19 Linear Programming

LEARNING OBJECTIVES

After completing this chapter, you should be able to:

LO19.1 Describe the type of problem that would lend itself to solution using linear programming.

LO19.2 Formulate a linear programming model from a description of a problem.

LO19.3 Solve simple linear programming problems using the graphical method.

LO19.4 Interpret computer solutions of linear programming problems.

LO19.5 Do sensitivity analysis on the solution of a linear programming problem.

CHAPTER OUTLINE

© Kupicco/Getty RF

Linear programming is a powerful quantitative tool used by operations managers and other managers to obtain optimal solutions to problems that involve restrictions or limitations, such as budgets and available materials, labor, and machine time. These problems are referred to as *constrained optimization* problems. There are numerous examples of linear programming applications to such problems, including:

- Establishing locations for emergency equipment and personnel that will minimize response time
- Determining optimal schedules for airlines for planes, pilots, and ground personnel
- Developing financial plans
- Determining optimal blends of animal feed mixes
- Determining optimal diet plans
- Identifying the best set of worker–job assignments
- Developing optimal production schedules
- Developing shipping plans that will minimize shipping costs
- Identifying the optimal mix of products in a factory
- Performing production and service planning

19.1 INTRODUCTION

Linear programming (LP) techniques consist of a sequence of steps that will lead to an optimal solution to linear-constrained problems, if an optimal solution exists. There are a number of different linear programming techniques; some are special-purpose (i.e., used to find solutions for specific types of problems) and others are more general in scope. This chapter covers the two general-purpose solution techniques: graphical linear programming and computer solutions. Graphical linear programming provides a visual portrayal of many of the important concepts of linear programming. However, it is limited to problems with only two variables. In practice, computers are used to obtain solutions for problems, some of which involve a large number of variables.

19.2 LINEAR PROGRAMMING MODELS

eXcel
mhhe.com/stevenson13e

SCREENCAM TUTORIAL

LO19.1 Describe the type of problem that would lend itself to solution using linear programming.

Linear programming models are mathematical representations of constrained optimization problems. These models have certain characteristics in common. Knowledge of these characteristics enables us to recognize problems that can be solved using linear programming. In addition, it also can help us formulate LP models. The characteristics can be grouped into two categories: components and assumptions. First, let's consider the components.

Four components provide the structure of a linear programming model:

1. Objective function
2. Decision variables
3. Constraints
4. Parameters

Objective function Mathematical statement of profit (or cost, etc.) for a given solution.

Decision variables Amounts of either inputs or outputs.

Constraints Limitations that restrict the available alternatives.

Linear programming algorithms require that a single goal or *objective,* such as the maximization of profits, be specified. The two general types of objectives are maximization and minimization. A maximization objective might involve profits, revenues, efficiency, or rate of return. Conversely, a minimization objective might involve cost, time, distance traveled, or scrap. The **objective function** is a mathematical expression that can be used to determine the total profit (or cost, etc., depending on the objective) for a given solution.

Decision variables represent choices available to the decision maker in terms of amounts of either inputs or outputs. For example, some problems require choosing a combination of inputs to minimize total costs, while others require selecting a combination of outputs to maximize profits or revenues.

Constraints are limitations that restrict the alternatives available to decision makers. The three types of constraints are less than or equal to ($\leq$), greater than or equal to ($\geq$), and simply equal to ($=$). A $\leq$ constraint implies an upper limit on the amount of some scarce resource (e.g., machine hours, labor hours, materials) available for use. A $\geq$ constraint specifies a minimum that must be achieved in the final solution (e.g., must contain at least 10 percent real fruit juice, must get at least 30 MPG on the highway). The $=$ constraint is more restrictive in the sense that it specifies *exactly* what a decision variable should equal (e.g., make 200 units of product A). A linear programming model can consist of one or more constraints. The constraints of a given problem define the set of combinations of the decision variables that satisfy all constraints; this set is referred to as the **feasible solution space**. Linear programming algorithms are designed to search the feasible solution space for the combination of decision variables that will yield an optimum in terms of the objective function.

Feasible solution space The set of all feasible combinations of decision variables as defined by the constraints.

An LP model consists of a mathematical statement of the objective and a mathematical statement of each constraint. These statements consist of symbols (e.g., x_1, x_2) that represent the decision variables and numerical values, called **parameters**. The parameters are fixed values; the model is solved *given* those values.

Parameters Numerical constants.

Example 1 illustrates an LP model.

EXAMPLE 1

Linear Programming Models Explained

Here is an LP model of a situation that involves the production of three possible products, each of which will yield a certain profit per unit, and each requires a certain use of two resources that are in limited supply: labor and materials. The objective is to determine how much of each product to make to achieve the greatest possible profit while satisfying all constraints.

$$\text{Decision variables} \quad \begin{cases} x_1 = \text{Quantity of product 1 to produce} \\ x_2 = \text{Quantity of product 2 to produce} \\ x_3 = \text{Quantity of product 3 to produce} \end{cases}$$

$$\text{Maximize} \quad 5x_1 + 8x_2 + 4x_3 \text{ (profit)} \quad \text{(Objective function)}$$

Subject to

Labor	$2x_1 + 4x_2 + 8x_3 \le 250$ hours	
Material	$7x_1 + 6x_2 + 5x_3 \le 100$ pounds	(Constraints)
Product 1	$x_1 \ge 10$ units	
	$x_1, x_2, x_3 \ge 0$	(Nonnegativity constraints)

First, the model lists and defines the decision variables. These typically represent *quantities*. In this case, they are quantities of three different products that might be produced.

Next, the model states the objective function. It includes every decision variable in the model and the contribution (profit per unit) of each decision variable. Thus, product x_1 has a profit of $5 per unit. The profit from product x_1 for a given solution will be 5 times the value of x_1 specified by the solution; the total profit from all products will be the sum of the individual product profits. Thus, if $x_1 = 10$, $x_2 = 0$, and $x_3 = 6$, the value of the objective function would be:

$$5(10) + 8(0) + 4(6) = 74$$

The objective function is followed by a list (in no particular order) of three constraints. Each constraint has a right-hand-side numerical value (e.g., the labor constraint has a right-hand-side value of 250) that indicates the amount of the constraint and a relation sign that indicates whether that amount is a maximum ($\le$), a minimum ($\ge$), or an equality ($=$). The left-hand side of each constraint consists of the variables subject to that particular constraint and a coefficient for each variable that indicates how much of the right-hand-side quantity *one unit* of the decision variable represents. For instance, for the labor constraint, one unit of x_1 will require two hours of labor. The sum of the values on the left-hand side of each constraint represents the amount of that constraint used by a solution. $x_1 = 10$, $x_2 = 0$, and $x_3 = 6$, the amount of labor used would be:

$$2(10) + 4(0) + 8(6) = 68 \text{ hours}$$

Because this amount does not exceed the quantity on the right-hand side of the constraint, it is said to be *feasible*.

Note that the third constraint refers to only a single variable; x_1 must be at least 10 units. Its implied coefficient is 1, although that is not shown.

Finally, there are the nonnegativity constraints. These are listed on a single line; they reflect the condition that no decision variable is allowed to have a negative value.

In order for LP models to be used effectively, certain *assumptions* must be satisfied:

1. **Linearity:** The impact of decision variables is linear in constraints and the objective function.
2. **Divisibility:** Noninteger values of decision variables are acceptable.
3. **Certainty:** Values of parameters are known and constant.
4. **Nonnegativity:** Negative values of decision variables are unacceptable.

Model Formulation

LO19.2 Formulate a linear programming model from a description of a problem.

An understanding of the components of linear programming models is necessary for model formulation. This helps provide organization to the process of assembling information about a problem into a model.

Naturally, it is important to obtain valid information on what constraints are appropriate, as well as on what values of the parameters are appropriate. If this is not done, the usefulness of the model will be questionable. Consequently, in some instances, considerable effort must be expended to obtain that information.

In formulating a model, use the format illustrated in Example 1. Begin by identifying the decision variables. Very often, decision variables are "the quantity of" something, such as

x_1 = the quantity of product 1. Generally, decision variables have profits, costs, times, or a similar measure of value associated with them. Knowing this can help you identify the decision variables in a problem.

Constraints are restrictions or requirements on one or more decision variables, and they refer to available amounts of resources such as labor, material, or machine time, or to minimal requirements, such as "Make at least 10 units of product 1." It can be helpful to give a name to each constraint, such as "labor" or "material 1." Let's consider some of the different kinds of constraints you will encounter.

1. A constraint that refers to one or more decision variables. This is the most common kind of constraint. The constraints in Example 1 are of this type.

2. A constraint that specifies a ratio. For example, "The ratio of x_1 to x_2 must be at least 3 to 2." To formulate this, begin by setting up the following ratio:

$$\frac{x_1}{x_2} \geq \frac{3}{2}$$

Then, cross multiply, obtaining

$$2x_1 \geq 3x_2$$

This is not yet in a suitable form because all variables in a constraint must be on the left-hand side of the inequality (or equality) sign, leaving only a constant on the right-hand side. To achieve this, we must subtract the variable amount that is on the right side from both sides. That yields

$$2x_1 - 3x_2 \geq 0$$

(Note that the direction of the inequality remains the same.)

3. A constraint that specifies a percentage for one or more variables relative to one or more other variables. For example, "x_1 cannot be more than 20 percent of the mix." Suppose that the mix consists of variables x_1, x_2, and x_3. In mathematical terms, this would be

$$x_1 \leq .20(x_1 + x_2 + x_3)$$

As always, all variables must appear on the left-hand side of the relationship. To accomplish that, we can expand the right-hand side, and then subtract the result from both sides. Expanding yields

$$x_1 \leq .20x_1 + .20x_2 + .20x_3$$

Subtracting yields

$$.80x_1 - .20x_2 - .20x_3 \leq 0$$

Once you have formulated a model, the next task is to solve it. The following sections describe two approaches to problem solution: graphical solutions and computer solutions.

19.3 GRAPHICAL LINEAR PROGRAMMING

LO19.3 Solve simple linear programming problems using the graphical method.

Graphical linear programming is a method for finding optimal solutions to two-variable problems. This section describes that approach.

Graphical linear programming Graphical method for finding optimal solutions to two-variable problems.

Outline of Graphical Procedure

The graphical method of linear programming involves plotting the constraint lines on a graph and identifying an area on the graph that satisfies all of the constraints. The area is referred to as the *feasible solution space*. Next, the objective function is plotted and used to identify the optimal point in the feasible solution space. The coordinates of the point can sometimes be read directly from the graph, although generally an algebraic determination of the coordinates of the point is necessary.

The general procedure followed in the graphical approach is as follows:

1. Set up the objective function and the constraints in mathematical format.
2. Plot the constraints.
3. Identify the feasible solution space.
4. Plot the objective function.
5. Determine the optimum solution.

The technique can best be illustrated through solution of a typical problem. Consider the problem described in Example 2.

EXAMPLE 2

Graphing the Problem and Finding the Optimal Solution

General description: A firm that assembles computers and computer equipment is about to start production of two new types of microcomputers. Each type will require assembly time, inspection time, and storage space. The amounts of each of these resources that can be devoted to the production of the microcomputers is limited. The manager of the firm would like to determine the quantity of each microcomputer to produce in order to maximize the profit generated by sales of these microcomputers.

Additional information: In order to develop a suitable model of the problem, the manager has met with design and production personnel. As a result of those meetings, the manager has obtained the following information.

	Type 1	Type 2
Profit per unit	$60	$50
Assembly time per unit	4 hours	10 hours
Inspection time per unit	2 hours	1 hour
Storage space per unit	3 cubic feet	3 cubic feet

The manager also has acquired information on the availability of company resources. These (daily) amounts are as follows.

Resource	Amount Available
Assembly time	100 hours
Inspection time	22 hours
Storage space	39 cubic feet

The manager met with the firm's marketing manager and learned that demand for the microcomputers was such that whatever combination of these two types of microcomputers is produced, all of the output can be sold.

In terms of meeting the assumptions, it would appear that the relationships are *linear:* The contribution to profit per unit of each type of computer and the time and storage space per unit of each type of computer are the same regardless of the quantity produced. Therefore, the total impact of each type of computer on the profit and each constraint is a linear function of the quantity of that variable. There may be a question of *divisibility* because, presumably, only whole units of computers will be sold. However, because this is a recurring process (i.e., the computers will be produced daily; a noninteger solution such as 3.5 computers per day will result in 7 computers every other day), this does not seem to pose a problem. The question of *certainty* cannot be explored here; in practice, the manager could be questioned to determine if there are any other possible constraints and whether the values shown for assembly times, and so forth, are known with certainty. For the purposes of discussion, we will assume certainty. Last, the assumption of *nonnegativity* seems justified; negative values for production quantities would not make sense.

Because we have concluded that linear programming is appropriate, let us now turn our attention to constructing a model of the microcomputer problem. First, we must define the decision variables. Based on the statement "The manager . . . would like to determine the quantity of each microcomputer to produce," the decision variables are the quantities of each type of computer. Thus,

x_1 = quantity of type 1 to produce
x_2 = quantity of type 2 to produce

Next, we can formulate the objective function. The profit per unit of type 1 is listed as \$60, and the profit per unit of type 2 is listed as \$50, so the appropriate objective function is

$$\text{Maximize } Z = 60x_1 + 50x_2$$

where Z is the value of the objective function, given values of x_1 and x_2. Theoretically, a mathematical function requires such a variable for completeness. However, in practice, the objective function often is written without the Z as sort of a shorthand version. (That approach is underscored by the fact that computer input does not call for Z: It is understood. The output of a computerized model does include a Z, though.)

Now for the constraints. There are three resources with limited availability: assembly time, inspection time, and storage space. The fact that availability is limited means that these constraints will all be ≤ constraints. Suppose we begin with the assembly constraint. The type 1 microcomputer requires 4 hours of assembly time per unit, whereas the type 2 microcomputer requires 10 hours of assembly time per unit. Therefore, with a limit of 100 hours available, the assembly constraint is

$$4x_1 + 10x_2 \leq 100 \text{ hours}$$

Similarly, each unit of type 1 requires 2 hours of inspection time, and each unit of type 2 requires 1 hour of inspection time. With 22 hours available, the inspection constraint is

$$2x_1 + 1x_2 \leq 22$$

(*Note:* The coefficient of 1 for x_2 need not be shown. Thus, an alternative form for this constraint is $2x_1 + x_2 \leq 22$.) The storage constraint is determined in a similar manner:

$$3x_1 + 3x_2 \leq 39$$

There are no other system or individual constraints. The nonnegativity constraints are

$$x_1, x_2 \geq 0$$

In summary, the mathematical model of the microcomputer problem is

x_1 = quantity of type 1 to produce
x_2 = quantity of type 2 to produce

Maximize $60x_1 + 50x_2$
Subject to

Assembly	$4x_1 + 10x_2$	$\leq$ 100 hours
Inspection	$2x_1 + 1x_2$	$\leq$ 22 hours
Storage	$3x_1 + 3x_2$	$\leq$ 39 cubic feet
	x_1, x_2	≥ 0

The next step is to plot the constraints.

Plotting Constraints

Begin by placing the nonnegativity constraints on a graph, as in Figure 19.1. The procedure for plotting the other constraints is simple:

1. Replace the inequality sign with an equal sign. This transforms the constraint into an *equation of a straight line.*

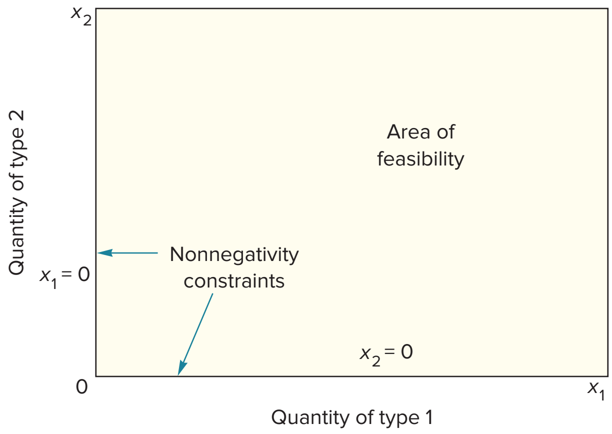


FIGURE 19.1
Graph showing the nonnegativity constraints

2. Determine where the line intersects each axis.
   a. To find where it crosses the x_2 axis, set x_1 equal to zero and solve the equation for the value of x_2.
   b. To find where it crosses the x_1 axis, set x_2 equal to zero and solve the equation for the value of x_1
3. Mark these intersections on the axes, and connect them with a straight line. (*Note:* If a constraint has only one variable, it will be a vertical line on a graph if the variable is x_1, or a horizontal line if the variable is x_2.)
4. Indicate by shading (or by arrows at the ends of the constraint line) whether the inequality is greater than or less than. (A general rule to determine which side of the line satisfies the inequality is to pick a point that is not on line, such as 0,0, solve the equation using these values, and see whether it is greater than or less than the constraint amount.)
5. Repeat steps 1–4 for each constraint.

Consider the assembly time constraint:

$$4x_1 + 10x_2 \leq 100$$

Removing the inequality portion of the constraint produces this straight line:

$$4x_1 + 10x_2 = 100$$

Next, identify the points where the line intersects each axis, as step 2 describes. Thus with $x_2 = 0$, we find

$$4x_1 + 10(0) = 100$$

Solving, we find that $4x_1 = 100$, so $x_1 = 25$ when $x_2 = 0$. Similarly, we can solve the equation for x_2 when $x_1 = 0$:

$$4(0) + 10x_2 = 100$$

Solving for x_2, we find $x_2 = 10$ when $x_1 = 0$.

Thus, we have two points: $x_1 = 0$, $x_2 = 10$, and $x_1 = 25$, $x_2 = 0$. We can now add this line to our graph of the nonnegativity constraints by connecting these two points (see Figure 19.2).

Next we must determine which side of the line represents points that are less than 100. To do this, we can select a test point that is not on the line, and we can substitute the x_1 and x_2 values of that point into the left-hand side of the equation of the line. If the result is less than 100, this tells us that all points on that side of the line are less than the value of the line (e.g., 100). Conversely, if the result is greater than 100, this indicates that the other side of the line represents the set of points that will yield values that are less than 100. A relatively simple test point to use is the origin (i.e., $x_1 = 0$, $x_2 = 0$). Substituting these values into the equation yields obviously this is less than 100. Hence, the side of the line closest to the origin represents the “less than” area (i.e., the feasible region).

$$4(0) + 10(0) = 0$$

FIGURE 19.2 Plot of the first constraint (assembly time)

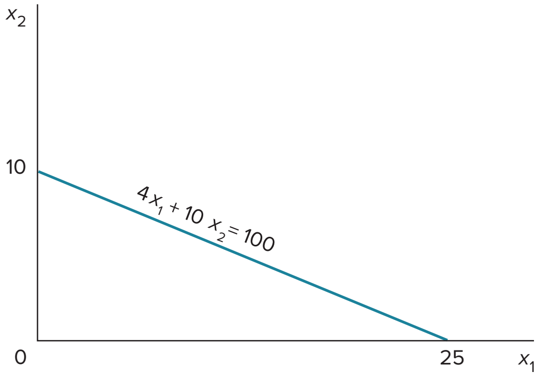
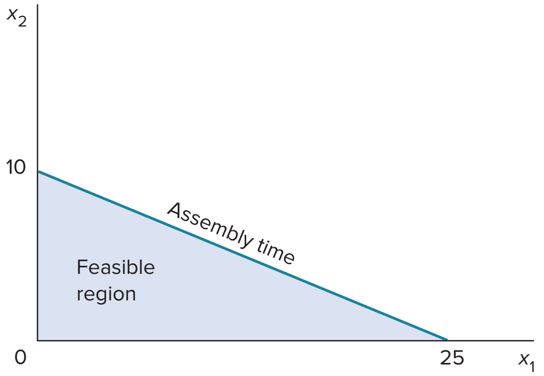


FIGURE 19.3 The feasible region, given the first constraint and the nonnegativity constraints



The feasible region for this constraint and the nonnegativity constraints then becomes the shaded portion shown in Figure 19.3.

For the sake of illustration, suppose we try one other point, say $x_1 = 10$, $x_2 = 10$. Substituting these values into the assembly constraint yields

$$4(10) + 10(10) = 140$$

Clearly this is greater than 100. Therefore, all points on this side of the line are greater than 100 (see Figure 19.4).

Continuing with the problem, we can add the two remaining constraints to the graph. For the inspection constraint:

1. Convert the constraint into the equation of a straight line by replacing the inequality sign with an equality sign:

   $2x_1 + 1x_2 \leq 22$ becomes $2x_1 + 1x_2 = 22$

2. Set x_1 equal to zero and solve for x_2:

   $2(0) + 1x_2 = 22$

   Solving, we find $x_2 = 22$. Thus, the line will intersect the x_2 axis at 22.

3. Next, set x_2 equal to zero and solve for x_1:

   $2x_1 + 1(0) = 22$

   Solving, we find $x_1 = 11$. Thus, the other end of the line will intersect the x_1 axis at 11.

4. Add the line to the graph (see Figure 19.5).

Note that the area of feasibility for this constraint is below the line (Figure 19.5). Again the area of feasibility at this point is shaded in for illustration, although when graphing problems, it is more practical to refrain from shading in the feasible region until all constraint lines have been drawn. However, because constraints are plotted one at a time, using a small arrow at the end of each constraint to indicate the direction of feasibility can be helpful.

The storage constraint is handled in the same manner:

1. Convert it into an equality:

   $3x_1 + 3x_2 = 39$

2. Set x_1 equal to zero and solve for x_2:

   $3(0) + 3x_2 = 39$

FIGURE 19.4 The point (10, 10) is above the constraint line

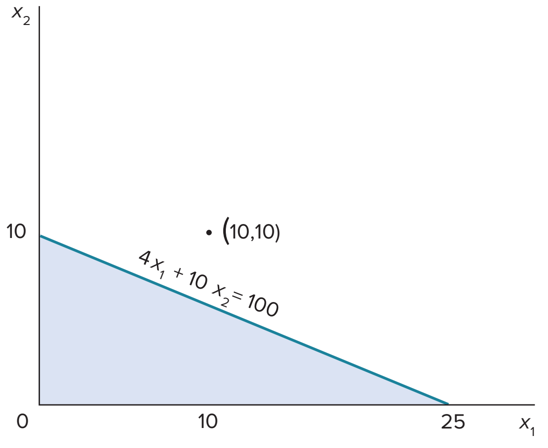
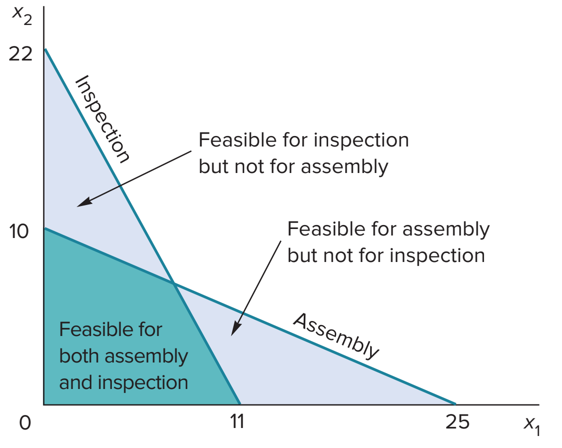


FIGURE 19.5 Partially completed graph, showing the assembly, inspection, and nonnegativity constraints



Solving, $x_2 = 13$. Thus, $x_2 = 13$ when $x_1 = 0$.

3. Set x_2 equal to zero and solve for x_1:

   $3x_1 + 3(0) = 39$

   Solving, $x_1 = 13$. Thus, $x_1 = 13$ when x_2 0.

4. Add the line to the graph (see Figure 19.6).

Identifying the Feasible Solution Space

The feasible solution space is the set of all points that satisfies *all* constraints. (Recall that the x_1 and x_2 axes form nonnegativity constraints.) The heavily shaded area shown in Figure 19.6 is the feasible solution space for our problem.

The next step is to determine which point in the feasible solution space will produce the optimal value of the objective function. This determination is made using the objective function.

Plotting the Objective Function Line

Plotting an objective function line involves the same logic as plotting a constraint line: Determine where the line intersects each axis. Recall that the objective function for the microcomputer problem is

$$60x_1 + 50x_2$$

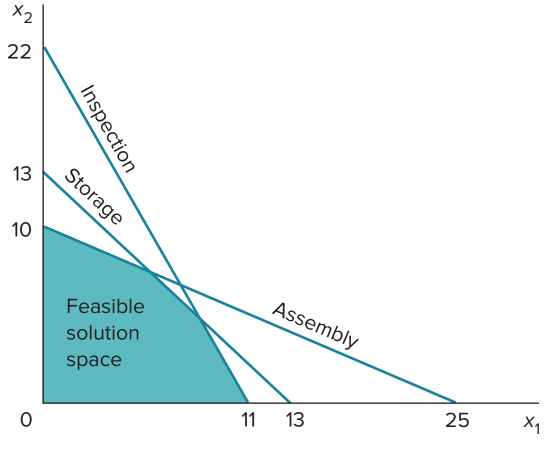


FIGURE 19.6 Completed graph of the microcomputer problem showing all constraints and the feasible solution space

This is not an equation because it does not include an equal sign. We can get around this by simply setting it equal to some quantity. Any quantity will do, although one that is evenly divisible by both coefficients is desirable.

Suppose we decide to set the objective function equal to 300. That is,

$$60x_1 + 50x_2 = 300$$

We can now plot the line on our graph. As before, we can determine the x_1 and x_2 intercepts of the line by setting one of the two variables equal to zero, solving for the other, and then reversing the process. Thus, with $x_1 = 0$, we have

$$60(0) + 50x_2 = 300$$

Solving, we find $x_2 = 6$. Similarly, with $x_2 = 0$, we have

$$60x_1 + 50(0) = 300$$

Solving, we find $x_1 = 5$. This line is plotted in Figure 19.7.

The profit line can be interpreted in the following way: It is an *isoprofit* line; every point on the line (i.e., every combination of x_1 and x_2 that lies on the line) will provide a profit of \$300. We can see from the graph many combinations that are both on the \$300 profit line and within the feasible solution space. In fact, considering noninteger as well as integer solutions, the possibilities are infinite.

Suppose we now consider another line, say the \$600 line. To do this, we set the objective function equal to this amount. Thus,

$$60x_1 + 50x_2 = 600$$

Solving for the x_1 and x_2 intercepts yields these two points:

x_1 intercept	x_2 intercept
$x_1 = 10$	$x_1 = 0$
$x_2 = 0$	$x_2 = 12$

This line is plotted in Figure 19.8, along with the previous \$300 line for purposes of comparison.

Two things are evident in Figure 19.8 regarding the profit lines. One is that the \$600 line *is farther* from the origin than the \$300 line; the other is that the two lines are *parallel.* The lines are parallel because they both have the same slope. The slope is not affected by the right side of the equation. Rather, it is determined solely by the coefficients 60 and 50. It would

FIGURE 19.7 Microcomputer problem with \$300 profit line added

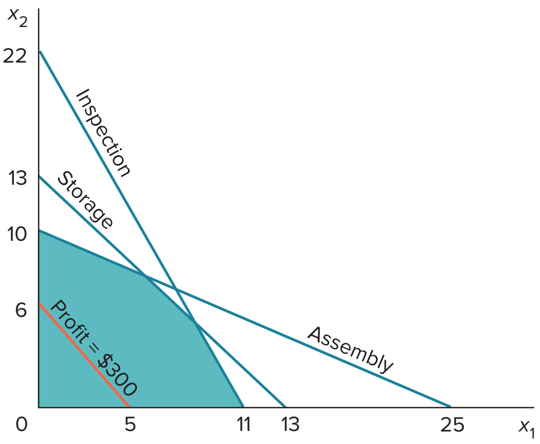
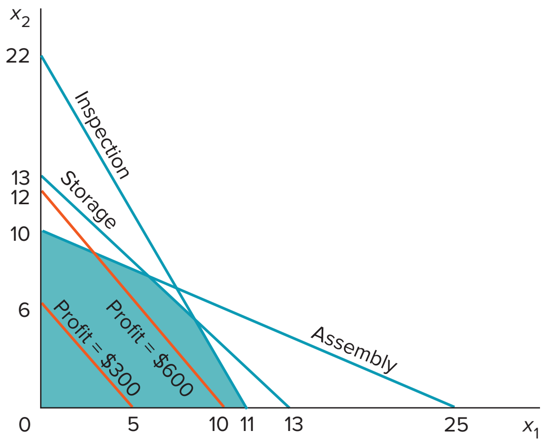


FIGURE 19.8 Microcomputer problem with profit lines of \$300 and \$600



be correct to conclude that regardless of the quantity we select for the value of the objective function, the resulting line will be parallel to these two lines. Moreover, if the amount is greater than 600, the line will be even farther away from the origin than the $600 line. If the value is less than 300, the line will be closer to the origin than the $300 line. And if the value is between 300 and 600, the line will fall between the $300 and $600 lines. This knowledge will help in determining the optimal solution.

Consider a third line, one with the profit equal to $900. Figure 19.9 shows that line along with the previous two profit lines. As expected, it is parallel to the other two, and even farther away from the origin. However, the line does not touch the feasible solution space at all. Consequently, there is no feasible combination of x_1 and x_2 that will yield that amount of profit. Evidently, the maximum possible profit is an amount between $600 and $900, which we can see by referring to Figure 19.9. We could continue to select profit lines in this manner, and eventually, we could determine an amount that would yield the greatest profit. However, there is a much simpler alternative. We can plot just one line, say the $300 line. We know that all other lines will be parallel to it. Consequently, by moving this one line parallel to itself we can "test" other profit lines. We also know that as we move away from the origin, the profits get larger. What we want to know is how far the line can be moved out from the origin and still be touching the feasible solution space, and the values of the decision variables at that point of greatest profit (i.e., the optimal solution). Locate this point on the graph by placing a straight edge along the $300 line (or any other convenient line) and sliding it away from the origin, being careful to keep it parallel to the line. This approach is illustrated in Figure 19.10.

Once we have determined where the optimal solution is in the feasible solution space, we must determine the values of the decision variables at that point. Then, we can use that information to compute the profit for that combination.

Note that the optimal solution is at the intersection of the inspection boundary and the storage boundary (see Figure 19.10). In other words, the optimal combination of x_1 and x_2 must satisfy both boundary (equality) conditions. We can determine those values by solving the two equations *simultaneously.* The equations are:

Inspection $2x_1 + 1x_2 = 22$

Storage $3x_1 + 3x_2 = 39$

The idea behind solving two *simultaneous equations* is to algebraically eliminate one of the unknown variables (i.e., to obtain an equation with a single unknown). This can be accomplished by multiplying the constants of one of the equations by a fixed amount and then adding (or subtracting) the modified equation from the other. (Occasionally, it is easier to multiply each equation by a fixed quantity.) For example, we can eliminate x_2 by multiplying

FIGURE 19.9 Microcomputer problem with profit lines of $300, $600, and $900

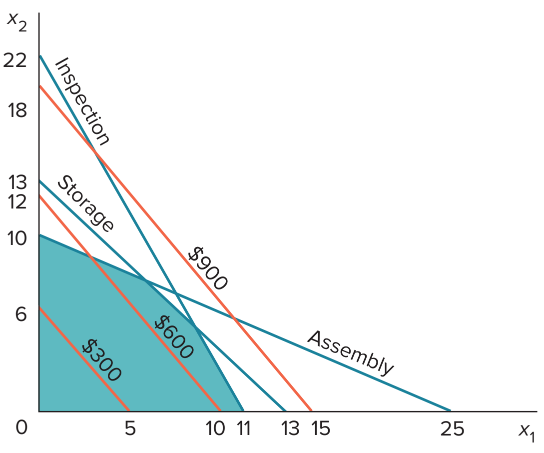
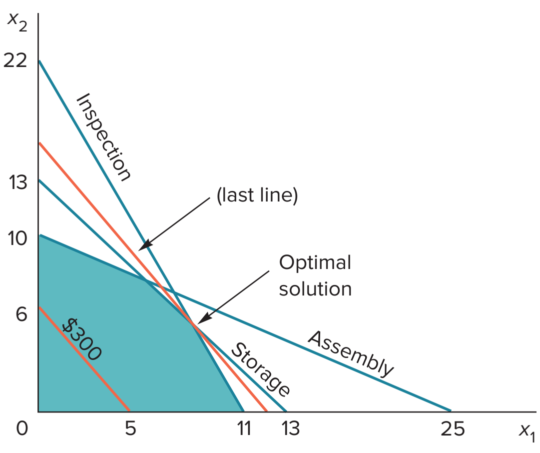


FIGURE 19.10 Finding the optimal solution to the microcomputer problem



the inspection equation by 3 and then subtracting the storage equation from the modified inspection equation. Thus,

$$3(2x_1 + 1x_2 = 22) \quad \text{becomes} \quad 6x_1 + 3x_2 = 66$$

Subtracting the storage equation from this produces

$$\begin{array}{r} 6x_1 + 3x_2 \ = 66 \\ -(3x_1 + 3x_2 \ = 39) \\ \hline 3x_1 + 0x_2 \ = 27 \end{array}$$

Solving the resulting equation yields $x_1 = 9$. The value of x_2 can be found by substituting $x_1 = 9$ into either of the original equations or the modified inspection equation. Suppose we use the original inspection equation. We have

$$2(9) + 1x_2 = 22$$

Solving, we find $x_2 = 4$.

Hence, the optimal solution to the microcomputer problem is to produce nine type 1 computers and four type 2 computers per day. We can substitute these values into the objective function to find the optimal profit:

$$\$60(9) + \$50(4) = \$740$$

Hence, the last line—the one that would last touch the feasible solution space as we moved away from the origin parallel to the $300 profit line—would be the line where profit equaled $740.

In this problem, the optimal values for both decision variables are integers. This will not always be the case; one or both of the decision variables may turn out to be noninteger. In some situations noninteger values would be of little consequence. This would be true if the decision variables were measured on a continuous scale, such as the amount of water, sand, sugar, fuel oil, time, or distance needed for optimality, or if the contribution per unit (profit, cost, etc.) were small, as with the number of nails or ball bearings to make. In some cases, the answer would simply be rounded down (maximization problems) or up (minimization problems) with very little impact on the objective function. Here, we assume that noninteger answers are acceptable as such.

Let's review the procedure for finding the optimal solution using the objective function approach:

1. Graph the constraints.
2. Identify the feasible solution space.
3. Set the objective function equal to some amount that is divisible by each of the objective function coefficients. This will yield integer values for the x_1 and x_2 intercepts and simplify plotting the line. Often, the product of the two objective function coefficients provides a satisfactory line. Ideally, the line will cross the feasible solution space close to the optimal point, and it will not be necessary to slide a straight edge because the optimal solution can be readily identified visually.
4. After identifying the optimal point, determine which two constraints intersect there. Solve their equations simultaneously to obtain the values of the decision variables at the optimum.
5. Substitute the values obtained in the previous step into the objective function to determine the value of the objective function at the optimum.

Redundant Constraints

Redundant constraint A constraint that does not form a unique boundary of the feasible solution space.

In some cases, a constraint does not form a unique boundary of the feasible solution space. Such a constraint is called a **redundant constraint.** Two such constraints are illustrated in Figure 19.11. Note that a constraint is redundant if it meets the following test: Its removal would not alter the feasible solution space.

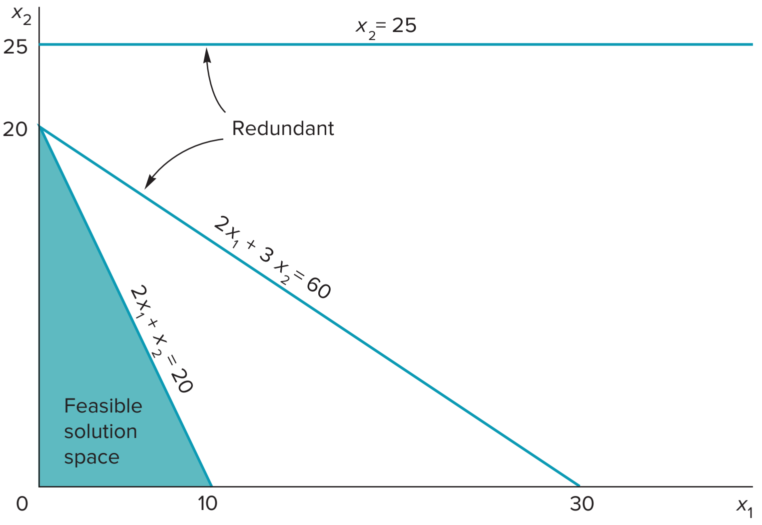


FIGURE 19.11
Examples of redundant constraints

When a problem has a redundant constraint, at least one of the other constraints in the problem is more restrictive than the redundant constraint.

Solutions and Corner Points

The feasible solution space in graphical linear programming is typically a polygon. Moreover, the solution to any problem will always be at a corner point (intersection of constraints) of the polygon. It is possible to determine the coordinates of each corner point of the feasible solution space, and use those values to compute the value of the objective function at those points. Because the solution is always at a corner point, comparing the values of the objective function at the corner points and identifying the best one (e.g., the maximum value) is another way to identify the optimal corner point. Using the graphical approach, it is much easier to plot the objective function and use that to identify the optimal corner point. However, for problems that have more than two decision variables, and the graphical method isn't appropriate, the "enumeration" approach is used to find the optimal solution.

With the **enumeration approach,** the coordinates of each corner point are determined, and then each set of coordinates is substituted into the objective function to determine its value at that corner point. After all corner points have been evaluated, the one with the maximum or minimum value (depending on whether the objective is to maximize or minimize) is identified as the optimal solution.

Enumeration approach Substituting the coordinates of each corner point into the objective function to determine which corner point is optimal.

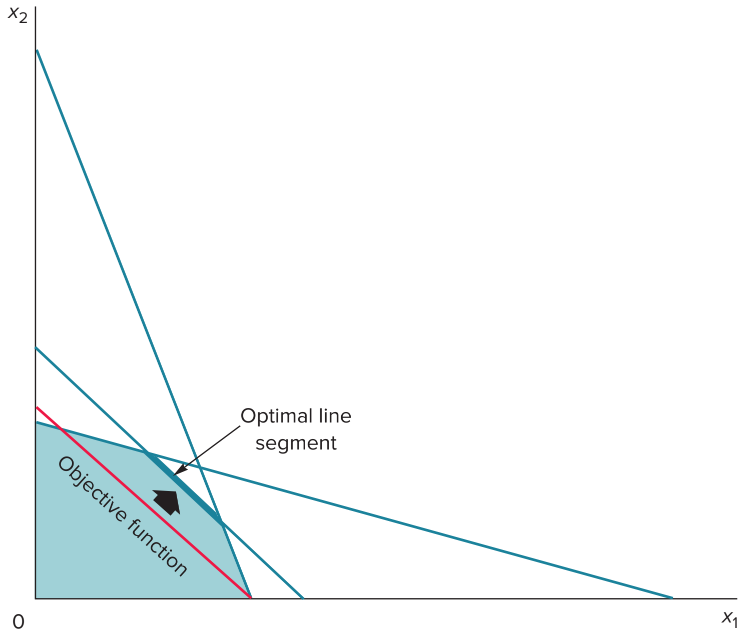
Thus, in the microcomputer problem, the corner points are $x_1 = 0$, $x_2 = 10$, $x_1 = 11$, $x_2 = 0$ (by inspection; see Figure 19.10), and $x_1 = 9$, $x_2 = 4$ and $x_1 = 5$, $x_2 = 8$ (using simultaneous equations, as illustrated on the previous page). Substituting into the objective function, the values are \$500 for (0,10); \$740 for (9,4); \$660 for (11,0), and \$700 for (5,8). Because (9,4) yields the highest value, that corner point is the optimal solution.

In some instances, the objective function will be *parallel* to one of the constraint lines that forms a *boundary of the feasible solution space*. When this happens, *every* combination of $x_1$and x_2 on the segment of the constraint that touches the feasible solution space represents an optimal solution. Hence, there are multiple optimal solutions to the problem. Even in such a case, the solution will also be a corner point—in fact, the solution will be at *two* corner points: those at the ends of the segment that touches the feasible solution space. Figure 19.12 illustrates an objective function line that is parallel to a constraint line.

Minimization

Graphical minimization problems are quite similar to maximization problems. There are, however, two important differences. One is that at least one of the constraints must be of the = or ≥ variety. This causes the feasible solution space to be away from the origin. The other difference is that the optimal point is the one closest to the origin. We find the optimal corner point by sliding the objective function (which is an *isocost* line) *toward* the origin instead of away from it.

FIGURE 19.12
Some LP problems have multiple optimal solutions



EXAMPLE 3

Solving a Minimization Problem

Solve the following problem using graphical linear programming.

Minimize $Z = 8x_1 + 12x_2$

Subject to

$$5x_1 + 2x_2 \geq 20$$
$$4x_1 + 3x_2 \geq 24$$
$$x_2 \geq 2$$
$$x_1 x_2 \geq 0$$

SOLUTION

1. Plot the constraints (shown in Figure 19.13).
   a. Change the constraints to equalities.
   b. For each constraint, set $x_1 = 0$ and solve for x_2, then set $x_2 = 0$ and solve for x_1.
   c. Graph each constraint. Note that $x_2 = 2$ is a horizontal line parallel to the x_1 axis and 2 units above it.
2. Shade the feasible solution space (see Figure 19.13).
3. Plot the objective function.
   a. Select a value for the objective function that causes it to cross the feasible solution space. Try $8 \times 12 = 96$; $8x_1 + 12x_2 = 96$ (acceptable).
   b. Graph the line (see Figure 19.14).
4. Slide the objective function toward the origin, being careful to keep it parallel to the original line.
5. The optimum (last feasible point) is shown in Figure 19.14. The x_2 coordinate ($x_2 = 2$) can be determined by inspection of the graph. Note that the optimum point is at the intersection of the line $x_2 = 2$ and the line $4x_1 + 3x_2 = 24$. Substituting the value of $x_2 = 2$ into the latter equation will yield the value of x_1 at the intersection:

   $4x_1 + 3(2) = 24 \quad x_1 = 4.5$

   Thus, the optimum is $x_1 = 4.5$ units and $x_2 = 2$.
6. Compute the minimum cost:

   $8x_1 + 12x_2 = 8(4.5) + 12(2) = 60$

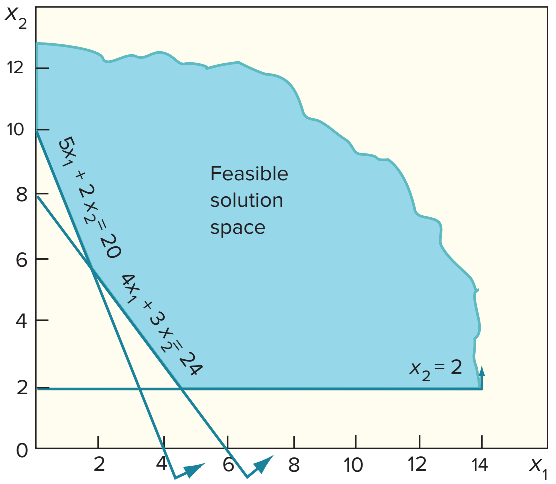


FIGURE 19.13 The constraints define the feasible solution space

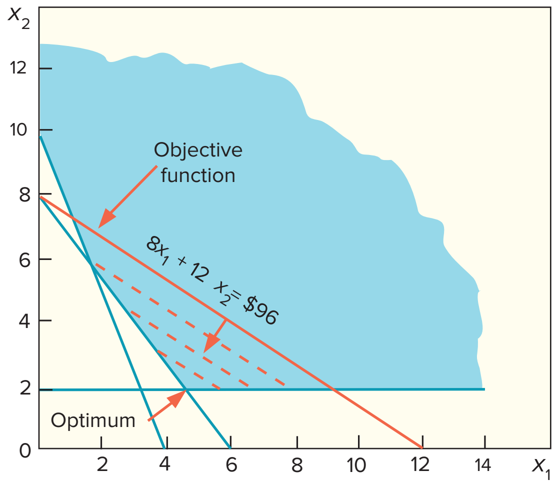


FIGURE 19.14 The optimum is the last point the objective function touches as it is moved toward the origin

Slack and Surplus

If a constraint forms the optimal corner point of the feasible solution space, it is called a **binding constraint.** In effect, it limits the value of the objective function; if the constraint could be relaxed (less restrictive), an improved solution would be possible. For constraints that are not binding, making them less restrictive will have no impact on the solution.

Binding constraint A constraint that forms the optimal corner point of the feasible solution space.

If the optimal values of the decision variables are substituted into the left-hand side of a binding constraint, the resulting value will exactly equal the right-hand value of the constraint. However, there will be a difference with a nonbinding constraint. If the left-hand side is greater than the right-hand side, we say that there is **surplus;** if the left-hand side is less than the right-hand side, we say that there is **slack.** Slack can only occur in a ≤ constraint; it is the amount by which the left-hand side is less than the right-hand side when the optimal values of the decision variables are substituted into the left-hand side. And surplus can only occur in a ≥ constraint; it is the amount by which the left-hand side exceeds the right-hand side of the constraint when the optimal values of the decision variables are substituted into the left-hand side.

Surplus When the values of decision variables are substituted into a ≥ constraint, the amount by which the resulting value exceeds the right-hand-side value.

Slack When the values of decision variables are substituted into a ≤ constraint, the amount by which the resulting value is less than the right-hand-side value.

For example, suppose the optimal values for a problem are $x_1 = 10$ and $x_2 = 20$. If one of the constraints is

$$3x_1 + 2x_2 \leq 100$$

substituting the optimal values into the left-hand side yields

$$3(10) + 2(20) = 70$$

Because the constraint is ≤, the difference between the values of 100 and 70 (i.e., 30) is *slack*. Suppose the optimal values had been $x_1 = 20$ and $x_2 = 20$. Substituting these values into the left-hand side of the constraint would yield $3(20) + 2(20) = 100$. Because the left-hand side equals the right-hand side, this is a binding constraint; slack is equal to zero.

Now consider this constraint:

$$4x_1 + x_2 \geq 50$$

Suppose the optimal values are $x_1 = 10$ and $x_2 = 15$; substituting into the left-hand side yields

$$4(10) + 15 = 55$$

Because this is a ≥ constraint, the difference between the left- and right-hand-side values is *surplus*. If the optimal values had been $x_1 = 12$ and $x_2 = 2$, substitution would result in the left-hand side being equal to 50. Hence, the constraint would be a binding constraint, and there would be no surplus (i.e., surplus would be zero).

19.4 THE SIMPLEX METHOD

Simplex A linear programming algorithm that can solve problems having more than two decision variables.

The **simplex** method is a general-purpose linear programming algorithm widely used to solve large-scale problems. Although it lacks the intuitive appeal of the graphical approach, its ability to handle problems with more than two decision variables makes it extremely valuable for solving problems often encountered in operations management.

Although manual solution of linear programming problems using simplex can yield a number of insights into how solutions are derived, space limitations preclude describing it here. However, it is available on the website that accompanies this book. The discussion here will focus on computer solutions.

19.5 COMPUTER SOLUTIONS

LO19.4 Interpret computer solutions of linear programming problems.

The microcomputer problem will be used to illustrate computer solutions. We repeat it here for ease of reference.

Maximize $60x_1 + 50x_2$ where x_1 = the number of type 1 computers
x_2 = the number of type 2 computers

Subject to

Assembly	$4x_1 + 10x_2$	$\leq$ 100 hours
Inspection	$2x_1 + 1x_2$	$\leq$ 22 hours
Storage	$3x_1 + 3x_2$	$\leq$ 39cubic feet
	x_1, x_2	≥ 0

Solving LP Models Using MS Excel

Solutions to linear programming models can be obtained from spreadsheet software such as Microsoft's Excel. Excel has a routine called Solver that performs the necessary calculations.

To use Solver:

SCREENCAM TUTORIAL

1. First, enter the problem in a worksheet, as shown in Figure 19.15. What is not obvious from the figure is the need to enter a formula for each cell where there is a zero (Solver automatically inserts the zero after you input the formula). The formulas are for the value of the objective function and the constraints, in the appropriate cells. Before you enter the formulas, designate the cells where you want the optimal values of x_1 and x_2. Here, cells D4 and E4 are used. To enter a formula, click on the cell that the formula will pertain to, and then enter the formula, starting with an equal sign. We want the optimal value of the objective function to appear in cell G4. For G4, enter the formula

   = 60*D4 + 50*E4

   The constraint formulas, using cells C7, C8, and C9, are

   for C7: = 4*D4 + 10*E4

   for C8: = 2*D4 + 1*E4

   for C9: = 3*D4 + 3*E4

2. Now, to access Solver in Excel 2010 or 2007, click Data at the top of the worksheet, and in that ribbon, click on Solver in the Analysis group. In Excel 2010 the Solver menu will appear as illustrated in Figure 19.16. If it does not appear there, it must be enabled using the Add-ins menu. Begin by setting the Objective (i.e., indicating the cell where you want the optimal value of the objective function to appear). Note, if the activated cell is the cell designated for the value of Z when you click on Solver, Solver will automatically set that cell as the Objective.

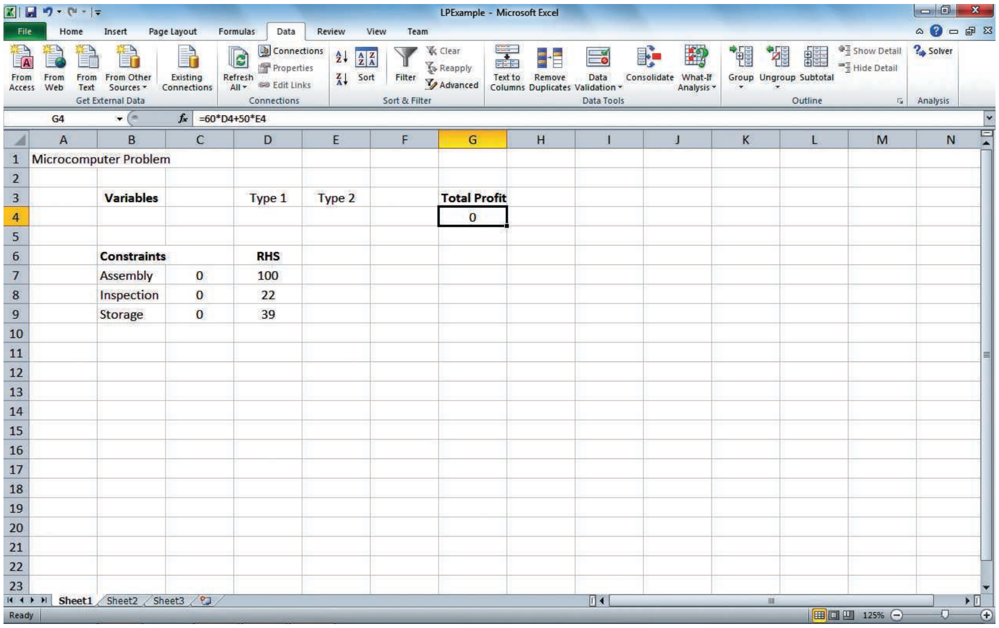


FIGURE 19.15
MS Excel worksheet for microcomputer problem

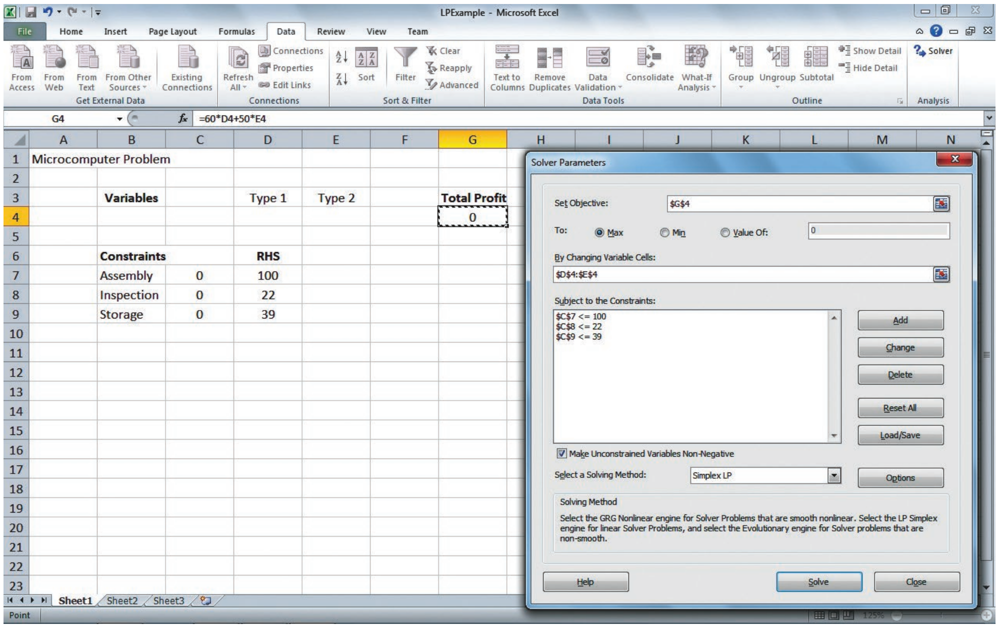


FIGURE 19.16
MS Excel Solver parameters for microcomputer problem

Select the Max radio button if it isn't already selected. The Changing Variable Cells are the cells where you want the optimal values of the decision variables to appear. Here, they are cells D4 and E4. We indicate this by the range D4:E4 (Solver will add the $ signs).

Finally, add the constraints by clicking on Add . . . When that menu appears, for each constraint, enter the cell that contains the formula for the left-hand side of the constraint, then select the appropriate inequality sign, and then enter the right-hand-side amount of the cell that has the right-hand-side amount. Here the right-hand-side amounts are used. After you have entered each constraint, either click on Add to add another constraint or click on OK to return to the Solver menu. (*Note:* Constraints can be entered in any order, and if cells are used for the right-hand side, then constraints with the same inequality could be grouped.) For the nonnegativity constraints simply check the checkbox to Make Unconstrained Variables Non-Negative. Also select Simplex LP as the Solving Method. Click on Solve.

3. The Solver Results menu will then appear, indicating that a solution has been found, or that an error has occurred. If there has been an error, go back to the Solver Parameters menu and check to see that your constraints refer to the correct changing cells, and that the inequality directions are correct. Make the corrections and click on Solve.

   Assuming everything is correct, in the Solver Results menu, in the Reports box, highlight both Answer and Sensitivity, and then click OK.

4. Solver will incorporate the optimal values of the decision variables and the objective function in your original layout on your worksheet (see Figure 19.17). We can see that the optimal values are type 1 = 9 units and type 2 = 4 units, and the total profit is 740. The answer report will also show the optimal values of the decision variables (middle part of Figure 19.18), and some information on the constraints (lower part of Figure 19.18). Of particular interest here is the indication of which constraints have slack and how much slack.

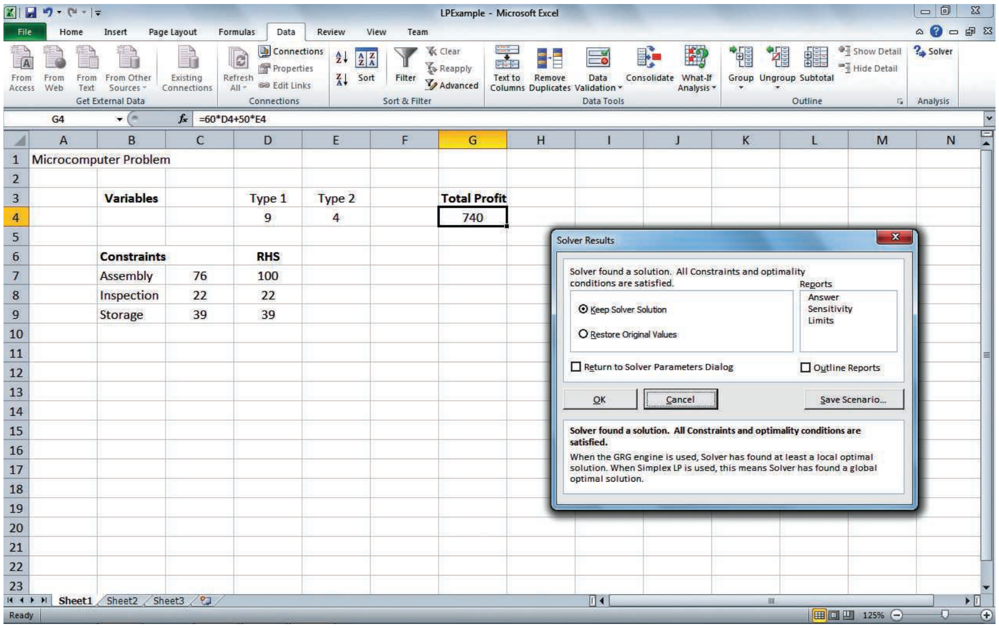


FIGURE 19.17
MS Excel worksheet solution for microcomputer problem

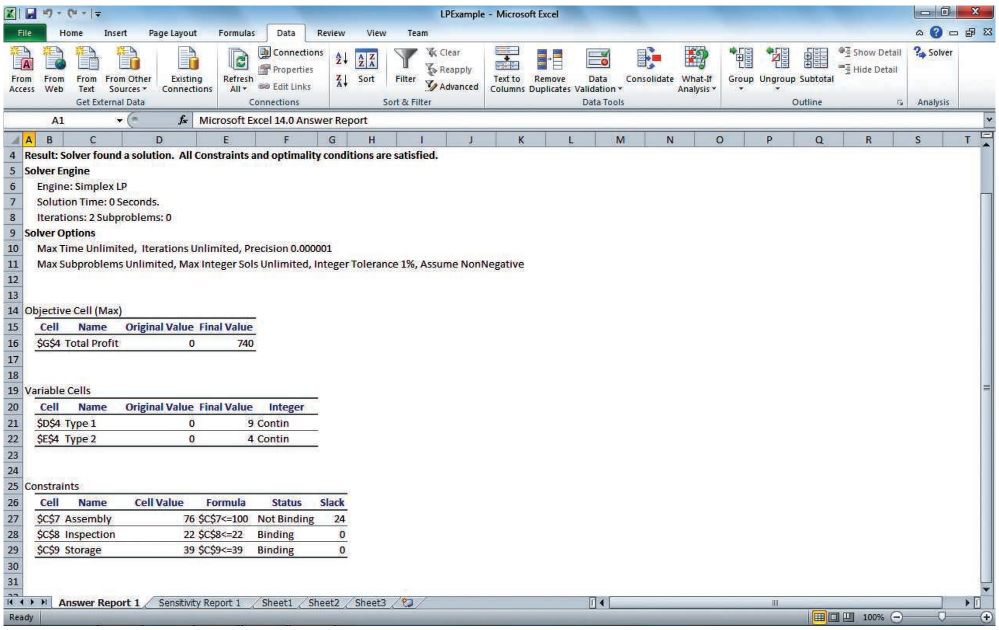


FIGURE 19.18
MS Excel Answer Report for microcomputer problem

We can see that the constraint entered in cell C7 (assembly) has a slack of 24, and that the constraints entered in cells C8 (inspection) and C9 (storage) have slack equal to zero, indicating that they are binding constraints.

19.6 SENSITIVITY ANALYSIS

Sensitivity analysis is a means of assessing the impact of potential changes to the parameters (the numerical values) of an LP model. Such changes may occur due to forces beyond a manager's control, or a manager may be contemplating making the changes, say, to increase profits or reduce costs.

Sensitivity analysis Assessing the impact of potential changes to the numerical values of an LP model.

There are three types of potential changes:

1. Objective function coefficients
2. Right-hand values of constraints
3. Constraint coefficients

LO19.5 Do sensitivity analysis on the solution to a linear programming problem.

We will consider the first two of these here. We begin with changes to objective function coefficients.

SCREENCAM TUTORIAL

Objective Function Coefficient Changes

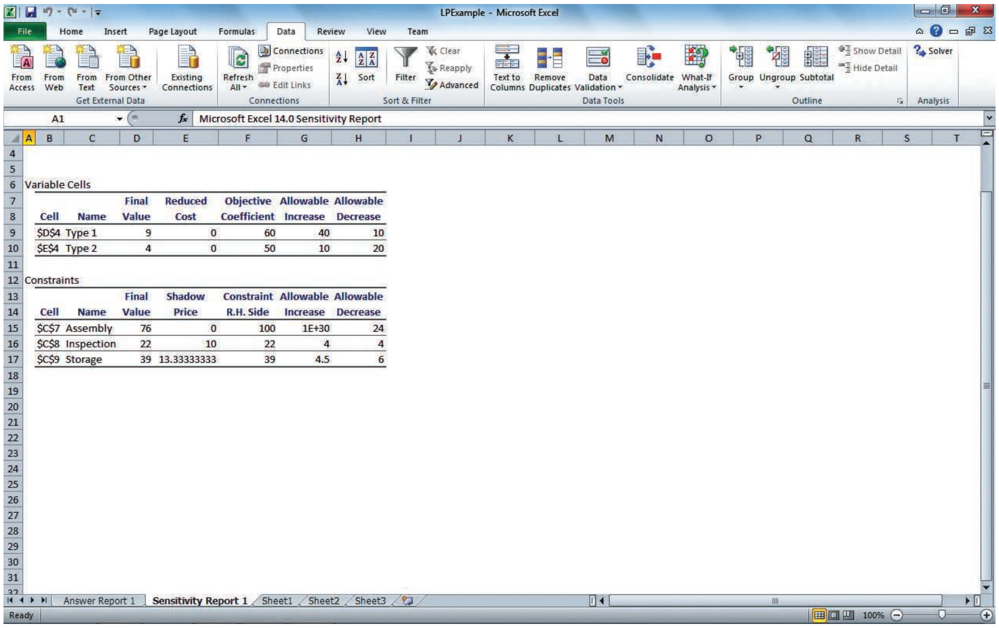
A change in the value of an objective function coefficient can cause a change in the optimal solution of a problem. In a graphical solution, this would mean a change to another corner point of the feasible solution space. However, not every change in the value of an objective function coefficient will lead to a changed solution; generally there is a *range of values for which the optimal values of the decision variables will not change.* For example, in the microcomputer problem, if the profit on type 1 computers increased from $60 per unit to, say, $65 per unit, the optimal solution would still be to produce nine units of type 1 and four units of type 2 computers. Similarly, if the profit per unit on type 1 computers decreased from $60 to, say, $58, producing nine of type 1 and four of type 2 would still be optimal. These sorts of changes are not uncommon; they may be the result of such things as price changes in raw materials, price discounts, cost reductions in production, and so on. Obviously, when a change does occur in the value of an objective function coefficient, it can be helpful for a manager to know if that change will affect the optimal values of the decision variables. The manager can quickly determine this by referring to that coefficient's **range of optimality**, which is the range in possible values of that objective function coefficient over which the optimal values of the decision variables will not change. Before we see how to determine the range, consider the implication of the range. The range of optimality for the type 1 coefficient in the microcomputer problem is 50 to 100. That means that as long as the coefficient's value is in that range, the optimal values will be nine units of type 1 and four units of type 2. Conversely, *if a change extends beyond the range of optimality, the solution will change.*

Range of optimality Range of values over which the solution quantities of all the decision variables remain the same.

Similarly, suppose instead that the coefficient (unit profit) of type 2 computers was to change. Its range of optimality is 30 to 60. As long as the change doesn't take it outside of this range, nine and four will still be the optimal values. Note, however, even for changes that are *within* the range of optimality, the optimal value of the objective function *will* change. If the type 1 coefficient increased from $60 to $61, and nine units of type 1 is still optimum, profit would increase by $9: nine units times $1 per unit. Thus, for a change that is within the range of optimality, a revised value of the objective function must be determined.

Now let's see how we can determine the range of optimality using computer output.

Using MS Excel. There is a table for the Changing Cells (see Figure 19.19). It shows the value of the objective function that was used in the problem for each type of computer (i.e., 60 and 50), and the allowable increase and allowable decrease for each coefficient. By subtracting the allowable decrease from the original value of the coefficient, and adding the allowable



Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
D4	Type 1	9	0	60	40	10
E4	Type 2	4	0	50	10	20

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
C7	Assembly	76	0	100	1E+30	24
C8	Inspection	22	10	22	4	4
C9	Storage	39	13.33333333	39	4.5	6

FIGURE 19.19
MS Excel sensitivity report for microcomputer problem

increase to the original value of the coefficient, we obtain the range of optimality for each coefficient. Thus, we find for type 1:

$$60 - 10 = 50 \text{ and } 60 + 40 = 100$$

Hence, the range for the type 1 coefficient is 50 to 100. For type 2:

$$50 - 20 = 30 \text{ and } 50 + 10 = 60$$

Hence the range for the type 2 coefficient is 30 to 60.

In this example, both of the decision variables are *basic* (i.e., nonzero). However, in other problems, one or more decision variables may be *nonbasic* (i.e., have an optimal value of zero). In such instances, unless the value of that variable's objective function coefficient increases by more than its *reduced cost,* it won't come into solution (i.e., become a basic variable). Hence, the range of optimality (sometimes referred to as the *range of insignificance)* for a nonbasic variable is from negative infinity to the sum of its current value and its reduced cost.

Now let's see how we can handle multiple changes to objective function coefficients, that is, a change in more than one coefficient. To do this, divide each coefficient's change by the allowable change in the same direction. Thus, if the change is a decrease, divide that amount by the allowable decrease. Treat all resulting fractions as positive. Sum the fractions. If the sum does not exceed 1.00, then multiple changes are within the range of optimality and will not result in any change to the optimal values of the decision variables.

Changes in the Right-Hand-Side (RHS) Value of a Constraint

In considering right-hand-side (RHS) changes, it is important to know if a particular constraint is binding on a solution. A constraint is binding if substituting the values of the decision variables of that solution into the left-hand side of the constraint results in a value that is equal to the RHS value. In other words, that constraint stops the objective function from achieving a better value (e.g., a greater profit or a lower cost). Each constraint has a corresponding **shadow price**, which is a marginal value that indicates the amount by which the value of the objective function would change if there were a one-unit change in the RHS value of that constraint. If a constraint is nonbinding, its shadow price is zero, meaning that increasing or decreasing its RHS value by one unit will have no impact on the value of the objective

Shadow price Amount by which the value of the objective function would change with a one-unit change in the RHS value of a constraint.

function. Nonbinding constraints have either slack (if the constraint is ≤) or surplus (if the constraint is ≥). Suppose a constraint has 10 units of slack in the optimal solution, which means 10 units that are unused. If we were to increase or decrease the constraint's RHS value by one unit, the only effect would be to increase or decrease its slack by one unit. But there is no profit associated with slack, so the value of the objective function wouldn't change. On the other hand, if the change is to the RHS value of a binding constraint, then the optimal value of the objective function would change. Any change in a binding constraint will cause the optimal values of the decision variables to change, and hence, cause the value of the objective function to change. For example, in the microcomputer problem, the inspection constraint is a binding constraint; it has a shadow price of 10. That means if there was one hour less of inspection time, total profit would decrease by $10, or if there was one more hour of inspection time available, total profit would increase by $10. In general, multiplying the amount of change in the RHS value of a constraint by the constraint's shadow price will indicate the change's impact on the optimal value of the objective function. However, this is only true over a limited range called the **range of feasibility**. In this range, the value of the shadow price remains constant. Hence, as long as a change in the RHS value of a constraint is within its range of feasibility, the shadow price will remain the same, and one can readily determine the impact on the objective function.

Range of feasibility Range of values for the RHS of a constraint over which the shadow price remains the same.

Let's see how to determine the range of feasibility from computer output.

Using MS Excel. In the sensitivity report there is a table labeled "Constraints" (see Figure 19.19). The table shows the shadow price for each constraint, its RHS value, and the allowable increase and allowable decrease. Adding the allowable increase to the RHS value and subtracting the allowable decrease will produce the range of feasibility for that constraint. For example, for the inspection constraint, the range would be

$$22 - 4 = 18; 22 + 4 = 26$$

Hence, the range of feasibility for inspection is 18 to 26 hours. Similarly, for the storage constraint, the range is

$$39 - 6 = 33 \text{ to } 39 + 4.5 = 43.5$$

The range for the assembly constraint is a little different; the assembly constraint is nonbinding (note the shadow price of 0) while the other two are binding (note their nonzero shadow prices). The assembly constraint has a slack of 24 (the difference between its RHS value of 100 and its final value of 76). With its slack of 24, its RHS value could be decreased by as much as 24 (to 76) before it would become binding. Conversely, increasing its right-hand side will only produce more slack. Thus, no amount of increase in the RHS value will make it binding, so there is no upper limit on the allowable increase. Excel indicates this by the large value (1E + 30) shown for the allowable increase. So its range of feasibility has a lower limit of 76 and no upper limit.

If there are changes to more than one constraint's RHS value, analyze these in the same way as multiple changes to objective function coefficients. That is, if the change is an increase, divide that amount by that constraint's allowable increase; if the change is a decrease, divide the decrease by the allowable decrease. Treat all resulting fractions as positives. Sum the fractions. As long as the sum does not exceed 1.00, the changes are within the range of feasibility for multiple changes, and the shadow prices won't change.

Table 19.1 summarizes the impacts of changes that fall within either the range of optimality or the range of feasibility.

Now let's consider what happens if a change goes beyond a particular range. In a situation involving the range of optimality, a change in an objective function that is beyond the range of optimality will result in a new solution. Hence, it will be necessary to recompute the solution. For a situation involving the range of feasibility, there are two cases to consider. The first case would be increasing the RHS value of a ≤ constraint to beyond the upper limit of its range of feasibility. This would produce slack equal to the amount by which the upper limit is exceeded. Hence, if the upper limit is 200, and the increase is 220, the result is that the

TABLE 19.1
Summary of the impact of changes within their respective ranges

Changes to objective function coefficients that are within the range of optimality	
Component	**Result**
Values of decision variables	No change
Value of objective function	Will change
Changes to RHS values of constraints that are within the range of feasibility	
Component	**Result**
Value of shadow price	No change
List of basic variables	No change
Values of basic variables	Will change
Value of objective function	Will change

constraint has a slack of 20. Similarly, for a $\geq$ constraint, going below its lower bound creates a surplus for that constraint. The second case for each of these would be exceeding the opposite limit (the lower bound for a $\leq$ constraint, or the upper bound for a $\geq$ constraint). In either instance, a new solution would have to be generated.

SUMMARY

Linear programming is a powerful tool used for constrained optimization situations. Components of LP problems include an objective function, decision variables, constraints, and numerical values (parameters) of the objective function and constraints.

The size of real-life problems and the burden of manual solution make computer solutions the practical way to solve real-life problems. Even so, much insight can be gained through the study of simple, two-variable problems and graphical solutions.

KEY POINTS

1. Optimizing techniques such as linear programming help business organizations make the best use of limited resources such as materials, time, and energy, to maximize profits or to minimize costs.
2. As with all techniques, it is important to confirm that the underlying assumptions on which the technique is based are reasonably satisfied by the model in order to achieve valid results.
3. Although the graphical technique has limited use due to the fact that it can only handle two-variable problems, it is very useful in conveying many of the important concepts associated with linear programming techniques.

KEY TERMS

binding constraint, 837
constraints, 824
decision variables, 824
enumeration approach, 835
feasible solution space, 824
graphical linear programming, 826
objective function, 824
parameters, 824
range of feasibility, 843
range of optimality, 841
redundant constraint, 834
sensitivity analysis, 841
shadow price, 842
simplex, 838
slack, 837
surplus, 837

SOLVED PROBLEMS

Problem 1

A small construction firm specializes in building and selling single-family homes. The firm offers two basic types of houses, model A and model B. Model A houses require 4,000 labor hours, 2 tons of stone, and 2,000 board feet of lumber. Model B houses require 10,000 labor hours, 3 tons of stone, and 2,000 board feet of lumber. Due to long lead times for ordering supplies and the scarcity of skilled and semiskilled workers in the area, the firm will be forced to rely on its present resources for the upcoming building season. It has 400,000 hours of labor, 150 tons of stone, and 200,000 board feet of lumber. What mix of model A and B houses should the firm construct if model A yields