing how it can be used to solve each of the examples in this chapter.

3.2 THE LINEAR PROGRAMMING MODEL

The Wyndor Glass Co. problem is intended to illustrate a typical linear programming problem (miniature version). However, linear programming is too versatile to be completely characterized by a single example. In this section we discuss the general characteristics of linear programming problems, including the various legitimate forms of the mathematical model for linear programming.

Let us begin with some basic terminology and notation. The first column of Table 3.2 summarizes the components of the Wyndor Glass Co. problem. The second column then introduces more general terms for these same components that will fit many linear programming problems. The key terms are *resources* and *activities,* where m denotes the number of different kinds of resources that can be used and n denotes the number of activities being considered. Some typical resources are money and particular kinds of machines, equipment, vehicles, and personnel. Examples of activities include investing in particular projects, advertising in particular media, and shipping goods from a particular source to a particular destination. In any application of linear programming, all the activities may be of one general kind (such as any one of these three examples), and then the individual activities would be particular alternatives within this general category.

As described in the introduction to this chapter, the most common type of application of linear programming involves allocating resources to activities. The amount available of each resource is limited, so a careful allocation of resources to activities must be made. Determining this allocation involves choosing the *levels* of the activities that achieve the best possible value of the *overall measure of performance.*

TABLE 3.2 Common terminology for linear programming

Prototype Example	General Problem
Production capacities of plants 3 plants	Resources m resources
Production of products 2 products Production rate of product j, x_j	Activities n activities Level of activity j, x_j
Profit Z	Overall measure of performance Z

Certain symbols are commonly used to denote the various components of a linear programming model. These symbols are listed below, along with their interpretation for the general problem of allocating resources to activities.

Z = value of overall measure of performance.

x_j = level of activity j (for $j = 1, 2, \ldots, n$).

c_j = increase in Z that would result from each unit increase in level of activity j.

b_i = amount of resource i that is available for allocation to activities (for $i = 1, 2, \ldots, m$).

a_{ij} = amount of resource i consumed by each unit of activity j.

The model poses the problem in terms of making decisions about the levels of the activities, so $x_1, x_2, \ldots, x_n$ are called the **decision variables.** As summarized in Table 3.3, the values of c_j, b_i, and a_{ij} (for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$) are the *input constants* for the model. The c_j, b_i, and a_{ij} are also referred to as the **parameters** of the model.

Notice the correspondence between Table 3.3 and Table 3.1.

A Standard Form of the Model

Proceeding as for the Wyndor Glass Co. problem, we can now formulate the mathematical model for this general problem of allocating resources to activities. In particular, this model is to select the values for $x_1, x_2, \ldots, x_n$ so as to

$$\text{Maximize} \qquad Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

subject to the restrictions

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\le b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\le b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\le b_m, \end{aligned}$$

TABLE 3.3 Data needed for a linear programming model involving the allocation of resources to activities

	Resource Usage per Unit of Activity				
	Activity				
Resource	1	2	. . .	*n*	Amount of Resource Available
1	a_{11}	a_{12}	. . .	a_{1n}	b_1
2	a_{21}	a_{22}	. . .	a_{2n}	b_2
.					.
.	. . .	. . .	. . .	. . .	.
.					.
m	a_{m1}	a_{m2}	. . .	a_{mn}	b_m
Contribution to Z per unit of activity	c_1	c_2	. . .	c_n	

and

$$x_1 \geq 0, \qquad x_2 \geq 0, \qquad \ldots, \qquad x_n \geq 0.$$

We call this *our standard form*[1] for the linear programming problem. Any situation whose mathematical formulation fits this model is a linear programming problem.

Notice that the model for the Wyndor Glass Co. problem fits our standard form, with $m = 3$ and $n = 2$.

Common terminology for the linear programming model can now be summarized. The function being maximized, $c_1x_1 + c_2x_2 + \cdots + c_nx_n$, is called the **objective function.** The restrictions normally are referred to as **constraints.** The first m constraints (those with a *function* of all the variables $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n$ on the left-hand side) are sometimes called **functional constraints** (or *structural constraints*). Similarly, the $x_j \geq 0$ restrictions are called **nonnegativity constraints** (or *nonnegativity conditions*).

Other Forms

We now hasten to add that the preceding model does not actually fit the natural form of some linear programming problems. The other *legitimate forms* are the following:

1. Minimizing rather than maximizing the objective function:

Minimize $\quad Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n.$

2. Some functional constraints with a greater-than-or-equal-to inequality:

$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq b_i$ for some values of *i.*

3. Some functional constraints in equation form:

$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i$ for some values of *i.*

4. Deleting the nonnegativity constraints for some decision variables:

x_j unrestricted in sign for some values of *j.*

Any problem that mixes some of or all these forms with the remaining parts of the preceding model is still a linear programming problem. Our interpretation of the words *allocating limited resources among competing activities* may no longer apply very well, if at all; but regardless of the interpretation or context, all that is required is that the mathematical statement of the problem fit the allowable forms.

Terminology for Solutions of the Model

You may be used to having the term *solution* mean the final answer to a problem, but the convention in linear programming (and its extensions) is quite different. Here, *any* specification of values for the decision variables $(x_1, x_2, \ldots, x_n)$ is called a **solution,** regardless of whether it is a desirable or even an allowable choice. Different types of solutions are then identified by using an appropriate adjective.

[1]This is called *our* standard form rather than *the* standard form because some textbooks adopt other forms.

A **feasible solution** is a solution for which *all* the constraints are *satisfied.*

An **infeasible solution** is a solution for which *at least one* constraint is *violated.*

In the example, the points (2, 3) and (4, 1) in Fig. 3.2 are *feasible solutions,* while the points $(-1, 3)$ and (4, 4) are *infeasible solutions.*

The **feasible region** is the collection of all feasible solutions.

The feasible region in the example is the entire shaded area in Fig. 3.2.

It is possible for a problem to have **no feasible solutions.** This would have happened in the example if the new products had been required to return a net profit of at least \$50,000 per week to justify discontinuing part of the current product line. The corresponding constraint, $3x_1 + 5x_2 \geq 50$, would eliminate the entire feasible region, so no mix of new products would be superior to the status quo. This case is illustrated in Fig. 3.4.

Given that there are feasible solutions, the goal of linear programming is to find a best feasible solution, as measured by the value of the objective function in the model.

An **optimal solution** is a feasible solution that has the *most favorable value* of the objective function.

The **most favorable value** is the *largest value* if the objective function is to be *maximized,* whereas it is the *smallest value* if the objective function is to be *minimized.*

Most problems will have just one optimal solution. However, it is possible to have more than one. This would occur in the example if the *profit per batch produced* of product 2 were changed to \$2,000. This changes the objective function to $Z = 3x_1 + 2x_2$, so that all the points

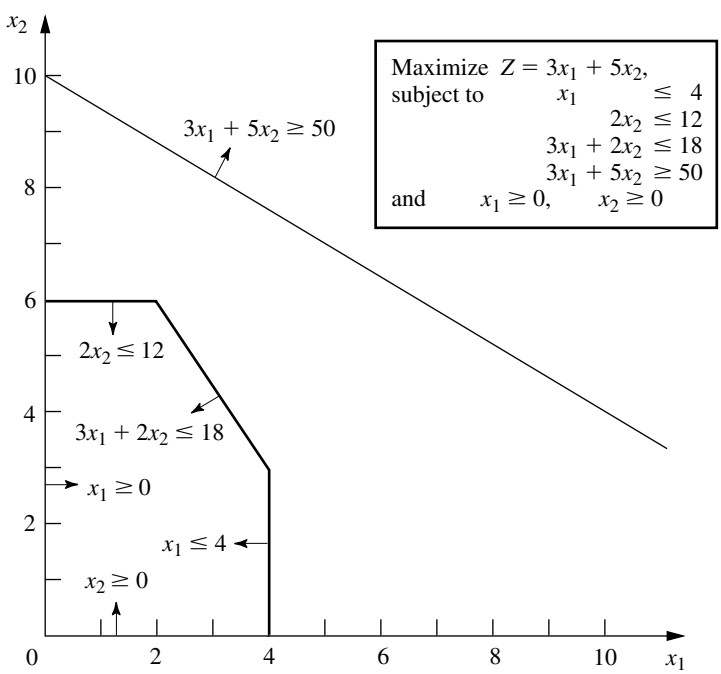


FIGURE 3.4
The Wyndor Glass Co. problem would have no feasible solutions if the constraint $3x_1 + 5x_2 \geq 50$ were added to the problem.

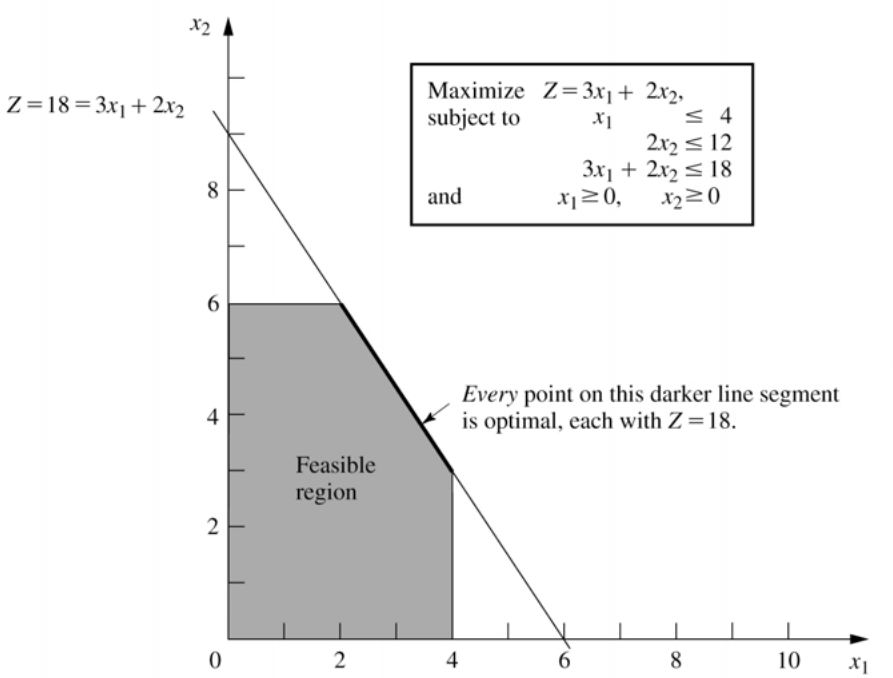


FIGURE 3.5
The Wyndor Glass Co. problem would have multiple optimal solutions if the objective function were changed to $Z = 3x_1 + 2x_2$.

on the line segment connecting (2, 6) and (4, 3) would be optimal. This case is illustrated in Fig. 3.5. As in this case, *any* problem having **multiple optimal solutions** will have an *infinite* number of them, each with the same optimal value of the objective function.

Another possibility is that a problem has **no optimal solutions.** This occurs only if (1) it has no feasible solutions or (2) the constraints do not prevent improving the value of the objective function (Z) indefinitely in the favorable direction (positive or negative). The latter case is referred to as having an **unbounded Z.** To illustrate, this case would result if the last two functional constraints were mistakenly deleted in the example, as illustrated in Fig. 3.6.

We next introduce a special type of feasible solution that plays the key role when the simplex method searches for an optimal solution.

> A **corner-point feasible (CPF) solution** is a solution that lies at a corner of the feasible region.

Figure 3.7 highlights the five CPF solutions for the example.

Sections 4.1 and 5.1 will delve into the various useful properties of CPF solutions for problems of any size, including the following relationship with optimal solutions.

> **Relationship between optimal solutions and CPF solutions:** Consider any linear programming problem with feasible solutions and a bounded feasible region. The problem must possess CPF solutions and at least one optimal solution. Furthermore, the best CPF solution *must* be an optimal solution. Thus, if a problem has exactly one optimal solution, it *must* be a CPF solution. If the problem has multiple optimal solutions, at least two *must* be CPF solutions.

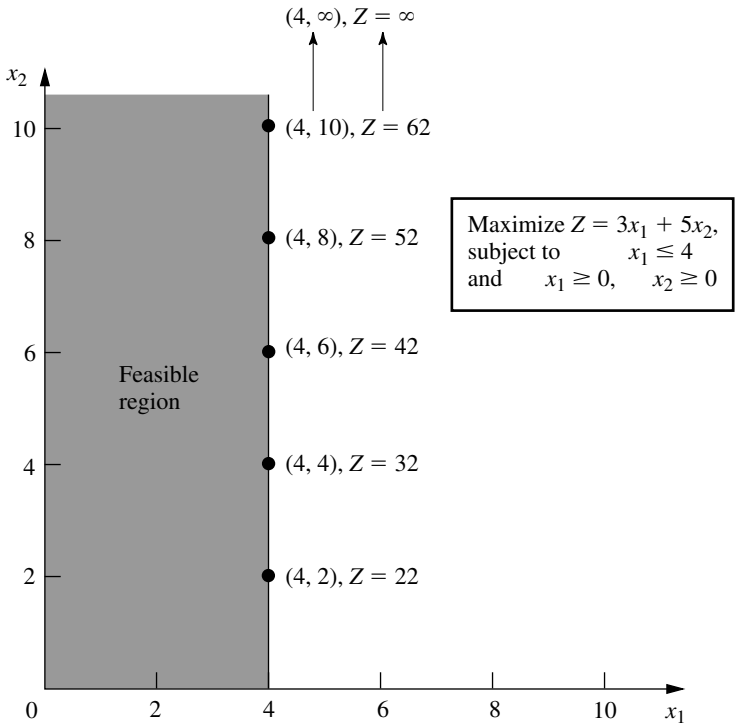


FIGURE 3.6
The Wyndor Glass Co. problem would have no optimal solutions if the only functional constraint were $x_1 \leq 4$, because x_2 then could be increased indefinitely in the feasible region without ever reaching the maximum value of $Z = 3x_1 + 5x_2$.

The example has exactly one optimal solution, $(x_1, x_2) = (2, 6)$, which is a CPF solution. (Think about how the graphical method leads to the one optimal solution being a CPF solution.) When the example is modified to yield multiple optimal solutions, as shown in Fig. 3.5, two of these optimal solutions—(2, 6) and (4, 3)—are CPF solutions.

3.3 ASSUMPTIONS OF LINEAR PROGRAMMING

All the assumptions of linear programming actually are implicit in the model formulation given in Sec. 3.2. However, it is good to highlight these assumptions so you can more easily evaluate how well linear programming applies to any given problem. Furthermore, we still need to see why the OR team for the Wyndor Glass Co. concluded that a linear programming formulation provided a satisfactory representation of the problem.

Proportionality

Proportionality is an assumption about both the objective function and the functional constraints, as summarized below.

> **Proportionality assumption:** The contribution of each activity to the *value of the objective function Z* is *proportional* to the *level of the activity* x_j, as represented by the $c_j x_j$ term in the objective function. Similarly, the contribution of each activity to the *left-hand side of each functional constraint* is *proportional* to the *level of the activity* x_j, as represented by the $a_{ij}x_j$ term in the constraint.

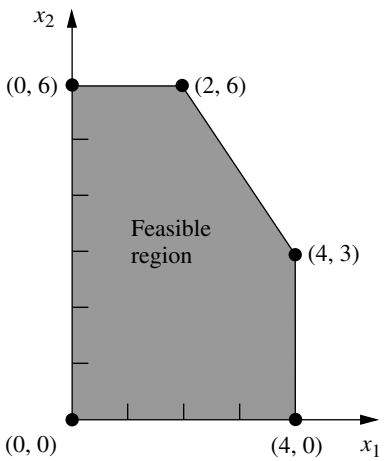


FIGURE 3.7
The five dots are the five CPF solutions for the Wyndor Glass Co. problem.

> Consequently, this assumption rules out any exponent other than 1 for any variable in any term of any function (whether the objective function or the function on the left-hand side of a functional constraint) in a linear programming model.[1]

To illustrate this assumption, consider the first term ($3x_1$) in the objective function ($Z = 3x_1 + 5x_2$) for the Wyndor Glass Co. problem. This term represents the profit generated per week (in thousands of dollars) by producing product 1 at the rate of x_1 batches per week. The *proportionality satisfied* column of Table 3.4 shows the case that was assumed in Sec. 3.1, namely, that this profit is indeed proportional to x_1 so that $3x_1$ is the appropriate term for the objective function. By contrast, the next three columns show different hypothetical cases where the proportionality assumption would be violated.

Refer first to the *Case 1* column in Table 3.4. This case would arise if there were *start-up costs* associated with initiating the production of product 1. For example, there

[1]When the function includes any *cross-product terms,* proportionality should be interpreted to mean that *changes* in the function value are proportional to *changes* in each variable (x_j) individually, given any fixed values for all the other variables. Therefore, a cross-product term satisfies proportionality as long as each variable in the term has an exponent of 1. (However, any cross-product term violates the *additivity assumption,* discussed next.)

TABLE 3.4 Examples of satisfying or violating proportionality

	Profit from Product 1 ($000 per Week)			
		Proportionality Violated		
x_1	Proportionality Satisfied	Case 1	Case 2	Case 3
0	0	0	0	0
1	3	2	3	3
2	6	5	7	5
3	9	8	12	6
4	12	11	18	6

might be costs involved with setting up the production facilities. There might also be costs associated with arranging the distribution of the new product. Because these are one-time costs, they would need to be amortized on a per-week basis to be commensurable with Z (profit in thousands of dollars per week). Suppose that this amortization were done and that the total start-up cost amounted to reducing Z by 1, but that the profit without considering the start-up cost would be $3x_1$. This would mean that the contribution from product 1 to Z should be $3x_1 - 1$ for $x_1 > 0$, whereas the contribution would be $3x_1 = 0$ when $x_1 = 0$ (no start-up cost). This profit function,[1] which is given by the solid curve in Fig. 3.8, certainly is *not* proportional to x_1.

At first glance, it might appear that *Case 2* in Table 3.4 is quite similar to Case 1. However, Case 2 actually arises in a very different way. There no longer is a start-up cost, and the profit from the first unit of product 1 per week is indeed 3, as originally assumed. However, there now is an *increasing marginal return;* i.e., the *slope* of the *profit function* for product 1 (see the solid curve in Fig. 3.9) keeps increasing as x_1 is increased. This violation of proportionality might occur because of economies of scale that can sometimes be achieved at higher levels of production, e.g., through the use of more efficient high-volume machinery, longer production runs, quantity discounts for large purchases of raw materials, and the learning-curve effect whereby workers become more efficient as they gain experience with a particular mode of production. As the incremental cost goes down, the incremental profit will go up (assuming constant marginal revenue).

[1]If the contribution from product 1 to Z were $3x_1 - 1$ for *all* $x_1 \geq 0$, including $x_1 = 0$, then the fixed constant, -1, could be deleted from the objective function without changing the optimal solution and proportionality would be restored. However, this "fix" does not work here because the -1 constant does not apply when $x_1 = 0$.

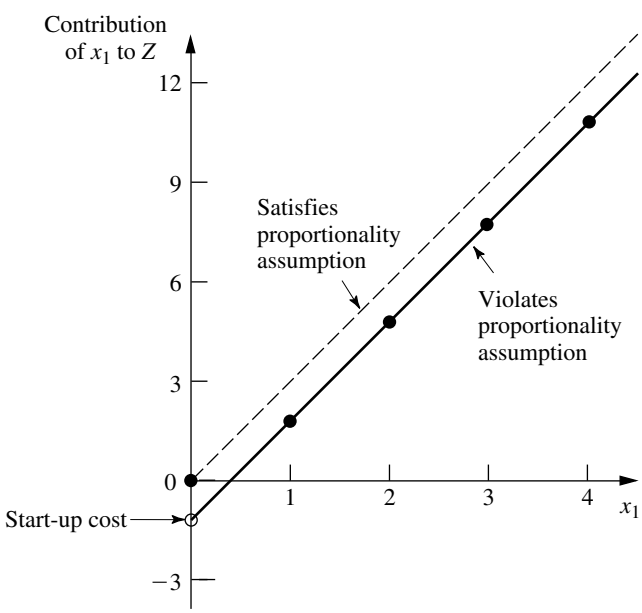


FIGURE 3.8
The solid curve violates the proportionality assumption because of the start-up cost that is incurred when x_1 is increased from 0. The values at the dots are given by the Case 1 column of Table 3.4.

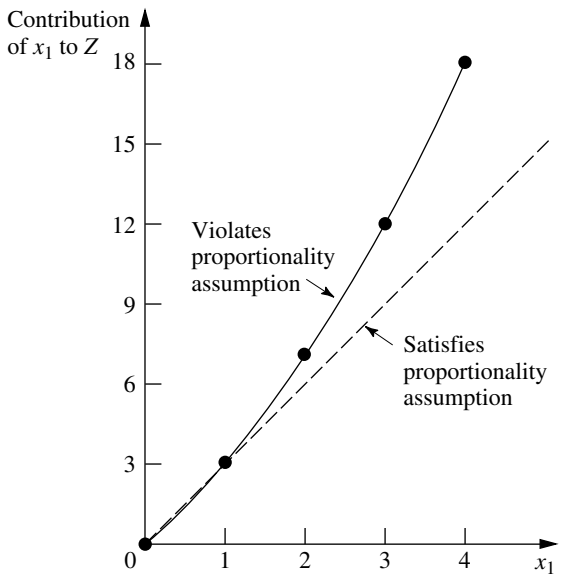


FIGURE 3.9
The solid curve violates the proportionality assumption because its slope (the *marginal return* from product 1) keeps increasing as x_1 is increased. The values at the dots are given by the Case 2 column of Table 3.4.

Referring again to Table 3.4, the reverse of Case 2 is *Case 3,* where there is a *decreasing marginal return.* In this case, the *slope* of the *profit function* for product 1 (given by the solid curve in Fig. 3.10) keeps decreasing as x_1 is increased. This violation of proportionality might occur because the *marketing costs* need to go up more than proportionally to attain increases in the level of sales. For example, it might be possible to sell product 1 at the rate of 1 per week ($x_1 = 1$) with no advertising, whereas attaining sales to sustain a production rate of $x_1 = 2$ might require a moderate amount of advertising, $x_1 = 3$ might necessitate an extensive advertising campaign, and $x_1 = 4$ might require also lowering the price.

All three cases are hypothetical examples of ways in which the proportionality assumption could be violated. What is the actual situation? The actual profit from produc-

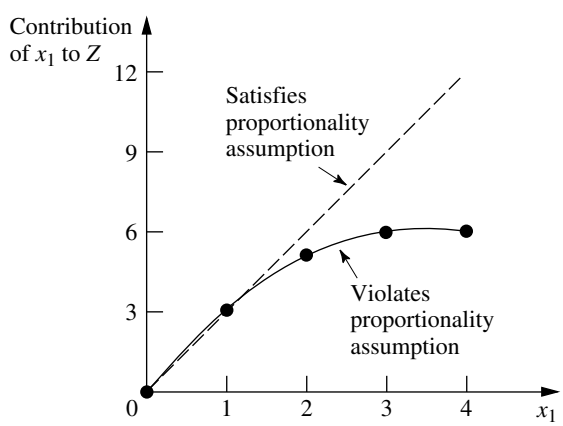


FIGURE 3.10
The solid curve violates the proportionality assumption because its slope (the *marginal return* from product 1) keeps decreasing as x_1 is increased. The values at the dots are given by the Case 3 column in Table 3.4.

ing product 1 (or any other product) is derived from the sales revenue minus various direct and indirect costs. Inevitably, some of these cost components are not strictly proportional to the production rate, perhaps for one of the reasons illustrated above. However, the real question is whether, after all the components of profit have been accumulated, proportionality is a reasonable approximation for practical modeling purposes. For the Wyndor Glass Co. problem, the OR team checked both the objective function and the functional constraints. The conclusion was that proportionality could indeed be assumed without serious distortion.

For other problems, what happens when the proportionality assumption does not hold even as a reasonable approximation? In most cases, this means you must use *nonlinear programming* instead (presented in Chap. 13). However, we do point out in Sec. 13.8 that a certain important kind of nonproportionality can still be handled by linear programming by reformulating the problem appropriately. Furthermore, if the assumption is violated only because of start-up costs, there is an extension of linear programming (*mixed integer programming*) that can be used, as discussed in Sec. 12.3 (the fixed-charge problem).

Additivity

Although the proportionality assumption rules out exponents other than 1, it does not prohibit *cross-product terms* (terms involving the product of two or more variables). The additivity assumption does rule out this latter possibility, as summarized below.

> **Additivity assumption:** *Every* function in a linear programming model (whether the objective function or the function on the left-hand side of a functional constraint) is the *sum* of the *individual contributions* of the respective activities.

To make this definition more concrete and clarify why we need to worry about this assumption, let us look at some examples. Table 3.5 shows some possible cases for the objective function for the Wyndor Glass Co. problem. In each case, the *individual contributions* from the products are just as assumed in Sec. 3.1, namely, $3x_1$ for product 1 and $5x_2$ for product 2. The difference lies in the last row, which gives the *function value* for Z when the two products are produced jointly. The *additivity satisfied* column shows the case where this *function value* is obtained simply by adding the first two rows ($3 + 5 = 8$), so that $Z = 3x_1 + 5x_2$ as previously assumed. By contrast, the next two columns show hypothetical cases where the additivity assumption would be violated (but not the proportionality assumption).

TABLE 3.5 Examples of satisfying or violating additivity for the objective function

	Value of Z		
		Additivity Violated	
(x_1, x_2)	Additivity Satisfied	Case 1	Case 2
(1, 0)	3	3	3
(0, 1)	5	5	5
(1, 1)	8	9	7

Referring to the *Case 1* column of Table 3.5, this case corresponds to an objective function of $Z = 3x_1 + 5x_2 + x_1x_2$, so that $Z = 3 + 5 + 1 = 9$ for $(x_1, x_2) = (1, 1)$, thereby violating the additivity assumption that $Z = 3 + 5$. (The proportionality assumption still is satisfied since after the value of one variable is fixed, the increment in Z from the other variable is proportional to the value of that variable.) This case would arise if the two products were *complementary* in some way that *increases* profit. For example, suppose that a major advertising campaign would be required to market either new product produced by itself, but that the same single campaign can effectively promote both products if the decision is made to produce both. Because a major cost is saved for the second product, their joint profit is somewhat more than the *sum* of their individual profits when each is produced by itself.

Case 2 in Table 3.5 also violates the additivity assumption because of the extra term in the corresponding objective function, $Z = 3x_1 + 5x_2 - x_1x_2$, so that $Z = 3 + 5 - 1 = 7$ for $(x_1, x_2) = (1, 1)$. As the reverse of the first case, Case 2 would arise if the two products were *competitive* in some way that *decreased* their joint profit. For example, suppose that both products need to use the same machinery and equipment. If either product were produced by itself, this machinery and equipment would be dedicated to this one use. However, producing both products would require switching the production processes back and forth, with substantial time and cost involved in temporarily shutting down the production of one product and setting up for the other. Because of this major extra cost, their joint profit is somewhat less than the *sum* of their individual profits when each is produced by itself.

The same kinds of interaction between activities can affect the additivity of the constraint functions. For example, consider the third functional constraint of the Wyndor Glass Co. problem: $3x_1 + 2x_2 \leq 18$. (This is the only constraint involving both products.) This constraint concerns the production capacity of Plant 3, where 18 hours of production time per week is available for the two new products, and the function on the left-hand side $(3x_1 + 2x_2)$ represents the number of hours of production time per week that would be used by these products. The *additivity satisfied* column of Table 3.6 shows this case as is, whereas the next two columns display cases where the function has an extra cross-product term that violates additivity. For all three columns, the *individual contributions* from the products toward using the capacity of Plant 3 are just as assumed previously, namely, $3x_1$ for product 1 and $2x_2$ for product 2, or $3(2) = 6$ for $x_1 = 2$ and $2(3) = 6$ for

TABLE 3.6 Examples of satisfying or violating additivity for a functional constraint

	Amount of Resource Used		
		Additivity Violated	
(x_1, x_2)	Additivity Satisfied	Case 3	Case 4
(2, 0)	6	6	6
(0, 3)	6	6	6
(2, 3)	12	15	10.8

$x_2 = 3$. As was true for Table 3.5, the difference lies in the last row, which now gives the *total function value* for production time used when the two products are produced jointly.

For Case 3 (see Table 3.6), the production time used by the two products is given by the function $3x_1 + 2x_2 + 0.5x_1x_2$, so the *total function value* is $6 + 6 + 3 = 15$ when $(x_1, x_2) = (2, 3)$, which violates the additivity assumption that the value is just $6 + 6 = 12$. This case can arise in exactly the same way as described for Case 2 in Table 3.5; namely, extra time is wasted switching the production processes back and forth between the two products. The extra cross-product term ($0.5x_1x_2$) would give the production time wasted in this way. (Note that wasting time switching between products leads to a positive cross-product term here, where the total function is measuring production time used, whereas it led to a negative cross-product term for Case 2 because the total function there measures profit.)

For Case 4 in Table 3.6, the function for production time used is $3x_1 + 2x_2 - 0.1x_1^2x_2$, so the *function value* for $(x_1, x_2) = (2, 3)$ is $6 + 6 - 1.2 = 10.8$. This case could arise in the following way. As in Case 3, suppose that the two products require the same type of machinery and equipment. But suppose now that the time required to switch from one product to the other would be relatively small. Because each product goes through a sequence of production operations, individual production facilities normally dedicated to that product would incur occasional idle periods. During these otherwise idle periods, these facilities can be used by the other product. Consequently, the total production time used (including idle periods) when the two products are produced jointly would be less than the *sum* of the production times used by the individual products when each is produced by itself.

After analyzing the possible kinds of interaction between the two products illustrated by these four cases, the OR team concluded that none played a major role in the actual Wyndor Glass Co. problem. Therefore, the additivity assumption was adopted as a reasonable approximation.

For other problems, if additivity is not a reasonable assumption, so that some of or all the mathematical functions of the model need to be *nonlinear* (because of the cross-product terms), you definitely enter the realm of nonlinear programming (Chap. 13).

Divisibility

Our next assumption concerns the values allowed for the decision variables.

> **Divisibility assumption:** Decision variables in a linear programming model are allowed to have *any* values, including *noninteger* values, that satisfy the functional and nonnegativity constraints. Thus, these variables are *not* restricted to just integer values. Since each decision variable represents the level of some activity, it is being assumed that the activities can be run at *fractional levels.*

For the Wyndor Glass Co. problem, the decision variables represent production rates (the number of batches of a product produced per week). Since these production rates can have *any* fractional values within the feasible region, the divisibility assumption does hold.

In certain situations, the divisibility assumption does not hold because some of or all the decision variables must be restricted to *integer values.* Mathematical models with this restriction are called *integer programming* models, and they are discussed in Chap. 12.

Certainty

Our last assumption concerns the *parameters* of the model, namely, the coefficients in the objective function c_j, the coefficients in the functional constraints a_{ij}, and the right-hand sides of the functional constraints b_i.

> **Certainty assumption:** The value assigned to each parameter of a linear programming model is assumed to be a *known constant.*

In real applications, the certainty assumption is seldom satisfied precisely. Linear programming models usually are formulated to select some future course of action. Therefore, the parameter values used would be based on a prediction of future conditions, which inevitably introduces some degree of uncertainty.

For this reason it is usually important to conduct **sensitivity analysis** after a solution is found that is optimal under the assumed parameter values. As discussed in Sec. 2.3, one purpose is to identify the *sensitive* parameters (those whose value cannot be changed without changing the optimal solution), since any later change in the value of a sensitive parameter immediately signals a need to change the solution being used.

Sensitivity analysis plays an important role in the analysis of the Wyndor Glass Co. problem, as you will see in Sec. 6.7. However, it is necessary to acquire some more background before we finish that story.

Occasionally, the degree of uncertainty in the parameters is too great to be amenable to sensitivity analysis. In this case, it is necessary to treat the parameters explicitly as *random variables.* Formulations of this kind have been developed, as discussed in Secs. 23.6 and 23.7 on the book's web site, www.mhhe.com/hillier.

The Assumptions in Perspective

We emphasized in Sec. 2.2 that a mathematical model is intended to be only an idealized representation of the real problem. Approximations and simplifying assumptions generally are required in order for the model to be tractable. Adding too much detail and precision can make the model too unwieldy for useful analysis of the problem. All that is really needed is that there be a reasonably high correlation between the prediction of the model and what would actually happen in the real problem.

This advice certainly is applicable to linear programming. It is very common in real applications of linear programming that almost *none* of the four assumptions hold completely. Except perhaps for the *divisibility assumption,* minor disparities are to be expected. This is especially true for the *certainty assumption,* so sensitivity analysis normally is a must to compensate for the violation of this assumption.

However, it is important for the OR team to examine the four assumptions for the problem under study and to analyze just how large the disparities are. If any of the assumptions are violated in a major way, then a number of useful alternative models are available, as presented in later chapters of the book. A disadvantage of these other models is that the algorithms available for solving them are not nearly as powerful as those for linear programming, but this gap has been closing in some cases. For some applications, the powerful linear programming approach is used for the initial analysis, and then a more complicated model is used to refine this analysis.

As you work through the examples in the next section, you will find it good practice to analyze how well each of the four assumptions of linear programming applies.


MPL

3.4 ADDITIONAL EXAMPLES

The Wyndor Glass Co. problem is a prototype example of linear programming in several respects: It involves allocating limited resources among competing activities, its model fits our standard form, and its context is the traditional one of improved business planning. However, the applicability of linear programming is much wider. In this section we begin broadening our horizons. As you study the following examples, note that it is their underlying mathematical model rather than their context that characterizes them as linear programming problems. Then give some thought to how the same mathematical model could arise in many other contexts by merely changing the names of the activities and so forth.

These examples are scaled-down versions of actual applications (including two that are included in the case studies presented in the next section).

Design of Radiation Therapy

MARY has just been diagnosed as having a cancer at a fairly advanced stage. Specifically, she has a large malignant tumor in the bladder area (a "whole bladder lesion").

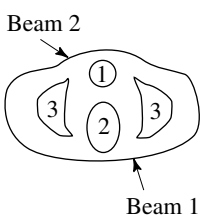
Mary is to receive the most advanced medical care available to give her every possible chance for survival. This care will include extensive *radiation therapy.*

Radiation therapy involves using an external beam treatment machine to pass ionizing radiation through the patient's body, damaging both cancerous and healthy tissues. Normally, several beams are precisely administered from different angles in a two-dimensional plane. Due to attenuation, each beam delivers more radiation to the tissue near the entry point than to the tissue near the exit point. Scatter also causes some delivery of radiation to tissue outside the direct path of the beam. Because tumor cells are typically microscopically interspersed among healthy cells, the radiation dosage throughout the tumor region must be large enough to kill the malignant cells, which are slightly more radiosensitive, yet small enough to spare the healthy cells. At the same time, the aggregate dose to critical tissues must not exceed established tolerance levels, in order to prevent complications that can be more serious than the disease itself. For the same reason, the total dose to the entire healthy anatomy must be minimized.

Because of the need to carefully balance all these factors, the design of radiation therapy is a very delicate process. The goal of the design is to select the combination of beams to be used, and the intensity of each one, to generate the best possible dose distribution. (The dose strength at any point in the body is measured in units called *kilorads.*) Once the treatment design has been developed, it is administered in many installments, spread over several weeks.

In Mary's case, the size and location of her tumor make the design of her treatment an even more delicate process than usual. Figure 3.11 shows a diagram of a cross section of the tumor viewed from above, as well as nearby critical tissues to avoid. These tissues include critical organs (e.g., the rectum) as well as bony structures (e.g., the femurs and pelvis) that will attenuate the radiation. Also shown are the entry point and direction for the only two beams that can be used with any modicum of safety in this case. (Actually,

FIGURE 3.11
Cross section of Mary's tumor (viewed from above), nearby critical tissues, and the radiation beams being used.



1. Bladder and tumor
2. Rectum, coccyx, etc.
3. Femur, part of pelvis, etc.

we are simplifying the example at this point, because normally dozens of possible beams must be considered.)

For any proposed beam of given intensity, the analysis of what the resulting radiation absorption by various parts of the body would be requires a complicated process. In brief, based on careful anatomical analysis, the energy distribution within the two-dimensional cross section of the tissue can be plotted on an isodose map, where the contour lines represent the dose strength as a percentage of the dose strength at the entry point. A fine grid then is placed over the isodose map. By summing the radiation absorbed in the squares containing each type of tissue, the average dose that is absorbed by the tumor, healthy anatomy, and critical tissues can be calculated. With more than one beam (administered sequentially), the radiation absorption is additive.

After thorough analysis of this type, the medical team has carefully estimated the data needed to design Mary's treatment, as summarized in Table 3.7. The first column lists the areas of the body that must be considered, and then the next two columns give the fraction of the radiation dose at the entry point for each beam that is absorbed by the respective areas on average. For example, if the dose level at the entry point for beam 1 is 1 kilorad, then an average of 0.4 kilorad will be absorbed by the entire healthy anatomy in the two-dimensional plane, an average of 0.3 kilorad will be absorbed by nearby critical tissues, an average of 0.5 kilorad will be absorbed by the various parts of the tumor, and 0.6 kilorad will be absorbed by the center of the tumor. The last column gives the restrictions on the total dosage from both beams that is absorbed on average by the respective areas of the body. In particular, the average dosage absorption for the healthy anatomy must be *as small as possible,* the critical tissues must *not exceed* 2.7 kilorads, the average over the entire tumor must *equal* 6 kilorads, and the center of the tumor must be *at least* 6 kilorads.

Formulation as a Linear Programming Problem. The two decision variables x_1 and x_2 represent the dose (in kilorads) at the entry point for beam 1 and beam 2, respectively. Because the total dosage reaching the healthy anatomy is to be minimized, let Z denote this quantity. The data from Table 3.7 can then be used directly to formulate the following linear programming model.[1]

[1]Actually, Table 3.7 simplifies the real situation, so the real model would be somewhat more complicated than this one and would have dozens of variables and constraints. For details about the general situation, see D. Sonderman and P. G. Abrahamson, "Radiotherapy Treatment Design Using Mathematical Programming Models," *Operations Research,* **33:**705–725, 1985, and its ref. 1.

TABLE 3.7 Data for the design of Mary's radiation therapy

Area	Fraction of Entry Dose Absorbed by Area (Average)		Restriction on Total Average Dosage, Kilorads
	Beam 1	Beam 2	
Healthy anatomy	0.4	0.5	Minimize
Critical tissues	0.3	0.1	≤ 2.7
Tumor region	0.5	0.5	$= 6$
Center of tumor	0.6	0.4	≥ 6

$$\text{Minimize} \qquad Z = 0.4x_1 + 0.5x_2,$$

subject to

$$\begin{aligned} 0.3x_1 + 0.1x_2 &\leq 2.7 \\ 0.5x_1 + 0.5x_2 &= 6 \\ 0.6x_1 + 0.4x_2 &\geq 6 \end{aligned}$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0.$$

Notice the differences between this model and the one in Sec. 3.1 for the Wyndor Glass Co. problem. The latter model involved *maximizing Z,* and all the functional constraints were in $\leq$ form. This new model does not fit this same standard form, but it does incorporate three other *legitimate* forms described in Sec. 3.2, namely, *minimizing Z,* functional constraints in $=$ form, and functional constraints in $\geq$ form.

However, both models have only two variables, so this new problem also can be solved by the *graphical method* illustrated in Sec. 3.1. Figure 3.12 shows the graphical solution. The *feasible region* consists of just the dark line segment between (6, 6) and (7.5, 4.5), because the points on this segment are the only ones that simultaneously satisfy all the constraints. (Note that the equality constraint limits the feasible region to the line containing this line segment, and then the other two functional constraints determine the two endpoints of the line segment.) The dashed line is the objective function line that passes through the optimal solution $(x_1, x_2) = (7.5, 4.5)$ with $Z = 5.25$. This solution is optimal rather than the point (6, 6) because *decreasing Z* (for positive values of Z) pushes the objective function line toward the origin (where $Z = 0$). And $Z = 5.25$ for (7.5, 4.5) is less than $Z = 5.4$ for (6, 6).

Thus, the optimal design is to use a total dose at the entry point of 7.5 kilorads for beam 1 and 4.5 kilorads for beam 2.



MPL

Regional Planning

The SOUTHERN CONFEDERATION OF KIBBUTZIM is a group of three kibbutzim (communal farming communities) in Israel. Overall planning for this group is done in its Coordinating Technical Office. This office currently is planning agricultural production for the coming year.

The agricultural output of each kibbutz is limited by both the amount of available irrigable land and the quantity of water allocated for irrigation by the Water Commissioner (a national government official). These data are given in Table 3.8.

TABLE 3.8 Resource data for the Southern Confederation of Kibbutzim

Kibbutz	Usable Land (Acres)	Water Allocation (Acre Feet)
1	400	600
2	600	800
3	300	375

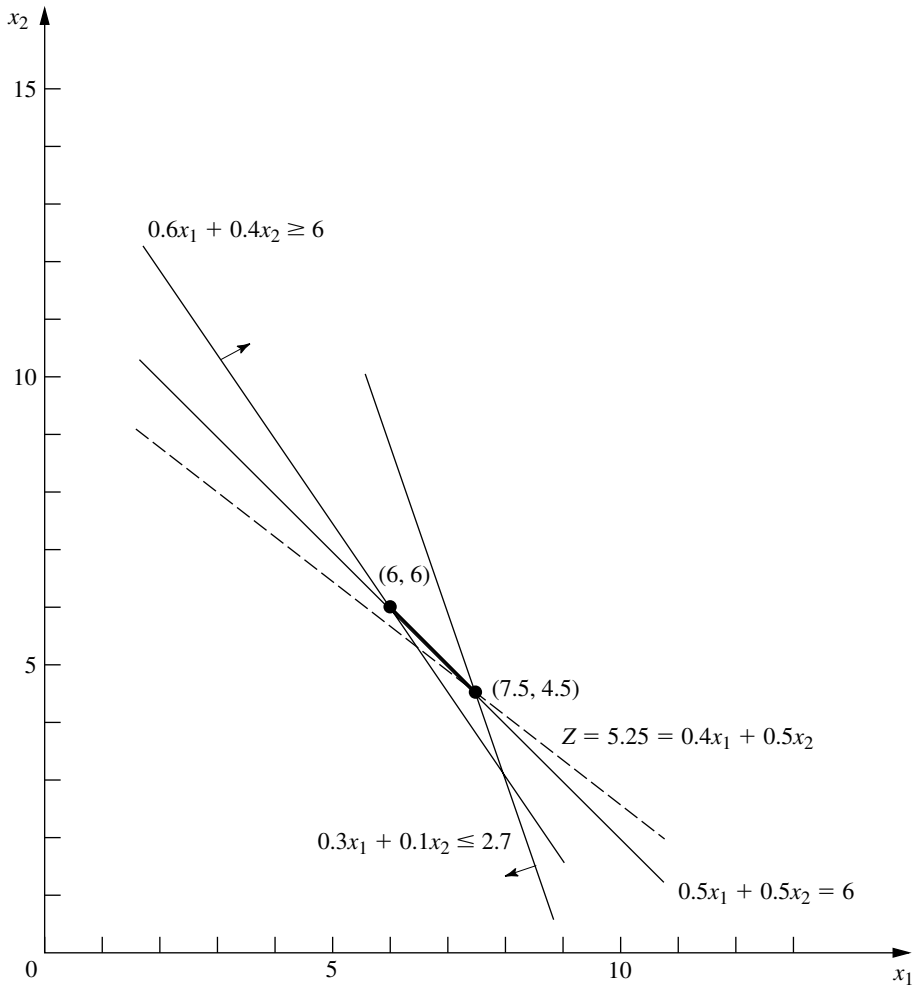


FIGURE 3.12
Graphical solution for the design of Mary's radiation therapy.

The crops suited for this region include sugar beets, cotton, and sorghum, and these are the three being considered for the upcoming season. These crops differ primarily in their expected net return per acre and their consumption of water. In addition, the Ministry of Agriculture has set a maximum quota for the total acreage that can be devoted to each of these crops by the Southern Confederation of Kibbutzim, as shown in Table 3.9.

TABLE 3.9 Crop data for the Southern Confederation of Kibbutzim

Crop	Maximum Quota (Acres)	Water Consumption (Acre Feet/Acre)	Net Return ($/Acre)
Sugar beets	600	3	1,000
Cotton	500	2	750
Sorghum	325	1	250

Because of the limited water available for irrigation, the Southern Confederation of Kibbutzim will not be able to use all its irrigable land for planting crops in the upcoming season. To ensure equity between the three kibbutzim, it has been agreed that every kibbutz will plant the same proportion of its available irrigable land. For example, if kibbutz 1 plants 200 of its available 400 acres, then kibbutz 2 must plant 300 of its 600 acres, while kibbutz 3 plants 150 acres of its 300 acres. However, any combination of the crops may be grown at any of the kibbutzim. The job facing the Coordinating Technical Office is to plan how many acres to devote to each crop at the respective kibbutzim while satisfying the given restrictions. The objective is to maximize the total net return to the Southern Confederation of Kibbutzim as a whole.

Formulation as a Linear Programming Problem. The quantities to be decided upon are the number of acres to devote to each of the three crops at each of the three kibbutzim. The decision variables x_j ($j = 1, 2, \ldots, 9$) represent these nine quantities, as shown in Table 3.10.

Since the measure of effectiveness Z is the total net return, the resulting linear programming model for this problem is

$$\text{Maximize} \qquad Z = 1{,}000(x_1 + x_2 + x_3) + 750(x_4 + x_5 + x_6) + 250(x_7 + x_8 + x_9),$$

subject to the following constraints:

1. Usable land for each kibbutz:

$$\begin{aligned} x_1 + x_4 + x_7 &\le 400 \\ x_2 + x_5 + x_8 &\le 600 \\ x_3 + x_6 + x_9 &\le 300 \end{aligned}$$

2. Water allocation for each kibbutz:

$$\begin{aligned} 3x_1 + 2x_4 + x_7 &\le 600 \\ 3x_2 + 2x_5 + x_8 &\le 800 \\ 3x_3 + 2x_6 + x_9 &\le 375 \end{aligned}$$

3. Total acreage for each crop:

$$\begin{aligned} x_1 + x_2 + x_3 &\le 600 \\ x_4 + x_5 + x_6 &\le 500 \\ x_7 + x_8 + x_9 &\le 325 \end{aligned}$$

TABLE 3.10 Decision variables for the Southern Confederation of Kibbutzim problem

	Allocation (Acres)		
	Kibbutz		
Crop	**1**	**2**	**3**
Sugar beets	x_1	x_2	x_3
Cotton	x_4	x_5	x_6
Sorghum	x_7	x_8	x_9

4. Equal proportion of land planted:

$$\frac{x_1 + x_4 + x_7}{400} = \frac{x_2 + x_5 + x_8}{600}$$

$$\frac{x_2 + x_5 + x_8}{600} = \frac{x_3 + x_6 + x_9}{300}$$

$$\frac{x_3 + x_6 + x_9}{300} = \frac{x_1 + x_4 + x_7}{400}$$

5. Nonnegativity:

$$x_j \geq 0, \qquad \text{for } j = 1, 2, \ldots, 9.$$

This completes the model, except that the equality constraints are not yet in an appropriate form for a linear programming model because some of the variables are on the right-hand side. Hence, their final form[1] is

$$\begin{aligned} 3(x_1 + x_4 + x_7) - 2(x_2 + x_5 + x_8) &= 0 \\ (x_2 + x_5 + x_8) - 2(x_3 + x_6 + x_9) &= 0 \\ 4(x_3 + x_6 + x_9) - 3(x_1 + x_4 + x_7) &= 0 \end{aligned}$$

The Coordinating Technical Office formulated this model and then applied the simplex method (developed in the next chapter) to find an optimal solution

$$(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) = \left(133\frac{1}{3}, 100, 25, 100, 250, 150, 0, 0, 0\right),$$

as shown in Table 3.11. The resulting optimal value of the objective function is $Z = 633{,}333\frac{1}{3}$, that is, a total net return of \$633,333.33.

[1]Actually, any one of these equations is redundant and can be deleted if desired. Also, because of these equations, any two of the usable land constraints also could be deleted because they automatically would be satisfied when both the remaining usable land constraint and these equations are satisfied. However, no harm is done (except a little more computational effort) by including unnecessary constraints, so you don't need to worry about identifying and deleting them in models you formulate.

TABLE 3.11 Optimal solution for the Southern Confederation of Kibbutzim problem

	Best Allocation (Acres)		
	Kibbutz		
Crop	**1**	**2**	**3**
Sugar beets	$133\frac{1}{3}$	100	25
Cotton	100	250	150
Sorghum	0	0	0



MPL

Controlling Air Pollution

The NORI & LEETS CO., one of the major producers of steel in its part of the world, is located in the city of Steeltown and is the only large employer there. Steeltown has grown and prospered along with the company, which now employs nearly 50,000 residents. Therefore, the attitude of the townspeople always has been, "What's good for Nori & Leets is good for the town." However, this attitude is now changing; uncontrolled air pollution from the company's furnaces is ruining the appearance of the city and endangering the health of its residents.

A recent stockholders' revolt resulted in the election of a new enlightened board of directors for the company. These directors are determined to follow socially responsible policies, and they have been discussing with Steeltown city officials and citizens' groups what to do about the air pollution problem. Together they have worked out stringent air quality standards for the Steeltown airshed.

The three main types of pollutants in this airshed are particulate matter, sulfur oxides, and hydrocarbons. The new standards require that the company reduce its annual emission of these pollutants by the amounts shown in Table 3.12. The board of directors has instructed management to have the engineering staff determine how to achieve these reductions in the most economical way.

The steelworks has two primary sources of pollution, namely, the blast furnaces for making pig iron and the open-hearth furnaces for changing iron into steel. In both cases the engineers have decided that the most effective types of abatement methods are (1) increasing the height of the smokestacks,[1] (2) using filter devices (including gas traps) in the smokestacks, and (3) including cleaner, high-grade materials among the fuels for the furnaces. Each of these methods has a technological limit on how heavily it can be used (e.g., a maximum feasible increase in the height of the smokestacks), but there also is considerable flexibility for using the method at a fraction of its technological limit.

Table 3.13 shows how much emission (in millions of pounds per year) can be eliminated from each type of furnace by fully using any abatement method to its technological limit. For purposes of analysis, it is assumed that each method also can be used less fully to achieve any fraction of the emission-rate reductions shown in this table. Furthermore, the fractions can be different for blast furnaces and for open-hearth furnaces. For either type of furnace, the emission reduction achieved by each method is not substantially affected by whether the other methods also are used.

[1]Subsequent to this study, this particular abatement method has become a controversial one. Because its effect is to reduce ground-level pollution by spreading emissions over a greater distance, environmental groups contend that this creates more acid rain by keeping sulfur oxides in the air longer. Consequently, the U.S. Environmental Protection Agency adopted new rules in 1985 to remove incentives for using tall smokestacks.

TABLE 3.12 Clean air standards for the Nori & Leets Co.

Pollutant	Required Reduction in Annual Emission Rate (Million Pounds)
Particulates	60
Sulfur oxides	150
Hydrocarbons	125

TABLE 3.13 Reduction in emission rate (in millions of pounds per year) from the maximum feasible use of an abatement method for Nori & Leets Co.

	Taller Smokestacks		Filters		Better Fuels	
Pollutant	Blast Furnaces	Open-Hearth Furnaces	Blast Furnaces	Open-Hearth Furnaces	Blast Furnaces	Open-Hearth Furnaces
Particulates	12	9	25	20	17	13
Sulfur oxides	35	42	18	31	56	49
Hydrocarbons	37	53	28	24	29	20

After these data were developed, it became clear that no single method by itself could achieve all the required reductions. On the other hand, combining all three methods at full capacity on both types of furnaces (which would be prohibitively expensive if the company's products are to remain competitively priced) is much more than adequate. Therefore, the engineers concluded that they would have to use some combination of the methods, perhaps with fractional capacities, based upon the relative costs. Furthermore, because of the differences between the blast and the open-hearth furnaces, the two types probably should not use the same combination.

An analysis was conducted to estimate the total annual cost that would be incurred by each abatement method. A method's annual cost includes increased operating and maintenance expenses as well as reduced revenue due to any loss in the efficiency of the production process caused by using the method. The other major cost is the *start-up cost* (the initial capital outlay) required to install the method. To make this one-time cost commensurable with the ongoing annual costs, the time value of money was used to calculate the annual expenditure (over the expected life of the method) that would be equivalent in value to this start-up cost.

This analysis led to the total annual cost estimates (in millions of dollars) given in Table 3.14 for using the methods at their full abatement capacities. It also was determined that the cost of a method being used at a lower level is roughly proportional to the fraction of the abatement capacity given in Table 3.13 that is achieved. Thus, for any given fraction achieved, the total annual cost would be roughly that fraction of the corresponding quantity in Table 3.14.

The stage now was set to develop the general framework of the company's plan for pollution abatement. This plan specifies which types of abatement methods will be used and at what fractions of their abatement capacities for (1) the blast furnaces and (2) the open-hearth furnaces. Because of the combinatorial nature of the problem of finding a

TABLE 3.14 Total annual cost from the maximum feasible use of an abatement method for Nori & Leets Co. ($ millions)

Abatement Method	Blast Furnaces	Open-Hearth Furnaces
Taller smokestacks	8	10
Filters	7	6
Better fuels	11	9

plan that satisfies the requirements with the smallest possible cost, an OR team was formed to solve the problem. The team adopted a linear programming approach, formulating the model summarized next.

Formulation as a Linear Programming Problem. This problem has six decision variables x_j, $j = 1, 2, \ldots, 6$, each representing the use of one of the three abatement methods for one of the two types of furnaces, expressed as a *fraction of the abatement capacity* (so x_j cannot exceed 1). The ordering of these variables is shown in Table 3.15. Because the objective is to minimize total cost while satisfying the emission reduction requirements, the data in Tables 3.12, 3.13, and 3.14 yield the following model:

$$\text{Minimize} \qquad Z = 8x_1 + 10x_2 + 7x_3 + 6x_4 + 11x_5 + 9x_6,$$

subject to the following constraints:

1. Emission reduction:

$$\begin{aligned} 12x_1 + 9x_2 + 25x_3 + 20x_4 + 17x_5 + 13x_6 &\geq 60 \\ 35x_1 + 42x_2 + 18x_3 + 31x_4 + 56x_5 + 49x_6 &\geq 150 \\ 37x_1 + 53x_2 + 28x_3 + 24x_4 + 29x_5 + 20x_6 &\geq 125 \end{aligned}$$

2. Technological limit:

$$x_j \leq 1, \qquad \text{for } j = 1, 2, \ldots, 6$$

3. Nonnegativity:

$$x_j \geq 0, \qquad \text{for } j = 1, 2, \ldots, 6.$$

The OR team used this model[1] to find a minimum-cost plan

$$(x_1, x_2, x_3, x_4, x_5, x_6) = (1, 0.623, 0.343, 1, 0.048, 1),$$

with $Z = 32.16$ (total annual cost of \$32.16 million). Sensitivity analysis then was conducted to explore the effect of making possible adjustments in the air standards given in Table 3.12, as well as to check on the effect of any inaccuracies in the cost data given in Table 3.14. (This story is continued in Case 6.1 at the end of Chap. 6.) Next came detailed planning and managerial review. Soon after, this program for controlling air pollution was fully implemented by the company, and the citizens of Steeltown breathed deep (cleaner) sighs of relief.

[1]An equivalent formulation can express each decision variable in natural units for its abatement method; for example, x_1 and x_2 could represent the number of *feet* that the heights of the smokestacks are increased.

TABLE 3.15 Decision variables (fraction of the maximum feasible use of an abatement method) for Nori & Leets Co.

Abatement Method	Blast Furnaces	Open-Hearth Furnaces
Taller smokestacks	x_1	x_2
Filters	x_3	x_4
Better fuels	x_5	x_6



MPL

Reclaiming Solid Wastes

The SAVE-IT COMPANY operates a reclamation center that collects four types of solid waste materials and treats them so that they can be amalgamated into a salable product. (Treating and amalgamating are separate processes.) Three different grades of this product can be made (see the first column of Table 3.16), depending upon the mix of the materials used. Although there is some flexibility in the mix for each grade, quality standards may specify the minimum or maximum amount allowed for the proportion of a material in the product grade. (This proportion is the weight of the material expressed as a percentage of the total weight for the product grade.) For each of the two higher grades, a fixed percentage is specified for one of the materials. These specifications are given in Table 3.16 along with the cost of amalgamation and the selling price for each grade.

The reclamation center collects its solid waste materials from regular sources and so is normally able to maintain a steady rate for treating them. Table 3.17 gives the quantities available for collection and treatment each week, as well as the cost of treatment, for each type of material.

The Save-It Co. is solely owned by Green Earth, an organization devoted to dealing with environmental issues, so Save-It's profits are used to help support Green Earth's activities. Green Earth has raised contributions and grants, amounting to $30,000 per week, to be used exclusively to cover the entire treatment cost for the solid waste materials. The board of directors of Green Earth has instructed the management of Save-It to divide this money among the materials in such a way that *at least half* of the amount available of each material is actually collected and treated. These additional restrictions are listed in Table 3.17.

Within the restrictions specified in Tables 3.16 and 3.17, management wants to determine the *amount* of each product grade to produce *and* the exact *mix* of materials to be used for each grade. The objective is to maximize the net weekly profit (total sales income *minus* total amalgamation cost), exclusive of the fixed treatment cost of $30,000 per week that is being covered by gifts and grants.

Formulation as a Linear Programming Problem. Before attempting to construct a linear programming model, we must give careful consideration to the proper definition of the decision variables. Although this definition is often obvious, it sometimes becomes

TABLE 3.16 Product data for Save-It Co.

Grade	Specification	Amalgamation Cost per Pound ($)	Selling Price per Pound ($)
A	Material 1: Not more than 30% of total Material 2: Not less than 40% of total Material 3: Not more than 50% of total Material 4: Exactly 20% of total	3.00	8.50
B	Material 1: Not more than 50% of total Material 2: Not less than 10% of total Material 4: Exactly 10% of total	2.50	7.00
C	Material 1: Not more than 70% of total	2.00	5.50

TABLE 3.17 Solid waste materials data for the Save-It Co.

Material	Pounds per Week Available	Treatment Cost per Pound ($)	Additional Restrictions
1	3,000	3.00	1. For each material, at least half of the pounds per week available should be collected and treated.
2	2,000	6.00	
3	4,000	4.00	
4	1,000	5.00	2. $30,000 per week should be used to treat these materials.

the crux of the entire formulation. After clearly identifying what information is really desired and the most convenient form for conveying this information by means of decision variables, we can develop the objective function and the constraints on the values of these decision variables.

In this particular problem, the decisions to be made are well defined, but the appropriate means of conveying this information may require some thought. (Try it and see if you first obtain the following *inappropriate* choice of decision variables.)

Because one set of decisions is the *amount* of each product grade to produce, it would seem natural to define one set of decision variables accordingly. Proceeding tentatively along this line, we define

y_i = number of pounds of product grade i produced per week $\quad (i = A, B, C)$.

The other set of decisions is the *mix* of materials for each product grade. This mix is identified by the proportion of each material in the product grade, which would suggest defining the other set of decision variables as

z_{ij} = proportion of material j in product grade i $\quad (i = A, B, C; j = 1, 2, 3, 4)$.

However, Table 3.17 gives both the treatment cost and the availability of the materials by *quantity* (pounds) rather than *proportion,* so it is this *quantity* information that needs to be recorded in some of the constraints. For material j ($j = 1, 2, 3, 4$),

Number of pounds of material j used per week $= z_{Aj}y_A + z_{Bj}y_B + z_{Cj}y_C$.

For example, since Table 3.17 indicates that 3,000 pounds of material 1 is available per week, one constraint in the model would be

$$z_{A1}y_A + z_{B1}y_B + z_{C1}y_C \leq 3{,}000.$$

Unfortunately, this is not a legitimate linear programming constraint. The expression on the left-hand side is *not* a linear function because it involves products of variables. Therefore, a linear programming model cannot be constructed with these decision variables.

Fortunately, there is another way of defining the decision variables that will fit the linear programming format. (Do you see how to do it?) It is accomplished by merely replacing each *product* of the old decision variables by a single variable! In other words, define

$x_{ij} = z_{ij}y_i \quad$ (for $i = A, B, C; j = 1, 2, 3, 4$)
$\quad$ = number of pounds of material j allocated to product grade i per week,

and then we let the x_{ij} be the decision variables. Combining the x_{ij} in different ways yields the following quantities needed in the model (for $i = A, B, C$; $j = 1, 2, 3, 4$).

$$x_{i1} + x_{i2} + x_{i3} + x_{i4} = \text{number of pounds of product grade } i \text{ produced per week.}$$
$$x_{Aj} + x_{Bj} + x_{Cj} = \text{number of pounds of material } j \text{ used per week.}$$
$$\frac{x_{ij}}{x_{i1} + x_{i2} + x_{i3} + x_{i4}} = \text{proportion of material } j \text{ in product grade } i.$$

The fact that this last expression is a *nonlinear* function does not cause a complication. For example, consider the first specification for product grade *A* in Table 3.16 (the proportion of material 1 should not exceed 30 percent). This restriction gives the nonlinear constraint

$$\frac{x_{A1}}{x_{A1} + x_{A2} + x_{A3} + x_{A4}} \leq 0.3.$$

However, multiplying through both sides of this inequality by the denominator yields an *equivalent* constraint

$$x_{A1} \leq 0.3(x_{A1} + x_{A2} + x_{A3} + x_{A4}),$$

so

$$0.7x_{A1} - 0.3x_{A2} - 0.3x_{A3} - 0.3x_{A4} \leq 0,$$

which is a legitimate linear programming constraint.

With this adjustment, the three quantities given above lead directly to all the functional constraints of the model. The objective function is based on management's objective of maximizing net weekly profit (total sales income *minus* total amalgamation cost) from the three product grades. Thus, for each product grade, the profit per pound is obtained by subtracting the amalgamation cost given in the third column of Table 3.16 from the selling price in the fourth column. These *differences* provide the coefficients for the objective function.

Therefore, the complete linear programming model is

$$\text{Maximize} \quad Z = 5.5(x_{A1} + x_{A2} + x_{A3} + x_{A4}) + 4.5(x_{B1} + x_{B2} + x_{B3} + x_{B4}) + 3.5(x_{C1} + x_{C2} + x_{C3} + x_{C4}),$$

subject to the following constraints:

1. Mixture specifications (second column of Table 3.16):

$$x_{A1} \leq 0.3(x_{A1} + x_{A2} + x_{A3} + x_{A4}) \quad \text{(grade } A\text{, material 1)}$$
$$x_{A2} \geq 0.4(x_{A1} + x_{A2} + x_{A3} + x_{A4}) \quad \text{(grade } A\text{, material 2)}$$
$$x_{A3} \leq 0.5(x_{A1} + x_{A2} + x_{A3} + x_{A4}) \quad \text{(grade } A\text{, material 3)}$$
$$x_{A4} = 0.2(x_{A1} + x_{A2} + x_{A3} + x_{A4}) \quad \text{(grade } A\text{, material 4).}$$
$$x_{B1} \leq 0.5(x_{B1} + x_{B2} + x_{B3} + x_{B4}) \quad \text{(grade } B\text{, material 1)}$$
$$x_{B2} \geq 0.1(x_{B1} + x_{B2} + x_{B3} + x_{B4}) \quad \text{(grade } B\text{, material 2)}$$
$$x_{B4} = 0.1(x_{B1} + x_{B2} + x_{B3} + x_{B4}) \quad \text{(grade } B\text{, material 4).}$$
$$x_{C1} \leq 0.7(x_{C1} + x_{C2} + x_{C3} + x_{C4}) \quad \text{(grade } C\text{, material 1).}$$

2. Availability of materials (second column of Table 3.17):

$$\begin{aligned} x_{A1} + x_{B1} + x_{C1} &\leq 3{,}000 \quad \text{(material 1)} \\ x_{A2} + x_{B2} + x_{C2} &\leq 2{,}000 \quad \text{(material 2)} \\ x_{A3} + x_{B3} + x_{C3} &\leq 4{,}000 \quad \text{(material 3)} \\ x_{A4} + x_{B4} + x_{C4} &\leq 1{,}000 \quad \text{(material 4).} \end{aligned}$$

3. Restrictions on amounts treated (right side of Table 3.17):

$$\begin{aligned} x_{A1} + x_{B1} + x_{C1} &\geq 1{,}500 \quad \text{(material 1)} \\ x_{A2} + x_{B2} + x_{C2} &\geq 1{,}000 \quad \text{(material 2)} \\ x_{A3} + x_{B3} + x_{C3} &\geq 2{,}000 \quad \text{(material 3)} \\ x_{A4} + x_{B4} + x_{C4} &\geq \quad 500 \quad \text{(material 4).} \end{aligned}$$

4. Restriction on treatment cost (right side of Table 3.17):

$$\begin{aligned} 3(x_{A1} + x_{B1} + x_{C1}) + 6(x_{A2} + x_{B2} + x_{C2}) + 4(x_{A3} + x_{B3} + x_{C3}) \\ + 5(x_{A4} + x_{B4} + x_{C4}) = 30{,}000. \end{aligned}$$

5. Nonnegativity constraints:

$$x_{A1} \geq 0, \qquad x_{A2} \geq 0, \qquad \ldots, \qquad x_{C4} \geq 0.$$

This formulation completes the model, except that the constraints for the mixture specifications need to be rewritten in the proper form for a linear programming model by bringing all variables to the left-hand side and combining terms, as follows:

Mixture specifications:

$$\begin{aligned} 0.7x_{A1} - 0.3x_{A2} - 0.3x_{A3} - 0.3x_{A4} &\leq 0 \quad \text{(grade } A\text{, material 1)} \\ -0.4x_{A1} + 0.6x_{A2} - 0.4x_{A3} - 0.4x_{A4} &\geq 0 \quad \text{(grade } A\text{, material 2)} \\ -0.5x_{A1} - 0.5x_{A2} + 0.5x_{A3} - 0.5x_{A4} &\leq 0 \quad \text{(grade } A\text{, material 3)} \\ -0.2x_{A1} - 0.2x_{A2} - 0.2x_{A3} + 0.8x_{A4} &= 0 \quad \text{(grade } A\text{, material 4).} \\ 0.5x_{B1} - 0.5x_{B2} - 0.5x_{B3} - 0.5x_{B4} &\leq 0 \quad \text{(grade } B\text{, material 1)} \\ -0.1x_{B1} + 0.9x_{B2} - 0.1x_{B3} - 0.1x_{B4} &\geq 0 \quad \text{(grade } B\text{, material 2)} \\ -0.1x_{B1} - 0.1x_{B2} - 0.1x_{B3} + 0.9x_{B4} &= 0 \quad \text{(grade } B\text{, material 4).} \\ 0.3x_{C1} - 0.7x_{C2} - 0.7x_{C3} - 0.7x_{C4} &\leq 0 \quad \text{(grade } C\text{, material 1).} \end{aligned}$$

An optimal solution for this model is shown in Table 3.18, and then these x_{ij} values are used to calculate the other quantities of interest given in the table. The resulting optimal value of the objective function is $Z = 35{,}108.90$ (a total weekly profit of \$35,108.90).

The Save-It Co. problem is an example of a **blending problem.** The objective for a blending problem is to find the best blend of ingredients into final products to meet certain specifications. Some of the earliest applications of linear programming were for *gasoline blending,* where petroleum ingredients were blended to obtain various grades of gasoline. The award-winning OR study at Texaco discussed at the end of

TABLE 3.18 Optimal solution for the Save-It Co. problem

Grade	Pounds Used per Week: Material 1	2	3	4	Number of Pounds Produced per Week
A	412.3 (19.2%)	859.6 (40%)	447.4 (20.8%)	429.8 (20%)	2149
B	2587.7 (50%)	517.5 (10%)	1552.6 (30%)	517.5 (10%)	5175
C	0	0	0	0	0
Total	3000	1377	2000	947	

Sec. 2.5 dealt with gasoline blending (although Texaco used a *nonlinear* programming model). Other blending problems involve such final products as steel, fertilizer, and animal feed.



MPL

Personnel Scheduling

UNION AIRWAYS is adding more flights to and from its hub airport, and so it needs to hire additional customer service agents. However, it is not clear just how many more should be hired. Management recognizes the need for cost control while also consistently providing a satisfactory level of service to customers. Therefore, an OR team is studying how to schedule the agents to provide satisfactory service with the smallest personnel cost.

Based on the new schedule of flights, an analysis has been made of the *minimum* number of customer service agents that need to be on duty at different times of the day to provide a satisfactory level of service. The rightmost column of Table 3.19 shows the number of agents needed for the time periods given in the first column. The other entries

TABLE 3.19 Data for the Union Airways personnel scheduling problem

Time Period	Time Periods Covered: Shift 1	2	3	4	5	Minimum Number of Agents Needed
6:00 A.M. to 8:00 A.M.	✔					48
8:00 A.M. to 10:00 A.M.	✔	✔				79
10:00 A.M. to noon	✔	✔				65
Noon to 2:00 P.M.	✔	✔	✔			87
2:00 P.M. to 4:00 P.M.		✔	✔			64
4:00 P.M. to 6:00 P.M.			✔	✔		73
6:00 P.M. to 8:00 P.M.			✔	✔		82
8:00 P.M. to 10:00 P.M.				✔		43
10:00 P.M. to midnight				✔	✔	52
Midnight to 6:00 A.M.					✔	15
Daily cost per agent	$170	$160	$175	$180	$195	

in this table reflect one of the provisions in the company's current contract with the union that represents the customer service agents. The provision is that each agent work an 8-hour shift 5 days per week, and the authorized shifts are

Shift 1: 6:00 A.M. to 2:00 P.M.
Shift 2: 8:00 A.M. to 4:00 P.M.
Shift 3: Noon to 8:00 P.M.
Shift 4: 4:00 P.M. to midnight
Shift 5: 10:00 P.M. to 6:00 A.M.

Checkmarks in the main body of Table 3.19 show the hours covered by the respective shifts. Because some shifts are less desirable than others, the wages specified in the contract differ by shift. For each shift, the daily compensation (including benefits) for each agent is shown in the bottom row. The problem is to determine how many agents should be assigned to the respective shifts each day to minimize the *total* personnel cost for agents, based on this bottom row, while meeting (or surpassing) the service requirements given in the rightmost column.

Formulation as a Linear Programming Problem. Linear programming problems always involve finding the best *mix of activity levels.* The key to formulating this particular problem is to recognize the nature of the activities.

Activities correspond to shifts, where the *level* of each activity is the number of agents assigned to that shift. Thus, this problem involves finding the *best mix of shift sizes.* Since the decision variables always are the levels of the activities, the five decision variables here are

x_j = number of agents assigned to shift j, for $j = 1, 2, 3, 4, 5$.

The main restrictions on the values of these decision variables are that the number of agents working during each time period must satisfy the minimum requirement given in the rightmost column of Table 3.19. For example, for 2:00 P.M. to 4:00 P.M., the total number of agents assigned to the shifts that cover this time period (shifts 2 and 3) must be at least 64, so

$$x_2 + x_3 \geq 64$$

is the functional constraint for this time period.

Because the objective is to minimize the total cost of the agents assigned to the five shifts, the coefficients in the objective function are given by the last row of Table 3.19.

Therefore, the complete linear programming model is

$$\text{Minimize} \quad Z = 170x_1 + 160x_2 + 175x_3 + 180x_4 + 195x_5,$$

subject to

$$\begin{array}{lll}
x_1 & \geq 48 & \text{(6–8 A.M.)} \\
x_1 + x_2 & \geq 79 & \text{(8–10 A.M.)} \\
x_1 + x_2 & \geq 65 & \text{(10 A.M. to noon)} \\
x_1 + x_2 + x_3 & \geq 87 & \text{(Noon–2 P.M.)} \\
x_2 + x_3 & \geq 64 & \text{(2–4 P.M.)} \\
x_3 + x_4 & \geq 73 & \text{(4–6 P.M.)}
\end{array}$$

$$
\begin{aligned}
x_3 + x_4 &\geq 82 \quad \text{(6–8 P.M.)} \\
x_4 &\geq 43 \quad \text{(8–10 P.M.)} \\
x_4 + x_5 &\geq 52 \quad \text{(10 P.M.–midnight)} \\
x_5 &\geq 15 \quad \text{(Midnight–6 A.M.)}
\end{aligned}
$$

and

$$x_j \geq 0, \qquad \text{for } j = 1, 2, 3, 4, 5.$$

With a keen eye, you might have noticed that the third constraint, $x_1 + x_2 \geq 65$, actually is not necessary because the second constraint, $x_1 + x_2 \geq 79$, ensures that $x_1 + x_2$ will be larger than 65. Thus, $x_1 + x_2 \geq 65$ is a *redundant* constraint that can be deleted. Similarly, the sixth constraint, $x_3 + x_4 \geq 73$, also is a *redundant* constraint because the seventh constraint is $x_3 + x_4 \geq 82$. (In fact, three of the nonnegativity constraints—$x_1 \geq 0$, $x_4 \geq 0$, $x_5 \geq 0$—also are redundant constraints because of the first, eighth, and tenth functional constraints: $x_1 \geq 48$, $x_4 \geq 43$, and $x_5 \geq 15$. However, no computational advantage is gained by deleting these three nonnegativity constraints.)

The optimal solution for this model is $(x_1, x_2, x_3, x_4, x_5) = (48, 31, 39, 43, 15)$. This yields $Z = 30{,}610$, that is, a total daily personnel cost of \$30,610.

This problem is an example where the divisibility assumption of linear programming actually is not satisfied. The number of agents assigned to each shift needs to be an integer. Strictly speaking, the model should have an additional constraint for each decision variable specifying that the variable must have an integer value. Adding these constraints would convert the linear programming model to an integer programming model (the topic of Chap. 12).

Without these constraints, the optimal solution given above turned out to have integer values anyway, so no harm was done by not including the constraints. (The form of the functional constraints made this outcome a likely one.) If some of the variables had turned out to be noninteger, the easiest approach would have been to *round up* to integer values. (Rounding up is feasible for this example because all the functional constraints are in $\geq$ form with nonnegative coefficients.) Rounding up does not ensure obtaining an optimal solution for the integer programming model, but the error introduced by rounding up such large numbers would be negligible for most practical situations. Alternatively, integer programming techniques described in Chap. 12 could be used to solve exactly for an optimal solution with integer values.

Section 3.5 includes a case study of how United Airlines used linear programming to develop a personnel scheduling system on a vastly larger scale than this example.



MPL

Distributing Goods through a Distribution Network

The Problem. The DISTRIBUTION UNLIMITED CO. will be producing the same new product at two different factories, and then the product must be shipped to two warehouses, where either factory can supply either warehouse. The distribution network available for shipping this product is shown in Fig. 3.13, where F1 and F2 are the two factories, W1 and W2 are the two warehouses, and DC is a distribution center. The amounts to be shipped from F1 and F2 are shown to their left, and the amounts to be received at W1 and W2 are shown to their right. Each arrow represents a feasible shipping lane. Thus, F1 can ship directly to W1 and has three possible routes (F1 $\rightarrow$ DC $\rightarrow$ W2, F1 $\rightarrow$ F2 $\rightarrow$

DC → W2, and F1 → W1 → W2) for shipping to W2. Factory F2 has just one route to W2 (F2 → DC → W2) and one to W1 (F2 → DC → W2 → W1). The cost per unit shipped through each shipping lane is shown next to the arrow. Also shown next to F1 → F2 and DC → W2 are the maximum amounts that can be shipped through these lanes. The other lanes have sufficient shipping capacity to handle everything these factories can send.

The decision to be made concerns how much to ship through each shipping lane. The objective is to minimize the total shipping cost.

Formulation as a Linear Programming Problem. With seven shipping lanes, we need seven decision variables ($x_{F1\text{-}F2}$, $x_{F1\text{-}DC}$, $x_{F1\text{-}W1}$, $x_{F2\text{-}DC}$, $x_{DC\text{-}W2}$, $x_{W1\text{-}W2}$, $x_{W2\text{-}W1}$) to represent the amounts shipped through the respective lanes.

There are several restrictions on the values of these variables. In addition to the usual nonnegativity constraints, there are two *upper-bound constraints,* $x_{F1\text{-}F2} \leq 10$ and $x_{DC\text{-}W2} \leq 80$, imposed by the limited shipping capacities for the two lanes, F1 → F2 and DC → W2. All the other restrictions arise from five *net flow constraints,* one for each of the five locations. These constraints have the following form.

Net flow constraint for each location:

Amount shipped out − amount shipped in = required amount.

As indicated in Fig. 3.13, these required amounts are 50 for F1, 40 for F2, −30 for W1, and −60 for W2.

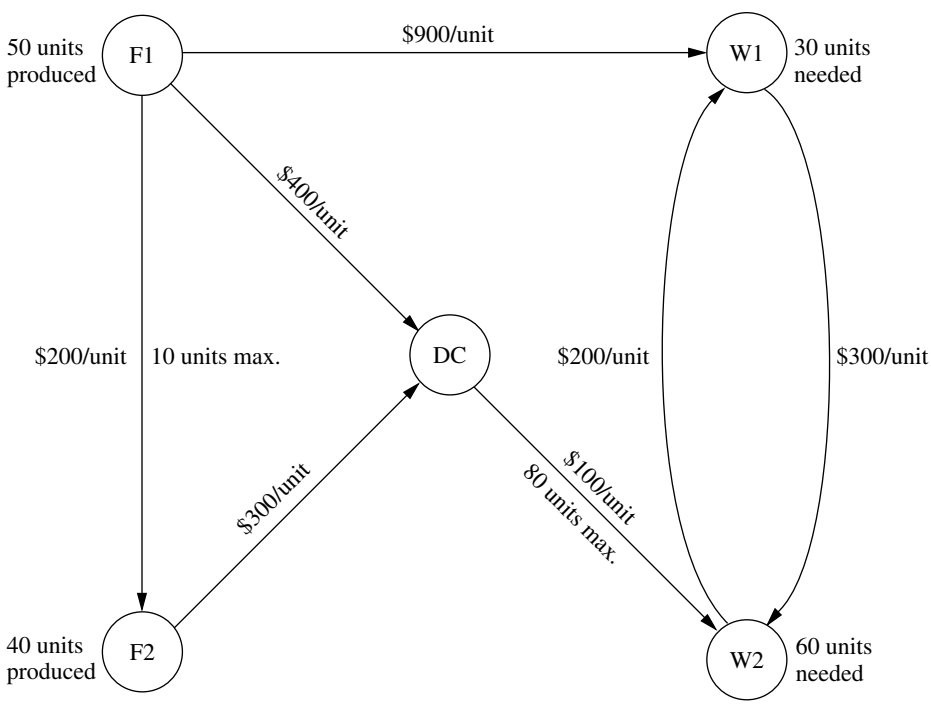


FIGURE 3.13
The distribution network for Distribution Unlimited Co.

What is the required amount for DC? All the units produced at the factories are ultimately needed at the warehouses, so any units shipped from the factories to the distribution center should be forwarded to the warehouses. Therefore, the total amount shipped from the distribution center to the warehouses should *equal* the total amount shipped from the factories to the distribution center. In other words, the *difference* of these two shipping amounts (the required amount for the net flow constraint) should be *zero.*

Since the objective is to minimize the total shipping cost, the coefficients for the objective function come directly from the unit shipping costs given in Fig. 3.13. Therefore, by using money units of hundreds of dollars in this objective function, the complete linear programming model is

$$\text{Minimize} \quad Z = 2x_{\text{F1-F2}} + 4x_{\text{F1-DC}} + 9x_{\text{F1-W1}} + 3x_{\text{F2-DC}} + x_{\text{DC-W2}} + 3x_{\text{W1-W2}} + 2x_{\text{W2-W1}},$$

subject to the following constraints:

1. Net flow constraints:

$$\begin{aligned}
x_{\text{F1-F2}} + x_{\text{F1-DC}} + x_{\text{F1-W1}} &= 50 \text{ (factory 1)} \\
-x_{\text{F1-F2}} + x_{\text{F2-DC}} &= 40 \text{ (factory 2)} \\
-x_{\text{F1-DC}} - x_{\text{F2-DC}} + x_{\text{DC-W2}} &= 0 \text{ (distribution center)} \\
-x_{\text{F1-W1}} + x_{\text{W1-W2}} - x_{\text{W2-W1}} &= -30 \text{ (warehouse 1)} \\
-x_{\text{DC-W2}} - x_{\text{W1-W2}} + x_{\text{W2-W1}} &= -60 \text{ (warehouse 2)}
\end{aligned}$$

2. Upper-bound constraints:

$$x_{\text{F1-F2}} \le 10, \qquad x_{\text{DC-W2}} \le 80$$

3. Nonnegativity constraints:

$$x_{\text{F1-F2}} \ge 0, \quad x_{\text{F1-DC}} \ge 0, \quad x_{\text{F1-W1}} \ge 0, \quad x_{\text{F2-DC}} \ge 0, \quad x_{\text{DC-W2}} \ge 0, \quad x_{\text{W1-W2}} \ge 0, \quad x_{\text{W2-W1}} \ge 0.$$

You will see this problem again in Sec. 9.6, where we focus on linear programming problems of this type (called the *minimum cost flow problem*). In Sec. 9.7, we will solve for its optimal solution:

$$x_{\text{F1-F2}} = 0, \quad x_{\text{F1-DC}} = 40, \quad x_{\text{F1-W1}} = 10, \quad x_{\text{F2-DC}} = 40, \quad x_{\text{DC-W2}} = 80, \quad x_{\text{W1-W2}} = 0, \quad x_{\text{W2-W1}} = 20.$$

The resulting total shipping cost is $49,000.

You also will see a case study involving a *much* larger problem of this same type at the end of the next section.

3.5 SOME CASE STUDIES

To give you a better perspective about the great impact linear programming can have, we now present three case studies of *real* applications. Each of these is a *classic* application, initiated in the early 1980s, that has come to be regarded as a standard of excellence for future applications of linear programming. The first one will bear some strong similarities to the Wyndor Glass Co. problem, but on a realistic scale. Similarly, the second and

third are realistic versions of the last two examples presented in the preceding section (the Union Airways and Distribution Unlimited examples).

Choosing the Product Mix at Ponderosa Industrial[1]

Until its sale in 1988, PONDEROSA INDUSTRIAL was a plywood manufacturer based in Anhuac, Chihuahua, that supplied 25 percent of the plywood in Mexico. Like any plywood manufacturer, Ponderosa's many products were differentiated by thickness and by the quality of the wood. The plywood market in Mexico is competitive, so the market establishes the prices of the products. The prices can fluctuate considerably from month to month, and there may be great differences between the products in their price movements from even one month to the next. As a result, each product's contribution to Ponderosa's total profit was continually varying, and in different ways for different products.

Because of its pronounced effect on profits, a critical issue facing management was the choice of *product mix*—how much to produce of each product—on a monthly basis. This choice was a very complex one, since it had to take into account the current amounts available of various resources needed to produce the products. The most important resources were logs in four quality categories and production capacities for both the pressing operation and the polishing operation.

Beginning in 1980, linear programming was used on a monthly basis to guide the product-mix decision. The linear programming model had an objective of maximizing the total profit from all products. The model's constraints included the various resource constraints as well as other relevant restrictions such as the minimum amount of a product that must be provided to regular customers and the maximum amount that can be sold. (To aid planning for the procurement of raw materials, the model also considered the impact of the product-mix decision for the upcoming month on production in the following month.) The model had 90 decision variables and 45 functional constraints.

This model was used each month to find the product mix for the upcoming month that would be optimal if the estimated values of the various parameters of the model prove to be accurate. However, since some of the parameter values could change quickly (e.g., the unit profits of the products), *sensitivity analysis* was done to determine the effect if the estimated values turned out to be inaccurate. The results indicated when adjustments in the product mix should be made (if time permitted) as unanticipated market changes occurred that affected the price (and so the unit profit) of certain products.

One key decision each month concerned the number of logs in each of the four quality categories to purchase. The amounts available for the upcoming month's production actually were parameters of the model. Therefore, after the purchase decision was made and then the corresponding optimal product mix was determined, *postoptimality analysis* was conducted to investigate the effect of adjusting the purchase decision. For example, it is very easy with linear programming to check what the impact on total profit would be if a quick purchase were to be made of additional logs in a certain quality category to enable increasing production for the upcoming month.

Ponderosa's linear programming system was *interactive,* so management received an immediate response to its "what-if questions" about the impact of encountering parame-

[1]A. Roy, E. E. DeFalomir, and L. Lasdon: "An Optimization-Based Decision Support System for a Product Mix Problem," *Interfaces,* **12**(2):26–33, April 1982.

ter values that differ from those in the original model. What if a quick purchase of logs of a certain kind were made? What if product prices were to fluctuate in a certain way? A variety of such scenarios can be investigated. Management effectively used this power to reach better decisions than the "optimal" product mix from the original model.

The impact of linear programming at Ponderosa was reported to be "tremendous." It led to a dramatic shift in the types of plywood products emphasized by the company. The improved product-mix decisions were credited with increasing the overall profitability of the company by 20 percent. Other contributions of linear programming included better utilization of raw material, capital equipment, and personnel.

Two factors helped make this application of linear programming so successful. One factor is that a *natural language* financial planning system was interfaced with the codes for finding an optimal solution for the linear programming model. Using natural language rather than mathematical symbols to display the components of the linear programming model and its output made the process understandable and meaningful for the managers making the product-mix decisions. Reporting to management in the language of managers is necessary for the successful application of linear programming.

The other factor was that the linear programming system used was *interactive.* As mentioned earlier, after an optimal solution was obtained for one version of the model, this feature enabled managers to ask a variety of "what-if" questions and receive immediate responses. Better decisions frequently were reached by exploring other plausible scenarios, and this process also gave managers more confidence that their decision would perform well under most foreseeable circumstances.

In any application, this ability to respond quickly to management's needs and queries through postoptimality analysis (whether interactive or not) is a vital part of a linear programming study.

Personnel Scheduling at United Airlines[1]

Despite unprecedented industry competition in 1983 and 1984, UNITED AIRLINES managed to achieve substantial growth with service to 48 new airports. In 1984, it became the only airline with service to cities in all 50 states. Its 1984 operating profit reached $564 million, with revenues of $6.2 billion, an increase of 6 percent over 1983, while costs grew by less than 2 percent.

Cost control is essential to competing successfully in the airline industry. In 1982, upper management of United Airlines initiated an OR study of its personnel scheduling as part of the cost control measures associated with the airline's 1983–1984 expansion. The goal was to schedule personnel at the airline's reservations offices and airports so as to minimize the cost of providing the necessary service to customers.

At the time, United Airlines employed over 4,000 reservations sales representatives and support personnel at its 11 reservations offices and about 1,000 customer service agents at its 10 largest airports. Some were part-time, working shifts from 2 to 8 hours; most were full-time, working 8- or 10-hour-shifts. Shifts start at several different times. Each reservations office was open (by telephone) 24 hours a day, as was each of the major airports. However, the number of employees needed at each location to provide the re-

[1]T. J. Holloran and J. E. Bryn, "United Airlines Station Manpower Planning System," *Interfaces,* **16**(1): 39–50, Jan.–Feb. 1986.

quired level of service varied greatly during the 24-hour day, and might fluctuate considerably from one half-hour to the next.

Trying to design the work schedules for all the employees at a given location to meet these service requirements most efficiently is a nightmare of combinatorial considerations. Once an employee begins working, he or she will be there continuously for the entire shift (2 to 10 hours, depending on the employee), *except* for either a meal break or short rest breaks every 2 hours. Given the *minimum* number of employees needed on duty for *each* half-hour interval over a 24-hour day (where these requirements change from day to day over a 7-day week), *how many* employees of *each shift length* should begin work at *what start time* over *each* 24-hour day of a 7-day week? Fortunately, linear programming thrives on such combinatorial nightmares.

Actually, several OR techniques described in this book were used in the computerized planning system developed to attack this problem. Both *forecasting* (Chap. 20) and *queuing theory* (Chaps. 17 and 18) were used to determine the minimum number of employees needed on duty for each half-hour interval. *Integer programming* (Chap. 12) was used to determine the times of day at which shifts would be allowed to start. However, the core of the planning system was *linear programming,* which did all the actual scheduling to provide the needed service with the smallest possible labor cost. A complete work schedule was developed for the first full week of a month, and then it was reused for the remainder of the month. This process was repeated each month to reflect changing conditions.

Although the details about the linear programming model have not been published, it is clear that the basic approach used is the one illustrated by the Union Airways example of personnel scheduling in Sec. 3.4. The objective function being minimized represents the total personnel cost for the location being scheduled. The main functional constraints require that the number of employees on duty during each time period will not fall below minimum acceptable levels.

However, the Union Airways example has only five decision variables. By contrast, the United Airlines model for some locations has over 20,000 decision variables! The difference is that a real application must consider myriad important details that can be ignored in a textbook example. For example, the United Airlines model takes into account such things as the meal and break assignment times for each employee scheduled, differences in shift lengths for different employees, and days off over a weekly schedule, among other scheduling details.

This application of linear programming was reported to have had "an overwhelming impact not only on United management and members of the manpower planning group, but also for many who had never before heard of management science (OR) or mathematical modeling." It earned rave reviews from upper management, operating managers, and affected employees alike. For example, one manager described the scheduling system as

> Magical, . . . just as the [customer] lines begin to build, someone shows up for work, and just as you begin to think you're overstaffed, people start going home.[1]

In more tangible terms, this application was credited with saving United Airlines more than $6 million *annually* in just direct salary and benefit costs. Other benefits included improved customer service and reduced need for support staff.

[1]Holloran and Bryn, "United Airlines Station Manpower Planning System," p. 49.

After some updating in the early 1990s, the system is providing similar benefits today.

One factor that helped make this application of linear programming so successful was "the support of operational managers and their staffs." This was a lesson learned by experience, because the OR team initially failed to establish a good line of communication with the operating managers, who then resisted the team's initial recommendations. The team leaders described their mistake as follows:

> The cardinal rule for earning the trust and respect of operating managers and support staffs—"getting them involved in the development process"—had been violated.[1]

The team then worked much more closely with the operating managers—with outstanding results.

Planning Supply, Distribution, and Marketing at Citgo Petroleum Corporation[2]

CITGO PETROLEUM CORPORATION specializes in refining and marketing petroleum. In the mid-1980s, it had annual sales of several billion dollars, ranking it among the 150 largest industrial companies in the United States.

After several years of financial losses, Citgo was acquired in 1983 by Southland Corporation, the owner of the 7-Eleven convenience store chain (whose sales include 2 billion gallons of quality motor fuels annually). To turn Citgo's financial losses around, Southland created a task force composed of Southland personnel, Citgo personnel, and outside consultants. An eminent OR consultant was appointed director of the task force to report directly to both the president of Citgo and the chairman of the board of Southland.

During 1984 and 1985, this task force applied various OR techniques (as well as information systems technologies) throughout the corporation. It was reported that these OR applications "have changed the way Citgo does business and resulted in approximately $70 million per year profit improvement."[3]

The two most important applications were both *linear programming systems* that provided management with powerful planning support. One, called the *refinery LP system,* led to great improvements in refinery yield, substantial reductions in the cost of labor, and other important cost savings. This system contributed approximately $50 million to profit improvement in 1985. (See the end of Sec. 2.4 for discussion of the key role that *model validation* played in the development of this system.)

However, we will focus here on the other linear programming system, called the supply, distribution, and marketing modeling system (or just the *SDM system*), that Citgo is continuing to use. The SDM system is particularly interesting because it is based on a special kind of linear programming model that uses *networks,* just like the model for the Distribution Unlimited example presented at the end of Sec. 3.4. The model for the SDM system provides a representation of Citgo's entire marketing and distribution network.

At the time the task force conducted its OR study, Citgo owned or leased 36 product storage terminals which were supplied through five distribution center terminals via a dis-

[1]Ibid, p. 47.

[2]See the references cited in footnote 2 on p. 10.

[3]See p. 4 of the second reference cited in footnote 2 on p. 10.

tribution network of pipelines, tankers, and barges. Citgo also sold product from over 350 exchange terminals that were shared with other petroleum marketers. To supply its customers, product might be acquired by Citgo from its refinery in Lake Charles, LA, or from spot purchases on one of five major spot markets, product exchanges, and trades with other industry refiners. These product acquisition decisions were made daily. However, the time from such a decision until the product reached the intended customers could be as long as 11 weeks. Therefore, the linear programming model used an 11-week planning horizon.

The SDM system is used to coordinate the supply, distribution, and marketing of each of Citgo's major products (originally four grades of motor fuel and No. 2 fuel oil) throughout the United States. Management uses the system to make decisions such as where to sell, what price to charge, where to buy or trade, how much to buy or trade, how much to hold in inventory, and how much to ship by each mode of transportation. Linear programming guides these decisions and when to implement them so as to minimize total cost or maximize total profit. The SDM system also is used in "what-if" sessions, where management asks what-if questions about scenarios that differ from those assumed in the original model.

The linear programming model in the SDM system has the same form as the model for the Distribution Unlimited example presented at the end of Sec. 3.4. In fact, both models fit an important special kind of linear programming problem, called the *minimum cost flow problem,* that will be discussed in Sec. 9.6. The main functional constraints for such models are *equality constraints,* where each one prescribes what the net flow of goods out of a specific location must be.

The Distribution Unlimited model has just seven decision variables and five equality constraints. By contrast, the Citgo model for each major product has about 15,000 decision variables and 3,000 equality constraints!

At the end of Sec. 2.1, we described the important role that *data collection* and *data verification* played in developing the Citgo models. With such huge models, a massive amount of data must be gathered to determine all the parameter values. A state-of-the-art management database system was developed for this purpose. Before using the data, a preloader program was used to check for data errors and inconsistencies. The importance of doing so was brought forcefully home to the task force when, as mentioned in Sec. 2.1, the initial run of the preloader program generated a paper log of error messages an inch thick! It was clear that the data collection process needed to be thoroughly debugged to help ensure the validity of the models.

The SDM linear programming system has greatly improved the efficiency of Citgo's supply, distribution, and marketing operations, enabling a huge reduction in product inventory with no drop in service levels. During its first year, the value of petroleum products held in inventory was reduced by $116.5 million. This huge reduction in capital tied up in carrying inventory resulted in saving about $14 million annually in interest expenses for borrowed capital, adding $14 million to Citgo's annual profits. Improvements in coordination, pricing, and purchasing decisions have been estimated to add at least another $2.5 million to annual profits. Many *indirect* benefits also are attributed to this application of linear programming, including improved data, better pricing strategies, and elimination of unnecessary product terminals, as well as improved communication and coordination between supply, distribution, marketing, and refinery groups.

Some of the factors that contributed to the success of this OR study are the same as for the two preceding case studies. Like Ponderosa Industrial, one factor was developing output reports in the language of managers to really meet their needs. These output reports are designed to be easy for managers to understand and use, and they address the issues that are important to management. Also like Ponderosa, another factor was enabling management to respond quickly to the dynamics of the industry by using the linear programming system extensively in "what-if" sessions. As in so many applications of linear programming, *postoptimality* analysis proved more important than the initial optimal solution obtained for one version of the model.

Much as in the United Airlines application, another factor was the enthusiastic support of operational managers during the development and implementation of this linear programming system.

However, the most important factor was the unlimited support provided to the task force by top management, ranging right up to the chief executive officer and the chairman of the board of Citgo's parent company, Southland Corporation. As mentioned earlier, the director of the task force (an eminent OR consultant) reported directly to both the president of Citgo and the chairman of the board of Southland. This backing by top management included strong organizational and financial support.

The organizational support took a variety of forms. One example was the creation and staffing of the position of senior vice-president of operations coordination to evaluate and coordinate recommendations based on the models which spanned organizational boundaries.

When discussing both this linear programming system and other OR applications implemented by the task force, team members described the financial support of top management as follows:

> The total cost of the systems implemented, $20 million to $30 million, was the greatest obstacle to this project. However, because of the information explosion in the petroleum industry, top management realized that numerous information systems were essential to gather, store, and analyze data. The incremental cost of adding management science (OR) technologies to these computers and systems was small, in fact very small in light of the enormous benefits they provided.[1]

3.6 DISPLAYING AND SOLVING LINEAR PROGRAMMING MODELS ON A SPREADSHEET

Spreadsheet software, such as Excel, is a popular tool for analyzing and solving small linear programming problems. The main features of a linear programming model, including all its parameters, can be easily entered onto a spreadsheet. However, spreadsheet software can do much more than just display data. If we include some additional information, the spreadsheet can be used to quickly analyze potential solutions. For example, a potential solution can be checked to see if it is feasible and what Z value (profit or cost) it achieves. Much of the power of the spreadsheet lies in its ability to immediately see the results of any changes made in the solution.

In addition, the Excel Solver can quickly apply the simplex method to find an optimal solution for the model.

[1]Ibid, p. 21.

To illustrate this process, we now return to the Wyndor example introduced in Sec. 3.1.

Displaying the Model on a Spreadsheet

After expressing profits in units of thousands of dollars, Table 3.1 in Sec. 3.1 gives all the parameters of the model for the Wyndor problem. Figure 3.14 shows the necessary additions to this table for an Excel spreadsheet. In particular, a row is added (row 9, labeled "Solution") to store the values of the decision variables. Next, a column is added (column E, labeled "Totals"). For each functional constraint, the number in column E is the numerical value of the left-hand side of that constraint. Recall that the left-hand side represents the actual amount of the resource used, given the values of the decision variables in row 9. For example, for the Plant 3 constraint in row 7, the amount of this resource used (in hours of production time per week) is

Production time used in Plant 3 = $3x_1 + 2x_2$.

In the language of Excel, the equivalent equation for the number in cell E7 is

E7 = C7*C9 + D7*D9.

Notice that this equation involves the sum of two products. There is a function in Excel, called SUMPRODUCT, that will sum up the product of each of the individual terms in two different ranges of cells. For instance, SUMPRODUCT(C7:D7,C9:D9) takes each of the individual terms in the range C7:D7, multiplies them by the corresponding term in the range C9:D9, and then sums up these individual products, just as shown in the above equation. Although optional with such short equations, this function is especially handy as a shortcut for entering longer linear programming equations.

Next, $\leq$ signs are entered in cells F5, F6, and F7 to indicate the form of the functional constraints. (When using a trial-and-error approach, the spreadsheet still will allow you to enter infeasible trial solutions that violate the $\leq$ signs, but these signs serve as a reminder to reject such trial solutions if no changes are made in the numbers in column G.)

FIGURE 3.14
The spreadsheet for the Wyndor problem before using the Excel Solver, so the values of the decision variables and the objective function are just entered as zeros.

	A	B	C	D	E	F	G
1	Wyndor Glass Co. Product-Mix Problem						
2							
3			Hours Used per Unit Produced				Hours
4			Doors	Windows	Totals		Available
5		Plant 1	1	0	0	≤	4
6		Plant 2	0	2	0	≤	12
7		Plant 3	3	2	0	≤	18
8		Unit Profit ($thousands)	3	5	0		
9		Solution	0	0			

	E
5	=SUMPRODUCT(C5:D5,C9:D9)
6	=SUMPRODUCT(C6:D6,C9:D9)
7	=SUMPRODUCT(C7:D7,C9:D9)
8	=SUMPRODUCT(C8:D8,C9:D9)

Finally, the value of the objective function is entered in cell E8. Much like the other values in column E, it is the sum of products. The equation for cell E8 is =SUMPRODUCT(C8:D8,C9:D9). The lower right-hand side of Fig. 3.14 shows all the formulas that need to be entered in the "Totals" column (column E) for the Wyndor problem.

Once the model is entered in this spreadsheet format, it is easy to analyze any potential solution. When values for the decision variables are entered in the spreadsheet, the "Totals" column immediately calculates the total amount of each resource used, as well as the total profit. Hence, by comparing column E with column G, it can be seen immediately whether the potential solution is feasible. If so, cell E8 shows how much profit it would generate. One approach to trying to solve a linear programming model would be trial and error, using the spreadsheet to analyze a variety of solutions. However, you will see next how Excel also can be used to quickly find an *optimal solution.*



Using the Excel Solver to Solve the Model

Excel includes a tool called Solver that uses the simplex method to find an optimal solution. (A more powerful version of Solver, called *Premium Solver,* also is available in your OR Courseware.) Before using Solver, all the following components of the model need to be included on the spreadsheet:

1. Each decision variable
2. The objective function and its value
3. Each functional constraint

The spreadsheet layout shown in Fig. 3.14 includes all these components. The parameters for the functional constraints are in rows 5, 6, and 7, and the coefficients for the objective function are in row 8. The values of the decision variables are in cells C9 and D9, and the value of the objective function is in cell E8. Since we don't know what the values of the decision variables should be, they are just entered as zeros. The Solver will then change these to the optimal values after solving the problem.

The Solver can be started by choosing "Solver" in the Tools menu. The Solver dialogue box is shown in Fig. 3.15. The "Target Cell" is the cell containing the value of the objective function, while the "Changing Cells" are the cells containing the values of the decision variables.

Before the Solver can apply the simplex method, it needs to know exactly where each component of the model is located on the spreadsheet. You can either type in the cell addresses or click on them. Since the target cell is cell E8 and the changing cells are in the range C9:D9, these addresses are entered into the Solver dialogue box as shown in Fig. 3.15. (Excel then automatically enters the dollar signs shown in the figure to fix these addresses.) Since the goal is to maximize the objective function, "Max" also has been selected.

Next, the addresses for the functional constraints need to be added. This is done by clicking on the "Add . . ." button on the Solver dialogue box. This brings up the "Add Constraint" dialogue box shown in Fig. 3.16. The location of the values of the left-hand sides and the right-hand sides of the functional constraints are specified in this dialogue box. The cells E5 through E7 all need to be less than or equal to the corresponding cells in G5 through G7. There also is a menu to choose between <=, =, or >=, so <= has been chosen for these constraints. (This choice is needed even though ≤ signs were pre-

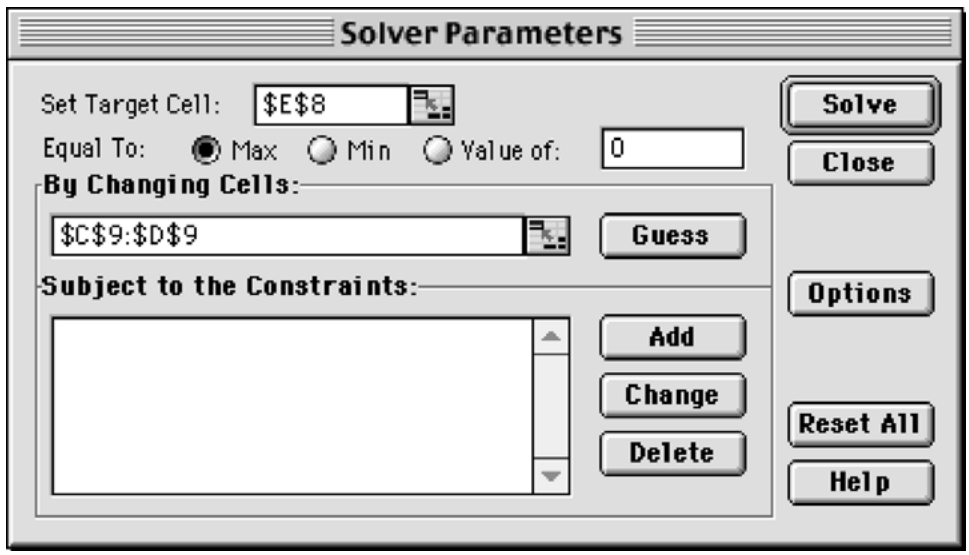


FIGURE 3.15
The Solver dialogue box after specifying which cells in Fig. 3.14 contain the values of the objective function and the decision variables, plus indicating that the objective function is to be maximized.

viously entered in column F of the spreadsheet because Solver uses only the functional constraints that are specified with the Add Constraint dialogue box.)

If there were more functional constraints to add, you would click on Add to bring up a new Add Constraint dialogue box. However, since there are no more in this example, the next step is to click on OK to go back to the Solver dialogue box.

The Solver dialogue box now summarizes the complete model (see Fig. 3.17) in terms of the spreadsheet in Fig. 3.14. However, before asking Solver to solve the model, one more step should be taken. Clicking on the Options . . . button brings up the dialogue box shown in Fig. 3.18. This box allows you to specify a number of options about how the problem will be solved. The most important of these are the Assume Linear Model option and the Assume Non-Negative option. Be sure that both options are checked as shown in the figure. This tells Solver that the problem is a *linear* programming problem with nonnegativity constraints for all the decision variables, and that the simplex method

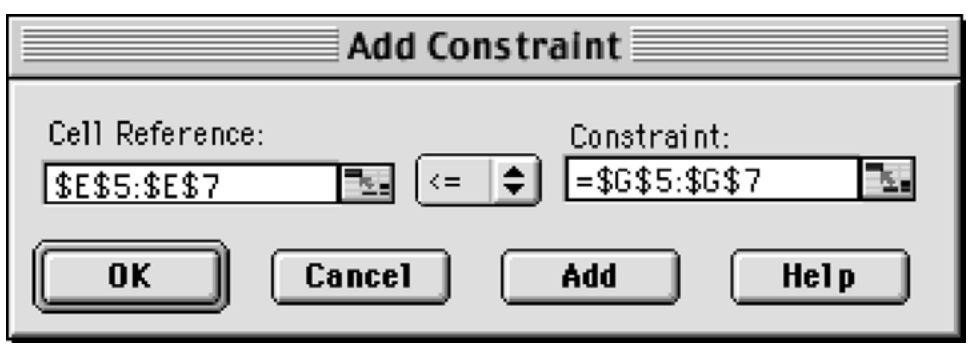


FIGURE 3.16
The Add Constraint dialogue box after specifying that cells E5, E6, and E7 in Fig. 3.14 are required to be less than or equal to cells G5, G6, and G7, respectively.

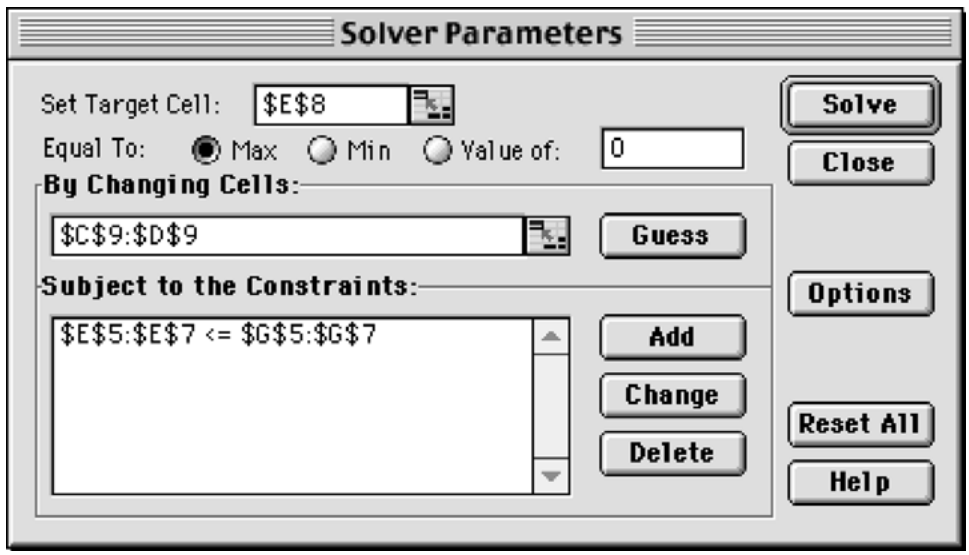


FIGURE 3.17
The Solver dialogue box after specifying the entire model in terms of the spreadsheet.

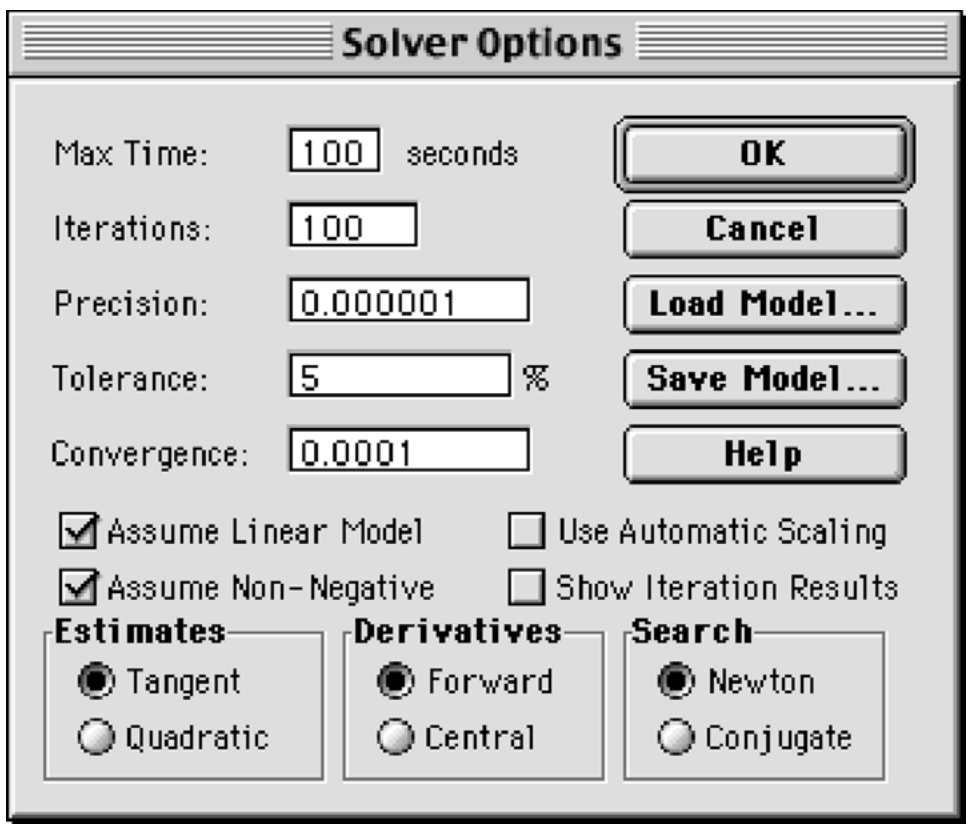


FIGURE 3.18
The Solver Options dialogue box after checking the Assume Linear Model and Assume Non-Negative options to indicate that we are dealing with a linear programming model with nonnegativity constraints that needs to be solved by the simplex method.

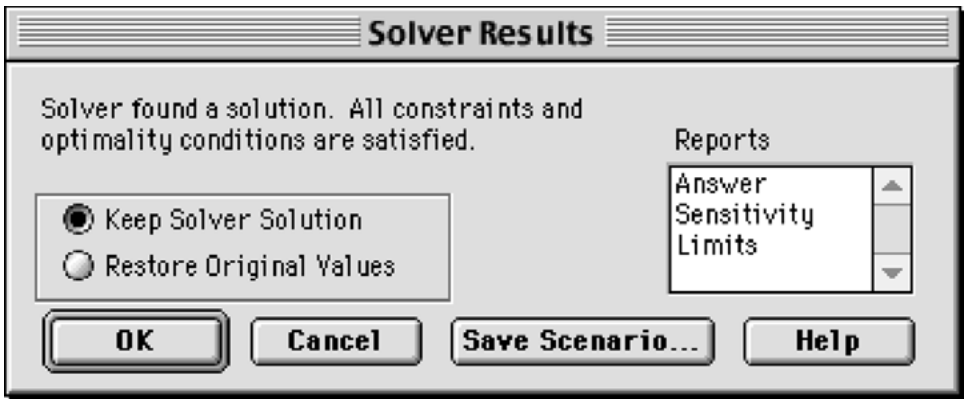


FIGURE 3.19
The Solver Results dialogue box that indicates that an optimal solution has been found.

should be used to solve the problem.[1] Regarding the other options, accepting the default values shown in the figure usually is fine for small problems. Clicking on the OK button then returns you to the Solver dialogue box.

Now you are ready to click on Solve in the Solver dialogue box, which will cause the Solver to execute the simplex method in the background. After a few seconds (for a small problem), Solver will then indicate the results. Typically, it will indicate that it has found an optimal solution, as specified in the Solver Results dialogue box shown in Fig. 3.19. If the model has no feasible solutions or no optimal solution, the dialogue box will indicate that instead by stating that "Solver could not find a feasible solution" or that "the Set Cell values do not converge." The dialogue box also presents the option of generating various reports. One of these (the Sensitivity Report) will be discussed in detail in Sec. 4.7.

After solving the model, the Solver replaces the original value of the decision variables in the spreadsheet with the optimal values, as shown in Fig. 3.20. The spreadsheet also indicates the value of the objective function, as well as the amount of each resource that is being used.

[1]In older versions of Excel prior to Excel 97, the Assume Non-Negative option is not available, so nonnegativity constraints have to be added with the Add Constraint dialogue box.

FIGURE 3.20
The spreadsheet obtained after solving the Wyndor problem.

	A	B	C	D	E	F	G
1	Wyndor Glass Co. Product-Mix Problem						
2							
3			Hours Used per Unit Produced				Hours
4			Doors	Windows	Totals		Available
5		Plant 1	1	0	2	≤	4
6		Plant 2	0	2	12	≤	12
7		Plant 3	3	2	18	≤	18
8		Unit Profit ($thousands)	3	5	36		
9		Solution	2	6			

3.7 FORMULATING VERY LARGE LINEAR PROGRAMMING MODELS

Linear programming models come in many different sizes. For the examples in Secs. 3.1 and 3.4, the model sizes range from three functional constraints and two decision variables (for the Wyndor and radiation therapy problems) up to 17 functional constraints and 12 decision variables (for the Save-It Company problem). The latter case may seem like a rather large model. After all, it does take a substantial amount of time just to write down a model of this size. However, by contrast, the models for the classic case studies presented in Sec. 3.5 are much, much larger. For example, the models in the Citgo case study typically have about 3,000 functional constraints and 15,000 decision variables.

The Citgo model sizes are not at all unusual. Linear programming models in practice commonly have hundreds or thousands of functional constraints. In fact, there have been some recently reported cases of a few hundred thousand constraints. The number of decision variables frequently is even larger than the number of functional constraints, and occasionally will range into the millions.

Formulating such monstrously large models can be a daunting task. Even a "medium-sized" model with a thousand functional constraints and a thousand decision variables has over a million parameters (including the million coefficients in these constraints). It simply is not practical to write out the algebraic formulation, or even to fill in the parameters on a spreadsheet, for such a model.

So how are these very large models formulated in practice? It requires the use of a *modeling language.*

Modeling Languages

A mathematical programming modeling language is software that has been specifically designed for efficiently formulating large linear programming models (and related models). Even with thousands of functional constraints, they typically are of a relatively few types where the constraints of the same type follow the same pattern. Similarly, the decision variables will fall into a small number of categories. Therefore, using large blocks of data in databases, a modeling language will simultaneously formulate all the constraints of the same type by simultaneously dealing with the variables of each type. We will illustrate this process soon.

In addition to efficiently formulating large models, a modeling language will expedite a number of model management tasks, including accessing data, transforming data into model parameters, modifying the model whenever desired, and analyzing solutions from the model. It also may produce summary reports in the vernacular of the decision makers, as well as document the model's contents.

Several excellent modeling languages have been developed over the last couple of decades. These include AMPL, MPL, GAMS, and LINGO.

The student version of one of these, MPL (short for mathematical programming language), is provided for you on the CD-ROM along with extensive tutorial material. The latest student version also can be downloaded from the website, maximal-usa.com. MPL is a product of Maximal Software, Inc. A new feature is extensive support for Excel in MPL. This includes both importing and exporting Excel ranges from MPL. Full support also is provided for the Excel VBA macro language through OptiMax 2000. (The student version of OptiMax 2000 is on the CD-ROM as well.) This product allows the user to

fully integrate MPL models into Excel and solve with any of the powerful solvers that MPL supports, including **CPLEX** (described in Sec. 4.8).

LINGO is a product of LINDO Systems, Inc. The latest student version of LINGO is available by downloading it from the website, www.lindo.com. LINDO Systems also provides a completely spreadsheet-oriented optimizer called What'sBest, also available on this website.

The CD-ROM includes MPL, LINGO, and What'sBest formulations for essentially every example in this book to which these modeling languages can be applied.

Now let us look at a simplified example that illustrates how a very large linear programming model can arise.

An Example of a Problem with a Huge Model

Management of the WORLDWIDE CORPORATION needs to address a *product-mix problem,* but one that is vastly more complex than the Wyndor product-mix problem introduced in Sec. 3.1. This corporation has 10 plants in various parts of the world. Each of these plants produces the same 10 products and then sells them within its region. The *demand* (sales potential) for each of these products from each plant is known for each of the next 10 months. Although the amount of a product sold by a plant in a given month cannot exceed the demand, the amount produced can be larger, where the excess amount would be stored in inventory (at some unit cost per month) for sale in a later month. Each unit of each product takes the same amount of space in inventory, and each plant has some upper limit on the total number of units that can be stored (the *inventory capacity*).

Each plant has the same 10 production processes (we'll refer to them as *machines*), each of which can be used to produce any of the 10 products. Both the production cost per unit of a product and the production rate of the product (number of units produced per day devoted to that product) depend on the combination of plant and machine involved (but not the month). The number of working days (*production days available*) varies somewhat from month to month.

Since some plants and machines can produce a particular product either less expensively or at a faster rate than other plants and machines, it is sometimes worthwhile to ship some units of the product from one plant to another for sale by the latter plant. For each combination of a plant being shipped from (the *fromplant*) and a plant being shipped to (the *toplant*), there is a certain cost per unit shipped of any product, where this unit shipping cost is the same for all the products.

Management now needs to determine how much of each product should be produced by each machine in each plant during each month, as well as how much each plant should sell of each product in each month and how much each plant should ship of each product in each month to each of the other plants. Considering the worldwide price for each product, the objective is to find the feasible plan that maximizes the total profit (total sales revenue *minus* the sum of the total production costs, inventory costs, and shipping costs).

The Structure of the Resulting Model

Because of the inventory costs and the limited inventory capacities, it is necessary to keep track of the amount of each product kept in inventory in each plant during each month. Consequently, the linear programming model has four types of decision variables: pro-

duction quantities, inventory quantities, sales quantities, and shipping quantities. With 10 plants, 10 machines, 10 products, and 10 months, this gives a total of 21,000 decision variables, as outlined below.

Decision Variables.

10,000 production variables: one for each combination of a plant, machine, product, and month
1,000 inventory variables: one for each combination of a plant, product, and month
1,000 sales variables: one for each combination of a plant, product, and month
9,000 shipping variables: one for each combination of a product, month, plant (the fromplant), and another plant (the toplant)

Multiplying each of these decision variables by the corresponding unit cost or unit revenue, and then summing over each type, the following objective function can be calculated:

Objective Function.

Maximize profit = total sales revenue − total cost,

where

Total cost = total production cost + total inventory cost + total shipping cost.

When maximizing this objective function, the 21,000 decision variables need to satisfy nonnegativity constraints as well as four types of functional constraints—production capacity constraints, plant balance constraints (equality constraints that provide appropriate values to the inventory variables), maximum inventory constraints, and maximum sales constraints. As enumerated below, there are a total of 3,100 functional constraints, but all the constraints of each type follow the same pattern.

Functional Constraints.

1,000 production capacity constraints (one for each combination of a plant, machine, and month):

Production days used $\leq$ production days available,

where the left-hand side is the sum of 10 fractions, one for each product, where each fraction is that product's production quantity (a decision variable) *divided* by the product's production rate (a given constant).

1,000 plant balance constraints (one for each combination of a plant, product, and month):

Amount produced + inventory last month + amount shipped in = sales + current inventory + amount shipped out,

where the *amount produced* is the sum of the decision variables representing the production quantities at the machines, the *amount shipped in* is the sum of the decision variables representing the shipping quantities in from the other plants, and the *amount shipped out* is the sum of the decision variables representing the shipping quantities out to the other plants.

100 maximum inventory constraints (one for each combination of a plant and month):

Total inventory $\leq$ inventory capacity,

where the left-hand side is the sum of the decision variables representing the inventory quantities for the individual products.

1,000 maximum sales constraints (one for each combination of a plant, product, and month):

Sales $\leq$ demand.

Now let us see how the MPL modeling language, a product of Maximal Software, Inc., can formulate this huge model very compactly.

MPL

Formulation of the Model in MPL

The modeler begins by assigning a title to the model and listing an *index* for each of the entities of the problem, as illustrated below.

```
TITLE
   Production_Planning;

INDEX
   product    := (A1, A2, A3, A4, A5, A6, A7, A8, A9, A10);
   month      := (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct);
   plant      := (p1, p2, p3, p4, p5, p6, p7, p8, p9, p10);
   fromplant  := plant;
   toplant    := plant;
   machine    := (m1, m2, m3, m4, m5, m6, m7, m8, m9, m10);
```

Except for the months, the entries on the right-hand side are arbitrary labels for the respective products, plants, and machines, where these same labels are used in the data files. Note that a colon is placed after the name of each entry and a semicolon is placed at the end of each statement (but a statement is allowed to extend over more than one line).

A big job with any large model is collecting and organizing the various types of data into data files. In this case, eight data files are needed to hold the product prices, demands, production costs, production rates, production days available, inventory costs, inventory capacities, and shipping costs. Numbering these data files as 1, 2, 3, . . . , 8, the next step is to give a brief suggestive name to each one and to identify (inside square brackets) the index or indexes over which the data in the file run, as shown below.

```
DATA
   Price [product]                      := DATAFILE 1;
   Demand [plant, product, month]       := DATAFILE 2;
   ProdCost [plant, machine, product]   := DATAFILE 3;
   ProdRate [plant, machine, product]   := DATAFILE 4;
   ProdDaysAvail [month]                := DATAFILE 5;
   InvtCost [product]                   := DATAFILE 6;
   InvtCapacity [plant]                 := DATAFILE 7;
   ShipCost [fromplant, toplant]        := DATAFILE 8;
```

Next, the modeler gives a short name to each type of decision variable. Following the name, inside square brackets, is the index or indexes over which the subscripts run.

```
VARIABLES
   Produce [plant, machine, product, month]      → Prod;
   Inventory [plant, product, month]             → Invt;
   Sales [plant, product, month]                 → Sale;
   Ship [product, month, fromplant, toplant]
       WHERE (fromplant <> toplant);
```

In the case of the decision variables with names longer than four letters, the arrows on the right point to four-letter abbreviations to fit the size limitations of many solvers. The last line indicates that the fromplant subscript and toplant subscript are not allowed to have the same value.

There is one more step before writing down the model. To make the model easier to read, it is useful first to introduce *macros* to represent the summations in the objective function.

```
MACROS
   Total Revenue  := SUM (plant, product, month: Price*Sales);
   TotalProdCost  := SUM (plant, machine, product, month:
                     ProdCost*Produce);
   TotalInvtCost  := SUM (plant, product, month:
                     InvtCost*Inventory);
   TotalShipCost  := SUM (product, month, fromplant, toplant:
                     ShipCost*Ship);
   TotalCost      := TotalProdCost + TotalInvtCost + TotalShipCost;
```

The first four macros use the MPL keyword SUM to execute the summation involved. Following each SUM keyword (inside the parentheses) is, first, the index or indexes over which the summation runs. Next (after the colon) is the vector product of a data vector (one of the data files) times a variable vector (one of the four types of decision variables).

Now this model with 3,100 functional constraints and 21,000 decision variables can be written down in the following compact form.

```
MODEL

   MAX  Profit = TotalRevenue − TotalCost;

SUBJECT TO
   ProdCapacity [plant, machine, month] → PCap;
      SUM (product: Produce/ProdRate) ≤ ProdDaysAvail;

   PlantBal [plant, product, month] → PBal;
        SUM (machine: Produce) + Inventory [month − 1]
      + SUM (fromplant: Ship[fromplant, toplant:= plant])
    =
        Sales + Inventory
      + SUM (toplant: Ship[from plant:= plant, toplant]);

   MaxInventory [plant, month] → MaxI:
      SUM (product: Inventory) ≤ InvtCapacity;

BOUNDS
   Sales ≤ Demand;

END
```

For each of the four types of constraints, the first line gives the name for this type. There is one constraint of this type for each combination of values for the indexes inside

the square brackets following the name. To the right of the brackets, the arrow points to a four-letter abbreviation of the name that a solver can use. Below the first line, the general form of constraints of this type is shown by using the SUM operator.

For each production capacity constraint, each term in the summation consists of a decision variable (the production quantity of that product on that machine in that plant during that month) divided by the corresponding production rate, which gives the number of production days being used. Summing over the products then gives the total number of production days being used on that machine in that plant during that month, so this number must not exceed the number of production days available.

The purpose of the plant balance constraint for each plant, product, and month is to give the correct value to the current inventory variable, given the values of all the other decision variables including the inventory level for the preceding month. Each of the SUM operators in these constraints involves simply a sum of decision variables rather than a vector product. This is the case also for the SUM operator in the maximum inventory constraints. By contrast, the left-hand side of the maximum sales constraints is just a single decision variable for each of the 1,000 combinations of a plant, product, and month. (Separating these upper-bound constraints on individual variables from the regular functional constraints is advantageous because of the computational efficiencies that can be obtained by using the *upper bound technique* described in Sec. 7.3.) No lower-bound constraints are shown here because MPL automatically assumes that all 21,000 decision variables have nonnegativity constraints unless nonzero lower bounds are specified. For each of the 3,100 functional constraints, note that the left-hand side is a linear function of the decision variables and the right-hand side is a constant taken from the appropriate data file. Since the objective function also is a linear function of the decision variables, this model is a legitimate linear programming model.

To solve the model, MPL supports various leading **solvers** (software packages for solving linear programming models and related models) that can be installed into MPL. As discussed in Sec. 4.8, **CPLEX** is a particularly prominent and powerful solver. The version of MPL in your OR Courseware already has installed the student version of CPLEX, which uses the simplex method to solve linear programming models. Therefore, to solve such a model formulated with MPL, all you have to do is choose *Solve CPLEX* from the *Run* menu or press the *Run Solve* button in the *Toolbar.* You then can display the solution file in a view window by pressing the *View* button at the bottom of the *Status Window.*

This brief introduction to MPL illustrates the ease with which modelers can use modeling languages to formulate huge linear programming models in a clear, concise way. To assist you in using MPL, an MPL Tutorial is included on the CD-ROM. This tutorial goes through all the details of formulating smaller versions of the production planning example considered here. You also can see elsewhere on the CD-ROM how all the other linear programming examples in this chapter and subsequent chapters would be formulated with MPL and solved by CPLEX.

The LINGO Modeling Language

LINGO is another popular modeling language that is featured in this book. The company that produces LINGO, LINDO Systems, also produces a widely used solver called **LINDO** as well as a spreadsheet solver, What'sBest. All three share a common set of solvers based

on the simplex method and, in more advanced versions, on the kind of algorithmic techniques introduced in Secs. 4.9 and 7.4 as well. (We will discuss LINDO further in Sec. 4.8 and Appendix 4.1.) As mentioned earlier, the student version of LINGO is available to you through downloading from the website, www.lindo.com.

Like MPL, LINGO enables a modeler to efficiently formulate a huge linear programming model in a clear, concise way. It also can be used for a wide variety of other models.

LINGO uses *sets* as its fundamental building block. For example, in the Worldwide Corp. production planning problem, the sets of interest include the collections of products, plants, machines, and months. Each member of a set may have one or more *attributes* associated with it, such as the price of a product, the inventory capacity of a plant, the production rate of a machine, and the number of production days available in a month. These attributes provide data for the model. Some set attributes, such as production quantities and shipping quantities, can be decision variables for the model. As with MPL, the SUM operator is commonly used to write the objective function and each constraint type in a compact form. After completing the formulation, the model can be solved by selecting the *Solve* command from the LINGO menu or pressing the *Solve* button on the toolbar.

An appendix to this chapter describes LINGO further and illustrates its use on a couple of small examples. A supplement on the CD-ROM shows how LINGO can be used to formulate the model for the Worldwide Corp. production planning example. A LINGO tutorial on the CD-ROM provides the details needed for doing basic modeling with this modeling language. The LINGO formulations and solutions for the various examples in both this chapter and many other chapters also are included on the CD-ROM.

3.8 CONCLUSIONS

Linear programming is a powerful technique for dealing with the problem of allocating limited resources among competing activities as well as other problems having a similar mathematical formulation. It has become a standard tool of great importance for numerous business and industrial organizations. Furthermore, almost any social organization is concerned with allocating resources in some context, and there is a growing recognition of the extremely wide applicability of this technique.

However, not all problems of allocating limited resources can be formulated to fit a linear programming model, even as a reasonable approximation. When one or more of the assumptions of linear programming is violated seriously, it may then be possible to apply another mathematical programming model instead, e.g., the models of integer programming (Chap. 12) or nonlinear programming (Chap. 13).

APPENDIX 3.1 THE LINGO MODELING LANGUAGE

LINGO is a mathematical modeling language designed particularly for formulating and solving a wide variety of optimization problems, including linear programming, integer programming (Chap. 12), and nonlinear programming (Chap. 13) problems. Extensive details and a downloadable student version can be found at www.lindo.com.

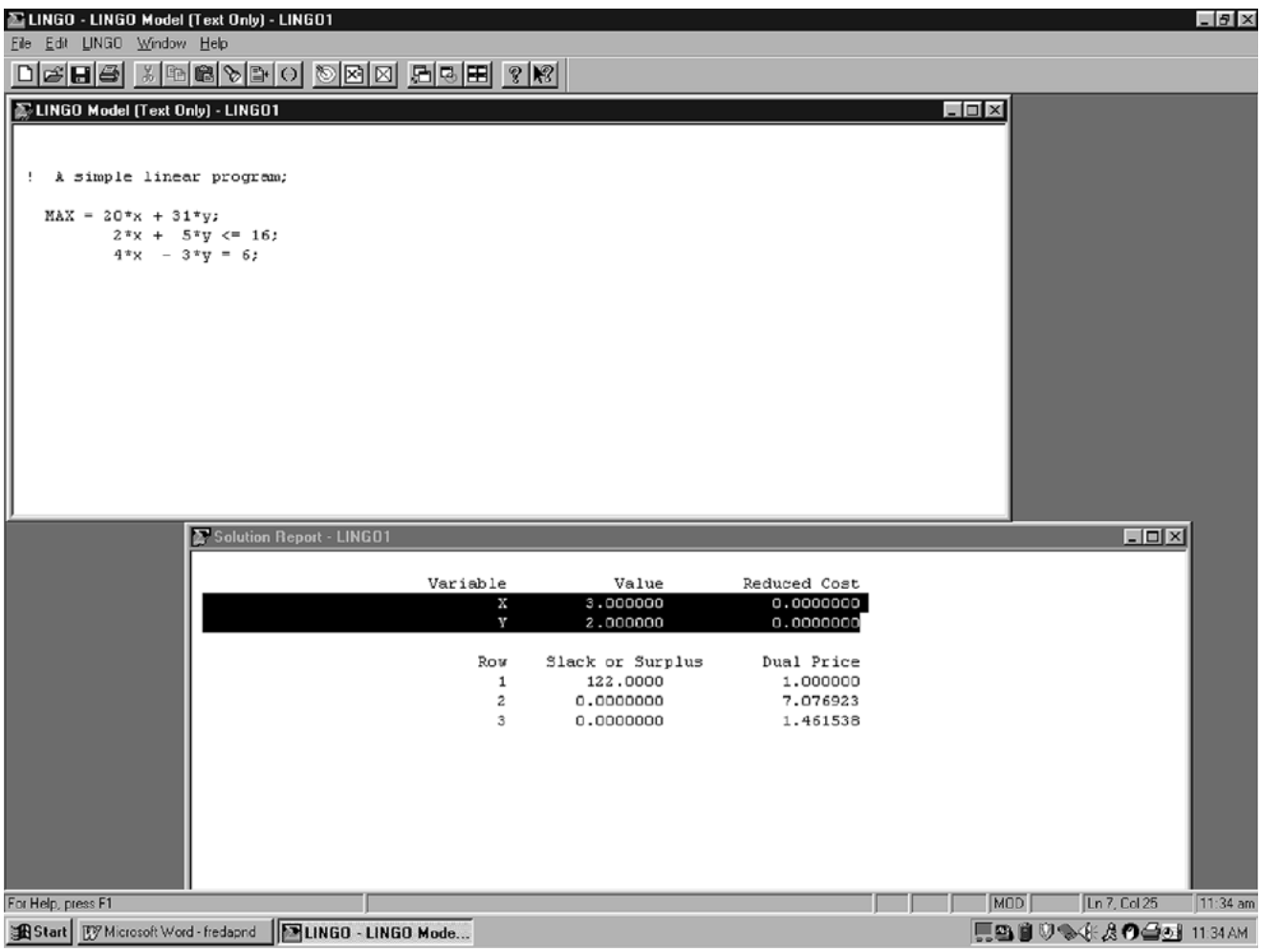
Simple problems are entered into LINGO in a fairly natural fashion. To illustrate, consider the following linear programming problem.

Maximize $Z = 20x + 31y$,

subject to

$2x + 5y \leq 16$
$4x - 3y = 6$

FIGURE A3.1
Screen shots showing the LINGO formulation and the LINGO solution report for a simple linear programming problem.



and

$$x \geq 0, \qquad y \geq 0.$$

The screen shot in the top half of Fig. A3.1 shows how this problem would be formulated with LINGO.

The first line of this formulation is just a comment describing the model. Note that the comment is preceded by an exclamation point and ended by a semicolon. This is a requirement for all comments in a LINGO formulation. The second line gives the objective function (without bothering to include the *Z* variable) and indicates that it is to be maximized. Note that each multiplication needs to be indicated by an asterisk. The objective function is ended by a semicolon, as is each of the functional constraints on the next two lines. The nonnegativity constraints are not shown in this formulation because these constraints are automatically assumed by LINGO. (If some variable *x* did *not* have a nonnegativity constraint, you would need to add @FREE(x); at the end of the formulation.)

Variables can be shown as either lowercase or uppercase, since LINGO is case-insensitive. For example, a variable x_1 can be typed in as either *x*1 or *X*1. Similarly, words can be either lowercase or uppercase (or a combination). For clarity, we will use uppercase for all reserved words that have a predefined meaning in LINGO.

Notice the menu bar at the top of the LINGO window in Fig. A3.1. The 'File' and 'Edit' menu items behave in a standard Windows fashion. To solve a model once it has been entered, click on the 'bullseye' icon. (If you are using a platform other than a Windows-based PC, instead type the GO command at the colon prompt and press the enter key.) Before attempting to solve the model, LINGO will first check whether your model has any syntax errors and, if so, will indicate where they occur. Assuming no such errors, a *solver* will begin solving the problem, during which time a *solver status* window will appear on the screen. (For linear programming models, the solver used is LINDO, which will be described in some detail in the appendix to the next chapter.) When the solver finishes, a *Solution Report* will appear on the screen.

The bottom half of Fig. A3.1 shows the solution report for our example. The *Value* column gives the optimal values of the decision variables. The first entry in the *Slack or Surplus* column shows the corresponding value of the objective function. The next two entries indicate the difference between the two sides of the respective constraints. The *Reduced Cost* and *Dual Price* columns provide some sensitivity analysis information for the problem. After discussing postoptimality analysis (including sensitivity analysis) in Sec. 4.7, we will explain what reduced costs and dual prices are while describing LINDO in Appendix 4.1. These quantities provide only a portion of the useful sensitivity analysis information. To generate a full sensitivity analysis report (such as shown in Appendix 4.1 for LINDO), the *Range* command in the LINGO menu would need to be chosen next.

Just as was illustrated with MPL in Sec. 3.7, LINGO is designed mainly for efficiently formulating very large models by simultaneously dealing with all constraints or variables of the same type. We soon will use the following example to illustrate how LINGO does this.

Example. Consider a *production-mix problem* where we are concerned with what mix of four products we should produce during the upcoming week. For each product, each unit produced requires a known amount of production time on each of three machines. Each machine has a certain number of hours of production time available per week. Each product provides a certain profit per unit produced.

Table A3.1 shows three types of data: machine-related data, product-related data, and data related to combinations of a machine and product. The objective is to determine how much to produce of each product so that total profit is maximized while not exceeding the limited production capacity of each machine.

TABLE A3.1 Data needed for the product-mix example

Machine	Production Time per Unit, Hours: Product P01	P02	P03	P04	Production Time Available per Week, Hours
Roll	1.7	2.1	1.4	2.4	28
Cut	1.1	2.5	1.7	2.6	34
Weld	1.6	1.3	1.6	0.8	21
Profit per unit	26	35	25	37	

In standard algebraic form, the structure of the linear programming model for this problem is to choose the nonnegative production levels (number of units produced during the upcoming week) for the four products so as to

$$\text{Maximize} \qquad \sum_{j=1}^{4} c_j x_j,$$

subject to

$$\sum_{j=1}^{4} a_{ij} x_j \leq b_j \qquad \text{for } i = 1, 2, 3;$$

where

x_j = production level for product P0j

c_j = unit profit for product P0j

a_{ij} = production time on machine i per unit of product P0j

b_i = production time available per week on machine i.

This model is small enough, with just 4 decision variables and 3 functional constraints, that it could be written out completely, term by term, but it would be tedious. In some similar applications, there might instead be hundreds of decision variables and functional constraints, so writing out a term-by-term version of this model each week would not be practical. LINGO provides a much more efficient and compact formulation, comparable to the above summary of the model, as we will see next.

Formulation of the Model in LINGO

This model has a repetitive nature. All the decision variables are of the same type and all the functional constraints are of the same type. LINGO uses *sets* to describe this repetitive nature.[1] The simple sets of interest in this case are

1. The set of machines, {*Roll, Cut, Weld*}.
2. The set of products, {P01, P02, P03, P04}.

[1]Order is implied in LINGO sets so, strictly speaking, they are not truly sets in the usual mathematical sense.

The attributes of interest for the members of these sets are

1. Attribute for each machine: Number of hours of production time available per week.
2. Attributes for each product: Profit per unit produced; Number of units produced per week.

Thus, the first two types of attributes are input data that will become parameters of the model, whereas the last type (number of units produced per week of the respective products) provides the decision variables for the model. Let us abbreviate these attributes as follows.

machine: ProdHoursAvail
product: Profit, Produce.

One other key type of information is the number of hours of production time that each unit of each product would use on each of the machines. This number can be viewed as an *attribute* for the members of the *set* of all combinations of a product and a machine. Since this set is derived from the two simple sets, it is referred to as a *derived set.* Let us abbreviate the attribute for members of this set as follows.

MaPr (machine, product): ProdHoursUsed

A LINGO formulation typically has three sections.

1. A SETS section that specifies the sets and their attributes. You can think of it as describing the structure of the data.
2. A DATA section that either provides the data to be used or indicates where it is to be obtained.
3. A section that provides the mathematical model itself.

We begin by showing the first two sections for the example below.

```
! LINGO3h;
! Product mix example;
! Notice: the SETS section says nothing about the number or names of
  the machines or products. That information is determined
  completely by supplied data;
SETS:
! The simple sets;
Machine: ProdHoursAvail;
Product: Profit, Produce;
! A derived set;
MaPr (Machine, Product): ProdHoursUsed;
ENDSETS
DATA:
! Get the names of the machines;
     Machine = Roll Cut Weld;
! Hours available on each machine;
ProdHoursAvail = 28  34  21;

! Get the names of the products;
  Product = P01  P02  P03  P04;
! Profit contribution per unit;
  Profit =   26   35   25   37;

! Hours needed per unit of product;
ProdHoursUsed = 1.7  2.1  1.4  2.4   ! Roll;
                1.1  2.5  1.7  2.6   ! Cut;
                1.6  1.3  1.6  0.8;  ! Weld;
ENDDATA
```

Before presenting the mathematical model itself, we need to introduce two key *set looping functions* that enable applying an operation to all members of a set by using a single statement. One is the @SUM function, which computes the sum of an expression over all members of a set. The general form of @SUM is `@SUM( set: expression)`. For every member of the set, the expression is computed, and then they are all added up. For example,

```
@SUM( Product(j): Profit(j)*Produce(j))
```

sums the expression following the colon—the unit profit of a product *times* the production rate of the product—over all members of the set preceding the colon. In particular, since this set is the set of products {Product(j) for $j = 1, 2, 3, 4$}, the sum is over the index j. Therefore, this specific @SUM function provides the objective function,

$$\sum_{j=1}^{4} c_j x_j,$$

given earlier for the model.

The second key set looping function is the @FOR function. This function is used to generate constraints over members of a set. The general form is `@FOR( set: constraint)`. For example,

```
@FOR(Machine(i):
   @SUM( Product(i): ProdHoursUsed(i, j)*Produce (j))
                          <= ProdHoursAvail (i, j);
      );
```

says to generate the constraint following the colon for each member of the set preceding the colon. (The "less than or equal to" symbol, ≤, is not on the standard keyboard, so LINGO treats the standard keyboard symbols `<=` as equivalent to ≤.) This set is the set of machines {Machine (i) for $i = 1, 2, 3$}, so this function loops over the index i. For each i, the constraint following the colon was expressed algebraically earlier as

$$\sum_{j=1}^{4} a_{ij} x_j \leq b_j.$$

Therefore, after the third section of the LINGO formulation (the mathematical model itself) is added, we obtain the complete formulation shown below:

```
! LINGO3h;
! Product mix example;
SETS:
!The simple sets;
   Machine: ProdHoursAvail;
   Product: Profit, Produce;
!A derived set;
   MaPr( Machine, Product): ProdHoursUsed;
ENDSETS
DATA:
!Get the names of the machines;
     Machine = Roll Cut Weld;
! Hours available on each machine;
ProdHoursAvail = 28  34  21;
```

```
! Get the names of the products;
     Product = P01  P02  P03  P04;
! Profit contribution per unit;
     Profit =   26   35   25   37;

! Hours needed per unit of product;
ProdHoursUsed = 1.7  2.1  1.4  2.4   ! Roll;
                1.1  2.5  1.7  2.6   ! Cut;
                1.6  1.3  1.6  0.8;  ! Weld;
ENDDATA
! Maximize total profit contribution;
MAX = @SUM( Product(i): Profit(i) * Produce(i));

! For each machine i;
   @FOR( Machine( i):
! Hours used must be <= hours available;
   @SUM( Product( j): ProdHoursUsed( i, j) * Produce( j))
                <= ProdHoursAvail;
     );
```

The model is solved by pressing the 'bullseye' button on the LINGO command bar. Pressing the 'x =' button on the command bar produces a report that looks in part as follows:

Variable	Value	Reduced Cost
PRODUCE(P01)	0.0000000	3.577921
PRODUCE(P02)	10.00000	0.0000000
PRODUCE(P03)	5.000000	0.0000000
PRODUCE(P04)	0.0000000	1.441558

Row	Slack or Surplus	Dual Price
1	475.0000	1.000000
2	0.0000000	15.25974
3	0.5000000	0.0000000
4	0.0000000	2.272727

Thus, we should produce 10 units of product P02 and 5 units of product P03, where Row 1 gives the resulting total profit of 475. Notice that this solution exactly uses the available capacity on the first and third machines (since Rows 2 and 4 give a Slack or Surplus of 0) and leaves the second machine with 0.5 hour of idleness. (We will discuss reduced costs and dual prices in Appendix 4.1 in conjunction with LINDO.)

The rows section of this report is slightly ambiguous in that you need to remember that Row 1 in the model concerns the objective function and the subsequent rows involve the constraints on machine capacities. This association can be made more clear in the report by giving names to each constraint in the model. This is done by enclosing the name in [ ], placed just in front of the constraint. See the following modified fragment of the model.

```
[Totprof] MAX = @SUM( Product: Profit * Produce);

! For each machine i;
   @FOR( Machine( i):
! Hours used must be <= hours available;
   [Capc] @SUM( Product( j): ProdHoursUsed( i, j) * Produce( j))
                <= ProdHoursAvail;
     );
```

The solution report now contains these row names.

Row	Slack or Surplus	Dual Price
TOTPROF	475.0000	1.000000
CAPC(ROLL)	0.0000000	15.25974
CAPC(CUT)	0.5000000	0.0000000
CAPC(WELD)	0.0000000	2.272727

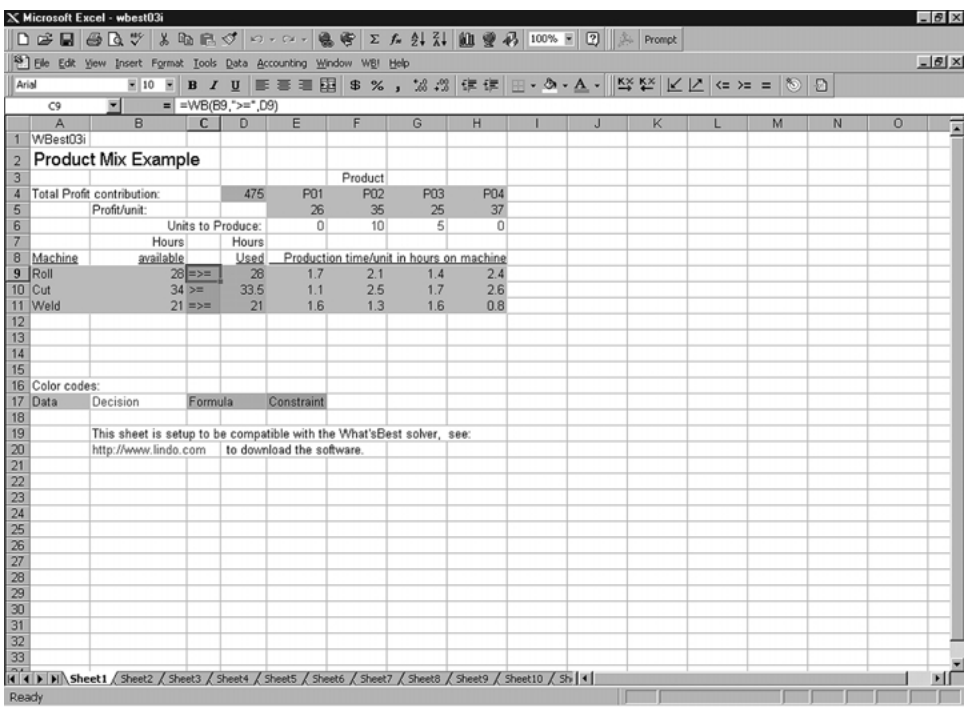
An important feature of a LINGO model like this one is that it is completely "scalable" in products and machines. In other words, if you wanted to solve another version of this product-mix problem with a different number of machines and products, you would only have to enter the new data in the DATA section. You would not need to change the SETS section or any of the equations. This conversion could be done by clerical personnel without any understanding of the model equations.

Importing and Exporting Spreadsheet Data with LINGO

The above example was completely self-contained in the sense that all the data were directly incorporated into the LINGO formulation. In some other applications, a large body of data will be stored in some source and will need to be entered into the model from that source. One popular place for storing data is spreadsheets.

LINGO has a simple function, `@OLE()`, for retrieving and placing data from and into spreadsheets. To illustrate, let us suppose the data for our product-mix problem were originally entered into a spreadsheet as shown in Fig. A3.2. For the moment we are interested only in the shaded cells

FIGURE A3.2
Screen shot showing data for the product-mix example entered in a spreadsheet.



in columns A-B and E-H. The data in these cells completely describe our little product-mix example. We want to avoid retyping these data into our LINGO model. Suppose that this spreadsheet is stored in the file d:\dirfred7\wbest03i.xls. The only part of the LINGO model that needs to be changed is the DATA section as shown below.

```
DATA:
! Get the names of the machines;
     Machine = @OLE( 'd:\dirfred7\wbest03i.xls');
! Hours available on each machine;
   ProdHoursAvail = @OLE( 'd:\dirfred7\wbest03i.xls');

! Get the names of the products;
     Product = @OLE( 'd:\dirfred7\wbest03i.xls');
! Profit contribution per unit;
     Profit = @OLE( 'd:\dirfred7\wbest03i.xls');

! Hours needed per unit of product;
   ProdHoursUsed = @OLE( 'd:\dirfred7\wbest03i.xls');

! Send the solution values back;
   @OLE( 'd:\dirfred7\wbest03i.xls') = Produce;
ENDDATA
```

The `@OLE()` function acts as your "plumbing contractor." It lets the data flow from the spreadsheet to LINGO and back to the spreadsheet. So-called Object Linking and Embedding (OLE) is a feature of the Windows operating system. LINGO exploits this feature to make a link between the LINGO model and a spreadsheet. The first five uses of `@OLE()` above illustrate that this function can be used on the right of an assignment statement to retrieve data from a spreadsheet. The last use above illustrates that this function can be placed on the left of an assignment statement to place solution results into the spreadsheet instead. Notice from Fig. A3.2 that the optimal solution has been placed back into the spreadsheet in cells E6:H6. One simple but hidden step that had to be done beforehand in the spreadsheet was to define range names for the various collections of cells containing the data. Range names can be defined in Excel by using the mouse and the *Insert, Name, Define* menu item. For example, the set of cells A9:A11 was given the range name of Machine. Similarly, the set of cells E4:H4 was given the range name Product.

Importing and Exporting from a Database with LINGO

Another common repository for data in a large firm is in a database. In a manner similar to `@OLE()`, LINGO has a connection function, `@ODBC()`, for transferring data from and to a database. This function is based around the Open DataBase Connectivity (ODBC) standard for communicating with SQL (Structured Query Language) databases. Most popular databases, such as Oracle, Paradox, DB/2, MS Access, and SQL Server, support the ODBC convention.

Let us illustrate the ODBC connection for our little product-mix example. Suppose that all the data describing our problem are stored in a database called acces03j. The modification required in the LINGO model is almost trivial. Only the DATA section needs to be changed, as illustrated in the following fragment from the LINGO model.

```
DATA:
! Get the names of the machines and available hours;
   Machine, ProdHoursAvail = @ODBC( 'acces03j');

! Get the names of the products and profits;
   Product, Profit = @ODBC( 'acces03j');
```

```
! Hours needed per unit of product;
   ProdHoursUsed   = @ODBC( 'acces03j');

! Send the solution values back;
   @ODBC( 'acces03j') = Produce;
ENDDATA
```

Notice that, similar to the spreadsheet-based model, the size of the model in terms of the number of variables and constraints is determined completely by what is found in the database. The LINGO model automatically adjusts to what is found in the database.

Now let us show what is in the database considered above. It contains three related tables. We give these tables names to match those in the LINGO model, namely, 'Machine,' to hold machine-related data, 'Product,' to hold product-related data, and 'MaPr,' to hold data related to combinations of machines and products. Here is what the tables look like on the screen:

Machine

Machine	ProdHoursAvail
Roll	28
Cut	34
Weld	21

Product

Product	Profit	Produce
P01	26	
P02	35	
P03	25	
P04	37	

MaPr

Machine	Product	ProdHoursUsed
Roll	P01	1.7
Roll	P02	2.1
Roll	P03	1.4
Roll	P04	2.4
Cut	P01	1.1
Cut	P02	2.5
Cut	P03	1.7
Cut	P04	2.6
Weld	P01	1.6
Weld	P02	1.3
Weld	P03	1.6
Weld	P04	0.8

Notice that the 'Produce' column has been left blank in the Product table. Once we solve the model, the 'Produce' amounts get inserted into the database and the Product table looks as follows:

Product

Product	Profit	Produce
P01	26	0
P02	35	10
P03	25	5
P04	37	0

There is one complication in using ODBC in Windows 95. The user must "register" the database with the Windows ODBC administrator. One does this by accessing (with mouse clicks) the My Computer/Control Panel/ODBC32 window. Once there, the user must give a name to the database (which may differ from the actual name of the file in which the data tables reside) and specify the directory in which the database file resides. It is this registered name that should be used in the LINGO model. Because the database has been registered, you did not see a directory specification in the `@ODBC( 'acces03j')` in the LINGO model. The ODBC manager knows the location of the database just from its name.

More about LINGO

Only some of the capabilities of LINGO have been illustrated in this appendix. More details can be found in the documentation that accompanies LINGO when it is downloaded. LINGO is available in a variety of sizes. The smallest version is the *demo* version that can be downloaded from www.lindo.com. It is designed for textbook-sized problems (currently a maximum of 150 functional constraints and 300 decision variables). However, the largest version (called the *extended* version) is limited only by the storage space available. Tens of thousands of functional constraints and hundreds of thousands of decision variables are not unusual.

If you would like to see how LINGO can formulate a huge model like the production planning example introduced in Sec. 3.7, a supplement to this appendix on the book's website, www.mhhe.com/hillier, shows the LINGO formulation of this example. By reducing the number of products, plants, machines, and months, the supplement also introduces actual data into the formulation and then shows the complete solution. The supplement goes on to discuss and illustrate the debugging and verification of this large model. The supplement also describes further how to retrieve data from external files (including spreadsheets) and how to insert results in existing files.

In addition to this supplement, the CD-ROM includes both a LINGO tutorial and LINGO/LINDO files with numerous examples of LINGO formulations.

SELECTED REFERENCES

1. Anderson, D. R., D. J. Sweeney, and T. A. Williams: *An Introduction to Management Science,* 9th ed., West, St. Paul, MN, 2000, chaps. 2, 4.
2. Gass, S.: *An Illustrated Guide to Linear Programming,* Dover Publications, New York, 1990.
3. Hillier, F. S., M. S. Hillier, and G. J. Lieberman: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets,* Irwin/McGraw-Hill, Burr Ridge, IL, 2000, chaps. 2, 3.

4. *LINGO User's Guide,* LINDO Systems, Inc., Chicago, IL, e-mail: info@lindo.com, 1999.
5. *MPL Modeling System* (*Release* 4.0) manual, Maximal Software, Inc., Arlington, VA, e-mail: info@maximal-usa.com, 1998.
6. Williams, H. P.: *Model Building in Mathematical Programming,* 3d ed., Wiley, New York, 1990.

LEARNING AIDS FOR THIS CHAPTER IN YOUR OR COURSEWARE

A Demonstration Example in OR Tutor:

Graphical Method

An Excel Add-In:

Premium Solver

"Ch. 3—Intro to LP" Files for Solving the Examples:

Excel File
LINGO/LINDO File
MPL/CPLEX File

Supplement to Appendix 3.1:

More about LINGO (appears on the book's website, www.mhhe.com/hillier).

See Appendix 1 for documentation of the software.

PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

D: The demonstration example listed above may be helpful.
C: Use the computer to solve the problem by applying the simplex method. The available software options for doing this include the Excel Solver or Premium Solver (Sec. 3.6), MPL/CPLEX (Sec. 3.7), LINGO (Appendix 3.1), and LINDO (Appendix 4.1), but follow any instructions given by your instructor regarding the option to use.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

D **3.1-1.*** For each of the following constraints, draw a separate graph to show the nonnegative solutions that satisfy this constraint.
(a) $x_1 + 3x_2 \leq 6$
(b) $4x_1 + 3x_2 \leq 12$
(c) $4x_1 + x_2 \leq 8$
(d) Now combine these constraints into a single graph to show the feasible region for the entire set of functional constraints plus nonnegativity constraints.

D **3.1-2.** Consider the following objective function for a linear programming model:

Maximize $Z = 2x_1 + 3x_2$

(a) Draw a graph that shows the corresponding objective function lines for $Z = 6$, $Z = 12$, and $Z = 18$.
(b) Find the slope-intercept form of the equation for each of these three objective function lines. Compare the slope for these three lines. Also compare the intercept with the x_2 axis.

3.1-3. Consider the following equation of a line:

$20x_1 + 40x_2 = 400$

(a) Find the slope-intercept form of this equation.

(b) Use this form to identify the slope and the intercept with the x_2 axis for this line.
(c) Use the information from part (*b*) to draw a graph of this line.

D **3.1-4.*** Use the graphical method to solve the problem:

Maximize $Z = 2x_1 + x_2$,

subject to

$$\begin{aligned} x_2 &\le 10 \\ 2x_1 + 5x_2 &\le 60 \\ x_1 + x_2 &\le 18 \\ 3x_1 + x_2 &\le 44 \end{aligned}$$

and

$$x_1 \ge 0, \quad x_2 \ge 0.$$

D **3.1-5.** Use the graphical method to solve the problem:

Maximize $Z = 10x_1 + 20x_2$,

subject to

$$\begin{aligned} -x_1 + 2x_2 &\le 15 \\ x_1 + x_2 &\le 12 \\ 5x_1 + 3x_2 &\le 45 \end{aligned}$$

and

$$x_1 \ge 0, \quad x_2 \ge 0.$$

3.1-6. The Whitt Window Company is a company with only three employees which makes two different kinds of hand-crafted windows: a wood-framed and an aluminum-framed window. They earn \$60 profit for each wood-framed window and \$30 profit for each aluminum-framed window. Doug makes the wood frames, and can make 6 per day. Linda makes the aluminum frames, and can make 4 per day. Bob forms and cuts the glass, and can make 48 square feet of glass per day. Each wood-framed window uses 6 square feet of glass and each aluminum-framed window uses 8 square feet of glass.

The company wishes to determine how many windows of each type to produce per day to maximize total profit.

(a) Describe the analogy between this problem and the Wyndor Glass Co. problem discussed in Sec. 3.1. Then construct and fill in a table like Table 3.1 for this problem, identifying both the activities and the resources.
(b) Formulate a linear programming model for this problem.
D **(c)** Use the graphical model to solve this model.
(d) A new competitor in town has started making wood-framed windows as well. This may force the company to lower the price they charge and so lower the profit made for each wood-framed window. How would the optimal solution change (if at all) if the profit per wood-framed window decreases from \$60 to \$40? From \$60 to \$20?
(e) Doug is considering lowering his working hours, which would decrease the number of wood frames he makes per day. How would the optimal solution change if he makes only 5 wood frames per day?

3.1-7. The Apex Television Company has to decide on the number of 27- and 20-inch sets to be produced at one of its factories. Market research indicates that at most 40 of the 27-inch sets and 10 of the 20-inch sets can be sold per month. The maximum number of work-hours available is 500 per month. A 27-inch set requires 20 work-hours and a 20-inch set requires 10 work-hours. Each 27-inch set sold produces a profit of \$120 and each 20-inch set produces a profit of \$80. A wholesaler has agreed to purchase all the television sets produced if the numbers do not exceed the maxima indicated by the market research.

(a) Formulate a linear programming model for this problem.
D **(b)** Use the graphical method to solve this model.

3.1-8. The WorldLight Company produces two light fixtures (products 1 and 2) that require both metal frame parts and electrical components. Management wants to determine how many units of each product to produce so as to maximize profit. For each unit of product 1, 1 unit of frame parts and 2 units of electrical components are required. For each unit of product 2, 3 units of frame parts and 2 units of electrical components are required. The company has 200 units of frame parts and 300 units of electrical components. Each unit of product 1 gives a profit of \$1, and each unit of product 2, up to 60 units, gives a profit of \$2. Any excess over 60 units of product 2 brings no profit, so such an excess has been ruled out.

(a) Formulate a linear programming model for this problem.
D **(b)** Use the graphical method to solve this model. What is the resulting total profit?

3.1-9. The Primo Insurance Company is introducing two new product lines: special risk insurance and mortgages. The expected profit is \$5 per unit on special risk insurance and \$2 per unit on mortgages.

Management wishes to establish sales quotas for the new product lines to maximize total expected profit. The work requirements are as follows:

Department	Work-Hours per Unit		Work-Hours Available
	Special Risk	Mortgage	
Underwriting	3	2	2400
Administration	0	1	800
Claims	2	0	1200

(a) Formulate a linear programming model for this problem.
D **(b)** Use the graphical method to solve this model.
(c) Verify the exact value of your optimal solution from part (*b*) by solving algebraically for the simultaneous solution of the relevant two equations.

3.1-10. Weenies and Buns is a food processing plant which manufactures hot dogs and hot dog buns. They grind their own flour for the hot dog buns at a maximum rate of 200 pounds per week. Each hot dog bun requires 0.1 pound of flour. They currently have a contract with Pigland, Inc., which specifies that a delivery of 800 pounds of pork product is delivered every Monday. Each hot dog requires $\frac{1}{4}$ pound of pork product. All the other ingredients in the hot dogs and hot dog buns are in plentiful supply. Finally, the labor force at Weenies and Buns consists of 5 employees working full time (40 hours per week each). Each hot dog requires 3 minutes of labor, and each hot dog bun requires 2 minutes of labor. Each hot dog yields a profit of $0.20, and each bun yields a profit of $0.10.

Weenies and Buns would like to know how many hot dogs and how many hot dog buns they should produce each week so as to achieve the highest possible profit.
(a) Formulate a linear programming model for this problem.
D **(b)** Use the graphical method to solve this model.

3.1-11.* The Omega Manufacturing Company has discontinued the production of a certain unprofitable product line. This act created considerable excess production capacity. Management is considering devoting this excess capacity to one or more of three products; call them products 1, 2, and 3. The available capacity on the machines that might limit output is summarized in the following table:

Machine Type	Available Time (Machine Hours per Week)
Milling machine	500
Lathe	350
Grinder	150

The number of machine hours required for each unit of the respective products is

Productivity coefficient (in machine hours per unit)

Machine Type	Product 1	Product 2	Product 3
Milling machine	9	3	5
Lathe	5	4	0
Grinder	3	0	2

The sales department indicates that the sales potential for products 1 and 2 exceeds the maximum production rate and that the sales potential for product 3 is 20 units per week. The unit profit would be $50, $20, and $25, respectively, on products 1, 2, and 3. The objective is to determine how much of each product Omega should produce to maximize profit.
(a) Formulate a linear programming model for this problem.
C **(b)** Use a computer to solve this model by the simplex method.

D **3.1-12.** Consider the following problem, where the value of c_1 has not yet been ascertained.

$$\text{Maximize} \qquad Z = c_1x_1 + x_2,$$

subject to

$$\begin{aligned} x_1 + \ \ x_2 &\le \ \ 6 \\ x_1 + 2x_2 &\le 10 \end{aligned}$$

and

$$x_1 \ge 0, \qquad x_2 \ge 0.$$

Use graphical analysis to determine the optimal solution(s) for (x_1, x_2) for the various possible values of $c_1 (-\infty < c_1 < \infty)$.

D **3.1-13.** Consider the following problem, where the value of k has not yet been ascertained.

$$\text{Maximize} \qquad Z = x_1 + 2x_2,$$

subject to

$$\begin{aligned} -x_1 + x_2 &\le 2 \\ x_2 &\le 3 \\ kx_1 + x_2 &\le 2k + 3, \qquad \text{where } k \ge 0 \end{aligned}$$

and

$$x_1 \ge 0, \qquad x_2 \ge 0.$$

The solution currently being used is $x_1 = 2$, $x_2 = 3$. Use graphical analysis to determine the values of k such that this solution actually is optimal.

D **3.1-14.** Consider the following problem, where the values of c_1 and c_2 have not yet been ascertained.

$$\text{Maximize} \qquad Z = c_1x_1 + c_2x_2,$$

subject to

$$\begin{aligned} 2x_1 + \ \ x_2 &\le 11 \\ -x_1 + 2x_2 &\le \ \ 2 \end{aligned}$$

and

$$x_1 \ge 0, \qquad x_2 \ge 0.$$

Use graphical analysis to determine the optimal solution(s) for (x_1, x_2) for the various possible values of c_1 and c_2. (*Hint:* Sepa-

rate the cases where $c_2 = 0$, $c_2 > 0$, and $c_2 < 0$. For the latter two cases, focus on the ratio of c_1 to c_2.)

3.2-1. The following table summarizes the key facts about two products, A and B, and the resources, Q, R, and S, required to produce them.

Resource	Resource Usage per Unit Produced		Amount of Resource Available
	Product A	Product B	
Q	2	1	2
R	1	2	2
S	3	3	4
Profit per unit	3	2	

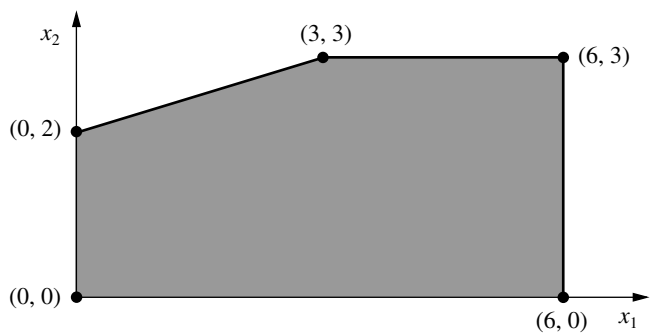
All the assumptions of linear programming hold.

(a) Formulate a linear programming model for this problem.

D **(b)** Solve this model graphically.

(c) Verify the exact value of your optimal solution from part (*b*) by solving algebraically for the simultaneous solution of the relevant two equations.

3.2-2. The shaded area in the following graph represents the feasible region of a linear programming problem whose objective function is to be maximized.



Label each of the following statements as True or False, and then justify your answer based on the graphical method. In each case, give an example of an objective function that illustrates your answer.

(a) If (3, 3) produces a larger value of the objective function than (0, 2) and (6, 3), then (3, 3) must be an optimal solution.

(b) If (3, 3) is an optimal solution and multiple optimal solutions exist, then either (0, 2) or (6, 3) must also be an optimal solution.

(c) The point (0, 0) cannot be an optimal solution.

3.2-3.* This is your lucky day. You have just won a $10,000 prize. You are setting aside $4,000 for taxes and partying expenses, but you have decided to invest the other $6,000. Upon hearing this news, two different friends have offered you an opportunity to become a partner in two different entrepreneurial ventures, one planned by each friend. In both cases, this investment would involve expending some of your time next summer as well as putting up cash. Becoming a *full* partner in the first friend's venture would require an investment of $5,000 and 400 hours, and your estimated profit (ignoring the value of your time) would be $4,500. The corresponding figures for the second friend's venture are $4,000 and 500 hours, with an estimated profit to you of $4,500. However, both friends are flexible and would allow you to come in at any *fraction* of a full partnership you would like. If you choose a fraction of a full partnership, all the above figures given for a full partnership (money investment, time investment, and your profit) would be multiplied by this same fraction.

Because you were looking for an interesting summer job anyway (maximum of 600 hours), you have decided to participate in one or both friends' ventures in whichever combination would maximize your total estimated profit. You now need to solve the problem of finding the best combination.

(a) Describe the analogy between this problem and the Wyndor Glass Co. problem discussed in Sec. 3.1. Then construct and fill in a table like Table 3.1 for this problem, identifying both the activities and the resources.

(b) Formulate a linear programming model for this problem.

D **(c)** Use the graphical method to solve this model. What is your total estimated profit?

D **3.2-4.** Use the graphical method to find all optimal solutions for the following model:

$$\text{Maximize} \quad Z = 500x_1 + 300x_2,$$

subject to

$$\begin{aligned} 15x_1 + 5x_2 &\le 300 \\ 10x_1 + 6x_2 &\le 240 \\ 8x_1 + 12x_2 &\le 450 \end{aligned}$$

and

$$x_1 \ge 0, \qquad x_2 \ge 0.$$

D **3.2-5.** Use the graphical method to demonstrate that the following model has no feasible solutions.

$$\text{Maximize} \quad Z = 5x_1 + 7x_2,$$

subject to

$$\begin{aligned} 2x_1 - x_2 &\le -1 \\ -x_1 + 2x_2 &\le -1 \end{aligned}$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0.$$

D **3.2-6.** Suppose that the following constraints have been provided for a linear programming model.

$$\begin{aligned} -x_1 + 3x_2 &\leq 30 \\ -3x_1 + x_2 &\leq 30 \end{aligned}$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0.$$

(a) Demonstrate that the feasible region is unbounded.
(b) If the objective is to maximize $Z = -x_1 + x_2$, does the model have an optimal solution? If so, find it. If not, explain why not.
(c) Repeat part (*b*) when the objective is to maximize $Z = x_1 - x_2$.
(d) For objective functions where this model has no optimal solution, does this mean that there are no good solutions according to the model? Explain. What probably went wrong when formulating the model?

3.3-1. Reconsider Prob. 3.2-3. Indicate why each of the four assumptions of linear programming (Sec. 3.3) appears to be reasonably satisfied for this problem. Is one assumption more doubtful than the others? If so, what should be done to take this into account?

3.3-2. Consider a problem with two decision variables, x_1 and x_2, which represent the levels of activities 1 and 2, respectively. For each variable, the permissible values are 0, 1, and 2, where the feasible combinations of these values for the two variables are determined from a variety of constraints. The objective is to maximize a certain measure of performance denoted by Z. The values of Z for the possibly feasible values of (x_1, x_2) are estimated to be those given in the following table:

	x_2		
x_1	**0**	**1**	**2**
0	0	4	8
1	3	8	13
2	6	12	18

Based on this information, indicate whether this problem completely satisfies each of the four assumptions of linear programming. Justify your answers.

3.4-1.* For each of the four assumptions of linear programming discussed in Sec. 3.3, write a one-paragraph analysis of how well you feel it applies to each of the following examples given in Sec. 3.4:
(a) Design of radiation therapy (Mary).
(b) Regional planning (Southern Confederation of Kibbutzim).
(c) Controlling air pollution (Nori & Leets Co.).

3.4-2. For each of the four assumptions of linear programming discussed in Sec. 3.3, write a one-paragraph analysis of how well it applies to each of the following examples given in Sec. 3.4.
(a) Reclaiming solid wastes (Save-It Co.).
(b) Personnel scheduling (Union Airways).
(c) Distributing goods through a distribution network (Distribution Unlimited Co.).

D **3.4-3.** Use the graphical method to solve this problem:

Maximize $\quad Z = 15x_1 + 20x_2,$

subject to

$$\begin{aligned} x_1 + 2x_2 &\geq 10 \\ 2x_1 - 3x_2 &\leq 6 \\ x_1 + x_2 &\geq 6 \end{aligned}$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0.$$

D **3.4-4.** Use the graphical method to solve this problem:

Minimize $\quad Z = 3x_1 + 2x_2,$

subject to

$$\begin{aligned} x_1 + 2x_2 &\leq 12 \\ 2x_1 + 3x_2 &= 12 \\ 2x_1 + x_2 &\geq 8 \end{aligned}$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0.$$

D **3.4-5.** Consider the following problem, where the value of c_1 has not yet been ascertained.

Maximize $\quad Z = c_1x_1 + 2x_2,$

subject to

$$\begin{aligned} 4x_1 + x_2 &\leq 12 \\ x_1 - x_2 &\geq 2 \end{aligned}$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0.$$

Use graphical analysis to determine the optimal solution(s) for (x_1, x_2) for the various possible values of c_1.

D **3.4-6.** Consider the following model:

Minimize $\quad Z = 40x_1 + 50x_2,$

subject to

$$2x_1 + 3x_2 \geq 30$$
$$x_1 + x_2 \geq 12$$
$$2x_1 + x_2 \geq 20$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0.$$

(a) Use the graphical method to solve this model.
(b) How does the optimal solution change if the objective function is changed to $Z = 40x_1 + 70x_2$?
(c) How does the optimal solution change if the third functional constraint is changed to $2x_1 + x_2 \geq 15$?

3.4-7. Ralph Edmund loves steaks and potatoes. Therefore, he has decided to go on a steady diet of only these two foods (plus some liquids and vitamin supplements) for all his meals. Ralph realizes that this isn't the healthiest diet, so he wants to make sure that he eats the right quantities of the two foods to satisfy some key nutritional requirements. He has obtained the following nutritional and cost information:

Ingredient	Grams of Ingredient per Serving		Daily Requirement (Grams)
	Steak	Potatoes	
Carbohydrates	5	15	$\geq$ 50
Protein	20	5	$\geq$ 40
Fat	15	2	$\leq$ 60
Cost per serving	\$4	\$2	

Ralph wishes to determine the number of daily servings (may be fractional) of steak and potatoes that will meet these requirements at a minimum cost.
(a) Formulate a linear programming model for this problem.
D **(b)** Use the graphical method to solve this model.
C **(c)** Use a computer to solve this model by the simplex method.

3.4-8. Dwight is an elementary school teacher who also raises pigs for supplemental income. He is trying to decide what to feed his pigs. He is considering using a combination of pig feeds available from local suppliers. He would like to feed the pigs at minimum cost while also making sure each pig receives an adequate supply of calories and vitamins. The cost, calorie content, and vitamin content of each feed is given in the table below.

Contents	Feed Type A	Feed Type B
Calories (per pound)	800	1,000
Vitamins (per pound)	140 units	70 units
Cost (per pound)	\$0.40	\$0.80

Each pig requires at least 8,000 calories per day and at least 700 units of vitamins. A further constraint is that no more than one-third of the diet (by weight) can consist of Feed Type A, since it contains an ingredient which is toxic if consumed in too large a quantity.
(a) Formulate a linear programming model for this problem.
D **(b)** Use the graphical method to solve this model. What is the resulting daily cost per pig?

3.4-9. Web Mercantile sells many household products through an on-line catalog. The company needs substantial warehouse space for storing its goods. Plans now are being made for leasing warehouse storage space over the next 5 months. Just how much space will be required in each of these months is known. However, since these space requirements are quite different, it may be most economical to lease only the amount needed each month on a month-by-month basis. On the other hand, the additional cost for leasing space for additional months is much less than for the first month, so it may be less expensive to lease the maximum amount needed for the entire 5 months. Another option is the intermediate approach of changing the total amount of space leased (by adding a new lease and/or having an old lease expire) at least once but not every month.

The space requirement and the leasing costs for the various leasing periods are as follows:

Month	Required Space (Sq. Ft.)
1	30,000
2	20,000
3	40,000
4	10,000
5	50,000

Leasing Period (Months)	Cost per Sq. Ft. Leased
1	\$ 65
2	\$100
3	\$135
4	\$160
5	\$190

The objective is to minimize the total leasing cost for meeting the space requirements.
(a) Formulate a linear programming model for this problem.
C **(b)** Solve this model by the simplex method.

3.4-10. Larry Edison is the director of the Computer Center for Buckly College. He now needs to schedule the staffing of the center. It is open from 8 A.M. until midnight. Larry has monitored the usage of the center at various times of the day, and determined that the following number of computer consultants are required:

Time of Day	Minimum Number of Consultants Required to Be on Duty
8 A.M.–noon	4
Noon–4 P.M.	8
4 P.M.–8 P.M.	10
8 P.M.–midnight	6

Two types of computer consultants can be hired: full-time and part-time. The full-time consultants work for 8 consecutive hours in any of the following shifts: morning (8 A.M.–4 P.M.), afternoon (noon–8 P.M.), and evening (4 P.M.–midnight). Full-time consultants are paid $14 per hour.

Part-time consultants can be hired to work any of the four shifts listed in the above table. Part-time consultants are paid $12 per hour.

An additional requirement is that during every time period, there must be at least 2 full-time consultants on duty for every part-time consultant on duty.

Larry would like to determine how many full-time and how many part-time workers should work each shift to meet the above requirements at the minimum possible cost.

(a) Formulate a linear programming model for this problem.

C **(b)** Solve this model by the simplex method.

3.4-11.* The Medequip Company produces precision medical diagnostic equipment at two factories. Three medical centers have placed orders for this month's production output. The table to the right shows what the cost would be for shipping each unit from each factory to each of these customers. Also shown are the number of units that will be produced at each factory and the number of units ordered by each customer. (Go to the next column.)

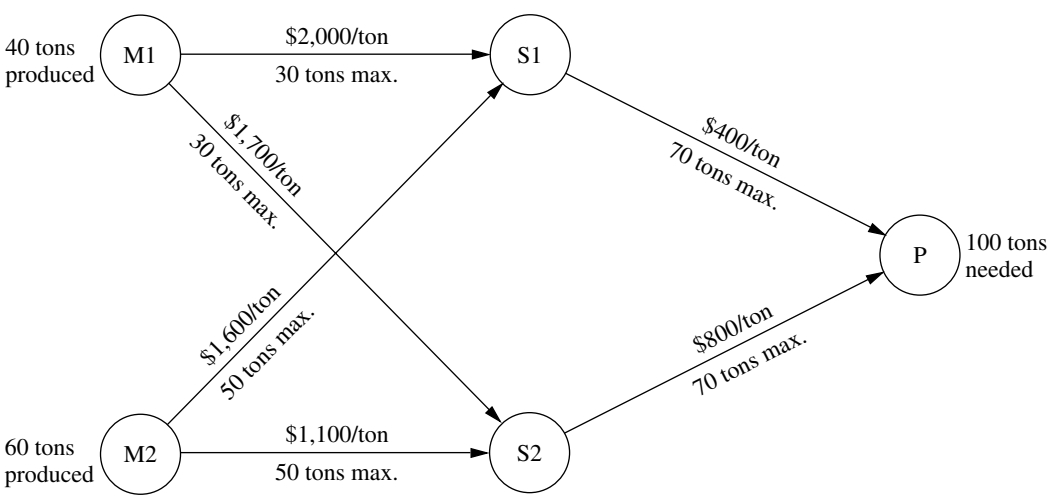
From \ To	Unit Shipping Cost			Output
	Customer 1	Customer 2	Customer 3	
Factory 1	$600	$800	$700	400 units
Factory 2	$400	$900	$600	500 units
Order size	300 units	200 units	400 units	

A decision now needs to be made about the shipping plan for how many units to ship from each factory to each customer.

(a) Formulate a linear programming model for this problem.

C **(b)** Solve this model by the simplex method.

3.4-12. The Fagersta Steelworks currently is working two mines to obtain its iron ore. This iron ore is shipped to either of two storage facilities. When needed, it then is shipped on to the company's steel plant. The diagram below depicts this distribution network, where M1 and M2 are the two mines, S1 and S2 are the two storage facilities, and P is the steel plant. The diagram also shows the monthly amounts produced at the mines and needed at the plant, as well as the shipping cost and the maximum amount that can be shipped per month through each shipping lane. (Go to the left column below the diagram.)



Management now wants to determine the most economical plan for shipping the iron ore from the mines through the distribution network to the steel plant.

(a) Formulate a linear programming model for this problem.

C **(b)** Solve this model by the simplex method.

3.4-13.* Al Ferris has $60,000 that he wishes to invest now in order to use the accumulation for purchasing a retirement annuity in 5 years. After consulting with his financial adviser, he has been offered four types of fixed-income investments, which we will label as investments A, B, C, D.

Investments A and B are available at the beginning of each of the next 5 years (call them years 1 to 5). Each dollar invested in A at the beginning of a year returns \$1.40 (a profit of \$0.40) 2 years later (in time for immediate reinvestment). Each dollar invested in B at the beginning of a year returns \$1.70 three years later.

Investments C and D will each be available at one time in the future. Each dollar invested in C at the beginning of year 2 returns \$1.90 at the end of year 5. Each dollar invested in D at the beginning of year 5 returns \$1.30 at the end of year 5.

Al wishes to know which investment plan maximizes the amount of money that can be accumulated by the beginning of year 6.

(a) All the functional constraints for this problem can be expressed as equality constraints. To do this, let A_t, B_t, C_t, and D_t be the amount invested in investment A, B, C, and D, respectively, at the beginning of year t for each t where the investment is available and will mature by the end of year 5. Also let R_t be the number of available dollars *not* invested at the beginning of year t (and so available for investment in a later year). Thus, the amount invested at the beginning of year t *plus* R_t must equal the number of dollars available for investment at that time. Write such an equation in terms of the relevant variables above for the beginning of each of the 5 years to obtain the five functional constraints for this problem.

(b) Formulate a complete linear programming model for this problem.

C **(c)** Solve this model by the simplex model.

3.4-14. The Metalco Company desires to blend a new alloy of 40 percent tin, 35 percent zinc, and 25 percent lead from several available alloys having the following properties:

	Alloy				
Property	**1**	**2**	**3**	**4**	**5**
Percentage of tin	60	25	45	20	50
Percentage of zinc	10	15	45	50	40
Percentage of lead	30	60	10	30	10
Cost (\$/lb)	22	20	25	24	27

The objective is to determine the proportions of these alloys that should be blended to produce the new alloy at a minimum cost.

(a) Formulate a linear programming model for this problem.

C **(b)** Solve this model by the simplex method.

3.4-15. The Weigelt Corporation has three branch plants with excess production capacity. Fortunately, the corporation has a new product ready to begin production, and all three plants have this capability, so some of the excess capacity can be used in this way. This product can be made in three sizes—large, medium, and small—that yield a net unit profit of \$420, \$360, and \$300, respectively. Plants 1, 2, and 3 have the excess capacity to produce 750, 900, and 450 units per day of this product, respectively, regardless of the size or combination of sizes involved.

The amount of available in-process storage space also imposes a limitation on the production rates of the new product. Plants 1, 2, and 3 have 13,000, 12,000, and 5,000 square feet, respectively, of in-process storage space available for a day's production of this product. Each unit of the large, medium, and small sizes produced per day requires 20, 15, and 12 square feet, respectively.

Sales forecasts indicate that if available, 900, 1,200, and 750 units of the large, medium, and small sizes, respectively, would be sold per day.

At each plant, some employees will need to be laid off unless most of the plant's excess production capacity can be used to produce the new product. To avoid layoffs if possible, management has decided that the plants should use the same percentage of their excess capacity to produce the new product.

Management wishes to know how much of each of the sizes should be produced by each of the plants to maximize profit.

(a) Formulate a linear programming model for this problem.

C **(b)** Solve this model by the simplex method.

3.4-16* A cargo plane has three compartments for storing cargo: front, center, and back. These compartments have capacity limits on both *weight* and *space,* as summarized below:

Compartment	**Weight Capacity (Tons)**	**Space Capacity (Cubic Feet)**
Front	12	7,000
Center	18	9,000
Back	10	5,000

Furthermore, the weight of the cargo in the respective compartments must be the same proportion of that compartment's weight capacity to maintain the balance of the airplane.

The following four cargoes have been offered for shipment on an upcoming flight as space is available:

Cargo	**Weight (Tons)**	**Volume (Cubic Feet/Ton)**	**Profit (\$/Ton)**
1	20	500	320
2	16	700	400
3	25	600	360
4	13	400	290

Any portion of these cargoes can be accepted. The objective is to determine how much (if any) of each cargo should be accepted and

how to distribute each among the compartments to maximize the total profit for the flight.

(a) Formulate a linear programming model for this problem.

C **(b)** Solve this model by the simplex method to find one of its multiple optimal solutions.

3.4-17. Comfortable Hands is a company which features a product line of winter gloves for the entire family—men, women, and children. They are trying to decide what mix of these three types of gloves to produce.

Comfortable Hands' manufacturing labor force is unionized. Each full-time employee works a 40-hour week. In addition, by union contract, the number of full-time employees can never drop below 20. Nonunion part-time workers can also be hired with the following union-imposed restrictions: (1) each part-time worker works 20 hours per week, and (2) there must be at least 2 full-time employees for each part-time employee.

All three types of gloves are made out of the same 100 percent genuine cowhide leather. Comfortable Hands has a long-term contract with a supplier of the leather, and receives a 5,000 square feet shipment of the material each week. The material requirements and labor requirements, along with the *gross profit* per glove sold (not considering labor costs) is given in the following table.

Glove	Material Required (Square Feet)	Labor Required (Minutes)	Gross Profit (per Pair)
Men's	2	30	$8
Women's	1.5	45	$10
Children's	1	40	$6

Each full-time employee earns $13 per hour, while each part-time employee earns $10 per hour. Management wishes to know what mix of each of the three types of gloves to produce per week, as well as how many full-time and how many part-time workers to employ. They would like to maximize their *net profit*—their gross profit from sales minus their labor costs.

(a) Formulate a linear programming model for this problem.

C **(b)** Solve this model by the simplex method.

3.4-18. Oxbridge University maintains a powerful mainframe computer for research use by its faculty, Ph.D. students, and research associates. During all working hours, an operator must be available to operate and maintain the computer, as well as to perform some programming services. Beryl Ingram, the director of the computer facility, oversees the operation.

It is now the beginning of the fall semester, and Beryl is confronted with the problem of assigning different working hours to her operators. Because all the operators are currently enrolled in the university, they are available to work only a limited number of hours each day, as shown in the following table.

		Maximum Hours of Availability				
Operators	Wage Rate	Mon.	Tue.	Wed.	Thurs.	Fri.
K. C.	$10.00/hour	6	0	6	0	6
D. H.	$10.10/hour	0	6	0	6	0
H. B.	$ 9.90/hour	4	8	4	0	4
S. C.	$ 9.80/hour	5	5	5	0	5
K. S.	$10.80/hour	3	0	3	8	0
N. K.	$11.30/hour	0	0	0	6	2

There are six operators (four undergraduate students and two graduate students). They all have different wage rates because of differences in their experience with computers and in their programming ability. The above table shows their wage rates, along with the maximum number of hours that each can work each day.

Each operator is guaranteed a certain minimum number of hours per week that will maintain an adequate knowledge of the operation. This level is set arbitrarily at 8 hours per week for the undergraduate students (K. C., D. H., H. B., and S. C.) and 7 hours per week for the graduate students (K. S. and N. K.).

The computer facility is to be open for operation from 8 A.M. to 10 P.M. Monday through Friday with exactly one operator on duty during these hours. On Saturdays and Sundays, the computer is to be operated by other staff.

Because of a tight budget, Beryl has to minimize cost. She wishes to determine the number of hours she should assign to each operator on each day.

(a) Formulate a linear programming model for this problem.

C **(b)** Solve this model by the simplex method.

3.4-19. Slim-Down Manufacturing makes a line of nutritionally complete, weight-reduction beverages. One of their products is a strawberry shake which is designed to be a complete meal. The strawberry shake consists of several ingredients. Some information about each of these ingredients is given below.

Ingredient	Calories from Fat (per tbsp)	Total Calories (per tbsp)	Vitamin Content (mg/ tbsp)	Thickeners (mg/ tbsp)	Cost (¢/ tbsp)
Strawberry flavoring	1	50	20	3	10
Cream	75	100	0	8	8
Vitamin supplement	0	0	50	1	25
Artificial sweetener	0	120	0	2	15
Thickening agent	30	80	2	25	6

The nutritional requirements are as follows. The beverage must total between 380 and 420 calories (inclusive). No more than 20 percent of the total calories should come from fat. There must be at least 50 milligrams (mg) of vitamin content. For taste reasons, there must be at least 2 tablespoons (tbsp) of strawberry flavoring for each tablespoon of artificial sweetener. Finally, to maintain proper thickness, there must be exactly 15 mg of thickeners in the beverage.

Management would like to select the quantity of each ingredient for the beverage which would minimize cost while meeting the above requirements.

(a) Formulate a linear programming model for this problem.

C **(b)** Solve this model by the simplex method.

3.4-20. Joyce and Marvin run a day care for preschoolers. They are trying to decide what to feed the children for lunches. They would like to keep their costs down, but also need to meet the nutritional requirements of the children. They have already decided to go with peanut butter and jelly sandwiches, and some combination of graham crackers, milk, and orange juice. The nutritional content of each food choice and its cost are given in the table below.

Food Item	Calories from Fat	Total Calories	Vitamin C (mg)	Protein (g)	Cost (¢)
Bread (1 slice)	10	70	0	3	5
Peanut butter (1 tbsp)	75	100	0	4	4
Strawberry jelly (1 tbsp)	0	50	3	0	7
Graham cracker (1 cracker)	20	60	0	1	8
Milk (1 cup)	70	150	2	8	15
Juice (1 cup)	0	100	120	1	35

The nutritional requirements are as follows. Each child should receive between 400 and 600 calories. No more than 30 percent of the total calories should come from fat. Each child should consume at least 60 milligrams (mg) of vitamin C and 12 grams (g) of protein. Furthermore, for practical reasons, each child needs exactly 2 slices of bread (to make the sandwich), at least twice as much peanut butter as jelly, and at least 1 cup of liquid (milk and/or juice).

Joyce and Marvin would like to select the food choices for each child which minimize cost while meeting the above requirements.

(a) Formulate a linear programming model for this problem.

C **(b)** Solve this model by the simplex method.

3.5-1. Read the article footnoted in Sec. 3.5 that describes the first case study presented in that section: "Choosing the Product Mix at Ponderosa Industrial."

(a) Describe the two factors which, according to the article, often hinder the use of optimization models by managers.

(b) Section 3.5 indicates without elaboration that using linear programming at Ponderosa "led to a dramatic shift in the types of plywood products emphasized by the company." Identify this shift.

(c) With the success of this application, management then was eager to use optimization for other problems as well. Identify these other problems.

(d) Photocopy the two pages of appendixes that give the mathematical formulation of the problem and the structure of the linear programming model.

3.5-2. Read the article footnoted in Sec. 3.5 that describes the second case study presented in that section: "Personnel Scheduling at United Airlines."

(a) Describe how United Airlines prepared shift schedules at airports and reservations offices prior to this OR study.

(b) When this study began, the *problem definition phase* defined five specific project requirements. Identify these project requirements.

(c) At the end of the presentation of the corresponding example in Sec. 3.4 (personnel scheduling at Union Airways), we pointed out that the divisibility assumption does not hold for this kind of application. An integer solution is needed, but linear programming may provide an optimal solution that is noninteger. How does United Airlines deal with this problem?

(d) Describe the flexibility built into the scheduling system to satisfy the group culture at each office. Why was this flexibility needed?

(e) Briefly describe the tangible and intangible benefits that resulted from the study.

3.5-3. Read the 1986 article footnoted in Sec. 2.1 that describes the third case study presented in Sec. 3.5: "Planning Supply, Distribution, and Marketing at Citgo Petroleum Corporation."

(a) What happened during the years preceding this OR study that made it vastly more important to control the amount of capital tied up in inventory?

(b) What geographical area is spanned by Citgo's distribution network of pipelines, tankers, and barges? Where do they market their products?

(c) What time periods are included in the model?

(d) Which computer did Citgo use to solve the model? What were typical run times?

(e) Who are the four types of model users? How does each one use the model?

(f) List the major types of reports generated by the SDM system.

(g) What were the major implementation challenges for this study?

(h) List the direct and indirect benefits that were realized from this study.

3.6-1.* You are given the following data for a linear programming problem where the objective is to maximize the profit from allocating three resources to two nonnegative activities.

Resource	Resource Usage per Unit of Each Activity		Amount of Resource Available
	Activity 1	Activity 2	
1	2	1	10
2	3	3	20
3	2	4	20
Contribution per unit	$20	$30	

Contribution per unit = profit per unit of the activity.

(a) Formulate a linear programming model for this problem.

D **(b)** Use the graphical method to solve this model.

(c) Display the model on an Excel spreadsheet.

(d) Use the spreadsheet to check the following solutions: $(x_1, x_2) = (2, 2), (3, 3), (2, 4), (4, 2), (3, 4), (4, 3)$. Which of these solutions are feasible? Which of these feasible solutions has the best value of the objective function?

C **(e)** Use the Excel Solver to solve the model by the simplex method.

3.6-2. Ed Butler is the production manager for the Bilco Corporation, which produces three types of spare parts for automobiles. The manufacture of each part requires processing on each of two machines, with the following processing times (in hours):

Machine	Part		
	A	B	C
1	0.02	0.03	0.05
2	0.05	0.02	0.04

Each machine is available 40 hours per month. Each part manufactured will yield a unit profit as follows:

	Part		
	A	B	C
Profit	$50	$40	$30

Ed wants to determine the mix of spare parts to produce in order to maximize total profit.

(a) Formulate a linear programming model for this problem.

(b) Display the model on an Excel spreadsheet.

(c) Make three guesses of your own choosing for the optimal solution. Use the spreadsheet to check each one for feasibility and, if feasible, to find the value of the objective function. Which feasible guess has the best objective function value?

(d) Use the Excel Solver to solve the model by the simplex method.

3.6-3. You are given the following data for a linear programming problem where the objective is to minimize the cost of conducting two nonnegative activities so as to achieve three benefits that do not fall below their minimum levels.

Benefit	Benefit Contribution per Unit of Each Activity		Minimum Acceptable Level
	Activity 1	Activity 2	
1	5	3	60
2	2	2	30
3	7	9	126
Unit cost	$60	$50	

(a) Formulate a linear programming model for this problem.

D **(b)** Use the graphical method to solve this model.

(c) Display the model on an Excel spreadsheet.

(d) Use the spreadsheet to check the following solutions: $(x_1, x_2) = (7, 7), (7, 8), (8, 7), (8, 8), (8, 9), (9, 8)$. Which of these solutions are feasible? Which of these feasible solutions has the best value of the objective function?

C **(e)** Use the Excel Solver to solve this model by the simplex method.

3.6-4.* Fred Jonasson manages a family-owned farm. To supplement several food products grown on the farm, Fred also raises pigs for market. He now wishes to determine the quantities of the available types of feed (corn, tankage, and alfalfa) that should be given to each pig. Since pigs will eat any mix of these feed types, the objective is to determine which mix will meet certain nutritional requirements at a *minimum cost.* The number of units of each type of basic nutritional ingredient contained within a kilogram of each feed type is given in the following table, along with the daily nutritional requirements and feed costs:

Nutritional Ingredient	Kilogram of Corn	Kilogram of Tankage	Kilogram of Alfalfa	Minimum Daily Requirement
Carbohydrates	90	20	40	200
Protein	30	80	60	180
Vitamins	10	20	60	150
Cost (¢)	84	72	60	

(a) Formulate a linear programming model for this problem.
(b) Display the model on an Excel spreadsheet.
(c) Use the spreadsheet to check if $(x_1, x_2, x_3) = (1, 2, 2)$ is a feasible solution and, if so, what the daily cost would be for this diet. How many units of each nutritional ingredient would this diet provide daily?
(d) Take a few minutes to use a trial-and-error approach with the spreadsheet to develop your best guess for the optimal solution. What is the daily cost for your solution?
C **(e)** Use the Excel Solver to solve the model by the simplex method.

3.6-5. Maureen Laird is the chief financial officer for the Alva Electric Co., a major public utility in the midwest. The company has scheduled the construction of new hydroelectric plants 5, 10, and 20 years from now to meet the needs of the growing population in the region served by the company. To cover at least the construction costs, Maureen needs to invest some of the company's money now to meet these future cash-flow needs. Maureen may purchase only three kinds of financial assets, each of which costs $1 million per unit. Fractional units may be purchased. The assets produce income 5, 10, and 20 years from now, and that income is needed to cover at least minimum cash-flow requirements in those years. (Any excess income above the minimum requirement for each time period will be used to increase dividend payments to shareholders rather than saving it to help meet the minimum cash-flow requirement in the next time period.) The following table shows both the amount of income generated by each unit of each asset and the minimum amount of income needed for each of the future time periods when a new hydroelectric plant will be constructed.

	Income per Unit of Asset			Minimum Cash Flow Required
Year	Asset 1	Asset 2	Asset 3	
5	$2 million	$1 million	$0.5 million	$400 million
10	$0.5 million	$0.5 million	$1 million	$100 million
20	0	$1.5 million	$2 million	$300 million

Maureen wishes to determine the mix of investments in these assets that will cover the cash-flow requirements while minimizing the total amount invested.
(a) Formulate a linear programming model for this problem.
(b) Display the model on a spreadsheet.
(c) Use the spreadsheet to check the possibility of purchasing 100 units of Asset 1, 100 units of Asset 2, and 200 units of Asset 3. How much cash flow would this mix of investments generate 5, 10, and 20 years from now? What would be the total amount invested?
(d) Take a few minutes to use a trial-and-error approach with the spreadsheet to develop your best guess for the optimal solution. What is the total amount invested for your solution?
C **(e)** Use the Excel Solver to solve the model by the simplex method.

3.7-1. The Philbrick Company has two plants on opposite sides of the United States. Each of these plants produces the same two products and then sells them to wholesalers within its half of the country. The orders from wholesalers have already been received for the next 2 months (February and March), where the number of units requested are shown below. (The company is not obligated to completely fill these orders but will do so if it can without decreasing its profits.)

	Plant 1		Plant 2	
Product	February	March	February	March
1	3,600	6,300	4,900	4,200
2	4,500	5,400	5,100	6,000

Each plant has 20 production days available in February and 23 production days available in March to produce and ship these products. Inventories are depleted at the end of January, but each plant has enough inventory capacity to hold 1,000 units total of the two products if an excess amount is produced in February for sale in March. In either plant, the cost of holding inventory in this way is $3 per unit of product 1 and $4 per unit of product 2.

Each plant has the same two production processes, each of which can be used to produce either of the two products. The production cost per unit produced of each product is shown below for each process in each plant.

	Plant 1		Plant 2	
Product	Process 1	Process 2	Process 1	Process 2
1	$62	$59	$61	$65
2	$78	$85	$89	$86

The production rate for each product (number of units produced per day devoted to that product) also is given below for each process in each plant.

	Plant 1		Plant 2	
Product	Process 1	Process 2	Process 1	Process 2
1	100	140	130	110
2	120	150	160	130

The net sales revenue (selling price minus normal shipping costs) the company receives when a plant sells the products to its own customers (the wholesalers in its half of the country) is $83 per unit of product 1 and $112 per unit of product 2. However, it also is possible (and occasionally desirable) for a plant to make a shipment to the other half of the country to help fill the sales of the other plant. When this happens, an extra shipping cost of $9 per unit of product 1 and $7 per unit of product 2 is incurred.

Management now needs to determine how much of each product should be produced by each production process in each plant during each month, as well as how much each plant should sell of each product in each month and how much each plant should ship of each product in each month to the other plant's customers. The objective is to determine which feasible plan would maximize the total profit (total net sales revenue minus the sum of the production costs, inventory costs, and extra shipping costs).

(a) Formulate a complete linear programming model in algebraic form that shows the individual constraints and decision variables for this problem.

C **(b)** Formulate this same model on an Excel spreadsheet instead. Then use the Excel Solver to solve the model.

C **(c)** Use MPL to formulate this model in a compact form. Then use the MPL solver CPLEX to solve the model.

C **(d)** Use LINGO to formulate this model in a compact form. Then use the LINGO solver to solve the model.

C **3.7-2.** Reconsider Prob. 3.1-11.

(a) Use MPL/CPLEX to formulate and solve the model for this problem.

(b) Use LINGO to formulate and solve this model.

C **3.7-3.** Reconsider Prob. 3.4-11.

(a) Use MPL/CPLEX to formulate and solve the model for this problem.

(b) Use LINGO to formulate and solve this model.

C **3.7-4.** Reconsider Prob. 3.4-15.

(a) Use MPL/CPLEX to formulate and solve the model for this problem.

(b) Use LINGO to formulate and solve this model.

C **3.7-5.** Reconsider Prob. 3.4-18.

(a) Use MPL/CPLEX to formulate and solve the model for this problem.

(b) Use LINGO to formulate and solve this model.

C **3.7-6.** Reconsider Prob. 3.6-4.

(a) Use MPL/CPLEX to formulate and solve the model for this problem.

(b) Use LINGO to formulate and solve this model.

C **3.7-7.** Reconsider Prob. 3.6-5.

(a) Use MPL/CPLEX to formulate and solve the model for this problem.

(b) Use LINGO to formulate and solve this model.

3.7-8. A large paper manufacturing company, the Quality Paper Corporation, has 10 paper mills from which it needs to supply 1,000 customers. It uses three alternative types of machines and four types of raw materials to make five different types of paper. Therefore, the company needs to develop a detailed production distribution plan on a monthly basis, with an objective of minimizing the total cost of producing and distributing the paper during the month. Specifically, it is necessary to determine jointly the amount of each type of paper to be made at each paper mill on each type of machine *and* the amount of each type of paper to be shipped from each paper mill to each customer.

The relevant data can be expressed symbolically as follows:

D_{jk} = number of units of paper type k demanded by customer j,

r_{klm} = number of units of raw material m needed to produce 1 unit of paper type k on machine type l,

R_{im} = number of units of raw material m available at paper mill i,

c_{kl} = number of capacity units of machine type l that will produce 1 unit of paper type k,

C_{il} = number of capacity units of machine type l available at paper mill i,

P_{ikl} = production cost for each unit of paper type k produced on machine type l at paper mill i,

T_{ijk} = transportation cost for each unit of paper type k shipped from paper mill i to customer j.

(a) Using these symbols, formulate a linear programming model for this problem by hand.

(b) How many functional constraints and decision variables does this model have?

C **(c)** Use MPL to formulate this problem.

C **(d)** Use LINGO to formulate this problem.

CASE 3.1 AUTO ASSEMBLY

Automobile Alliance, a large automobile manufacturing company, organizes the vehicles it manufactures into three families: a family of trucks, a family of small cars, and a family of midsized and luxury cars. One plant outside Detroit, MI, assembles two models from the family of midsized and luxury cars. The first model, the Family Thrillseeker, is a four-door sedan with vinyl seats, plastic interior, standard features, and excellent gas mileage. It is marketed as a smart buy for middle-class families with tight budgets, and each Family Thrillseeker sold generates a modest profit of $3,600 for the company. The second model, the Classy Cruiser, is a two-door luxury sedan with leather seats, wooden interior, custom features, and navigational capabilities. It is marketed as a privilege of affluence for upper-middle-class families, and each Classy Cruiser sold generates a healthy profit of $5,400 for the company.

Rachel Rosencrantz, the manager of the assembly plant, is currently deciding the production schedule for the next month. Specifically, she must decide how many Family Thrillseekers and how many Classy Cruisers to assemble in the plant to maximize profit for the company. She knows that the plant possesses a capacity of 48,000 labor-hours during the month. She also knows that it takes 6 labor-hours to assemble one Family Thrillseeker and 10.5 labor-hours to assemble one Classy Cruiser.

Because the plant is simply an assembly plant, the parts required to assemble the two models are not produced at the plant. They are instead shipped from other plants around the Michigan area to the assembly plant. For example, tires, steering wheels, windows, seats, and doors all arrive from various supplier plants. For the next month, Rachel knows that she will be able to obtain only 20,000 doors (10,000 left-hand doors and 10,000 right-hand doors) from the door supplier. A recent labor strike forced the shutdown of that particular supplier plant for several days, and that plant will not be able to meet its production schedule for the next month. Both the Family Thrillseeker and the Classy Cruiser use the same door part.

In addition, a recent company forecast of the monthly demands for different automobile models suggests that the demand for the Classy Cruiser is limited to 3,500 cars. There is no limit on the demand for the Family Thrillseeker within the capacity limits of the assembly plant.

(a) Formulate and solve a linear programming problem to determine the number of Family Thrillseekers and the number of Classy Cruisers that should be assembled.

Before she makes her final production decisions, Rachel plans to explore the following questions independently except where otherwise indicated.

(b) The marketing department knows that it can pursue a targeted $500,000 advertising campaign that will raise the demand for the Classy Cruiser next month by 20 percent. Should the campaign be undertaken?

(c) Rachel knows that she can increase next month's plant capacity by using overtime labor. She can increase the plant's labor-hour capacity by 25 percent. With the new assembly plant capacity, how many Family Thrillseekers and how many Classy Cruisers should be assembled?

(d) Rachel knows that overtime labor does not come without an extra cost. What is the maximum amount she should be willing to pay for all overtime labor beyond the cost of this labor at regular time rates? Express your answer as a lump sum.

(e) Rachel explores the option of using both the targeted advertising campaign and the overtime labor-hours. The advertising campaign raises the demand for the Classy Cruiser by 20 percent, and the overtime labor increases the plant's labor-hour capacity by 25 percent. How many Family Thrillseekers and how many Classy Cruisers should be assembled using the advertising campaign and overtime labor-hours if the profit from each Classy Cruiser sold continues to be 50 percent more than for each Family Thrillseeker sold?

(f) Knowing that the advertising campaign costs $500,000 and the maximum usage of overtime labor-hours costs $1,600,000 beyond regular time rates, is the solution found in part (*e*) a wise decision compared to the solution found in part (*a*)?

(g) Automobile Alliance has determined that dealerships are actually heavily discounting the price of the Family Thrillseekers to move them off the lot. Because of a profit-sharing agreement with its dealers, the company is therefore not making a profit of $3,600 on the Family Thrillseeker but is instead making a profit of $2,800. Determine the number of Family Thrillseekers and the number of Classy Cruisers that should be assembled given this new discounted price.

(h) The company has discovered quality problems with the Family Thrillseeker by randomly testing Thrillseekers at the end of the assembly line. Inspectors have discovered that in over 60 percent of the cases, two of the four doors on a Thrillseeker do not seal properly. Because the percentage of defective Thrillseekers determined by the random testing is so high, the floor supervisor has decided to perform quality control tests on every Thrillseeker at the end of the line. Because of the added tests, the time it takes to assemble one Family Thrillseeker has increased from 6 to 7.5 hours. Determine the number of units of each model that should be assembled given the new assembly time for the Family Thrillseeker.

(i) The board of directors of Automobile Alliance wishes to capture a larger share of the luxury sedan market and therefore would like to meet the full demand for Classy Cruisers. They ask Rachel to determine by how much the profit of her assembly plant would decrease as compared to the profit found in part (*a*). They then ask her to meet the full demand for Classy Cruisers if the decrease in profit is not more than $2,000,000.

(j) Rachel now makes her final decision by combining all the new considerations described in parts (*f*), (*g*), and (*h*). What are her final decisions on whether to undertake the advertising campaign, whether to use overtime labor, the number of Family Thrillseekers to assemble, and the number of Classy Cruisers to assemble?

CASE 3.2 CUTTING CAFETERIA COSTS

A cafeteria at All-State University has one special dish it serves like clockwork every Thursday at noon. This supposedly tasty dish is a casserole that contains sautéed onions, boiled sliced potatoes, green beans, and cream of mushroom soup. Unfortunately, students fail to see the special quality of this dish, and they loathingly refer to it as the Killer Casserole. The students reluctantly eat the casserole, however, because the cafeteria provides only a limited selection of dishes for Thursday's lunch (namely, the casserole).

Maria Gonzalez, the cafeteria manager, is looking to cut costs for the coming year, and she believes that one sure way to cut costs is to buy less expensive and perhaps lower-quality ingredients. Because the casserole is a weekly staple of the cafeteria menu, she concludes that if she can cut costs on the ingredients purchased for the casserole, she can significantly reduce overall cafeteria operating costs. She therefore de-

cides to invest time in determining how to minimize the costs of the casserole while maintaining nutritional and taste requirements.

Maria focuses on reducing the costs of the two main ingredients in the casserole, the potatoes and green beans. These two ingredients are responsible for the greatest costs, nutritional content, and taste of the dish.

Maria buys the potatoes and green beans from a wholesaler each week. Potatoes cost \$0.40 per pound, and green beans cost \$1.00 per pound.

All-State University has established nutritional requirements that each main dish of the cafeteria must meet. Specifically, the total amount of the dish prepared for all the students for one meal must contain 180 grams (g) of protein, 80 milligrams (mg) of iron, and 1,050 mg of vitamin C. (There are 453.6 g in 1 lb and 1,000 mg in 1 g.) For simplicity when planning, Maria assumes that only the potatoes and green beans contribute to the nutritional content of the casserole.

Because Maria works at a cutting-edge technological university, she has been exposed to the numerous resources on the World Wide Web. She decides to surf the Web to find the nutritional content of potatoes and green beans. Her research yields the following nutritional information about the two ingredients:

	Potatoes	Green Beans
Protein	1.5 g per 100 g	5.67 g per 10 ounces
Iron	0.3 mg per 100 g	3.402 mg per 10 ounces
Vitamin C	12 mg per 100 g	28.35 mg per 10 ounces

(There are 28.35 g in 1 ounce.)

Edson Branner, the cafeteria cook who is surprisingly concerned about taste, informs Maria that an edible casserole must contain at least a six to five ratio in the weight of potatoes to green beans.

Given the number of students who eat in the cafeteria, Maria knows that she must purchase enough potatoes and green beans to prepare a minimum of 10 kilograms (kg) of casserole each week. (There are 1,000 g in 1 kg.) Again for simplicity in planning, she assumes that only the potatoes and green beans determine the amount of casserole that can be prepared. Maria does not establish an upper limit on the amount of casserole to prepare, since she knows all leftovers can be served for many days thereafter or can be used creatively in preparing other dishes.

(a) Determine the amount of potatoes and green beans Maria should purchase each week for the casserole to minimize the ingredient costs while meeting nutritional, taste, and demand requirements.

Before she makes her final decision, Maria plans to explore the following questions independently except where otherwise indicated.

(b) Maria is not very concerned about the taste of the casserole; she is only concerned about meeting nutritional requirements and cutting costs. She therefore forces Edson to change the recipe to allow for only at least a one to two ratio in the weight of potatoes to green beans. Given the new recipe, determine the amount of potatoes and green beans Maria should purchase each week.

(c) Maria decides to lower the iron requirement to 65 mg since she determines that the other ingredients, such as the onions and cream of mushroom soup, also provide iron. Determine the amount of potatoes and green beans Maria should purchase each week given this new iron requirement.

(d) Maria learns that the wholesaler has a surplus of green beans and is therefore selling the green beans for a lower price of $0.50 per lb. Using the same iron requirement from part (*c*) and the new price of green beans, determine the amount of potatoes and green beans Maria should purchase each week.

(e) Maria decides that she wants to purchase lima beans instead of green beans since lima beans are less expensive and provide a greater amount of protein and iron than green beans. Maria again wields her absolute power and forces Edson to change the recipe to include lima beans instead of green beans. Maria knows she can purchase lima beans for $0.60 per lb from the wholesaler. She also knows that lima beans contain 22.68 g of protein per 10 ounces of lima beans, 6.804 mg of iron per 10 ounces of lima beans, and no vitamin C. Using the new cost and nutritional content of lima beans, determine the amount of potatoes and lima beans Maria should purchase each week to minimize the ingredient costs while meeting nutritional, taste, and demand requirements. The nutritional requirements include the reduced iron requirement from part (*c*).

(f) Will Edson be happy with the solution in part (*e*)? Why or why not?

(g) An All-State student task force meets during Body Awareness Week and determines that All-State University's nutritional requirements for iron are too lax and that those for vitamin C are too stringent. The task force urges the university to adopt a policy that requires each serving of an entrée to contain at least 120 mg of iron and at least 500 mg of vitamin C. Using potatoes and lima beans as the ingredients for the dish and using the new nutritional requirements, determine the amount of potatoes and lima beans Maria should purchase each week.

CASE 3.3 STAFFING A CALL CENTER[1]

California Children's Hospital has been receiving numerous customer complaints because of its confusing, decentralized appointment and registration process. When customers want to make appointments or register child patients, they must contact the clinic or department they plan to visit. Several problems exist with this current strategy. Parents do not always know the most appropriate clinic or department they must visit to address their children's ailments. They therefore spend a significant amount of time on the phone being transferred from clinic to clinic until they reach the most appropriate clinic for their needs. The hospital also does not publish the phone numbers of all clinic and departments, and parents must therefore invest a large amount of time in detective work to track down the correct phone number. Finally, the various clinics and departments do not communicate with each other. For example, when a doctor schedules a referral with a colleague located in another department or clinic, that department or clinic almost never receives word of the referral. The parent must contact the correct department or clinic and provide the needed referral information.

[1]This case is based on an actual project completed by a team of master's students in the Department of Engineering-Economic Systems and Operations Research at Stanford University.

In efforts to reengineer and improve its appointment and registration process, the children's hospital has decided to centralize the process by establishing one call center devoted exclusively to appointments and registration. The hospital is currently in the middle of the planning stages for the call center. Lenny Davis, the hospital manager, plans to operate the call center from 7 A.M. to 9 P.M. during the weekdays.

Several months ago, the hospital hired an ambitious management consulting firm, Creative Chaos Consultants, to forecast the number of calls the call center would receive each hour of the day. Since all appointment and registration-related calls would be received by the call center, the consultants decided that they could forecast the calls at the call center by totaling the number of appointment and registration-related calls received by all clinics and departments. The team members visited all the clinics and departments, where they diligently recorded every call relating to appointments and registration. They then totaled these calls and altered the totals to account for calls missed during data collection. They also altered totals to account for repeat calls that occurred when the same parent called the hospital many times because of the confusion surrounding the decentralized process. Creative Chaos Consultants determined the average number of calls the call center should expect during each hour of a weekday. The following table provides the forecasts.

Work Shift	Average Number of Calls
7 A.M.–9 A.M.	40 calls per hour
9 A.M.–11 A.M.	85 calls per hour
11 A.M.–1 P.M.	70 calls per hour
1 P.M.–3 P.M.	95 calls per hour
3 P.M.–5 P.M.	80 calls per hour
5 P.M.–7 P.M.	35 calls per hour
7 P.M.–9 P.M.	10 calls per hour

After the consultants submitted these forecasts, Lenny became interested in the percentage of calls from Spanish speakers since the hospital services many Spanish patients. Lenny knows that he has to hire some operators who speak Spanish to handle these calls. The consultants performed further data collection and determined that on average, 20 percent of the calls were from Spanish speakers.

Given these call forecasts, Lenny must now decide how to staff the call center during each 2 hour shift of a weekday. During the forecasting project, Creative Chaos Consultants closely observed the operators working at the individual clinics and departments and determined the number of calls operators process per hour. The consultants informed Lenny that an operator is able to process an average of six calls per hour. Lenny also knows that he has both full-time and part-time workers available to staff the call center. A full-time employee works 8 hours per day, but because of paperwork that must also be completed, the employee spends only 4 hours per day on the phone. To balance the schedule, the employee alternates the 2-hour shifts between answering phones and completing paperwork. Full-time employees can start their day either by answering phones or by completing paperwork on the first shift. The full-time em-

ployees speak either Spanish or English, but none of them are bilingual. Both Spanish-speaking and English-speaking employees are paid $10 per hour for work before 5 P.M. and $12 per hour for work after 5 P.M. The full-time employees can begin work at the beginning of the 7 A.M. to 9 A.M. shift, 9 A.M. to 11 A.M. shift, 11 A.M. to 1 P.M. shift, or 1 P.M. to 3 P.M. shift. The part-time employees work for 4 hours, only answer calls, and only speak English. They can start work at the beginning of the 3 P.M. to 5 P.M. shift or the 5 P.M. to 7 P.M. shift, and like the full-time employees, they are paid $10 per hour for work before 5 P.M. and $12 per hour for work after 5 P.M.

For the following analysis consider only the labor cost for the time employees spend answering phones. The cost for paperwork time is charged to other cost centers.

(a) How many Spanish-speaking operators and how many English-speaking operators does the hospital need to staff the call center during each 2-hour shift of the day in order to answer all calls? Please provide an integer number since half a human operator makes no sense.

(b) Lenny needs to determine how many full-time employees who speak Spanish, full-time employees who speak English, and part-time employees he should hire to begin on each shift. Creative Chaos Consultants advise him that linear programming can be used to do this in such a way as to minimize operating costs while answering all calls. Formulate a linear programming model of this problem.

(c) Obtain an optimal solution for the linear programming model formulated in part (*b*) to guide Lenny's decision.

(d) Because many full-time workers do not want to work late into the evening, Lenny can find only one qualified English-speaking operator willing to begin work at 1 P.M. Given this new constraint, how many full-time English-speaking operators, full-time Spanish-speaking operators, and part-time operators should Lenny hire for each shift to minimize operating costs while answering all calls?

(e) Lenny now has decided to investigate the option of hiring bilingual operators instead of monolingual operators. If all the operators are bilingual, how many operators should be working during each 2-hour shift to answer all phone calls? As in part (*a*), please provide an integer answer.

(f) If all employees are bilingual, how many full-time and part-time employees should Lenny hire to begin on each shift to minimize operating costs while answering all calls? As in part (*b*), formulate a linear programming model to guide Lenny's decision.

(g) What is the maximum percentage increase in the hourly wage rate that Lenny can pay bilingual employees over monolingual employees without increasing the total operating costs?

(h) What other features of the call center should Lenny explore to improve service or minimize operating costs?