CHAPTER 14
SOLUTION CONCEPTS FOR LINEAR PROGRAMMING

Learning Objectives

After completing this chapter, you should be able to

1. Describe where an optimal solution can be located in the feasible region of a linear programming problem.
2. Identify the different possibilities for how many optimal solutions a linear programming problem can have.
3. Explain how a linear programming problem could have no optimal solution.
4. Describe the role of corner points in searching for an optimal solution of a linear programming problem.
5. Summarize how the simplex method uses corner points to find an optimal solution for a linear programming problem.
6. Identify the six key solution concepts that make the simplex method so efficient.
7. Use the simplex method by hand to solve small linear programming problems.
8. Discuss the computer implementation of the simplex method.
9. Identify the key solution concept for the interior-point approach to solving linear programming problems.
10. Describe the complementary roles of the simplex method and the interior-point approach.

Chapters 2, 3, and 5 emphasized the variety of managerial problems that can be formulated and analyzed as linear programming problems. Now we take the next step. Once we have formulated an appropriate linear programming model, how do we solve it to find an *optimal solution*?

The easy answer is "click on the Solve button," just as you started doing with the Excel Solver in Chapter 2. For some, that answer is sufficient. After all, managers do not need to know what makes their computer routines run. For those who are content with clicking on the Solve button, this entire chapter can be skipped.

This chapter is aimed instead at those students (or the students of those instructors) who wish to go a little deeper to gain some idea about what lies behind that Solve button.

In the first three sections, we will offer you an intuitive feeling for how linear programming problems are solved. Although the procedures are algebraic in nature, we will focus on the key geometric insights that make them click. The following four sections then illuminate the connections between these geometric insights and the algebraic procedures.

Although software packages for solving linear programming problems can be used without knowing anything about solution procedures, there are six reasons for gaining a basic intuitive understanding of these procedures:

1. To satisfy your curiosity about what are the key ideas that enable the procedures to solve complex problems.
2. To illustrate the systematic solution procedures (called *algorithms*) that are so widely used throughout management science.
3. To gain confidence in the validity and power of these algorithms.
4. To learn something about the limitations of these algorithms.
5. To understand the meaning of some of the unusual outcomes from using these algorithms (e.g., finding that the problem has no optimal solution).

6. To lay the basis for further study if you intend to participate in technical aspects of management science studies.

Although these six teaching goals are being addressed in this chapter in the context of linear programming, many of the same ideas carry over to algorithms used in other areas of management science as well.

So how are linear programming problems typically solved? You saw in Chapter 2 that the *graphical method* is one convenient solution procedure. Alas, this method is limited to problems with just two decision variables. Most real problems have many more decision variables, so we need another approach.

In 1947, George Dantzig developed the *simplex method* for solving linear programming problems with any number of decision variables. Ever since, it has been the standard solution procedure for such problems, and it is the one used by the Excel Solver. There are two reasons why. One is that the simplex method is such an exceptionally efficient procedure. Although spreadsheets are not designed to deal with very large problems, because of the time required to formulate a large model on a spreadsheet, the Excel Solver will quickly solve the model once it has been formulated. With a more powerful software package that does not use spreadsheets (but does use modeling software for efficiently formulating the model and transferring data into the model), the simplex method can solve even huge problems with many thousands of functional constraints and many thousands of decision variables (occasionally even millions of functional constraints and decision variables). The other reason for the use of the simplex method is that it provides the information needed for conducting the what-if analysis described in Chapter 5. All the data given in Solver's sensitivity report are obtained directly from the output of the simplex method.

However, the long reign of the simplex method as the undisputed champion of linear programming solution procedures has ended. In 1984, Narendra Karmarkar discovered a powerful *interior-point approach* to solving linear programming problems. Although this approach still is being refined many years later, it already has enabled solving some massive problems that are beyond the scope of the simplex method. Nevertheless, the interior-point approach will not be supplanting the simplex method as the usual procedure for solving routine linear programming problems. Instead, we anticipate that the two approaches will be playing complementary roles in the coming decades, as described in Section 14.9.

The first section provides background on the characteristics of optimal solutions. The following two sections explore the concepts that enable the simplex method to find an optimal solution so efficiently. Sections 14.4 to 14.7 delve further into the details of the simplex method. Section 14.8 touches on the computer implementation of the simplex method. Section 14.9 introduces the key solution concept for the interior-point approach, and then discusses how this approach fits into the overall picture.

14.1 SOME KEY FACTS ABOUT OPTIMAL SOLUTIONS

You saw in Section 2.4 how the graphical method can be used to find an optimal solution for linear programming problems with two decision variables. However, we did not take the time then to reflect much on the *characteristics* of such solutions. In this section, we will turn our attention to describing some key facts about the characteristics of optimal solutions.

Although we will continue to use the same two-variable Wyndor problem to motivate and illustrate these characteristics, our conclusions also apply for problems with more than two decision variables. Real applications of linear programming typically involve problems with hundreds or thousands of decision variables! These characteristics of optimal solutions will provide some valuable clues about how to analyze large problems.

The Location of Optimal Solutions

The first few key facts concern the question of where an optimal solution can (or cannot) be located in the feasible region.

> **Key Fact 1**: An optimal solution *must* lie on the *boundary* of the feasible region. Algebraically, this says that an optimal solution *must* satisfy the boundary equation for one or more of the constraints (either functional or nonnegativity).

To illustrate this key fact, consider again the Wyndor Glass Co. product-mix problem analyzed in Sections 2.1-2.5. Its linear programming model (in algebraic form) and feasible region are shown again in Figure 14.1. The five dark line segments in this figure form the boundary of the feasible region for this problem. Each dark line segment is a portion of the constraint boundary line for one of the five constraints. [Remember, the boundary lines for the two nonnegativity constraints are the W axis (for $D \geq 0$) and the D axis (for $W \geq 0$).] Thus, the points on each line segment satisfy the corresponding constraint boundary equation (where $D = 0$ and $W = 0$ are the constraint boundary equations for $D \geq 0$ and $W \geq 0$, respectively).

The optimal solution for this problem was found by the graphical method to be $(D, W) = (2, 6)$. This solution does indeed lie on the boundary of the feasible region in Figure 14.1. Furthermore, if the graphical method were to be reapplied after changing the objective function (without changing the constraints), the optimal solution *always* would be a point on this boundary. (Try it.) In fact, we will describe a little later how, by changing the objective function in just the right way, it is possible to make *any* specific point on this boundary an optimal solution. (Do you see how?)

However, the concept we want to emphasize now is that it is *impossible* for a point *inside* the boundary of the feasible region to be an optimal solution. For example, one point lying inside in Figure 14.1 is $(D, W) = (1, 5)$. From *any* such point, it is possible to move in *any* direction by at least a tiny amount and still be feasible. Moving in certain directions will improve the value of the objective function. Therefore, when starting from *any* point *inside* the boundary of the feasible region, it always is possible to move a tiny amount in a direction that improves the value of the objective function and thereby obtain a better feasible solution, so the original point cannot be optimal. This concept applies regardless of how many decision variables the problem has.

Most linear programming problems have just one optimal solution. We will describe some exceptions later. However, by restricting our attention for now to problems with exactly one optimal solution, much more can be said about just where on the boundary of the feasible region this solution must lie.

> **Key Fact 2**: If a linear programming problem has exactly one optimal solution, this solution *must* be a *corner point*.

For a problem with two decision variables, the **corner points** are the corners of the feasible region, i.e., points on the boundary of the feasible region where two constraint boundary lines intersect. The Wyndor problem has five corner points, as shown in Figure 14.2. Since the horizontal and vertical axes are the constraint boundary lines for the nonnegativity constraints, one corner point, $(D, W) = (0,0)$, lies at the intersection of these two axes. Each of the other four corner points also lies at the intersection of a pair of constraint boundary lines.

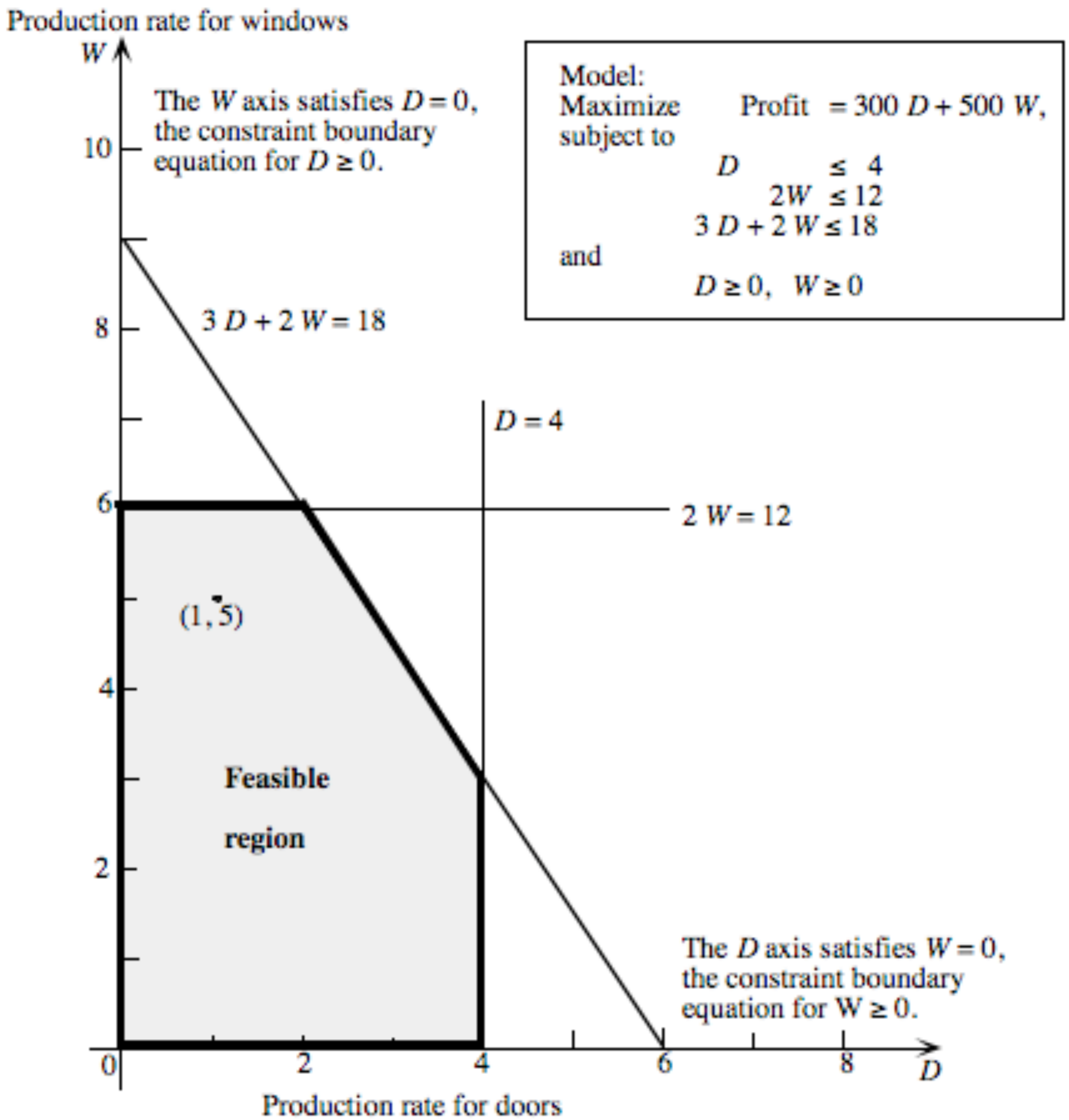


Figure 14.1 Graph highlighting the boundary of the feasible region for the Wyndor problem. Also shown is the constraint boundary equation for each of the five constraint boundary lines.

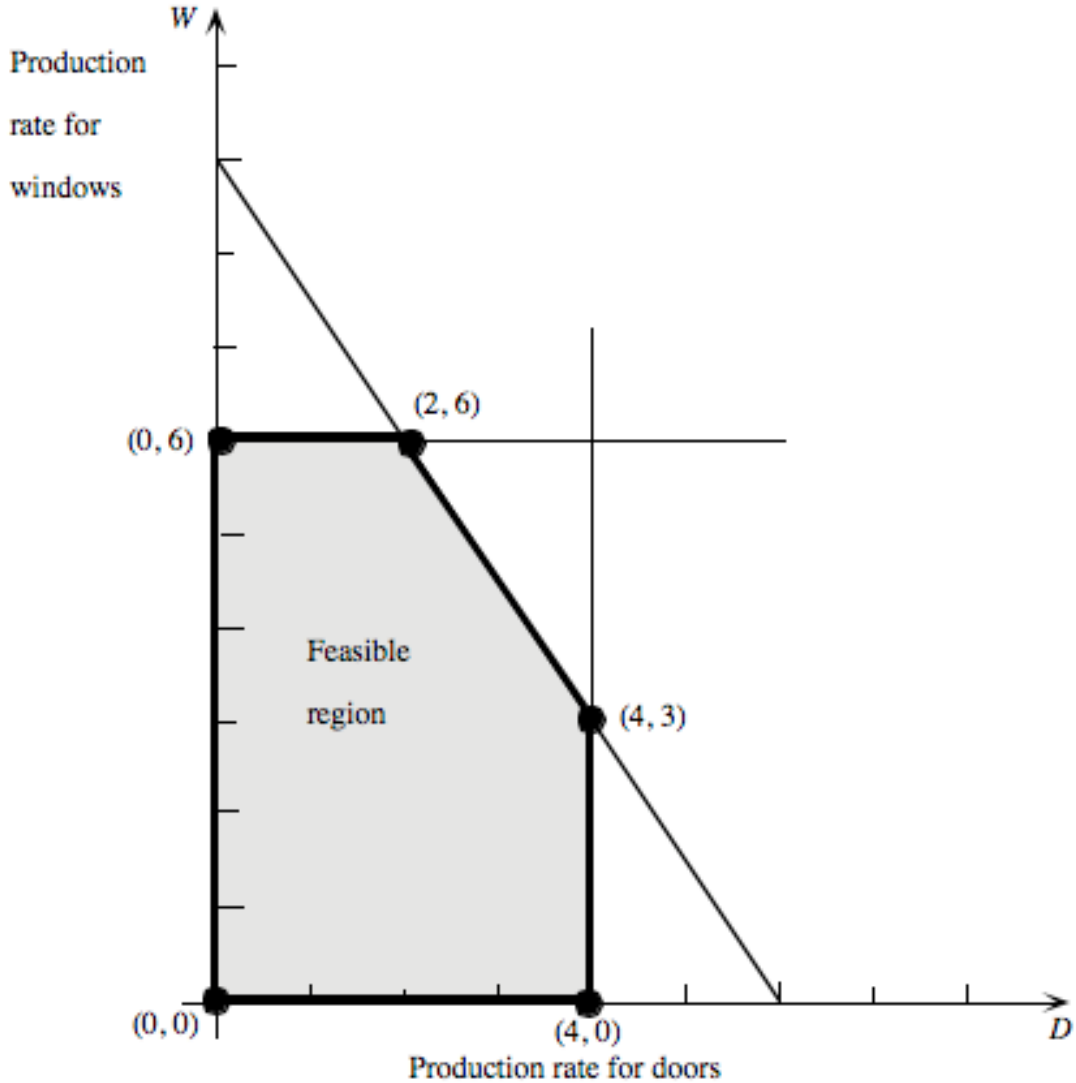


Figure 14.2 Graph highlighting the corner points for the Wyndor problem. Each corner point lies at the intersection of two constraint boundary lines while also satisfying all the other constraints.

This problem has just one optimal solution, (D, W) = (2, 6), which is one of the five corner points. Reviewing how the graphical method obtained this optimal solution (see Figure 2.7) suggests how moving the objective function line in the favorable direction as far as possible leads to the optimal objective function line passing through a corner point. The same thing would happen regardless of what the objective function is. For example, Table 14.1 gives a sampling of cases for the Wyndor problem where the unit profits from the two products are different from the original estimates given in Section 2.1. The coefficients of D and W in the objective function are simply these unit profits, as shown in the third column of the table. The last column gives the resulting optimal solution. Note (and check with the graphical method or Solver) how each of the five corner points in Figure 14.2 is the optimal solution for one of these five cases.

Table 14.1 The Optimal Solution with Different Unit Profits for the Wyndor Problem

Unit Profit		Objective	Optimal
Doors	Windows	Function	Solution
\$400	\$400	Profit = 400D + 400W	(2, 6)
\$500	\$300	Profit = 500D + 300W	(4, 3)
\$300	- \$100	Profit = 300D - 100W	(4, 0)
- \$100	\$500	Profit = -100D + 500W	(0, 6)
- \$100	- \$100	Profit = -100D - 100W	(0, 0)

Key Fact 2 provides a convenient way of finding the optimal solution. You simply need to check each corner point, calculate its objective function value (Profit), and select the one with the largest profit. Since the best corner point is optimal, *all* other feasible solutions can be ignored. Table 14.2 illustrates the application of this method to the original version of the Wyndor problem. This is called the **enumeration-of-corner-points method**.

Table 14.2 Applying the Enumeration-of-Corner-Points Method to Find the Optimal Solution for the Wyndor Problem

	Corner Point	Profit = 300D + 500 W	
	(D, W) = (0, 0)	Profit = 300 (0) + 500 (0) = 0	
	(D, W) = (0, 6)	Profit = 300 (0) + 500 (6) = \$3,000	
Optimal →	(D, W) = (2, 6)	Profit = 300 (2) + 500 (6) = \$3,600	← Best
	(D, W) = (4, 3)	Profit = 300 (4) + 500 (3) = \$2,700	
	(D, W) = (4, 0)	Profit = 300 (4) + 500 (0) = \$1,200	

The *enumeration-of-corner-points method* also can be used to solve linear programming problems with more than two decision variables. Furthermore, it is possible to greatly streamline this method to solve such problems very efficiently. We introduce the streamlined method in Key Fact 3.

> **Key Fact 3**: The **simplex method** is an extremely efficient solution procedure for solving linear programming problems with many thousands (or even millions) of decision variables. It is a greatly streamlined version of the *enumeration-of-corner-points method*, since it also only evaluates corner points (in a very efficient way). However, the simplex method has a way of quickly getting to the best corner point *and* detecting that this point is optimal, so it then stops without needing to evaluate the rest of the corner points.

For the Wyndor problem, the simplex method evaluates the first three corner points in Table 14.2 (in the same order), detects that (D, W) = (2, 6) is optimal, and so stops. The next two sections will describe the concepts that make the simplex method so efficient.

To apply the simplex method to problems with more than two decision variables (so the feasible region cannot be graphed), we need a procedure for identifying corner points.

> **Key Fact 4**: Let n denote the number of decision variables for a problem. (For example, $n = 2$ for the Wyndor problem.) From a geometric viewpoint, a **corner point** is a feasible solution that lies at the intersection of n constraint boundaries. From an algebraic viewpoint, a corner point is a feasible solution that satisfies n constraint boundary equations simultaneously.

Thus, using the algebraic viewpoint, one procedure for obtaining a corner point is to solve a system of n equations (constraint boundary equations) in n unknowns (the decision variables). If this simultaneous solution also satisfies the *other* constraints (those *not* providing the constraint boundary equations), then it is a corner point. The simplex method uses a greatly streamlined variant of this procedure to move from the last corner point obtained to the next one.

Thus far, we have restricted our attention to linear programming problems that have exactly one optimal solution. What are all the possibilities for the number of optimal solutions?

> **Key Fact 5**: The only possibilities for a linear programming problem are that it has
> (1) exactly *one* optimal solution,
> or (2) an *infinite* number of optimal solutions,
> or (3) *no* optimal solution.

The second case is referred to as having *multiple optimal solutions*. We first will describe how this case can arise, and why it always gives an *infinite* number of optimal solutions rather than just a few. Later, we will discuss the two ways in which the third case can occur.

Multiple Optimal Solutions

For the Wyndor problem, suppose that the unit profit for doors will be \$200 instead of \$500. This changes the objective function (expressed in units of dollars) to

$$\text{Profit} = 300\,D + 200\,W.$$

Note that the coefficients of *both* D and W now are exactly 100 times as large as for the problem's third functional constraint,

$$3\,D + 2\,W \leq 18,$$

which has the constraint boundary line given by the equation,

$$3\,D + 2\,W = 18.$$

Consequently, when the graphical method is applied to this new version of the problem, as shown in Figure 14.3, the optimal objective function line, Profit = \$1800 = 300 D + 200 W, now *coincides* with this constraint boundary line. Therefore, *two* corner points, (2, 6) and (4, 3), now are optimal since they both produce the largest Profit.

$$(D, W) = (2, 6):\ \ \text{Profit} = 300\,(2) + 200\,(6) = \$1800.$$

$$(D, W) = (4, 3):\ \ \text{Profit} = 300\,(4) + 200\,(3) = \$1800.$$

Furthermore, *every* point on the line segment between (2, 6) and (4, 3) also is an optimal solution with an objective function value of Profit = \$1800.

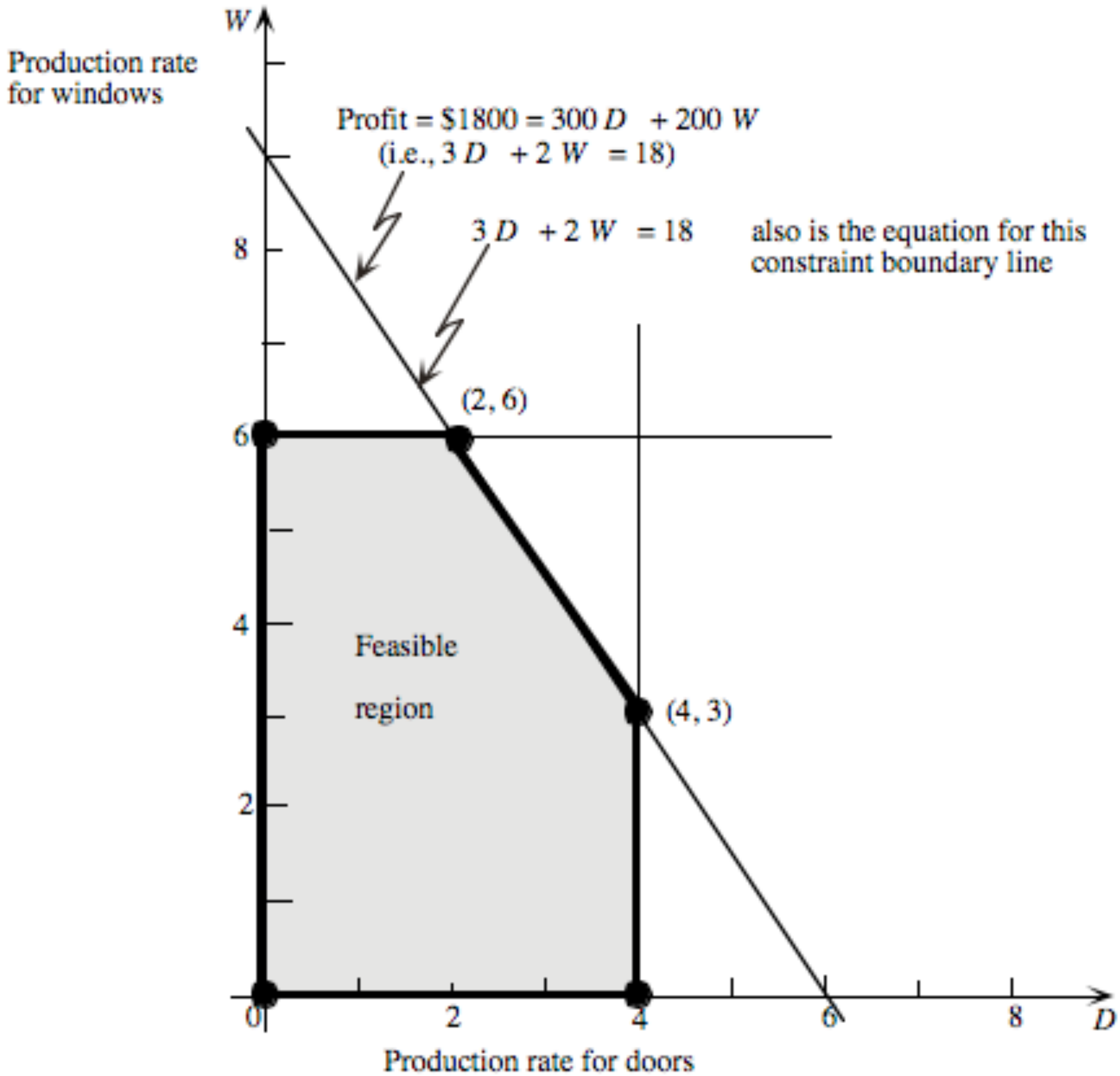


Figure 14.3 Graph resulting from applying the graphical method to the Wyndor problem after the objective function is changed to Profit = 300 D + 200 W. Since the optimal objective function line (Profit = \$1800 = 300 D + 200 W) passes through the entire line segment between (2, 6) and (4, 3), the infinite number of points on this line segment are all optimal.

Like any other line segment, there are an *infinite* number of points on the line segment between (2, 6) and (4, 3). This is what provides an infinite number of optimal solutions for this problem. However, this mathematically curious fact is not the important conclusion here. All that really matters is that there are *multiple* optimal solutions. It is valuable for management to have several options of product mixes that will maximize profitability—say (D, W) = (2, 6), or (3, 4.5), or (4, 3) — because factors not incorporated into the mathematical model may make one of the options more attractive than the others. For example, these factors might include management's desires to (1) highlight a particularly prestigious new product, (2) use a certain new product to initiate a family of similar products, (3) meet the needs of the company's most important customers, and (4) counter new products being introduced by the company's competitors.

This example illustrates how problems with multiple optimal solutions always have two optimal corner points—the two end points of a line segment on the boundary of the feasible region where every point on that line segment is an optimal solution. When a problem has more than two decision variables, it even is possible to have *more* than two corner points that are optimal solutions. (For example, with three decision variables, it is possible for all three vertices of a triangle on the boundary of the feasible region to be optimal corner points, in which case every point in the triangle is an optimal solution.)

Because a linear programming problem can have multiple optimal solutions, we generally will speak of *an* optimal solution rather than *the* optimal solution if we don't know yet how many there are.

> **Key Fact 6:** If a linear programming problem has multiple optimal solutions, at least two of these optimal solutions *must* be *corner points*.

Since the simplex method only evaluates corner points until it finds the best one (the one with the best value of the objective function), Key Fact 6 ensures that it still will find one of the optimal solutions. Furthermore, the next key fact points out that it provides additional information as well.

> **Key Fact 7**: If a linear programming problem has multiple optimal solutions, the simplex method automatically will find one of the optimal corner points *and* signal that there are one or more others. If desired, the simplex method also can find the other optimal corner points quickly after finding the first one.

To illustrate how the simplex method signals that there are other optimal corner points, Figure 14.4 shows the spreadsheet model for the same version of the Wyndor problem as considered in Figure 14.3. Thus, both (2, 6) and (4, 3) are optimal corner points, but the simplex method as executed by the Excel Solver happens to show the latter one in the changing cells UnitsProduced (C12:D12). The signal that this is not the only optimal solution appears in the sensitivity report shown in Figure 14.5. Specifically, look at the last two columns in the top part of the report. The signal is that 0 appears in either of these columns. The fact that the objective coefficient (unit profit) for doors has an allowable decrease of 0 indicates that the optimal solution now shown in the third column would no longer be optimal if this unit profit is decreased by even a tiny amount. The same conclusion holds if the unit profit for windows is *increased* by even a tiny amount, since this objective coefficient has an allowable increase of 0. Making either or both of these tiny changes in the unit profits and re-solving would yield $(D, W) = (2, 6)$, which is the only optimal solution after making these changes but is tied with (4, 3) as an optimal solution before making the changes.

	A	B	C	D	E	F	G
1	**Wyndor Glass Co. Product-Mix Problem**						
2							
3			Doors	Windows			
4		Unit Profit	$300	$200			
5					Hours		Hours
6			Hours Used Per Unit Produced		Used		Available
7		Plant 1	1	0	4	≤	4
8		Plant 2	0	2	6	≤	12
9		Plant 3	3	2	18	≤	18
10							
11			Doors	Windows			Total Profit
12		Units Produced	4	3			$1,800

Solver Parameters

Set Objective Cell: TotalProfit
To: Max
By Changing Variable Cells:
UnitsProduced
Subject to the Constraints:
Hours Used <= HoursAvailable
Solver Options:
Make Variables Nonnegative
Solving Method: Simplex LP

	E
5	Hours
6	Used
7	=SUMPRODUCT(C7:D7,UnitsProduced)
8	=SUMPRODUCT(C8:D8,UnitsProduced)
9	=SUMPRODUCT(C9:D9,UnitsProduced)

	G
11	Total Profit
12	=SUMPRODUCT(UnitProfit,UnitsProduced)

Range Name	Cells
HoursAvailable	G7:G9
HoursUsed	E7:E9
HoursUsedPerUnitProduced	C7:D9
TotalProfit	G12
UnitProfit	C4:D4
UnitsProduced	C12:D12

Figure 14.4 The spreadsheet model for the Wyndor problem after the unit profit for windows is decreased from $500 to $200, which leads to having an infinite number of optimal solutions as depicted in Figure 14.3.

Adjustable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
C12	Units Produced Doors	4	0	300	1E+30	0
D12	Units Produced Windows	3	0	200	0	200

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
E7	Plant 1 Used	4	0	4	2	2
E8	Plant 2 Used	6	0	12	1E+30	6
E9	Plant 3 Used	18	100	18	6	6

Figure 14.5 The sensitivity report for the spreadsheet model shown in Figure 14.4.

When more than two changing cells have an objective coefficient with an allowable decrease or allowable increase of 0, trying the various combinations of a tiny decrease or increase and re-solving will yield the various optimal corner points. The *other* optimal solutions include every point on *each* line segment connecting a pair of optimal corner points. Since a sampling of a few optimal solutions generally is fully adequate for management, we will not burden you with the details of how to identify even more optimal solutions by interpolating simultaneously between three or more optimal corner points.

The example portrayed in Figure 14.3 illustrates how, by changing the objective function in a certain way, one of the five line segments forming the boundary of this feasible region can become optimal. Table 14.3 summarizes how each of the other four line segments can become optimal instead by changing the objective function in another way. (Check each of these four cases in Figure 14.3 to see why the optimal objective function line passes through the indicated line segment.)

Table 14.3 Multiple Optimal Solutions with Different Unit Profits for the Wyndor Problem

Unit Profit: Doors	Unit Profit: Windows	Objective Function	Multiple Optimal Solutions
$300	$200	Profit = 300 *D* + 200 *W*	Line segment between (2, 6) and (4, 3)
$300	0	Profit = 300 *D*	Line segment between (4, 3) and (4, 0)
0	$500	Profit = 500 *W*	Line segment between (0, 6) and (2, 6)
0	-$100	Profit = -100 *W*	Line segment between (0, 0) and (4, 0)
-$100	0	Profit = -100 *D*	Line segment between (0, 0) and (0, 6)

Having seen how a linear programming problem can have *more* than one optimal solution, now let us look in turn at the two ways in which a problem can have *no* optimal solution at all.

No Feasible Solutions

Recall that a *feasible* solution is one that satisfies *all* the constraints —including both functional and nonnegativity constraints —*simultaneously*. Thus far, we have taken for granted that there will be some feasible solutions. But this is not necessarily the case.

> **Key Fact 8**: The constraints of a linear programming problem can be so restrictive that it is impossible for a solution to satisfy all the constraints simultaneously. Thus, there are *no feasible solutions* and so *no optimal solution*. A situation like this is readily detected by the simplex method.

To illustrate how this could happen, let us introduce another consideration into the Wyndor problem. Suppose a substantial investment is required to initiate the production and marketing of the two new products. To justify this investment, management feels that the combined production rate of the two products should be at least 10 units per week. This consideration imposes the additional constraint,

$$D + W \geq 10.$$

The resulting graph of all the constraints is shown in Figure 14.6, where each arrow indicates which side of the corresponding constraint boundary line is permitted by that constraint. Note how all the solutions permitted by the new constraint lie beyond the original feasible region. What that means is there are no feasible solutions. And the fact that there are no feasible solutions means that the two proposed new products should not be undertaken at all.

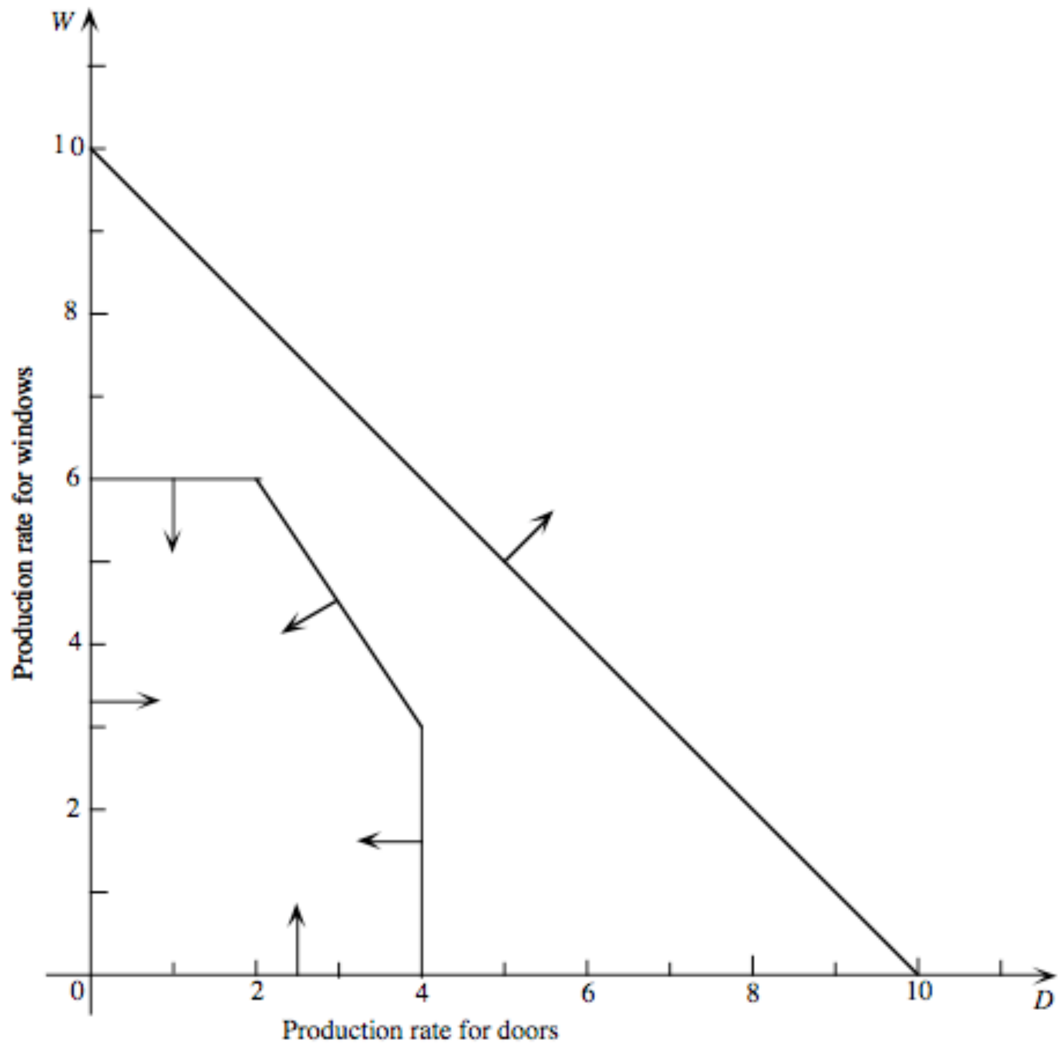


Figure 14.6	Graph showing that there are no feasible solutions for the Wyndor problem after adding the constraint $D + W \geq 10$, because this constraint goes not permit any of the solutions in the original feasible region

Figure 14.7 shows the spreadsheet model for this situation. Using the previous optimal solution of (2, 6) in the changing cells yields a value of 8 in cell E13, which does not satisfy the constraint that this total of the production rates must be at least 10 (as indicated by cells E14 and E15). Clicking on the Solve button then brings up the message that "Solver could not find a feasible solution," as shown in Figure 14.8.

	A	B	C	D	E	F	G
1	**Wyndor Glass Co. Product-Mix Problem**						
2							
3			Doors	Windows			
4		Unit Profit	$300	$500			
5					Hours		Hours
6			Hours Used Per Unit Produced		Used		Available
7		Plant 1	1	0	2	≤	4
8		Plant 2	0	2	12	≤	12
9		Plant 3	3	2	18	≤	18
10							
11					Total		
12			Doors	Windows	Produced		Total Profit
13		Units Produced	2	6	8		$3,600
14					≥		
15				Minimum Production	10		

Solver Parameters
Set Objective Cell: TotalProfit
To: Max
By Changing Variable Cells:
UnitsProduced
Subject to the Constraints:
Hours Used <= HoursAvailable
TotalProduced >= MinimumProduction
Solver Options:
Make Variables Nonnegative
Solving Method: Simplex LP

Range Name	Cells
HoursAvailable	G7:G9
HoursUsed	E7:E9
HoursUsedPerUnitProduced	C7:D9
TotalProfit	G12
UnitProfit	C4:D4
UnitsProduced	C12:D12

	E
5	Hours
6	Used
7	=SUMPRODUCT(C7:D7,UnitsProduced)
8	=SUMPRODUCT(C8:D8,UnitsProduced)
9	=SUMPRODUCT(C9:D9,UnitsProduced)

	G
12	Total Profit
13	=SUMPRODUCT(UnitProfit,UnitsProduced)

Figure 14.7 The spreadsheet model for the Wyndor problem after adding the constraint, $D + W \geq 10$, which leads to having no feasible solutions as depicted in Figure 14.6.

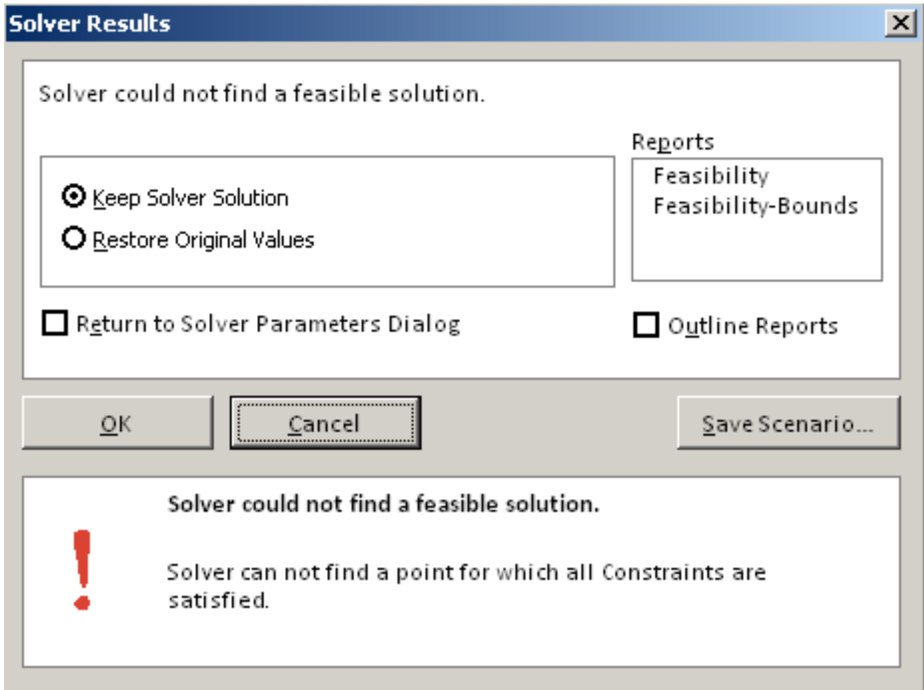


Figure 14.8 The message given by Solver for the model in Figure 14.7 that has no feasible solutions.

Before drawing the conclusion that the two proposed new products should not be undertaken at all, management probably will want to review the situation further. Are any of the constraints more restrictive than really needed? For example, if the minimum combined production rate permitted for the two products is reduced to 8 units per week, the resulting constraint is $D + W \geq 8$, and the solution, $(D, W) = (2, 6)$, becomes both feasible and optimal. Another possibility is to increase the production capacity made available to the two new products in some of the plants by cutting back on the production of a current product.

In other applications, the explanation for having no feasible solutions may be that an error was made in formulating the model or in inputting data into the computer. Perhaps one of the constraints was inadvertently made more restrictive than intended.

For some applications of linear programming in the *public sector*, certain constraints entered into the model may reflect restrictions requested by special-interest groups. These restrictions tend to be far tighter than would be considered reasonable by other affected groups, so the model with these constraints may well have no feasible solutions. If so, the message that "Solver could not find a feasible solution" then provides strong ammunition for negotiating more reasonable restrictions that will yield some feasible solutions.

No Bound on the Best Objective Function Value

So having no feasible solutions is one way a problem can have no optimal solution. We now will look at a second way.

Whereas having no feasible solutions occurs because the constraints are too restrictive, this next case arises when the constraints are too *unrestrictive*. A typical reason a set of constraints is too unrestrictive is that one or more of them that should have been in the model were inadvertently omitted. As you might guess, such an omission can cause strange things to happen in an incomplete model. One bizarre possibility is that there might be no limit on how much the value of the objective function can be improved without violating any of the constraints that were included in the model.

> **Key Fact 9**: If some necessary constraints were not included in the linear programming model, it is possible to have *no limit* on the best objective function value for solutions in the (supposed) feasible region. If this occurs, then no feasible solution can be optimal because there always is a better feasible solution. This is another situation readily detected by the simplex method.

To illustrate this case, suppose that the production capacity constraints for Plants 2 and 3 were inadvertently omitted from the linear programming model for the Wyndor problem. This would leave the nonnegativity constraints and the production capacity constraint for Plant 1, $D \le 4$, as the only constraints included in the model. Thus, *any* nonnegative value of W, no matter how large, is permitted. The feasible region for this (incomplete) model is the shaded region (and its extension upward off the page) shown in Figure 14.9, which also includes a series of objective function lines that were drawn in a vain attempt to reach an optimal solution. Even if D = 4 and some huge value is chosen for W (say, W = 1,000), the value of Profit = 300 D + 500 W can be increased even further just by making W even larger (say, W = 2,000). Therefore, even though there are many feasible solutions that yield a huge value of the objective function, none of them are *the* best feasible solution, so none can be classified as an optimal solution.

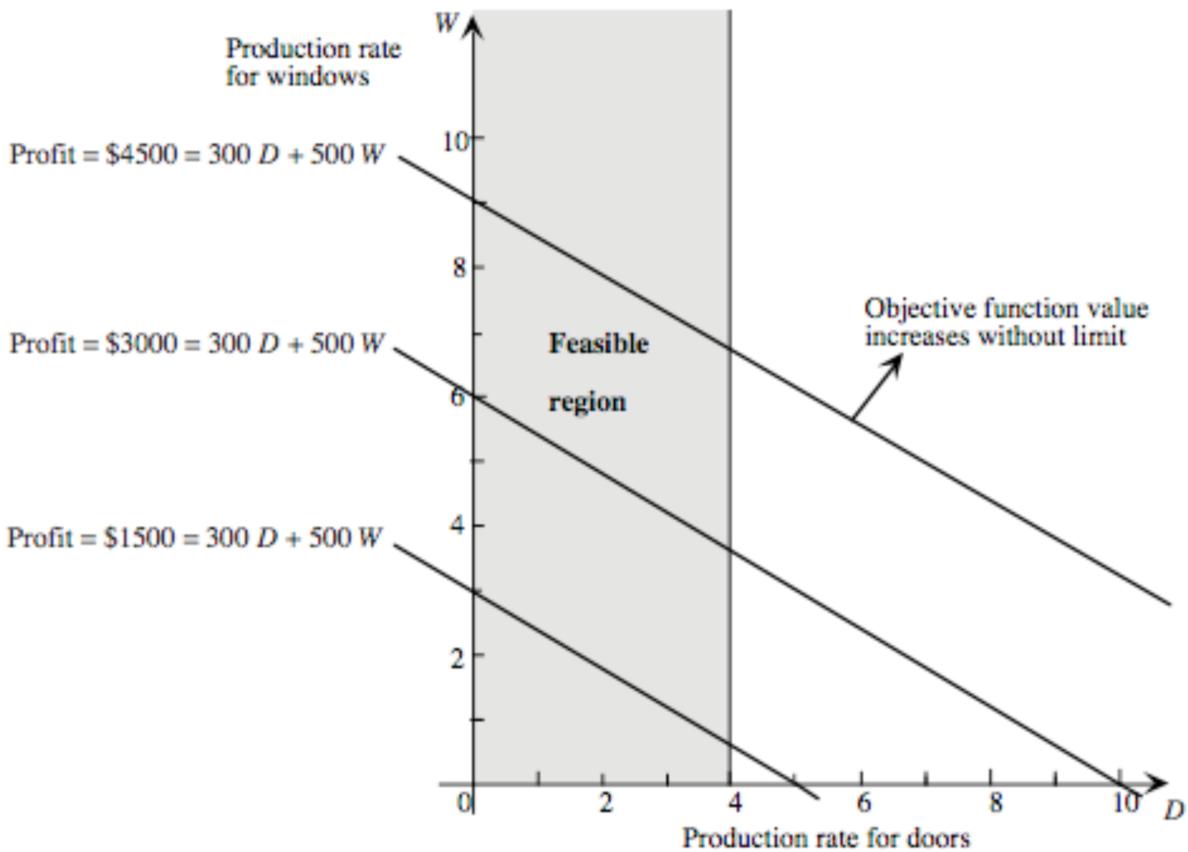


Figure 14.9	Graph showing that there would be no bound on the best objective function value in the model for the Wyndor problem if the only constraints were $D \ge 0$, $W \ge 0$, and $D \le 4$, because nothing prevents increasing W and Profit indefinitely.

Figure 14.10 shows the corresponding spreadsheet model. When the previous optimal solution of (2, 6) is entered into the changing cells UnitsProduced (C10:D10), everything appears to be normal. However,

clicking on the Solve button then brings up the message shown in Figure 14.11, namely, "The Objective Cell values do not converge." In this case, this message is saying that the process of setting the values of the changing cells at an optimal solution does not converge because the total profit appearing in the objective cell TotalProfit (G10) continues to increase indefinitely as the production rate for windows (cell D7) is increased.

	A	B	C	D	E	F	G
1	**Wyndor Glass Co. Product-Mix Problem**						
2							
3			Doors	Windows			
4		Unit Profit	$300	$500			
5					Hours		Hours
6			Hours Used Per Unit Produced		Used		Available
7		Plant 1	1	0	2	≤	4
8							
9			Doors	Windows			Total Profit
10		Units Produced	2	6			$3,600

Solver Parameters
Set Objective Cell: TotalProfit
To: Max
By Changing Variable Cells:
UnitsProduced
Subject to the Constraints:
Hours Used <= HoursAvailable
Solver Options:
Make Variables Nonnegative
Solving Method: Simplex LP

Range Name	Cells
HoursAvailable	G7
HoursUsed	E7
HoursUsedPerUnitProduced	C7:D7
TotalProfit	G10
UnitProfit	C4:D4
UnitsProduced	C10:D10

	E
5	Hours
6	Used
7	=SUMPRODUCT(HoursUsedPerUnitProduced,UnitsProduced)

	G
9	Total Profit
10	=SUMPRODUCT(UnitProfit,UnitsProduced)

Figure 14.10 The spreadsheet model for the Wyndor problem if the only constraints were nonnegativity constraints and $D \leq 4$, which leads to having no bound on the best objective function value as depicted in Figure 14.9.

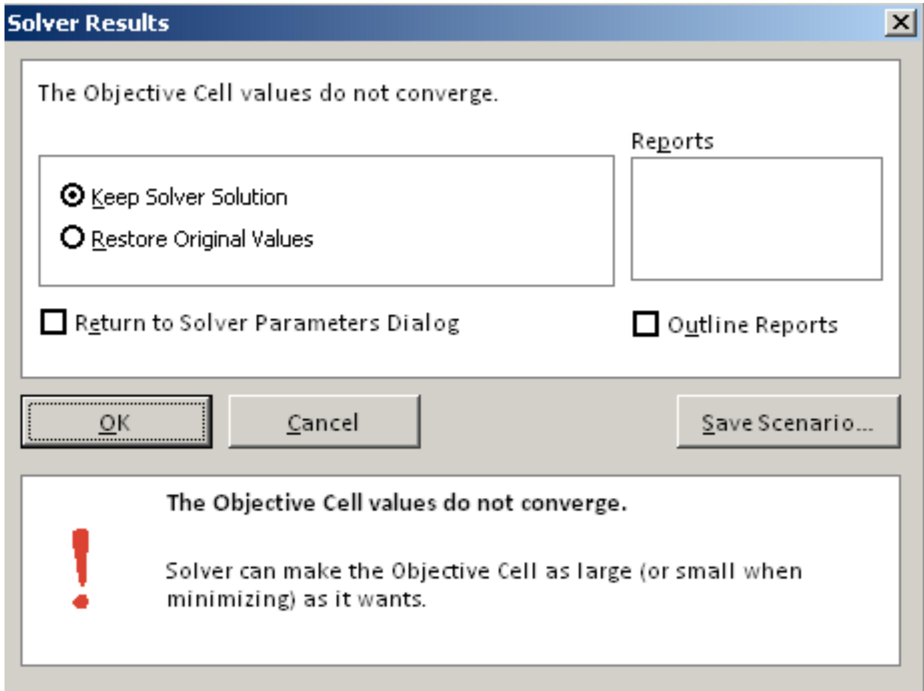


Figure 14.11 The message given by Solver for the model in Figure 14.10 that has no bound on the best objective function value.

Since not even linear programming has discovered a way to generate astronomically large profits, the real message when there is no bound on the best objective function value is that a mistake has been made in formulating the model. Typically, this outcome only occurs during the stage of the study when the model is still being tested and refined. When it occurs, the functional constraints should be checked to ascertain where they do not accurately represent the restrictions in the real problem.

Besides omitting necessary constraints, another possibility is that constraints included in the model have been formulated improperly. For example, perhaps a ≥ sign has been used in a functional constraint when it should have been a ≤ sign.

The feasible region in Figure 14.9 is said to be *unbounded*, because there is no bound on the distance one can move in a certain direction (straight up in this case). *Any* linear programming model that has no bound on the best objective function value *must* have an unbounded feasible region. Furthermore, the feasible region must be unbounded in a direction that improves the objective function value. However, it is possible to have a model whose feasible region is unbounded only in directions that do *not* improve the objective function value, so the problem still has an optimal solution. (The feasible region shown at the end of the Supplement to Chapter 2 for the Profit & Gambit problem is an example.)

A Managerial Perspective

After focusing so much on optimal solutions, it would be easy to conclude that the whole point of a linear programming study is to find and implement an optimal solution. In reality, much more is involved.

> **Key Fact 10**: An optimal solution is only optimal with respect to a particular mathematical model that provides only a rough representation of the real problem. A manager is interested in far more than just finding such a solution. The purpose of a linear programming study is to help guide management's final decision by providing insights into the likely consequences of pursuing various managerial options under different assumptions about future conditions. Changing these assumptions requires changing the initial version of the linear programming model. Most of the important insights are gained while conducting the analysis done *after* finding an optimal solution for the initial version of the model. Management scientists often refer to this analysis as **postoptimality analysis**. Another term (the one being used in this book) is **what-if analysis**, because questions are being addressed about *what* would happen to the conclusions from the model *if* future conditions turn out to be such and such instead.

Chapter 5 focused on this key type of analysis.

REVIEW QUESTIONS

1. What is a corner point?
2. What can be said about the corner point with the best value of the objective function?
3. How does the simplex method differ from the enumeration-of-corner-points method?
4. Can a linear programming problem have exactly two optimal solutions?
5. Can a linear programming problem have exactly two corner points that are optimal solutions?
6. What can happen to cause a linear programming problem to have no feasible solutions?
7. What can happen to cause a linear programming problem to have no bound on the best objective function value?

14.2 THE ROLE OF CORNER POINTS IN SEARCHING FOR AN OPTIMAL SOLUTION

In the preceding section, we presented 10 *key facts* about optimal solutions. We now turn to the role that some of these facts play in determining how to search for an optimal solution.

Finding the Best Corner Point Solves the Problem

Notice that *half* of these key facts about optimal solutions (Key Facts 2, 3, 4, 6 and 7) talk about *corner points*. This is an indication of the close relationship between corner points and optimal solutions. In fact, Key Fact 2 says that the optimal solution (when there is just one) for any linear programming problem *must* be a corner point. Even when the problem has multiple optimal solutions, Key Fact 6 indicates that some of these optimal solutions also *must* be corner points.

Key Facts 2 and 6 are important because they greatly simplify the search for an optimal solution. Together, they imply that this search only requires finding the *best* corner point (the corner point with the best value of the objective function):

> **Optimality of the Best Corner Point**: For *any* linear programming problem with an optimal solution[1], the best corner point *must* be an optimal solution. (When two or more corner points tie for being the best one, *all* these best corner points *must* be optimal solutions.)

The big advantage of searching for an optimal solution simply by finding the *best corner point* is that it tremendously reduces the number of solutions that need to be considered. Linear programming problems normally have a vast number (literally an *infinite* number) of feasible solutions. By contrast, the number of corner points is relatively small.

[1] It is possible (but highly unusual) for a linear programming problem *not* to have an optimal solution, as described in Key Facts 8 and 9.

To illustrate how a linear programming problem can be solved by finding its best corner point, consider again the Wyndor Glass Co. case study introduced in Section 2.1. Both the linear programming model (in algebraic form) and the feasible region for the Wyndor problem are again shown in Figure 14.12, where we have highlighted the five corner points. Each corner point lies at a corner of the feasible region where two constraint boundary lines intersect.

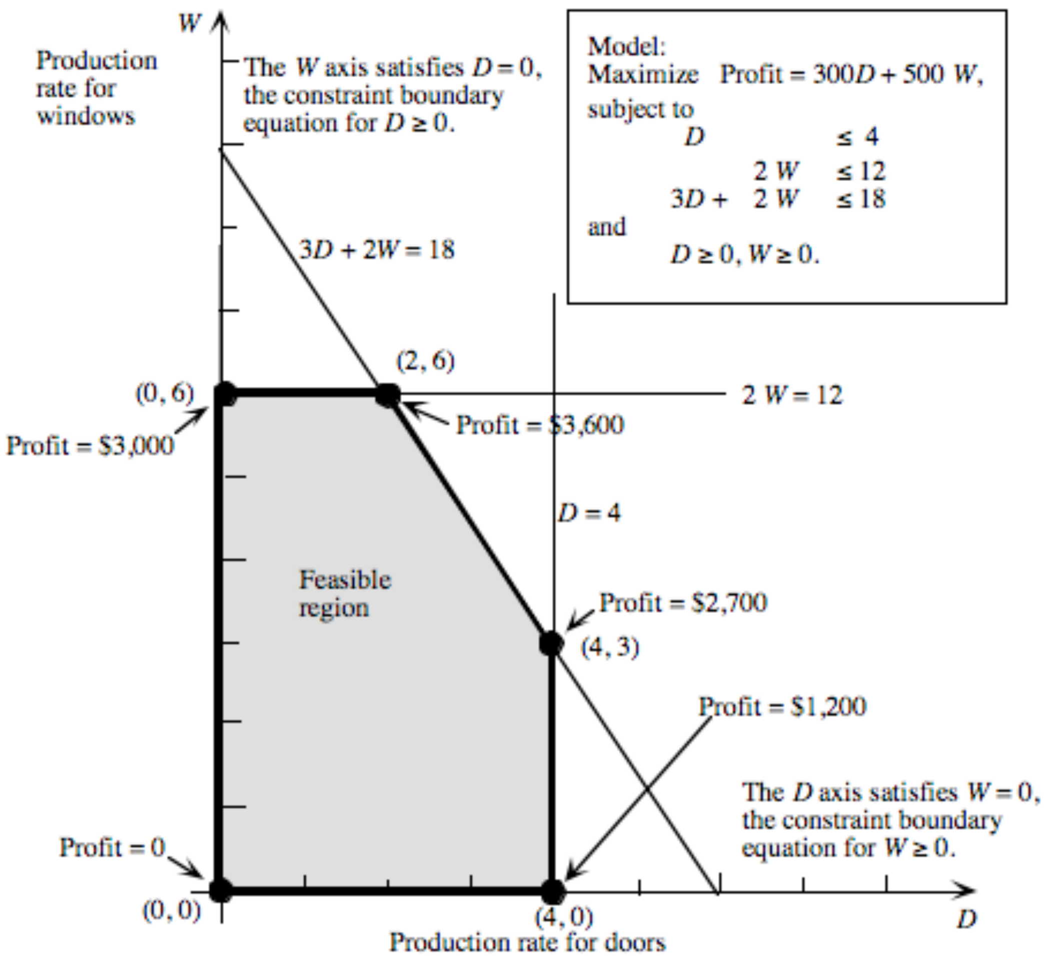


Figure 14.12 The five corner points are the key feasible solutions for the Wyndor problem.

In the preceding section, we showed in Table 14.2 how the *enumeration-of-corner-points method* solves this problem by calculating the value of the objective function, Profit = 300 D + 500 W, for all five corner points. This work is summarized in Figure 14.12, which shows the value of the objective function for each corner point. Since (2, 6) has Profit = \$3,600, which is larger than for the other corner points, (2, 6) is the *best corner point*. Therefore, (D, W) = (2, 6) is the *optimal solution* for this example. (Recall that this solution was found to be optimal by both the graphical method and the Excel Solver in Chapter 2.)

Like the enumeration-of-corner-points method, the simplex method only looks at corner points. However, the big difference is that the simplex method usually is able to identify the best corner point after examining only a portion (often only a tiny portion) of all the corner points, as described next.

How the Simplex Method Solves the Wyndor Problem

The simplex method is an algebraic procedure. However, when the problem has only two decision variables, we can describe what the algebra of the simplex method accomplishes in graphical terms. In particular, referring to Figure 14.12, here is what the simplex method does to solve the Wyndor problem.

1. It begins by examining the corner point at the origin, (0, 0), and concludes that this is not an optimal solution.
2. It then moves to the corner point, (0, 6), and concludes that this also is not an optimal solution.
3. It then moves to the corner point, (2, 6), and concludes that this is the optimal solution, so it stops.

This sequence of corner points examined is shown in Figure 14.13.

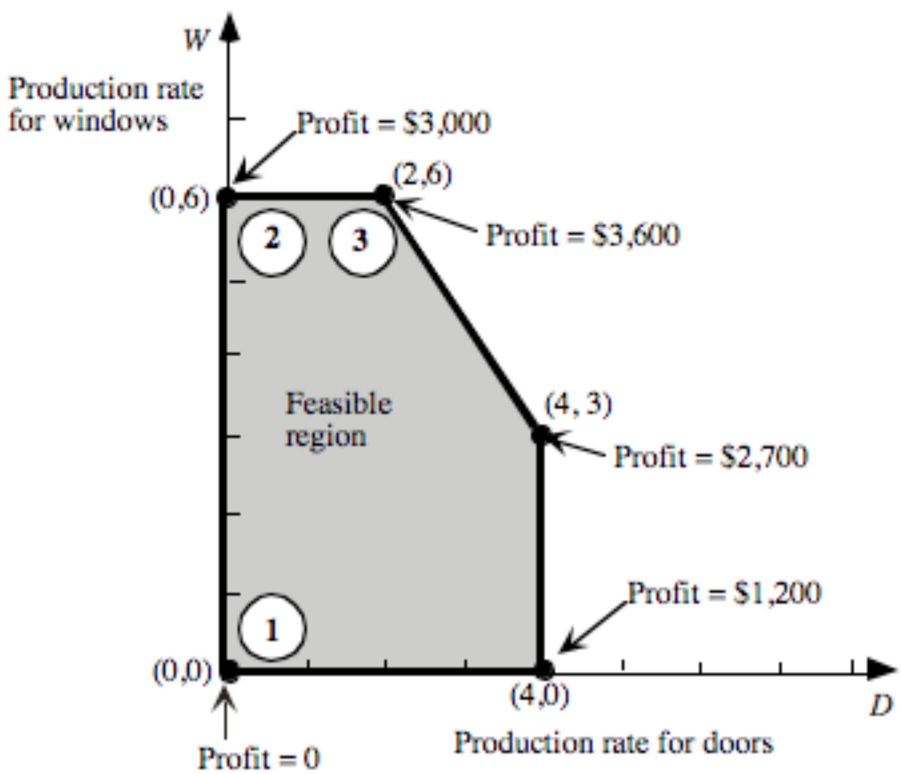


Figure 14.13 This graph shows the sequence of corner points (①, ②, ③) examined by the simplex method for the Wyndor problem. The optimal solution, (2, 6), is found after examining just three points.

Now let us look at the rationale behind these three steps and the conclusions drawn.

Why start with the corner point at the origin, (0, 0)? Simply because this is a convenient place to begin. (When solving for corner points algebraically, this one can be identified without doing any algebra.) The only situation where (0, 0) cannot be chosen as the initial corner point is when it lies outside the feasible region and so is not a corner point. In this case, any real corner point can be chosen instead.

At step 1, how is the simplex method able to conclude that (0, 0) is not an optimal solution? It does this by checking the *adjacent* corner points, (4, 0) and (0, 6). Both of these points are *better* because they produce better values of the objective function than (0, 0). So (0, 0) cannot be optimal.

The only situation where (0, 0) would be optimal is when *neither* adjacent corner point is better. (Check how this would happen if the objective function in the model were changed to Profit = - D - W.)

In step 2, why does the simplex method move to (0, 6)? As discussed more fully later in this section, the simplex method only moves to *adjacent* corner points. Therefore, from (0, 0), the only alternatives are to move next to either (4, 0) or (0, 6). (4, 0) gives Profit = $1,200 and (0, 6) gives Profit = $3,000. Both are an improvement over (0, 0) with Profit = 0, so either alternative would move us *toward* an optimal solution. Since we want to move toward an optimal solution as quickly as possible, the simplex method compares the *rate of improvement* in the value of the objective function (Profit) when moving from (0, 0) toward (4, 0) or (0, 6). Since the unit profits for doors and windows are $300 and $500, respectively, increasing the production rate of windows (W) provides a better rate of improvement ($500 per unit increase in W) than increasing the production rate of doors. Therefore, (0, 6) is chosen as the next corner point to consider.

To finish step 2, (0, 6) is not optimal because one of its adjacent corner points, (2, 6), is better than (0, 6). Profit = $3,600 for (2, 6) is larger than Profit = $3,000 for (0, 6).

To begin step 3, the simplex method notes that (0, 6) has the two adjacent corner points, (0, 0) and (2, 6). (0, 0) was just examined and discarded before moving to (0, 6), so (2, 6) becomes the automatic choice to move to next.

How does the simplex method then conclude that (2, 6) is the optimal solution? The reason is that both adjacent corner points, (0, 6) and (4, 3), are *not* better than (2, 6). We already know from the previous work that (0, 6) is not as good. Furthermore, (4, 3) gives Profit = $2,700, which is less than Profit = $3,600 for (2, 6). Since (4, 3) is worse than (2, 6), continuing to move clockwise beyond (4, 3) cannot possibly lead to a corner point better than (2, 6).

A Summary of the Simplex Method

The procedure for the simplex method we just illustrated is typical. Here is a summary for problems with either two or more decision variables.

Getting Started: Select some corner point as the *initial corner point* to be examined. If the *origin* is a corner point of the feasible region, this is a convenient choice.

Checking for Optimality: Check each of the corner points *adjacent* to the current corner point. If *none* of the adjacent corner points are *better* (as measured by the value of the objective function) than the current corner point, then stop because the current corner point is an *optimal solution*. However, if *one or more* of the adjacent corner points are *better* than the current corner point, then continue as described in the next step.

Moving On: One of the adjacent corner points that is better than the current corner point needs to be selected as the next current point to be examined. When more than one is better, the conventional selection method is to choose the one that provides the **best rate of improvement**[2] in the value of the objective function while moving toward that adjacent corner point. After making the selection, return to the Checking for Optimality step above.

---

[2]The rate of improvement is the improvement in the value of the objective function per unit of distance moved from the current corner point to the adjacent corner point. As an algebraic procedure, the simplex method uses a nongeometric definition of distance, but we will not delve into this technicality.

Finding Corner Points Algebraically

For the Wyndor problem, it is easy to identify the corner points after graphing the feasible region, as shown in Figures 14.12 and 14.13. However, for problems with more than two (or possibly three) decision variables, it is not possible to find the corner points *graphically*. But we can have a computer do it *algebraically* (as indicated by Key Fact 4). This is what the simplex method does. The details are described in the box "How Are Corner Points Found Algebraically?"

How Are Corner Points Found Algebraically?

Consider the Wyndor problem and its graphical representation in Figure 14.12. Each constraint has a corresponding boundary equation, as shown below.

Table 14.4 The Constraint Boundary Equation for Each Constraint of the Wyndor Problem

Constraint	Constraint Boundary Equation
$D \geq 0$	$D = 0$
$W \geq 0$	$W = 0$
$D \leq 4$	$D = 4$
$2W \leq 12$	$2W = 12$
$3D + 2W \leq 18$	$3D + 2W = 18$

Each constraint boundary equation is the equation for a *constraint boundary line*, as shown in Figure 14.12.

Since each corner point lies at the intersection of two constraint boundary lines, the corner point must satisfy the corresponding two constraint boundary equations, as shown in the following table.

Table 14.5 The Pair of Constraint Boundary Equations Satisfied by Each Corner Point of the Wyndor Problem

Corner Point	Corresponding Constraint Boundary Equations	Corner Point Satisfies the Equations
(0, 0)	$D = 0$ $W = 0$	$0 = 0$ $0 = 0$
(0, 6)	$D = 0$ $2W = 12$	$0 = 0$ $2(6) = 12$
(2, 6)	$2W = 12$ $3D + 2W = 18$	$2(6) = 12$ $3(2) + 2(6) = 18$
(4, 3)	$3D + 2W = 18$ $D = 4$	$3(4) + 2(3) = 18$ $4 = 4$
(4, 0)	$D = 4$ $W = 0$	$4 = 4$ $0 = 0$

Thus, each corner point is found algebraically by simultaneously solving its corresponding pair of constraint boundary equations.

Not every pair of constraint boundary equations yields a simultaneous solution which is a corner point of the feasible region. For example, the pair,

$$D = 0$$

$$3D + 2W = 18,$$

yields $D = 0$, $W = 9$ as its simultaneous solution. Although this solution lies at the intersection of the *constraint boundary lines* corresponding to these equations, it is not a corner point of the feasible region because it does not satisfy the constraint, $2W \leq 12$. (Now see if you can find the other two pairs of constraint boundary equations whose simultaneous solutions lie outside the feasible region.)

Another possibility is that a pair of constraint boundary equations might yield no simultaneous solution at all, as illustrated by the pair,

$$D = 0$$

$$D = 4.$$

Since these two equations cannot be satisfied simultaneously (the corresponding two lines never intersect), this pair of equations cannot yield a corner point. (See if you can find the one other pair of constraint boundary equations that does not have a simultaneous solution.)

When a problem has *more than two decision variables*, the procedure for finding its corner points algebraically is basically the same as the one here (as described by Key Fact 4). However, instead of solving just a pair of constraint boundary equations simultaneously, the number of constraint boundary equations must equal the number of decision variables. If this number of constraint boundary equations yields a simultaneous solution that is a feasible solution, that solution is a corner point.

Adjacent Corner Points

One key to the efficiency of the simplex method is the *order* in which it examines corner points. As indicated earlier, each corner point in this order always is "adjacent" to the preceding one. We now will focus on the characteristics of adjacent corner points and how they impact efficiency.

To review what we mean by *adjacent* corner points, look at Figure 14.12. Suppose that you start at any of the corner points in this figure and then begin moving around the boundary of the feasible region (in either direction). The first corner point you reach is *adjacent* to the one from which you started. Thus, each corner point has two *adjacent corner points* —the first one reached in the clockwise direction and the first one reached in the counter-clockwise direction. Table 14.6 identifies these adjacent corner points.

Table 14.6 Adjacent Corner Points for Each Corner Point of the Wyndor Problem

Corner Point	Its Adjacent Corner Points
(0, 0)	(0, 6) and (4, 0)
(0, 6)	(2, 6) and (0, 0)
(2, 6)	(4, 3) and (0, 6)
(4, 3)	(4, 0) and (2, 6)
(4, 0)	(0, 0) and (4, 3)

A useful feature of adjacent corner points is that they share some of the same constraint boundaries. For example, the corner point (0, 0) lies on the constraint boundaries, $D = 0$ and $W = 0$. The adjacent corner point (0, 6) also lies on the constraint boundary $D = 0$ (along with $2W = 12$). The other adjacent corner point (4, 0) shares the constraint boundary $W = 0$ (its other constraint boundary is $D = 4$).

This Wyndor example has just two decision variables. However, regardless of the number of decision variables, adjacent corner points have the same characteristics, as summarized in the following definition.

> Two corner points are **adjacent corner points** if they share all but one of the same constraint boundaries. (The number of shared constraint boundaries is one less than the number of decision variables in the model.) The two adjacent corner points are connected by a line segment that lies on these same shared constraint boundaries. Such a line segment is referred to as an **edge of the feasible region**.

With two decision variables, this definition implies that two adjacent corner points always share *one* constraint boundary. Check this out in Figure 14.12. In this same figure, you should also be able to see that the *edge* of the feasible region connecting each pair of adjacent corner points lies on the shared constraint boundary.

Because adjacent corner points share all but one of the same *constraint boundaries*, they also share all but one of the same *constraint boundary equations*. This fact *greatly* streamlines the algebra when the simplex method solves the system of constraint boundary equations to identify the corner point being moved to next. Rather than having to solve the entire system of equations from scratch, only a few quick algebraic operations are needed to modify the solution of the previous system of equations after replacing just one of the equations. This great streamlining of the algebra adds greatly to the efficiency of the simplex method.

The Number of Corner Points

Focusing solely on corner points tremendously reduces the number of solutions that need to be considered in the search for an optimal solution. This concept is another of the keys to the efficiency of the simplex method.

However, this concept by itself does not nearly explain the remarkable efficiency of the simplex method. The *enumeration-of-corner-points method* introduced in the preceding section also only considers corner points. But it is so inefficient that it is only capable of solving rather small problems with even a fast computer.

The difficulty is that the number of corner points tends to grow very rapidly as the problem size is increased. For example, the Wyndor problem with its 2 decision variables and 3 functional constraints has only 5 corner points. By contrast, a similar problem with 20 decision variables and 30 functional constraints might have more than a *billion* corner points! Imagine the vast number of corner points with 100 decision variables and 100 functional constraints. Nevertheless, the simplex method routinely solves problems with even many *thousands* of decision variables and functional constraints!

How is this possible? How can the simplex method solve problems with astronomical numbers of corner points? The secret lies in the clever way it is able to reach and identify the optimal corner point for a large problem after examining only a tiny, tiny fraction of all the corner points. For example, on a problem with a *billion* corner points, it probably will examine less than a *hundred* of them before finding the optimal solution.

We next turn to outlining the solution concepts that make the simplex method so efficient.

REVIEW QUESTIONS

1. When is the best corner point (the one with the best value of the objective function) an optimal solution?
2. What is the advantage of only considering corner points when searching for an optimal solution?
3. When the simplex method is ready to move from the current corner point to the next one, which corner points are candidates to be this next one?
4. How does the simplex method determine if the current corner point is an optimal solution?
5. When is the simultaneous solution of a set of constraint boundary equations a corner point?
6. When are two corner points adjacent to each other?
7. The simplex method examines corner points in an order such that each one has what relationship to the preceding one? Why?
8. How many corner points might a problem with 20 decision variables and 30 functional constraints have?
9. Why is the simplex method able to solve problems with astronomical numbers of corner points when the enumeration-of-corner-points method cannot?

14.3 SOLUTION CONCEPTS FOR THE SIMPLEX METHOD

In the preceding section, you have seen an outline of how the simplex method solves linear programming problems. Now we will review this same material from a broader perspective, focusing on the six key ideas —the *solution concepts* —that make the simplex method so efficient.

The Key Role of Corner Points

The preceding section began with a description of the key role of corner points in searching for an optimal solution. Recall that the simplex method only examines corner points, because finding the best corner point solves the problem. Thus:

> **Solution Concept 1**: Focus solely on the *corner points.* For *any* linear programming problem with an optimal solution, the best corner point *must* be the best feasible solution of any kind, i.e., an optimal solution.

This eliminates the need to consider all those feasible solutions that are not corner points.

The Simplex Method Is an Iterative Algorithm

The next solution concept concerns the general nature of the simplex method.

You probably have heard the term *algorithm* used before. It is one of those technical terms that have been entering the common vocabulary in this computer age. So exactly what is an algorithm?

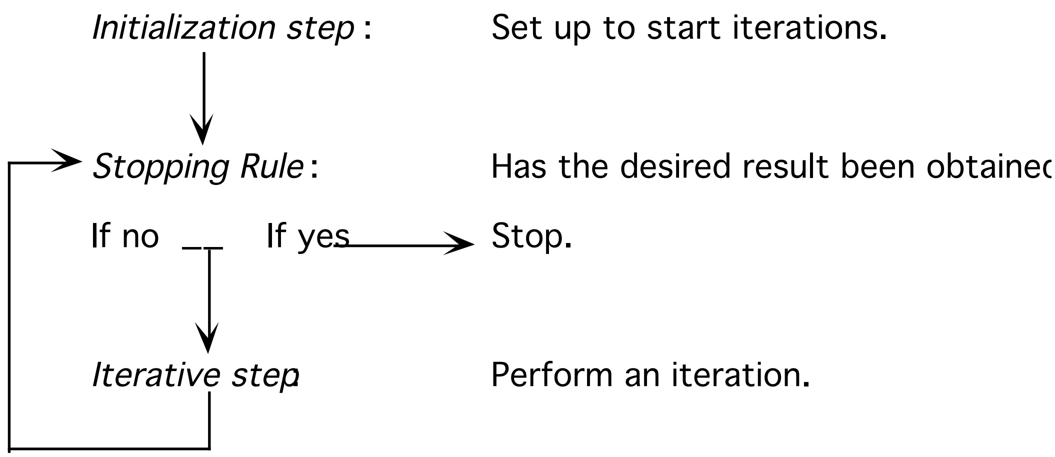
> An **algorithm** is a systematic procedure for solving a mathematical problem. The steps of an algorithm are so well specified that it can be (and frequently is) executed on a computer.

You can see from the definition that this term fits some solution procedures that you learned in school years ago. For example, the familiar procedure for *long division* is an algorithm because it is a *systematic* solution procedure and its steps are so *well specified* that it could be programmed for execution on a computer.

Remember that the *long division algorithm* involves performing some calculations to find *each* digit of the answer (quotient) in turn, moving from left to right. Each time a digit is found, a prescribed series of steps then is used to find the next digit. An important characteristic of the algorithm is that the *same* series of steps is repeated (iterated) over and over again to find the succession of digits. Each execution of the prescribed series of steps is called an **iteration**, so the algorithm is referred to as an **iterative algorithm**.

Iterative algorithms also include a procedure for getting started (the **initialization step**) and a criterion for determining when to stop (the **stopping rule**), as depicted in the following diagram:

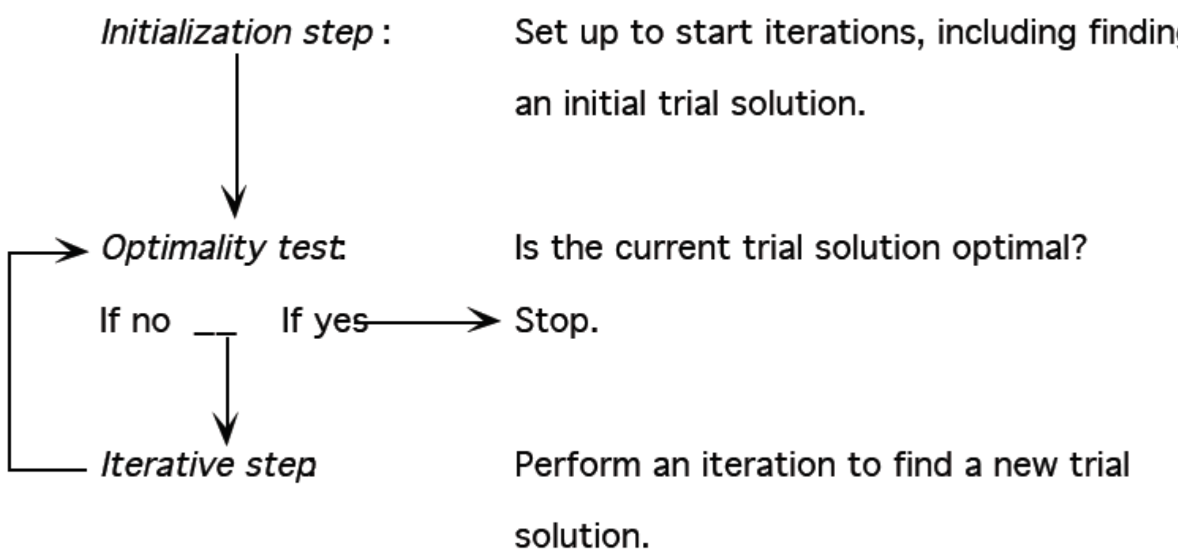
Structure of Iterative Algorithms



For the long division algorithm, the *initialization step* involves writing down the problem in the usual format, moving the decimal point (if any) of the divisor and dividend to the right as needed, and identifying the initial digit of the quotient. Iterations then find the additional digits needed. The last digit of the quotient is the one over the last digit of the dividend, so the stopping rule says to stop iterating when that digit is found. The remainder (if any) is then attached to the quotient, and the algorithm is finished.

Management science algorithms typically are iterative algorithms. For most of these algorithms, each iteration begins with a *trial solution* for the problem under consideration, and then the series of steps constituting an iteration lead to finding a better trial solution. The goal is to find a trial solution that is an *optimal* solution. Therefore, the "desired result" specified in the stopping rule is that the current trial solution is optimal. This means that the stopping rule actually is an **optimality test**, as shown below.

Structure of Most Management Science Algorithms



As summarized in our next solution concept, the simplex method fits right into this structure of most management science algorithms. In this case, the *trial solutions* being examined are *corner points* .

> **Solution Concept 2**: The simplex method is an *iterative algorithm* . The *initialization step* finds an initial corner point. Each *iteration* then consists of a prescribed series of steps for moving from the current corner point to a new corner point. The *optimality test* stops the algorithm when the new corner point is an optimal solution.

Thus, the three components of the simplex method are the initialization step, the optimality test, and the iterative step. Most of the time typically is spent repeating the iterative step. Each execution of the iterative step is one iteration of the algorithm.

The summary of the simplex method in the preceding section further describes these components, using the following less technical names:

Getting Started	=	Initialization
Moving On	=	Iteration
Checking for Optimality	=	Optimality test

The subsequent solution concepts clarify how the components of the simplex method achieve their objectives.

Find an Initial Corner Point Quickly

One reason the simplex method is so efficient is that it achieves the objectives of its components with a minimum of computational effort. This is illustrated by the *initialization step*, which often finds an initial corner point without doing any computations at all!

> **Solution Concept 3**: Whenever possible, the initialization step of the simplex method chooses the *origin* to be the initial corner point (trial solution). Thus, this corner point has all the decision variables equal to zero, so no work is required to solve algebraically for the corner point.

The one situation where it is not possible to choose the origin in this way is when the origin is not in the feasible region because it violates one or more of the functional constraints. We saw an example of this with the Profit & Gambit problem in Section 2.6 (see the end of the Supplement to Chapter 2 for graphs of the feasible region for this problem). In this situation, the simplex method uses another more complicated procedure (called "Phase 1") to find an initial corner point quickly. We will not dwell on this technicality.

Focus on Adjacent Corner Points to Perform an Iteration

The preceding section discusses the relationship between a corner point and its adjacent corner points. Our next solution concept describes why this relationship is such a useful one.

> **Solution Concept 4**: Given a corner point, it is much quicker computationally to gather information about its *adjacent* corner points than about other corner points. Therefore, each time the simplex method performs an iteration to move from the current corner point, it *always* chooses to move to an *adjacent* corner point. Other corner points are ignored. Consequently, the path being followed to an optimal solution is along the *edges* of the feasible region.

The simplex method never needs to take the time to “look” beyond the adjacent corner points. This partially explains the great efficiency of the simplex method.

Another reason the simplex method is so efficient is the clever way in which the *iterative step* chooses the next corner point. Among all the corner points that are adjacent to the current corner point, the iterative step chooses one that is on a “quick path” to an optimal solution, as described next.

Improve the Value of the Objective Function Rapidly

The simplex method seeks to follow a path that will reach an optimal solution as quickly as possible. On large problems, the chosen path normally will reach this optimal solution after passing through only a tiny, tiny fraction of all the corner points. (This fraction can be much larger on very small problems, where the total number of corner points is relatively small.) The reason that the simplex method reaches an optimal solution so quickly is because it focuses on *improving* the value of the objective function *rapidly*.

> **Solution Concept 5**: After identifying the current corner point, the simplex method uses algebra to next examine each of the edges of the feasible region that emanate from this corner point. Each of these edges leads to an *adjacent* corner point at the other end, but the simplex method does not even take the time to do the algebra needed to identify the adjacent corner point. Instead, it simply identifies the *rate of improvement* in the value of the objective function that would be obtained by moving along the edge. Among the edges with a *positive* rate of improvement in the value of the objective function, it then chooses the one with the *largest* rate of improvement to actually move along. The iteration is completed by identifying the adjacent corner point at the other end of this one edge, and then relabeling this adjacent corner point as the *current* corner point for the *next* iteration.

To conclude, we'll now see how the optimality test is performed efficiently by comparing the *current corner point* with its *adjacent corner points*.

Focus on Adjacent Corner Points to Perform the Optimality Test

The final key to the efficiency of the simplex method is its simple optimality test. To check whether the current corner point is optimal, it is only necessary to determine whether any of its *adjacent* corner points are *better* (a more favorable value of the objective function). If none are better, then the current corner point is optimal and the algorithm is finished.

The simplex method has a very quick way of checking whether any adjacent corner points are better, as follows.

> **Solution Concept 6**: Solution Concept 5 describes how the simplex method examines each of the edges of the feasible region that emanate from the current corner point. This examination of an edge leads to quickly identifying the *rate of improvement* in the value of the objective function that would be obtained by moving along the edge toward the adjacent corner point at the other end. A *positive* rate of improvement implies that the adjacent corner point is *better* than the current corner point, whereas a *negative* rate of improvement implies that the adjacent corner point is *worse*. Therefore, the optimality test consists simply of checking whether *any* of the edges give a *positive* rate of improvement in the value of the objective function. If *none* do, then the current corner point is optimal.

Our goal in presenting these six solution concepts has been to give you some intuitive insight into how the simplex method operates and what makes it so efficient. We feel that gaining this insight is important before learning the algebraic details of the simplex method. The next four sections lead into the connections between this insight and the algebraic details.

REVIEW QUESTIONS

1. The simplex method focuses solely on what kind of solutions?
2. What does an iteration of the simplex method consist of?
3. What is the structure of most management science algorithms, including the simplex method?
4. Whenever possible, the initialization step of the simplex method chooses which solution to be the initial corner point to be examined?
5. When the simplex method finishes examining a corner point, it is relatively quick computationally to then gather information about which other corner points? What information is gathered?
6. How is the optimality test for the simplex method performed?

14.4 THE SIMPLEX METHOD WITH TWO DECISION VARIABLES

The simplex method is an algebraic procedure that examines a sequence of corner points until the best one (the optimal solution) is found. However, when the problem has just two decision variables, the simplex method can be simplified to a geometric procedure that examines these corner points graphically. This section focuses on the latter procedure. We will continue using the example of the Wyndor Glass Co. case study introduced in Section 2.1.

How the Simplex Method Solves the Wyndor Problem

Section 14.2 discusses the key role that *corner points* play in searching for an optimal solution for the Wyndor problem (or any other linear programming problem). Figure 14.12 shows the feasible region for this problem and highlights the corner points.

Here is what the simplex method does with Figure 14.12 to solve the Wyndor problem.

1. It begins by examining the corner point at the origin, (0, 0), and concludes that this is not an optimal solution.

2. It then moves to the corner point, (0, 6), and concludes that this also is not an optimal solution.

3. It then moves to the corner point, (2, 6), and concludes that this is the optimal solution, so it stops.

This sequence of corner points examined is shown in Figure 14.14.

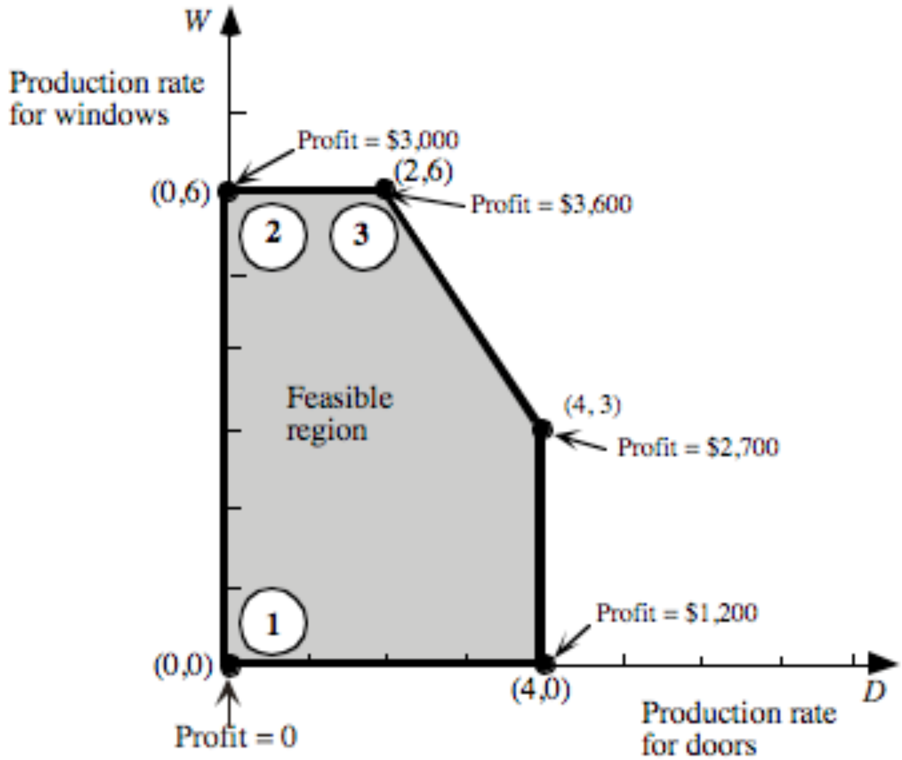


Figure 14.14 This graph shows the sequence of corner points (1, 2, 3) examined by the simplex method for the Wyndor problem. The optimal solution, (2, 6) is found after examining just three points.

Now let us look at the rationale behind these three steps and the conclusions drawn.

Why start with the corner point at the origin, (0, 0)? Simply because this is a convenient place to begin. (When solving for corner points algebraically, this one can be identified without doing any algebra.) The only situation where (0, 0) cannot be chosen as the initial corner point is when it lies outside the feasible region and so is not a corner point. In this case, any real corner point can be chosen instead.

At step 1, how is the simplex method able to conclude that (0, 0) is not an optimal solution? It does this by checking the two *adjacent* corner points,
(4, 0) and (0, 6). Both of these points are *better* because they produce higher values of *Profit* (the objective function) than (0, 0). So (0, 0) cannot be optimal.

The only situation where (0, 0) would be optimal is when *neither* adjacent corner point is better. Check how this would happen if the objective in the model were changed to *minimize* Cost = 3 D + 5 W.

In step 2, why does the simplex method move to (0, 6)? We pointed out in Section 14.2 that one key to the efficiency of the simplex method is that it only moves to *adjacent* corner points. Therefore, from (0, 0), the only alternatives are to move next to either (4, 0) or (0, 6). (4, 0) gives Profit = 1,200 and (0, 6) gives Profit = 3,000. Both are an improvement over (0, 0) with Profit = 0, so either alternative would move us *toward* an optimal solution. Since we want to move toward an optimal solution as quickly as possible, we choose the *better* adjacent corner point (the one with the larger value of Profit), namely, (0, 6). (Further discussion of this criterion and an alternate criterion for selecting the adjacent corner point is given in the box entitled "Which Adjacent Corner Point Should Be Chosen?".)

Which Adjacent Corner Point Should Be Chosen?

When the simplex method is ready to move from the current corner point to an adjacent corner point, either of the following criteria can be used to choose the adjacent corner point.

The Best Adjacent Corner Point Criterion: Choose the *best adjacent corner point*, i.e., the one with the *most favorable value of the objective function*. (*Most favorable* means *largest* when the objective is to *maximize Profit*, whereas it means *smallest* when the objective is to *minimize Cost*.)

The Best Rate of Improvement Criterion: Determine the "rate of improvement" for each adjacent corner point. This rate of improvement is defined as the improvement in the value of the objective function per unit of distance moved along the edge of the feasible region from the current corner point to the adjacent corner point. (*Improvement in this value* means *increase in Profit* when the objective is to *maximize Profit*, and it means *decrease in Cost* when the objective is to *minimize Cost*.) Choose the adjacent corner point with the *best* (largest) rate of improvement.

To illustrate the second criterion, consider the Wyndor example when the current point is (0, 0) and the adjacent corner points are (4, 0) and (6, 0). Since the objective function to be maximized is
Profit = 300 D + 500 W, each unit increase in D increases Profit by 300, whereas each unit increase in W increases Profit by 500. Therefore, the *rate of improvement* from moving from (0, 0) toward (4, 0) is 300, and the *rate of improvement* from moving from (0, 0) toward (0, 6) is 500. Since 500 is larger than 300, this criterion says to select (0, 6) as the adjacent corner point to move to next.

The algebraic form of the simplex method normally uses the *best rate of improvement criterion*. The reason is that the algebraic procedure has a very efficient method for identifying rates of improvement without even solving algebraically for the adjacent corner points and calculating their values of the objective function. (This method uses a special definition for *unit of distance*, as we will clarify in Section 14.7.)

However, the *best rate of improvement criterion* is not very convenient for the graphical form of the simplex method being presented here. Except when the current corner point is (0, 0), this criterion usually would require somewhat more work than the *best adjacent corner point criterion*. Therefore, we will use the *best adjacent corner point criterion* when applying the simplex method graphically.

To finish step 2, (0, 6) is not optimal because one of its adjacent corner points, (2, 6), is better than (0, 6). Profit = 3,600 for (2, 6) is larger than Profit = 3,000 for (0, 6).

To begin step 3, the simplex method notes that (0, 6) has the two adjacent corner points, (0, 0) and (2, 6). (0, 0) was just examined and discarded before moving to (0, 6), so (2, 6) becomes the automatic choice to move to next. (Once the decision has been made to begin moving around the boundary of the feasible region in either the clockwise or counter-clockwise direction, all subsequent movement will be in the same direction.)

How does the simplex method then conclude that (2, 6) is the optimal solution? The reason is that both adjacent corner points, (0, 6) and (4, 3), are *not* better than (2, 6). We already know from the previous work that (0, 6) is not as good. Furthermore, (4, 3) gives Profit = 2,700, which is less than Profit = 3,600 for (2, 6). Since (4, 3) is worse than (2, 6), continuing to move clockwise beyond (4, 3) cannot possibly lead to a corner point better than (2, 6).

A Summary of the Simplex Method

The procedure for the simplex method we just illustrated is typical. Here is a summary for problems with either two or more decision variables.

> **Getting Started**: Select some corner point as the *initial corner point* to be examined. This choice can be made arbitrarily. However, if the *origin* is a corner point of the feasible region, this is a convenient choice. Wherever you choose to start, label it (temporarily) as the *current corner point* and continue as described in the following step.
>
> **Checking for Optimality**: Check each of the corner points *adjacent* to the *current corner point*. If *none* of the adjacent corner points are *better* (as measured by the value of the objective function) than the current corner point, then stop because the current corner point is an *optimal solution*. (Any adjacent corner point with a value of the objective function *equal* to this optimal value also is an optimal solution.) However, if *one or more* of the adjacent corner points are *better* than the current corner point, then continue as described in the next step.
>
> **Moving On**: One of the adjacent corner points that is better than the current corner point needs to be selected as the next current point to be examined. When executing this procedure graphically, choose the *best adjacent corner point* according to the value of Profit. (When executing this procedure algebraically, as described in Section 14.7, the *best rate of improvement criterion* is used instead to make this choice.) Label it the new *current corner point* and return to the Checking for Optimality step above.

These three steps apply equally well whether the problem has *two* decision variables or *more than two*. However, when there are just two decision variables, the procedure can be streamlined somewhat, as follows.

> **A Shortcut with Just Two Decision Variables:** When the *current* corner point still is the *initial* corner point, the *Moving On* step leads to selecting one of the adjacent corner points as the next corner point to be examined. This corner point is reached by moving from the initial corner point along the boundary of the feasible region in either the *clockwise* or *counter-clockwise* direction. The procedure thereafter involves moving further around the boundary of the feasible region in the *same* direction (clockwise or counter-clockwise), from corner point to corner point, until the optimal solution is reached. Thus, each time the procedure returns to the *Checking for Optimality* step, only the next adjacent corner point in this direction needs to be checked. This adjacent corner point then is automatically selected as the next corner point to be examined in the *Moving On* step.

We refer to this streamlined procedure with two decision variables as the **graphical simplex method**.

To solidify your understanding of the graphical simplex method, we suggest that you now go back and check how this description fits what we did with the Wyndor example in the preceding subsection.

Although the Wyndor example is a maximization problem, this summary also applies to minimization problems, as you'll see in the next example.

A Minimization Example

Consider the Profit & Gambit advertising-mix problem back in Section 2.7 (see Figure 2.22 and the Supplement to Chapter 2 for the application of the graphical method to this problem). The linear programming model and the corresponding graph are repeated here as Figure 4.15. The feasible region is *unbounded* and has just three corner points.

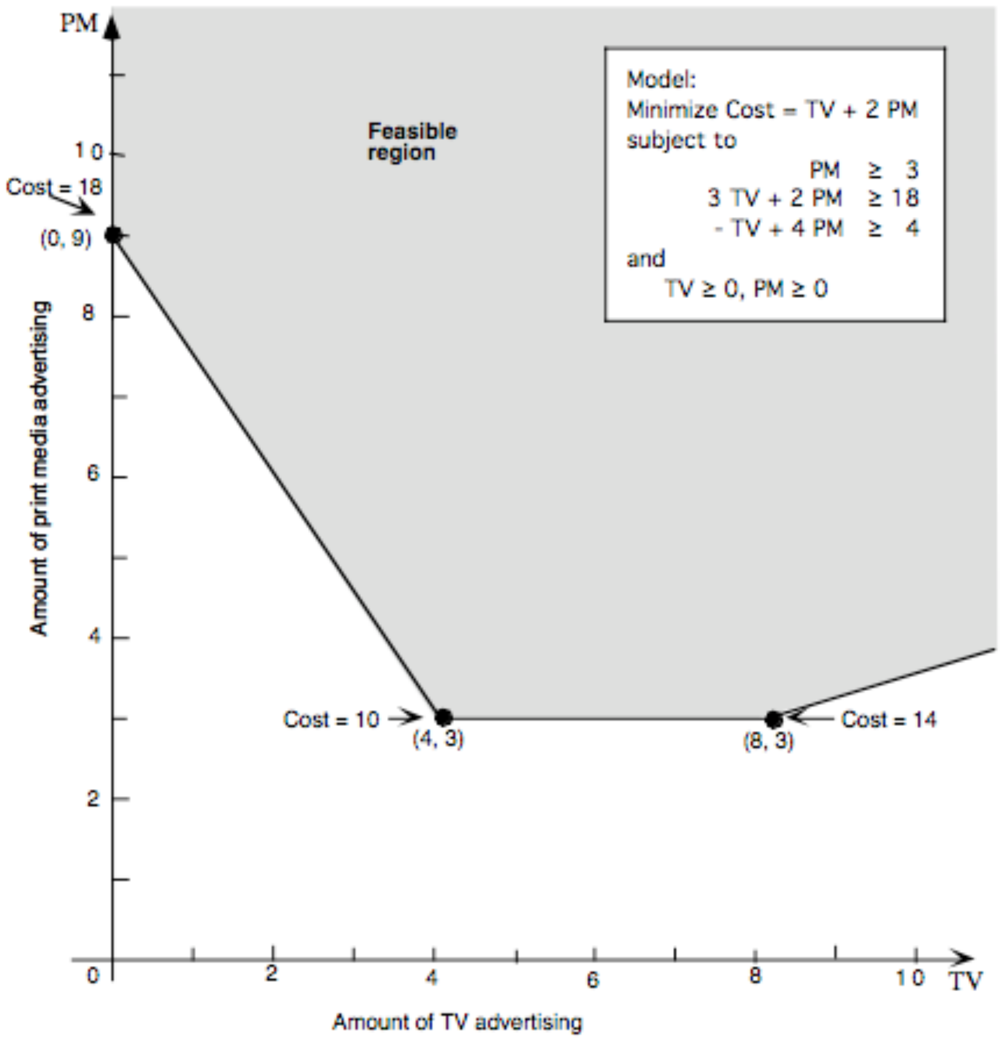


Figure 14.15 The three corner points to be considered by the simplex method to find the optimal solution for the Profit and Gambit advertising-mix problem.

Let's apply the simplex method in the format just given.

Getting Started: Since (0, 0) is not a corner point of the feasible region, we select the *initial corner point* arbitrarily, say, (0, 9) with Cost = 18.

Checking for Optimality: (0, 9) has just one *adjacent* corner point, (4, 3), since the feasible region is unbounded above (0, 9). Since (4, 3) gives Cost = 10, which is better than Cost = 18 for (0, 9), we conclude that (0, 9) is not optimal. (Remember the objective is to *minimize* Cost = TV + 2 PM.)

Moving On: (4, 3) is the *best* (and only) adjacent corner point of (0, 9), so the simplex method moves from (0, 9) to (4, 3). (Since this movement is along the boundary of the feasible region in the *counter-clockwise direction*, any subsequent movement will be in this same direction.)

Checking for Optimality: For (4, 3), we only need to check the adjacent corner point in the counter-clockwise direction, (8, 3). Since (8, 3) gives Cost = 14, which is worse than Cost = 10 for (4, 3), we conclude that (4, 3) is the optimal solution and the simplex method is finished.

You may check for yourself that the simplex method would have come to this same conclusion if the corner point selected to be the *initial corner point* had been either (4, 3) or (8, 3) instead of (0, 9).

REVIEW QUESTIONS

1. Does the simplex method examine all the corner points of a linear programming problem in order to find an optimal solution?
2. When the simplex method is ready to move from the current corner point to the next one, which corner points are candidates to be this next one?
3. What are the names of two criteria for selecting the next corner point?
4. How does the simplex method get started?
5. How does the simplex method determine if the current corner point is an optimal solution?
6. How does the graphical simplex method determine whether to move around the boundary of the feasible region in a clockwise or a counter-clockwise direction?
7. How does a minimization problem differ from a maximization problem when the graphical simplex method chooses the corner point to move to next?

14.5 THE SIMPLEX METHOD WITH THREE DECISION VARIABLES

Our main purpose in describing the simplex method with two decision variables is to provide a good intuitive insight into how it operates on linear programming problems with *more than two* decision variables. In fact, the *summary* given in Section 14.4 applies to larger problems as well. However, there are certain aspects of dealing with larger problems that are not well illuminated by looking at examples with two decision variables. Therefore, it is instructive to take a brief look at the case of *three* decision variables.

Thinking Three-Dimensionally

To illustrate the case of three decision variables, consider the case study of the Super Grain Corp. advertising-mix problem in Section 3.1. The linear programming model for this problem is

$$\text{Maximize} \quad \text{Exposure} = 1{,}300\ \text{TV} + 600\ M + 500\ \text{SS},$$

subject to

$$\begin{aligned} 300\ \text{TV} + 150\ M + 100\ \text{SS} &\leq 4{,}000 \\ 90\ \text{TV} + 30\ M + 40\ \text{SS} &\leq 1{,}000 \\ \text{TV} &\leq 5 \end{aligned}$$

and

$$\text{TV} \geq 0, \quad M \geq 0, \quad \text{SS} \geq 0.$$

By taking some care, it is possible to graph the feasible region for this problem as shown in Figure 14.16.

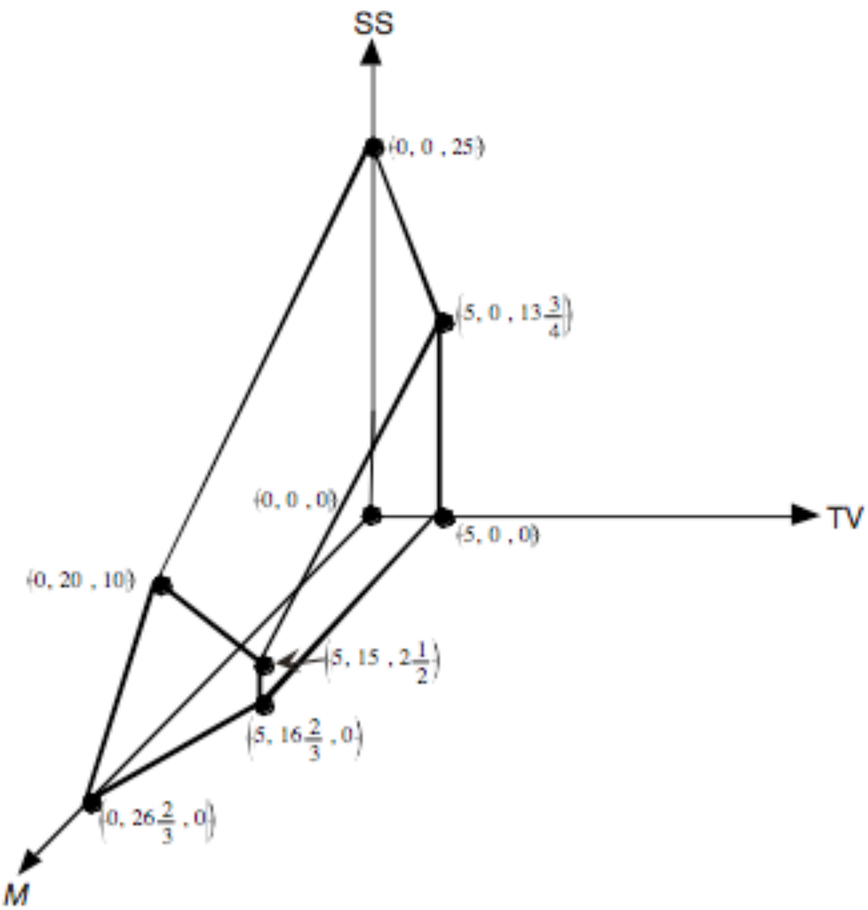


Figure 14.16 This three-dimensional graph shows the feasible region and its corner points for the Super Grain Corp. advertising-mix problem.

To help visualize this three-dimensional graph, think of yourself as standing in the middle of a room and looking toward one corner where two walls and the floor meet. The edge where the wall on your right meets the floor is the TV axis, and the edge where the wall to the left meets the floor is the *M* axis. The SS axis coincides with the edge where the two walls meet.

Now look at the feasible region drawn in Figure 14.16 and think of this as a *solid, three-dimensional object* that sits in this corner of the room. This object lies flat on the floor (the constraint boundary SS = 0) and has two vertical sides that are flush against the two walls (constraint boundaries TV = 0 and *M* = 0). The object also has a third vertical side that is parallel to the left-side wall and five units of distance from this wall (the constraint boundary TV = 5). Finally, the object has a roof with two slanting sections. The larger slanting section (given by the constraint boundary equation, 90 TV + 30 *M* + 40 SS = 1,000) has corners at (0, 0, 25), (5, 0, 13.75), (5, 15, 2.5), and (0, 20, 10). The smaller and steeper section (given by the constraint boundary equation, 300 TV + 150 *M* + 100 SS = 4,000) has corners at (0, 20, 10), (5, 15, 2.5), (5, 16.667, 0), and (0, 26.667, 0). The two sections meet at the edge between (0, 20, 10) and (5, 15, 2.5). The entire *solid* object (all the way back to the hidden sides) is the feasible region.

The object has eight *corners* that are highlighted by dots — the six corners of the two sections of the roof plus the two corners at floor level in the back, (0,0,0) and (5, 0, 0). These corners are the eight corner points of the feasible region for the linear programming problem.

Table 14.7 summarizes these corner points and their values of the objective function. Note that (5, 15, 2.5) is the corner point with the best value of the objective function. By examining all eight corner points and comparing their objective function values, the *enumeration-of-corner-points method* presented in Section 14.1 would conclude that (TV, *M*, SS) = (5, 15, 2.5) must be the optimal solution.

Table 14.7 The Corner Points and Their Objective Function Values for the Super Grain Problem

Corner Point	Value of Objective Function
(0, 0, 0)	0
(0, 26.667, 0)	16,000
(5, 16.66 , 0)	16,500
(5, 15, 2.5)	16,750
(0, 20,10)	17,000
(0, 0, 25)	10,000
(5, 0, 13.75)	13,375
(5, 0, 0)	6,500

Now let's see how the simplex method finds this same optimal solution without examining all the corner points.

Applying the Simplex Method

The following steps outline the application of the simplex method to this problem.

Getting Started: Since (0, 0, 0) is a corner point of the feasible region, it is selected to be the *initial corner point* to be examined.

Checking for Optimality: (0, 0, 0) has three *adjacent corner points*: (5, 0, 0) with Exposure = 6,500, (0, 26.667, 0) with Exposure = 16,000, and (0, 0, 25) with Exposure = 12,500. [Note that these three corner points indeed are *adjacent* to (0, 0, 0) since they each lie on all but one of the (0, 0, 0) constraint boundaries: TV = 0, *M* = 0, and SS = 0.] All three adjacent corner points are better than (0, 0, 0) with Exposure = 0, so (0, 0, 0) is not optimal.

Moving On: Since we are executing the procedure graphically, we choose the *best adjacent corner point,* (0, 26.667, 0) with Exposure = 16,000. The simplex method then moves along the *edge* of the feasible region from (0, 0, 0) to (0, 26.667, 0).

Checking for Optimality: (0, 26.667, 0) has three *adjacent corner points*: (0, 0, 0), (5, 16.667, 0), and (0, 20, 10). We already know from the preceding steps that (0, 0, 0) is not better than (0, 26.667, 0) with Exposure = 16,000. However, we need to check (5, 16.667, 0) with Exposure = 16,500 and (0, 20, 10) with Exposure = 17,000. Both are better than (0, 26.667, 0), so (0, 26.667, 0) is not optimal.

Moving On: We now want to choose the *best adjacent corner point*, which is (0, 20, 10) with Exposure = 17,000. With this choice, the simplex method moves along the *edge* of the feasible region from (0, 26.667 , 0) to (0, 20, 10).

Checking for Optimality: We now compare

Exposure = 17,000 for (0, 20, 10)

with the value of Exposure for the three *adjacent corner points*:

Exposure = 16,500 for (0, 26.667, 0)
Exposure = 16,750 for (5, 15, 2.5)
Exposure = 10,000 for (0, 0, 25)

Since all these adjacent corner points are worse than (0, 20, 10), we conclude that (0, 20, 10) is the optimal solution and the simplex method is finished.

To summarize, here is the path that was followed by the simplex method to reach this optimal solution.

Path:	(0, 0, 0) →	(0, 26.667 , 0) →	(0, 20 , 10)
Exposure:	0	16,000	17,000

When executing the simplex method algebraically, the *best rate of improvement criterion* is used instead of the *best adjacent corner point criterion* to choose the next corner point to be examined. The best rate of improvement criterion focuses on the objective of maximizing Exposure = 1,300 TV + 600 *M* + 500 SS. When starting from (0, 0, 0), this criterion says to begin by increasing TV because this gives the best rate of improvement in Exposure (1,300 is larger than either 600 or 500). This leads us along the *edge* of the feasible region from (0, 0, 0) to the adjacent corner point with TV increased from zero, (5, 0, 0). The simplex method with this criterion then follows the path to the optimal solution shown below.

Path:	(0, 0, 0) →	(5, 0 , 0) →	(5, 16.667, 0) →	(5, 15, 2.5) →	(0, 20, 10)
Exposure:	0	7,000	16,500	16,750	17,000

Note that this path with the best rate of improvement criterion involves examining two more corner points than the previous path with the best adjacent corner point criterion. On another problem, the reverse could well happen. Neither criterion has a substantial advantage in terms of the average number of corner points that need to be examined to reach an optimal solution.

General Characteristics

In Section 14.4, we saw what happens when a problem has just *two* decision variables: The constraint boundaries are simply *lines*, where each corner point lies at the intersection of *two* such lines. Solving algebraically for the corner point requires solving a system of *two* constraint boundary equations. There are *two* (at most) adjacent corner points. The corner point shares *all but one* (namely, 2 - 1 = 1) of its constraint boundaries with each of its adjacent corner points. Therefore, the edge of the feasible region connecting the corner point and any particular adjacent corner point lies on the shared constraint boundary.

In this section, we have illustrated how these characteristics change when a problem has *three* decision variables: The constraint boundaries now are *planes*, where each corner point lies at the intersection of *three* such planes. Solving algebraically for the corner point requires solving a system of *three* constraint boundary equations. There are *three* (at most) adjacent corner points. The corner point shares *all but one* (namely, 3 - 1 = 2) of its constraint boundaries with each of its adjacent corner points. Therefore, the edge of the feasible region connecting the corner point and any particular adjacent corner point lies at the *intersection* of these shared constraint boundaries.

The situation is very analogous when the number of decision variables *exceeds three*. In fact, *every* statement in the preceding paragraph still holds when the number *three* is replaced throughout by the actual number of decision variables. The only exception is that, in the second sentence, the word *planes* should be replaced by *hyperplanes*. (A "hyperplane" is a "flat" surface in higher dimensions that is analogous to a plane in three dimensions.)

Rest assured that we do not expect you to be able to visualize the geometry of higher dimensions. (We have trouble with that ourselves.) You are doing very well if you can visualize the three-dimensional graph in Figure 14.16. For problems with *more than three* decision variables, the important point is that the characteristics of the simplex method are very analogous to the characteristics with *three* decision variables.

To reinforce this point, it would be helpful for you to review the solution concepts for the simplex method presented in Section 14.3.

REVIEW QUESTIONS

1. Looking at the origin of a three-dimensional graph is analogous to doing what while standing in the middle of a room?
2. When a linear programming problem has three decision variables, how many *adjacent corner points* can a corner point have?
3. Is it possible for the best rate of improvement criterion and the best adjacent corner point criterion to follow a different path to an optimal solution?
4. With three decision variables, the constraint boundaries have what geometric form?
5. With *n* decision variables, solving algebraically for a corner point requires solving a system of how many constraint boundary equations?

14.6 THE ROLE OF SUPPLEMENTARY VARIABLES

In addition to decision variables, the simplex method also considers *slack variables* and *surplus variables* — supplementary variables that provide additional information and simplify the algebraic operations.

Slack Variables

To illustrate slack variables, consider again the Wyndor problem and its linear programming model summarized in Figure 14.12. The slack variables involve just the functional constraints. The three functional constraints for this problem are

$$\begin{aligned} D \quad\quad\; &\le 4 \\ 2W &\le 12 \\ 3D + 2W &\le 18. \end{aligned}$$

These three constraints specify the restrictions on the amount of production time that can be used in the three plants of the company for the two new products (special doors and windows) under consideration, where D is the number of doors produced per week and W is the number of windows produced per week. The numbers on the right-hand side of the constraints are the number of hours of production time *available* per week in the three respective plants. The left-hand sides represent the number of hours of production time per week *actually used* for the two products in the respective plants.

If we subtract the *time used* from the *time available*, we obtain:

Unused production time in Plant 1 = $4 - D$.
Unused production time in Plant 2 = $12 - 2W$.

Unused production time in Plant 3 = $18 - 3D - 2W$.

Let's introduce algebraic symbols — s_1, s_2, and s_3 — to be the *slack variables* that represent these quantities.

Slack variable for first constraint: $s_1 = 4 - D$.
Slack variable for second constraint: $s_2 = 12 - 2W$.
Slack variable for third constraint: $s_3 = 18 - 3D - 2W$.

The name derives from the fact that the slack variable for a ≤ constraint represents the *slack* (gap) between the two sides of the inequality. In this case, the slack is the unused production time.

For example, for the optimal solution, $(D, W) = (2, 6)$, the slack variables have the values:

$$s_1 = 4 - D = 4 - 2 = 2,$$
$$s_2 = 12 - 2W = 12 - 2(6) = 0,$$
$$s_3 = 18 - 3D - 2W = 18 - 3(2) - 2(6) = 0.$$

Thus, this solution would leave Plant 1 with some unused production time (2 hours per week), but none for Plants 2 and 3.

For any linear programming problem, the **slack variable** for a ≤ constraint is a variable that *equals* the right-hand side *minus* the left-hand side. The constraint is satisfied as long as the slack variable is *nonnegative*, since this implies that the left-hand side is not larger than the right-hand side.

There are several reasons why it is useful to introduce slack variables. One is that the values of the slack variables provide valuable information to management. For the Wyndor problem, management would like to know the impact of a proposed product mix on the unused production times in the various plants.

A second reason is that slack variables enable converting ≤ constraints into *equations*. The functional constraints with slack variables for the Wyndor problem are

$$\begin{array}{llllllll} \text{Plant 1:} & D + & & s_1 & & & = & 4 \\ \text{Plant 2:} & & 2W + & & s_2 & & = & 12 \\ \text{Plant 3:} & 3D + & 2W + & & & s_3 & = & 18, \end{array}$$

where all the decision variables (D, W) and slack variables (s_1, s_2, s_3) also are required to be *nonnegative*. This is a very convenient form of the problem for the simplex method, because it is much simpler for an algebraic procedure to deal with *equations* than with *inequalities*. For example, whenever two of the five variables in these three equations are set equal to zero, it is then straightforward on a computer to solve the system of three equations for the three remaining variables.

Still another reason is that having a slack variable equal zero for a solution immediately identifies a *constraint boundary* on which the solution must lie. For example, consider the optimal solution, $(D, W) = (2, 6)$, for the Wyndor problem. Since we already have calculated that $s_2 = 0$ and $s_3 = 0$ for this corner point, the above equations for Plants 2 and 3 immediately indicate that the corner point must satisfy

$$\begin{array}{rrl} & 2W & = 12 \\ 3D + & 2W & = 18. \end{array}$$

These are the **constraint boundary equations** for the constraints, $2W \leq 12$ and $3D + 2W \leq 18$, respectively, so the corner point must lie on these two constraint boundaries.

In fact, the simplex method always identifies the *current corner point* being examined for this problem by setting two of the five variables equal to zero. To begin, it sets $D = 0$ and $W = 0$ (the *origin* lies on the constraint boundaries, $D = 0$ and $W = 0$). After an iteration, it substitutes $s_2 = 0$ for $W = 0$, since the new corner point (0, 6) lies on the constraint boundaries, $D = 0$ and $2W = 12$. After one more iteration, it reaches the optimal solution (2, 6) by substituting $s_3 = 0$ for $D = 0$, which gives the constraint boundaries, $2W = 12$ and $3D + 2W = 18$.

Table 14.8 summarizes these iterations of the simplex method. The first two columns show the corner points in the order in which they are examined (just as was depicted in Figure 14.14). The third column indicates which two variables have been set equal to zero. The next column gives the solution of the system of three equations (Plants 1, 2, and 3) for the three remaining variables. The final column shows the equations of the constraint boundaries on which the corner point lies.

Table 14.8 The Progression of the Simplex Method on the Wyndor Problem

Order	Corner Point	Variables = 0	Other Variables	Corresponding Constraint Boundary Equations
1	$(D, W) = (0, 0)$	$D = 0$ $W = 0$	$s_1 = 4$ $s_2 = 12$ $s_3 = 18$	$D = 0$ $W = 0$
2	$(D, W) = (0, 6)$	$D = 0$ $s_2 = 0$	$s_1 = 4$ $W = 6$ $s_3 = 6$	$D = 0$ $2W = 12$
3	$(D, W) = (2, 6)$	$s_3 = 0$ $s_2 = 0$	$s_1 = 2$ $W = 6$ $D = 2$	$2W = 12$ $3D + 2W = 18$

The variables that the simplex method currently has set equal to zero (whether decision variables and/or slack variables) are called **nonbasic variables**. (These are the variables in the third column of Table 2.) The other variables (those in the fourth column) are called **basic variables**. The resulting solution for *all* the variables, *including* the slack variables, is called a **basic feasible solution**. (This solution combines the third and fourth columns.) A *basic feasible solution* is simply a *corner point* that has been augmented by including the values of the slack variables.

To illustrate these terms, consider the optimal solution for the Wyndor problem. This optimal solution can be expressed as either a *corner point* (no slack variables) or as a *basic feasible solution* (include the slack variables), as summarized below.

	Optimal Solution				
Corner point:	$D = 2$,	$W = 6$.			
Basic feasible solution:	$D = 2$,	$W = 6$,	$s_1 = 2$,	$s_2 = 0$,	$s_3 = 0$.
Nonbasic variables:	$s_2 = 0$,	$s_3 = 0$.			
Basic variables:	$D = 2$,	$W = 6$,	$s_1 = 2$.		

The solution is the same either way. The only difference is in the amount of information being given about the solution. The two names, *corner point* and *basic feasible solution*, are used just to differentiate between the amount of information being provided.

After it is initialized, the simplex method does not need to distinguish between decision variables and slack variables. All variables are treated alike.

Surplus Variables

Surplus variables can be thought of as the *mirror image* of *slack variables*. Surplus variables arise with ≥ functional constraints, whereas slack variables arise with ≤ constraints. Specifically, a **surplus variable** gives the amount by which the *left-hand side* of a ≥ constraint exceeds the *right-hand side* — the *surplus* on the left-hand side over the right-hand side. By contrast, a slack variable gives the *slack* by which the *right-hand side* of a ≤ constraint exceeds the *left-hand side*.

To illustrate, consider the Profit & Gambit advertising-mix problem that is graphed in Figure 14.15. Management has prescribed that the advertising campaign being planned must yield an increase in sales of *at least* 3%, 18%, and 4% for the stain remover, liquid detergent, and powder detergent, respectively. These requirements led to the following three functional constraints:

Stain remover:	PM	≥ 3
Liquid detergent:	3 TV + 2 PM	≥ 18
Powder detergent:	- TV + 4 PM	≥ 4

where TV is the number of units of advertising on television and PM is the number of units of advertising in the print media.

Let s_1, s_2, and s_3 denote the corresponding surplus variables, as summarized below.

Surplus variable for first constraint:	s_1 = PM - 3.
Surplus variable for second constraint:	s_2 = 3 TV + 2 PM - 18.
Surplus variable for third constraint:	s_3 = - TV + 4 PM - 4.

Thus, each surplus variable represents the surplus in the *actual increase* in sales over the *minimum required increase* in sales for that product. Note that the three functional constraints are satisfied as long as the corresponding surplus variables are *nonnegative*.

For example, at the optimal solution, (TV, PM) = (4, 3), the surplus variables have the values:

s_1 = PM - 3 = 3 - 3 = 0,
s_2 = 3 TV + 2 PM - 18 = 3 (4) + 2 (3) - 18 = 0,
s_3 = -TV + 4 PM - 4 = -4 + 4 (3) - 4 = 4.

The fact that these three surplus variables are *nonnegative* (along with TV and PM) immediately indicates that this solution is indeed feasible. The fact that $s_1 = 0$ and $s_2 = 0$ also indicates that the optimal solution is the corner point that lies on the constraint boundaries for the first two functional constraints. In other words, this corner point is the simultaneous solution of the two constraint boundary equations,

$$\begin{aligned} \text{PM} &= 3 \\ 3\ \text{TV} + 2\ \text{PM} &= 18 \end{aligned}$$

All the discussion in the preceding subsection about why it is useful to introduce *slack* variables also applies equally well to *surplus* variables, and will not be repeated here. The terminology introduced there about *nonbasic variables*, *basic variables*, and *basic feasible solutions* also applies with surplus variables. To illustrate, the optimal solution for the example can be expressed as either a *corner point* (without surplus variables) or a *basic feasible solution* (with surplus variables), as shown below.

Optimal Solution

Corner point:	TV = 4, PM = 3.		
Basic feasible solution:	TV = 4, PM = 3,	$s_1 = 0$, $s_2 = 0$,	$s_3 = 4$.
Nonbasic variables:	$s_1 = 0$, $s_2 = 0$.		
Basic variables:	TV = 4, PM = 3,	$s_3 = 4$.	

REVIEW QUESTIONS

1. The name *slack variable* is derived from what?
2. Why does a slack variable being nonnegative imply that the corresponding functional constraint is satisfied?
3. What do the slack variables for the Wyndor problem represent?
4. Why is it more convenient for the simplex method to have ≤ constraints converted into equations by introducing slack variables?
5. What value does a nonbasic variable have?
6. What is the difference between a *corner point* and the corresponding *basic feasible solution*?
7. What does a surplus variable for a ≥ constraint represent?

14.7 SOME ALGEBRAIC DETAILS FOR THE SIMPLEX METHOD

In Sections 14.4 and 14.5, we described the simplex method from a geometric viewpoint. The goal was to give a good intuitive feeling for what the simplex method does, and why, without worrying about the algebraic details. This section fills in these details.

We continue to use the Wyndor Co. problem summarized in Figure 14.12 to illustrate these details.

Connecting the Geometry and Algebra of the Simplex Method

Figure 14.12 shows that the feasible region for the Wyndor problem has five *constraint boundaries*. The *Constraint Boundary Equation* (CBE) for each one is

$$
\begin{aligned}
&\text{CBE 1:} \quad D &&= 4\\
&\text{CBE 2:} \quad 2W &&= 12\\
&\text{CBE 3:} \quad 3D + 2W &&= 18\\
&\text{CBE 4:} \quad D &&= 0\\
&\text{CBE 5:} \quad W &&= 0.
\end{aligned}
$$

Each corner point satisfies two of these constraint boundary equations, i.e., it lies at the intersection of the corresponding two constraint boundaries.

As described in Section 14.6, the simplex method begins by introducing *slack variables* (s_1, s_2, s_3) into the functional constraints to obtain the following system of equations:

$$
\begin{aligned}
&(1) \quad D + s_1 = 4\\
&(2) \quad 2W + s_2 = 12\\
&(3) \quad 3D + 2W + s_3 = 18,
\end{aligned}
$$

where all five variables must be nonnegative. At each iteration, the simplex method sets two of the five variables equal to zero (the *nonbasic variables*) and then solves this system of three equations for the three remaining variables (the *basic variables*). The resulting solution, called a *basic feasible solution*, is a corner point that has been augmented by including the values of the slack variables.

Section 14.6 also describes how setting a slack variable equal to zero (by choosing that slack variable to be a nonbasic variable) immediately identifies a *constraint boundary* on which the solution for the decision variables must lie. Choosing a decision variable to be a nonbasic variable does the same thing. This relationship between nonbasic variables and constraint boundaries is:

If s_1 is a nonbasic variable ($s_1 = 0$), then CBE 1 is satisfied.
If s_2 is a nonbasic variable ($s_2 = 0$), then CBE 2 is satisfied.
If s_3 is a nonbasic variable ($s_3 = 0$), then CBE 3 is satisfied.
If D is a nonbasic variable ($D = 0$), then CBE 4 is satisfied.
If W is a nonbasic variable ($W = 0$), then CBE 5 is satisfied.

Thus, the choice of the two variables to be the current nonbasic variables determines the two constraint boundaries on which the current corner point will lie.

As described in Section 14.4 (see Figure 14.14), the geometric path followed by the simplex method for this problem is from the initial corner point (0, 0) to the adjacent corner point (0, 6) to its adjacent corner point (2, 6). The left side of Table 14.9 summarizes this sequence, including the pair of constraint boundary equations yielding each of these corner points, where the first column indicates the number of completed iterations. The right side shows the same sequence when the slack variables also are included for the algebraic execution of the simplex method. Thus, the last column gives the current value of D, W, s_1, s_2, s_3 in that order. Note how each pair of nonbasic variables corresponds to the pair of constraint boundary equations in the manner prescribed in the immediately preceding paragraph. Also note how each basic feasible solution is just the corresponding corner point plus the resulting values of the slack variables.

Table 14.9 The Path Followed by the Simplex Method for the Wyndor Problem

	Geometric Progression		Algebraic Progression		
Iteration	Corner Point	CBE	Nonbasic Variables	Basic Variables	Basic Feasible Solution (D, W, s_1, s_2, s_3)
0	(0, 0)	4, 5	D, W	s_1, s_2, s_3	(0, 0, 4, 12, 18)
1	(0, 6)	4, 2	D, s_2	s_1, W, s_3	(0, 6, 4, 0, 6)
2	(2, 6)	3, 2	s_3, s_2	s_1, W, D	(2, 6, 2, 0, 0)

Now focus on the columns of Table 14.9 giving the nonbasic variables and the basic variables. When moving from the first row of the table to the second (i.e., performing the first iteration), one nonbasic variable (W) becomes a basic variable and one basic variable (s_2) becomes a nonbasic variable. When moving from the second row to the third (i.e., performing the second iteration), the same pattern recurs. One nonbasic variable (D) becomes a basic variable and one basic variable (s_3) becomes a nonbasic variable. This pattern is no coincidence. During *any* iteration of the simplex method for any problem, exactly *one* nonbasic variable becomes a basic variable and one basic variable becomes a nonbasic variable. (The reason is that an iteration *always* moves from the current corner point to an *adjacent* corner point, i.e., a corner point that shares all but one of the same constraint boundaries with the current corner point.)

When performing an iteration, the nonbasic variable that becomes a basic variable is called the **entering basic variable**. The basic variable that becomes a nonbasic variable is called the **leaving basic variable**. Thus, for the first iteration for the Wyndor problem, W is the entering basic variable and s_2 is the leaving basic variable. For the second iteration, D is the entering basic variable and s_3 is the leaving basic variable.

Each **iteration** of the simplex method consists of the steps outlined below.

OUTLINE OF AN ITERATION OF THE SIMPLEX METHOD

1. Determine the entering basic variable.
2. Determine the leaving basic variable.
3. Solve for the new basic feasible solution.

The overall flow of the algorithm, including these iterations, is summarized next.

STRUCTURE OF THE SIMPLEX METHOD

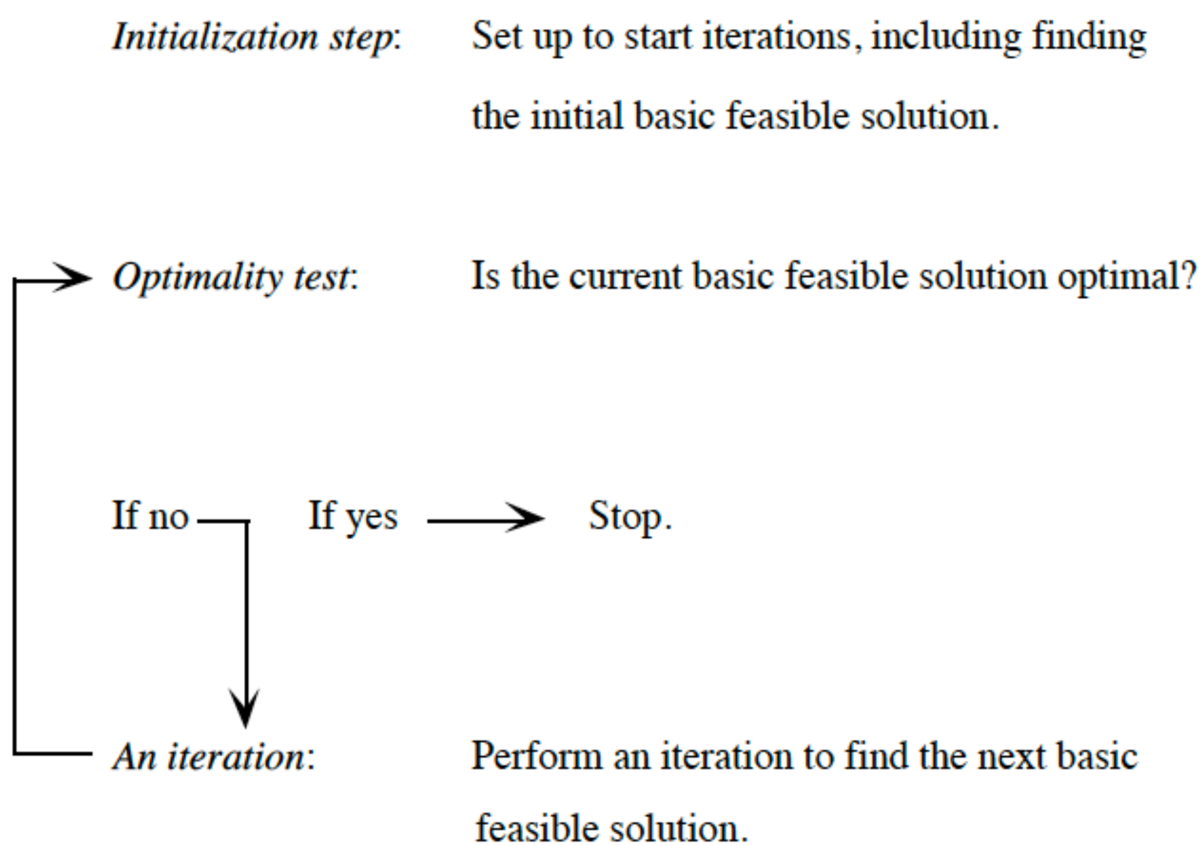


Table 14.9 has summarized the path followed by the simplex method for the Wyndor problem, but not how this path is found. We now describe how each of the steps are performed.

The Initialization Step

To simplify the notation, we now will let the symbol P represent the *value of the objective function*, i.e.,

P = Profit per week from the doors and windows

$= 300\,D + 500\,W.$

This objective function, $P = 300\,D + 500W$, can be rewritten with the decision variables on the left-hand side as

$$P - 300D - 500W = 0,$$

where P can be viewed as an additional variable in this equation. The initialization step begins by writing this equation along with the equations obtained by introducing slack variables into the functional constraints. This leads to the following equivalent statement of the original problem.

Maximize P,

subject to satisfying the following system of equations:

$$\begin{array}{llllllll} (0) & P - 300\,D - & 500W & & & & = & 0 \\ (1) & \quad D & & + s_1 & & & = & 4 \\ (2) & & 2W & & + s_2 & & = & 12 \\ (3) & \quad 3\,D + & 2W & & & + s_3 & = & 18 \end{array}$$

and

$$D \geq 0,\; W \geq 0,\; s_1 \geq 0,\; s_2 \geq 0,\; s_3 \geq 0.$$

(You will see after the first iteration why it is useful to include equation 0 in this system of equations.)

The initialization step next uses equations 1 - 3 to find a convenient *initial basic feasible solution*. This involves selecting two of the five variables (D, W, s_1, s_2, s_3) to be *nonbasic variables* (so set equal to zero) and the other three to be *basic variables*. These equations are then used to solve for the values of the basic variables.

The most convenient choice (since it avoids doing any algebra) is to select the *decision variables* (D, W) to be the nonbasic variables, and the *slack variables* to be the basic variables. After setting $D = 0$ and $W = 0$, equations 1 - 3 immediately yield:

Initial Basic Feasible Solution
Nonbasic variables: $D = 0$, $W = 0$
Basic variables: $s_1 = 4$, $s_2 = 12$, $s_3 = 18$
Value of objective function: $P = 0$.

Note why this solution for the basic variables can be read directly from equations 1 - 3, without performing any algebraic operations. The reason is that each of these equations has just one basic variable, which has a coefficient of 1, and this basic variable does not appear in any other equation (including equation 0). You will soon see that when subsequent iterations change the set of basic variables, the simplex method uses an algebraic procedure (Gaussian elimination) to convert the equations into this same convenient form for reading every subsequent basic feasible solution as well. This form is called **proper form from Gaussian elimination**.

The Optimality Test

The optimality test is applied quickly and easily by using the current equation 0 given above,

$$(0)\;\; P - 300\,D - 500W = 0.$$

The question being asked in examining this equation is whether the value of P can be increased by increasing the value of any of the nonbasic variables (D, W) from 0. The rule for answering this question is the following.

> **Rule for the Optimality Test**: Examine the current equation 0, which contains only P and the nonbasic variables (no basic variables) on the left-hand side along with a constant on the right-hand side. If *none* of the nonbasic variables have a *negative coefficient*, then the current basic feasible solution is *optimal*. If *one or more* of these coefficients are *negative*, then the current basic feasible solution is *not optimal*.

Since both D and W have a negative coefficient (-300 and -500), this test concludes that the current basic feasible solution ($D = 0$, $W = 0$, $s_1 = 4$, $s_2 = 12$, $s_3 = 18$) is *not* optimal.

The reasoning behind this test becomes more apparent when the nonbasic variables in the equation are brought over to the right-hand side (so negative coefficients become positive and vice-versa),

$$P = 0 + 300\,D + 500W.$$

At this point (before any iterations are performed), this just gives the original objective function. For the current basic solution, both $D = 0$ and $W = 0$, so $P = 0$. Increasing either D or W increases P, where the coefficient of that variable (300 or 500) gives the *rate* at which P increases per unit increase in the variable. Furthermore, either D or W can be increased by at least a small amount and still yield a feasible solution by adjusting the values of the basic variables to satisfy the system of equations. (Adjusting the values of the basic variables does not affect the value of P, because the basic variables are not present in equation 0.) For example, increasing the value of D from 0 to 1 changes the current solution from $D = 0$, $W = 0$, $s_1 = 4$, $s_2 = 12$, $s_3 = 18$ (with $P = 0$) to $D = 1$, $W = 0$, $s_1 = 3$, $s_2 = 12$, $s_3 = 15$ (with $P = 300$). Therefore, the former solution cannot be optimal.

Increasing the value of one of the nonbasic variables from 0 while adjusting the values of the basic variables accordingly corresponds to moving along an edge of the feasible region from the current corner point to one of its adjacent corner points. This leads to the interpretation of the optimality test given in Solution Concept 6 at the end of Section 14.3.

Determining the Entering Basic Variable

To begin the first iteration, the first step is to determine the *entering basic variable* (the current nonbasic variable that should become a basic variable for the next basic feasible solution). Since there currently are two nonbasic variables, D and W, one of these two variables must be chosen.

Just as for the optimality test, this step is executed by using the current equation 0,

$$(0) \qquad P - 300\,D - 500\,W = 0.$$

The question being addressed at this step is which nonbasic variable would increase P the most by increasing the value of that nonbasic variable from 0 to 1. This question is answered as follows.

> **Rule for Determining the Entering Basic Variable**: Examine the current equation 0, which contains only P and the nonbasic variables (no basic variables) on the left-hand side along with a constant on the right-hand side. Among the nonbasic variables with a *negative coefficient*, choose the one whose coefficient has the *largest absolute value* to be the entering basic variable.

Remember that one or more of the nonbasic variables currently must have a negative coefficient, since this is how the optimality test determined that the current basic feasible solution is not optimal.

In the current equation 0 of the example, both D and W have a negative coefficient (-300 and -500). The absolute value of a negative number is obtained by dropping the negative sign, so these coefficients have the absolute value, 300 and 500, respectively. Since 500 is larger than 300, W is chosen to be the entering basic variable.

As for the optimality test, the reasoning behind this rule becomes more apparent when the nonbasic variables in the equation are brought over to the right-hand side,

$$P = 0 + 300\,D + 500\,W.$$

Increasing D from 0 to 1 increases P by 300, whereas increasing W from 0 to 1 increases P by 500. (Increasing either D or W also necessitates adjusting the values of the basic variables to satisfy the system of equations, but these adjustments do not affect the value of P because the basic variables are not present in equation 0.) These increases in P actually are *rates of improvement in P* per unit increase in the nonbasic variable involved. Thus, increasing W gives a *better rate of improvement in P* than increasing D.

Referring back to Figure 14.12 in Section 14.2, increasing W from 0 corresponds to moving along the edge of the feasible region from the current corner point, (0, 0) with $P = 0$, toward one of the adjacent corner points, (0, 6) with $P = 500\,(6) = 3{,}000$. Increasing D from 0 corresponds to moving along the edge of the feasible region from (0, 0) toward the other adjacent corner point, (4, 0) with $P = 300\,(4) = 1{,}200$. Solution Concept 5 in Section 14.3 describes how the former alternative would be chosen because it gives the larger rate of improvement in P. Thus, the rule for determining the entering basic variable is based on the *best rate of improvement criterion* . (This criterion was first described in Section 1, but now we can be more specific in defining the *rate of improvement in P* as the increase in *P per unit increase in the nonbasic variable involved*.)

Determining the Leaving Basic Variable

The second step of an iteration involves determining the *leaving basic variable* (the current basic variable that should become a nonbasic variable for the next basic feasible solution). The current candidates are the three basic variables s_1, s_2 , s_3.

This step uses all the equations in the current system of equations except equation 0,

$$
\begin{array}{lrrrrrl}
(1) & D & & +\,s_1 & & & = 4 \\
(2) & & 2W & & +\,s_2 & & = 12 \\
(3) & 3\,D\ + & 2W & & & +\,s_3 & = 18
\end{array}
$$

The question being asked this time is which basic variable decreases to 0 first as the entering basic variable is increased. The answer is provided by the following rule.

> **Minimum Ratio Rule for Determining the Leaving Basic Variable**: For each equation that has a *strictly positive coefficient* (neither zero nor negative) for the entering basic variable, take the *ratio* of the right-hand side to this coefficient. Identify the equation that has the *minimum ratio*, and select the basic variable in this equation to be the leaving basic variable.

Notice how the last part of this rule ("select the basic variable in this equation") uses the fact that the system of equations is in *proper form from Gaussian elimination*. This form ensures that there is exactly one basic variable in each of these equations, namely, s_1 in equation 1, s_2 in equation 2, and s_3 in equation 3.

Since W is the entering basic variable, only equations 2 and 3 have a *strictly positive coefficient* for this variable. The resulting *ratios* for these equations are

$$(2)\ \frac{12}{2} = 6 \quad \leftarrow \text{minimum}$$

$$(3)\ \frac{18}{2} = 9.$$

Since equation 2 has the *minimum ratio*, its basic variable becomes the leaving basic variable.

The reasoning behind this rule is based on increasing the entering basic variable (and so increasing *P*) as much as possible without causing the resulting solution to become infeasible. From equation 2,

$$s_2 = 12 - 2W,$$

which implies that 12/2 = 6 is the upper bound on W without violating the nonnegativity constraint, $s_2 \geq 0$. Since $D = 0$, equation 3 yields

$$s_3 = 18 - 2W,$$

so 18/2 = 9 is the upper bound on W without violating $s_3 \geq 0$. However, using $W = 9$ would make $s_2 = -6$, which is not feasible. Therefore, the *minimum ratio* is used to select $W = 6$ and $s_2 = 0$ (a nonbasic variable) to be part of the next basic feasible solution (along with $D = 0$, $s_1 = 4$, $s_3 = 6$). Equation 1 does not enter into this analysis because it does not contain W, so increasing W would never cause s_1 to become negative.

The graphical interpretation of this line of reasoning is provided by referring back to Figure 14.12. Starting from (0, 0) and increasing W leads up the W axis toward the adjacent corner point (0, 6). Increasing W to $W = 6$ reaches (0, 6), which lies on the constraint boundary line, $2W = 12$, so $s_2 = 0$. Increasing W further would take you out of the feasible region. For example, increasing W to $W = 9$ would take you to the infeasible point (0, 9) which lies on the constraint boundary line, $3D + 2W = 18$ (so $s_3 = 0$). Therefore, in order to stop at the adjacent corner point, the *minimum ratio* is used to determine the leaving basic variable.

When applying the rule for determining the leaving basic variable, one rare possibility is that none of the equations have a strictly positive coefficient for the entering basic variable. Having this possibility occur means that both the entering basic variable and *P* can be increased indefinitely without ever leaving the feasible region. This circumstance is the one described in Key Fact 9 in Section 14.1.

Solving for the New Basic Feasible Solution

After determining the entering basic variable and the leaving basic variable, the final step of an iteration is to solve for the new basic feasible solution.

Actually, when we finished determining the leaving basic variable, we already were able to look ahead and see what this new basic feasible solution would be. In the third paragraph from the end of the preceding subsection, we identified this solution as $D = 0$, $W = 6$, $s_1 = 4$, $s_2 = 0$, $s_3 = 6$. What the current step does is to convert the system of equations into a form that (1) clearly exhibits the new basic feasible solution and (2) enables the optimality test and (if needed) the next iteration to be performed on this new solution. This form for the system of equations is called *proper form from Gaussian elimination* (as introduced earlier). The procedure used to obtain this form is called **Gaussian elimination**. Gaussian elimination is a standard algebraic procedure for finding a simultaneous solution of a system of linear equations.

Earlier in this section, we showed the initial system of equations as follows:

(0)	P - 300	D -	$500W$				= 0
(1)		D		+ $\mathbf{s_1}$			= 4
(2)			$2W$		+ $\mathbf{s_2}$		= 12
(3)		$3D$ +	$2W$			+ $\mathbf{s_3}$	= 18,

where the *initial* basic variables (s_1, s_2, s_3) now are shown in bold type. We also gave the following requirements to have *proper form from Gaussian elimination*.

REQUIREMENTS FOR PROPER FORM FROM GAUSSIAN ELIMINATION

1. Equation 0 does not contain any basic variables.
2. Each of the other equations contains exactly one basic variable.
3. An equation's one basic variable has a coefficient of 1. (This coefficient is not shown explicitly.)
4. An equation's one basic variable does not appear in any other equation (so each basic variable appears in exactly one equation).

Note how these four requirements are satisfied for the initial systems of equations when the basic variables were s_1, s_2 , s_3. However, for the new basic feasible solution, W has replaced s_2 as a basic variable. This system of equations no longer is in proper form from Gaussian elimination in terms of the new set of basic variables (s_1, W, s_3). Requirement 1 is violated because equation 0 contains W. Requirement 2 is violated because equation 3 contains both W and s_3. The fact that W has a coefficient of 2 in equation 2 violates requirement 3. Requirement 4 is violated because W appears in two equations besides equation 2.

To restore proper form, Gaussian elimination performs algebraic operations to accomplish two kinds of changes in the system of equations. First, since the entering basic variable W is replacing the leaving basic variable s_2 as the one basic variable in equation 2, we need to obtain a coefficient of 1 for W in this equation. This is accomplished by *dividing* equation 2,

$$(2) \qquad 2\,\mathbf{W} \quad + s_2 \quad = 12,$$

by 2, the coefficient of W. This yields

$$(2) \qquad \mathbf{W} \quad + 0.5\, s_2 = 6.$$

Second, W must be eliminated from the other equations (0 and 3) in which it appears. This is accomplished by *subtracting* the appropriate multiple of the new equation 2 from each of these other equations. The appropriate multiple is the coefficient of W in the other equation. (When this coefficient is *negative*, subtracting this multiple is equivalent to *adding* the *absolute value* of this multiple.)

In particular, consider equation 0. Since its coefficient of W is -500, we want to *add* 500 times the new equation 2. (This is equivalent to subtracting (-500) times this equation.) 500 times the new equation 2 is

$$500 \text{ (new eq. 2):} \qquad \mathbf{500\,W} + 250\, s_2 = 3{,}000.$$

Therefore, the complete algebraic operations are

Old eq. 0:	P - 300 D	- **500 W**		= 0
+ 500 (new eq. 2):	+(	**500 W**	+ 250 s_2	= 3,000)
= new eq. 0:	P - 300 D		+ 250 s_2	= 3,000.

Since W has a coefficient of 2 in equation 3, the algebraic operations needed to eliminate this coefficient are

$$\begin{array}{rlllll} \text{Old eq. 3:} & 3D & + \mathbf{2\,W} & & + s_3 & = 18 \\ -2\ \text{(new eq. 2):} & -(& \mathbf{2\,W} & + s_2 & & = 12) \\ \hline = \text{new eq. 3:} & 3D & & - s_2 & & = \ 6. \end{array}$$

Combining all these results gives the following new system of equations.

$$\begin{array}{llllllll} (0) & P - 300\,D & & & + 250\ s_2 & & = & 3{,}000 \\ (1) & D & & + \mathbf{s_1} & & & = & 4 \\ (2) & & \mathbf{W} & & + 0.5\ s_2 & & = & 6 \\ (3) & 3D & & & -\ s_2 & + \mathbf{s_3} & = & 6 \end{array}$$

Note that this system of equations does satisfy all the requirements to be in proper form from Gaussian elimination for the current set of basic variables (shown in bold type). Therefore, you can immediately read the value of each basic variable from the right-hand side of its equation to obtain the new basic feasible solution summarized below.

Second Basic Feasible Solution
Nonbasic variables: $D = 0$, $s_2 = 0$
Basic variables: $s_1 = 4$, $W = 6$, $s_3 = 6$
Value of objective function: $P = 3{,}000$

Furthermore, this system of equations now is in the form needed to perform the optimality test (no basic variables in equation 0) and the next iteration.

Summary of the Procedure for Solving for the New Basic Feasible Solution:

1. For the equation containing the leaving basic variable, divide that equation by the coefficient of the entering basic variable. The entering basic variable now becomes the one basic variable contained in this equation.
2. Now subtract the appropriate multiple of this equation from each of the other equations that contain the entering basic variable. The appropriate multiple is the coefficient of the entering basic variable in the other equation.
3. The system of equations now is in *proper form from Gaussian elimination*, so read the value of each basic variable from the right-hand side of its equation to obtain the new basic feasible solution.

Completing the Example

You now have seen all the parts of the simplex method (the initialization step, the optimality test, and the three steps of an iteration) in action in the Wyndor problem. To finish solving the problem, the simplex method now recycles through these parts (except for the initialization step) repeatedly until it completes its pilgrimage by reaching an optimal solution. We briefly outline the remainder of this pilgrimage below, using the summary (or rule) already given for each part of the simplex method.

Optimality Test for the Second Basic Feasible Solution:

Since the current equation 0,

$$(0) \qquad P - 300D + 250\, s_2 = 3{,}000,$$

has one *negative coefficient* (-300 for *D*), we conclude that the second basic feasible solution is *not optimal*. Because this equation contains no basic variables (the first requirement for proper form from Gaussian elimination), this negative coefficient implies that *P can* be increased by increasing the nonbasic variable *D* from its current value of *D* = 0, so another iteration is needed.

Determining the Entering Basic Variable:

Since the nonbasic variable *D* is the only variable with a *negative coefficient* in the current equation 0, *D* is the entering basic variable for the second iteration.

Determining the Leaving Basic Variable:

Referring back to the preceding subsection, look at the current system of equations that gives the current (second) basic feasible solution. The entering basic variable *D* has a *strictly positive coefficient* (1 and 3) in equations 1 and 3, so for each of these equations we take the ratio of the right-hand side to the coefficient.

$$(1) \quad \frac{4}{1} = 4$$

$$(3) \quad \frac{6}{3} = 2 \quad \leftarrow \text{ minimum}$$

Since equation 3 has a *minimum ratio*, we select the basic variable in this equation (s_3) to be the leaving basic variable.

Solving for the New Basic Feasible Solution:

Since equation 3 is the one containing the leaving basic variable, we begin by dividing that equation by 3, the coefficient of the entering basic variable (*D*) in that equation. This yields

$$(3) \quad \mathbf{D} - \frac{1}{3}s_2 + \frac{1}{3}s_3 = 2$$

Second, to eliminate *D* from the other equations that contain it (equations 0 and 1), we now subtract the appropriate multiple of this new equation 3 from these other equations. For equation 0, this appropriate multiple is -300, the coefficient of *D* in this equation. (Equivalently, this amounts to *adding* the multiple 300 of equation 3 to equation 0.) For equation 1, this appropriate multiple is 1, since this is the coefficient of *D* in this equation. After completing these algebraic operations to restore *proper form from Gaussian elimination* in terms of the new set of basic variable shown in bold type (s_1, *W*, *D*), the system of equations becomes

$$
\begin{array}{llllllll}
(0) & P & & & 150s_2 & +100s_3 & = & 3{,}600 \\
(1) & & & \boldsymbol{s_1} & +\frac{1}{3}s_2 & -\frac{1}{3}s_3 & = & 2 \\
(2) & & \mathbf{W} & & +\frac{1}{2}s_2 & & = & 6 \\
(3) & \mathbf{D} & & & -\frac{1}{3}s_2 & +\frac{1}{3}s_3 & = & 2
\end{array}
$$

Therefore, the value of each basic variable in the new basic feasible solution now can be read from the right-hand side of its equation, as summarized below.

Third Basic Feasible Solution:
Nonbasic variables: $s_3 = 0$, $s_2 = 0$
Basic variables: $s_1 = 2$, $D = 6$, $W = 2$
Value of objective function: $P = 3{,}600$

Optimality Test for the Third Basic Feasible Solution:

Since the current equation 0 given above has *no* negative coefficients, the current basic feasible solution is *optimal*, so the simplex method is finished.

The Tabular Form of the Simplex Method

After understanding the logic of the simplex method, some people prefer to switch to a more compact form of this method for solving small problems by hand. This more compact form is called the **tabular form** of the simplex method. The tabular form performs exactly the same steps as the "algebraic form" of the simplex method presented in this section, but records the information more compactly. Instead of writing down each system of equations in full detail, the tabular form uses a **simplex tableau** to record only the essential information. The simplex tableau is simply a table, where each row gives the essential information for one of the equations. For a particular row (equation), the various columns of the table record (1) the basic variable appearing in that equation, (2) the coefficients of the variables, and (3) the constant on the right-hand side of the equation. This saves writing the symbols for the variables in each of the equations. More importantly, it permits highlighting the numbers involved in the arithmetic calculations and recording the computations compactly.

We will not describe the tabular form further here. However, if your instructor presents this form, keep in mind that it is using exactly the same procedure in a shorthand way as the algebraic form outlined in this section.

REVIEW QUESTIONS

1. What is the three-step outline of an iteration of the simplex method?
2. What is meant by the *entering basic variable* during an iteration? What is the rule for determining the entering basic variable?
3. What is meant by the *leaving basic variable* during an iteration? What is the rule for determining the leaving basic variable?
4. What is accomplished during the initialization step?
5. What is the rule for the optimality test?
6. What are the requirements for proper form from Gaussian elimination?
7. How does the tabular form of the simplex method differ from the algebraic form?

14.8 COMPUTER IMPLEMENTATION OF THE SIMPLEX METHOD

If the electronic computer had never been invented, you undoubtedly would have never heard of linear programming and the simplex method. Even though it is possible to apply the simplex method by hand to solve tiny linear programming problems, the calculations involved are just too tedious to do this on a routine basis. However, the simplex method is ideally suited for execution on a computer. It is the *computer revolution* that has made possible the widespread application of linear programming in recent decades.

Computer codes for the simplex method, such as the one in the Excel Solver, now are widely available for essentially all modern computer systems. In fact, a considerable number of powerful software packages for linear programming have been developed by various software development companies.

The simplex method is used routinely to solve large linear programming problems. For example, a problem with many thousands of functional constraints and an even larger number of decision variables is not considered particularly large for a mainframe computer or for some workstations, and personal computers are not lagging too far behind. Occasionally, even vastly larger problems with millions of functional constraints and decision variables now are being successfully solved on fast computers, but not on a routine basis yet. The primary limiting factor is the number of functional constraints, since the number of decision variables does not affect the computation time very much.

With large linear programming problems, it is inevitable that some mistakes and faulty decisions will be made initially in formulating the model and inputting it into the computer. Therefore, a thorough process of testing and refining the model (**model validation**) is needed. The usual end-product is not a single static model that is solved once by the simplex method. Instead, the management science team and management typically consider a long series of variations on a basic model to examine different scenarios as part of the *what-if analysis* discussed in Chapter 5.

Imagine trying to do all this—both formulate a basic model and then repeatedly modify it for what-if analysis—when the model has thousands of functional constraints and decision variables. You certainly would not accomplish this by trying to fill in the *millions* of cells needed on an Excel spreadsheet. A much more powerful software package would be needed. One key requirement for this package is that it must help perform “model management." **Model management** encompasses a variety of activities, including *formulating* the model, *inputting* the model into the computer, *modifying* the model, *analyzing solutions* from the model, and *presenting results* in the language of management. The package commonly includes a **modeling language** to efficiently generate the model from existing databases. Such packages are widely available today and are frequently used by management scientists.

Until the mid-1980s, linear programming problems were solved almost exclusively on *mainframe computers*. Since then, there has been an explosion in the capability of doing linear programming on both *workstations* and *personal computers*.

REVIEW QUESTIONS

1. The simplex method routinely can solve linear programming problems of what size on a fast computer?
2. When dealing with large problems, the software package needs to help perform what kinds of model management activities?

14.9 THE INTERIOR-POINT APPROACH TO SOLVING LINEAR PROGRAMMING PROBLEMS

The most dramatic new development in management science during the 1980s was the discovery of the interior-point approach to solving linear programming problems. This discovery was made in 1984 by a young mathematician at AT&T Bell Laboratories, Narendra Karmarkar, when he successfully developed a new algorithm for linear programming with this kind of approach. Although this particular algorithm experienced only mixed success in competing with the simplex method, the key solution concept described below appeared to have great potential for solving *huge* linear programming problems beyond the reach of the simplex method. Many top researchers subsequently worked on modifying Karmarkar's algorithm to fully tap this potential. Much progress was made and a number of powerful algorithms using the interior-point approach have been developed. Today, the more powerful software packages that are designed for solving really large linear programming problems (such as CPLEX) include at least one algorithm using the interior-point approach along with the simplex method. As research continues on these algorithms, their computer implementations continue to improve.

Now let us look at the key idea behind Karmarkar's algorithm and its subsequent variants that use the interior-point approach.

The Key Solution Concept

Although radically different from the simplex method, Karmarkar's algorithm does share a few of the same characteristics. It is an *iterative* algorithm. It gets started by identifying a feasible *trial solution*. At each *iteration*, it moves from the current trial solution to a *better* trial solution in the feasible region. It then continues this process until it reaches a trial solution that is (essentially) optimal.

The big difference lies in the nature of these trial solutions. For the simplex method, the trial solutions are *corner points*, so all movement is along edges on the *boundary* of the feasible region. For Karmarkar's algorithm, the trial solutions are **interior points**, i.e., points *inside* the boundary of the feasible region. For this reason, Karmarkar's algorithm and its variants are referred to as **interior-point algorithms**. (The name *barrier algorithm* now is also commonly used for such algorithms.)

> **Solution Concept**: Interior-point algorithms shoot through the *interior* of the feasible region toward an optimal solution instead of taking a less direct path around the boundary of the feasible region.

Your MS Courseware includes a routine for generating the output of a typical interior-point algorithm for selected problems. To illustrate, Figure 14.17 shows the path followed by this algorithm when it is applied to the Wyndor problem, starting from the initial trial solution (1, 2). Note how all the trial solutions (dots) on this path are *inside* the boundary of the feasible region until the path reaches the optimal solution (2, 6). Contrast this path with the path followed by the simplex method (see Figure 14.13) around the *boundary* of the feasible region from (0, 0) to (0, 6) to (2, 6).

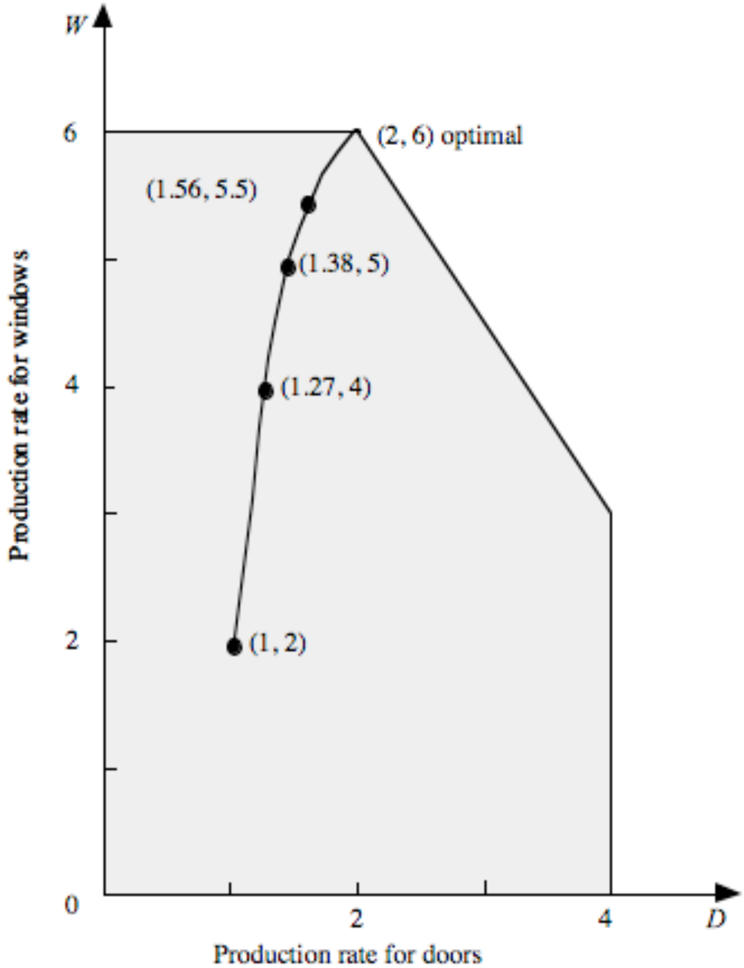


Figure 14.17 The curve from (1, 2) to (2, 6) shows a typical path followed by in interior-point algorithm, right through the *interior* of the feasible region for the Wyndor problem.

Figure 14.18 shows the output of this algorithm for this problem when using 15 iterations to obtain 15 new trial solutions. Note how the successive trial solutions keep getting closer and closer to the optimal solution, but never literally get there. However, the deviation becomes so infinitesimally small that the final trial solution can be taken to be the optimal solution for all practical purposes.

	A	B	C	D	E
1	Interior-Point Algorithm for Wyndor Problem				
2					
3		Iteration	D	W	Profit
4		0	1.00000	2.00000	$1,300.00
5		1	1.27298	4.00000	$2,381.89
6		2	1.37744	5.00000	$2,913.23
7		3	1.56291	5.50000	$3,218.87
8		4	1.80268	5.71816	$3,399.89
9		5	1.92134	5.82908	$3,490.94
10		6	1.96639	5.90595	$3,542.90
11		7	1.98385	5.95199	$3,571.15
12		8	1.99197	5.97594	$3,585.56
13		9	1.99599	5.98796	$3,592.78
14		10	1.99799	5.99398	$3,596.39
15		11	1.99900	5.99699	$3,598.19
16		12	1.99950	5.99850	$3,599.10
17		13	1.99975	5.99925	$3,599.50
18		14	1.99987	5.99962	$3,599.77
19		15	1.99994	5.99981	$3,599.89

Figure 14.18 The output of the interior-point algorithm in your MS Courseware when performing 15 iterations starting from the initial trial solution (D, W) = (1, 2).

Comparison with the Simplex Method

Interior-point algorithms are far more complicated than the simplex method. Considerably more extensive computations are required for each iteration to find the next trial solution. Therefore, the computer time per iteration for an interior-point algorithm is many times longer than for the simplex method.

For fairly small problems, the number of iterations needed by an interior-point algorithm and the simplex method tend to be somewhat comparable. For example, on a problem with 10 functional constraints, roughly 20 iterations would be typical for either kind of algorithm. Consequently, on problems of similar size, the total computer time for an interior-point algorithm will tend to be many times longer than for the simplex method.

On the other hand, a key advantage of interior point algorithms is that large problems do not require many more iterations than small problems, For example, a problem with 10,000 functional constraints probably will require well under 100 iterations. Even considering the very substantial computer time per iteration needed for a problem of this size, such a small number of iterations makes the problem reasonably easy to solve. By contrast, the simplex method might need 20,000 iterations, and so might struggle to finish within a reasonable amount of computer time. Therefore, interior-point algorithms often are faster than the simplex method for such huge problems.

The reason for this very large difference in the number of iterations on huge problems is the difference in the paths followed. At each iteration, the simplex method moves from the current corner point to an adjacent corner point along an edge on the boundary of the feasible region. Huge problems have an astronomical number of corner points. The path from the initial corner point to an optimal solution may be a very circuitous one around the boundary, taking numerous small steps to the next adjacent corner point. By contrast, an interior-point algorithm bypasses all this by shooting through the interior of the feasible region toward an optimal solution. Adding more functional constraints adds more constraint boundaries to the feasible region, but has little effect on the number of trial solutions needed on this path through the interior. This makes it possible for interior-point algorithms to solve problems with a huge number of functional constraints.

A final key comparison concerns the ability to perform the various kinds of what-if analysis described in Chapter 5. The simplex method and its variants are very well suited and are widely used for this kind of analysis by providing the kind of information given in Excel Solver's sensitivity report. Unfortunately, the interior-point approach currently has limited capability in this area (although research progress is being made). Given the great importance of what-if analysis, this is a crucial drawback of interior-point algorithms. However, we point out below how the simplex method can be combined with the interior-point approach to overcome this drawback.

The Complementary Roles of the Simplex Method and the Interior-Point Approach

We anticipate that the simplex method will continue to be the standard algorithm for the routine use of linear programming. It should continue to be the most efficient algorithm for problems of moderate size (say, less than 10,000 functional constraints) and occasionally for considerably larger problems as well. However, the interior-point approach should be faster than the simplex method for many really big problems. As the size grows huge (say, hundreds of thousands of functional constraints), the interior-point approach sometimes may be the only one capable of solving the problem with today's computers. It is dangerous to generalize, however, since recent research advances have enabled the latest state-of-the-art computer packages for linear programming to use the simplex method and its variants to solve some problems with hundreds of thousands, or even millions, of functional constraints and decision variables!

To overcome the drawback of the interior-point approach that it has limited capability for performing what-if analysis, researchers have developed procedures for switching over to the simplex method after an interior-point algorithm has finished. Recall that the trial solutions obtained by an interior-point algorithm keep getting closer and closer to an optimal solution (the best corner point), but never quite get there. Therefore, a switching procedure requires identifying a corner point that is very close to the final trial solution.

For example, by looking at Figures 14.17 and 14.18, it is easy to see that the final trial solution in the latter figure is very near the corner point (2, 6). Unfortunately, on problems with thousands of decision variables (so no graph is available), identifying a nearby corner point is a very challenging and time-consuming task. However, good progress has been made in developing procedures to do this.

Once this nearby corner point has been found, the optimality test for the simplex method is applied to check whether it actually is the corner point that is an optimal solution. If it is not optimal, some iterations of the simplex method are conducted to move from this corner point to an optimal solution. Generally, only a very few iterations (perhaps one) are needed because the interior-point algorithm has brought us so close to an optimal solution. Therefore, doing these iterations should be quite fast even on problems that are too huge to be solved from scratch. After actually reaching an optimal solution, the simplex method and its variants then are applied to help perform what-if analysis.

REVIEW QUESTIONS

1. Who developed a new linear programming algorithm using the interior-point approach in 1984?
2. Do today's more powerful software packages for linear programming typically include an interior-point algorithm? Do they include the simplex method?
3. What is the key solution concept for interior-point algorithms?
4. How do interior-point algorithms and the simplex method compare regarding the computer time needed per iteration?
5. How do interior-point algorithms and the simplex method compare regarding the number of iterations needed on small problems (say, 10 functional constraints)? On huge problems (say, 10,000 functional constraints)? What accounts for the difference on huge problems?
6. How do interior-point algorithms and the simplex method compare regarding their capability for performing what-if analysis?
7. What drawback of an interior-point algorithm can be overcome by switching over to the simplex method after the interior-point algorithm is finished?
8. Switching over in this way begins by identifying what kind of solution?

14.10 SUMMARY

The simplex method continues to be the main procedure for routinely solving linear programming problems. In addition, good progress has been made on developing powerful interior-point algorithms that often are more efficient than the simplex method for solving very big problems, including even huge problems with hundreds of thousands, or even millions, of functional constraints and decision variables.

The simplex method is an algebraic procedure that focuses on *corner points* of the feasible region, since the best corner point must be an optimal solution (unless the problem has no optimal solutions). At each *iteration*, the simplex method moves along an *edge* of the boundary of the feasible region from the current corner point to a better *adjacent corner point*. It continues these iterations until it reaches a corner point that has no *better* adjacent corner points. The *optimality test* then says that this corner point is an optimal solution.

The simplex method uses supplementary variables to convert inequality constraints into equations called constraint boundary equations. By setting certain decision variables and supplementary variables to zero (the nonbasic variables), the simplex method solves for a basic feasible solution that corresponds to a particular corner point of the feasible region by solving this system of constraint boundary equations for the remaining variables (the basic variables). To move to a better adjacent corner point, the simplex method performs an iteration that consists of determining the entering basic variable (the nonbasic variable that will become a basic variable), then determining the leaving basic variable (the basic variable that will become a nonbasic variable), and then solving for the new basic feasible solution.

Interior-point algorithms follow a very different path than the simplex method. They shoot through the *interior* of the feasible region toward an optimal solution. Even on huge problems, this path enables these algorithms to reach what is essentially an optimal solution after a relatively small number of iterations (trial solutions). This characteristic explains why interior-point algorithms often is considerably faster than the simplex method on huge problems despite being slower on smaller problems.

Software packages based on the simplex method and its extensions are widely available for mainframes, workstations, and personal computers. Packages that include the interior-point approach are commonly available as well.

The simplex method and interior-point algorithms now play complementary roles for solving linear programming problems. For problems of small to moderate size (even with many thousands of functional constraints and decision variables), the simplex method should be used. For much larger problems, an interior-point algorithm may be the best option. It also would be reasonable on such problems to use an interior-point algorithm to approach an optimal solution and then to switch over to the simplex method to finish finding this solution and to perform what-if analysis.

Glossary

Adjacent corner points: Corner points that share all but one of the same constraint boundaries. (Section 14.2)

Algorithm: A systematic solution procedure for solving a mathematical problem (frequently on a computer). (Section 14.3)

Basic feasible solution: A solution that the simplex method obtains by setting the nonbasic variables equal to zero and solving for the basic variables. The resulting solution is a corner point that has been augmented by the values of the supplementary variables. (Section 14.6)

Basic variable: One of the variables that the simplex method solves for after setting the nonbasic variables equal to zero. (Section 14.6)

Best adjacent corner point criterion: When applying the simplex method graphically, this criterion chooses the next corner point by selecting the adjacent corner point that gives the most favorable value of the objective function. (Section 14.4)

Best rate of improvement : A simplex method criterion for choosing the next corner point by selecting the adjacent corner point that gives the best rate of improvement in the value of the objective function per unit of distance moved. (Sections 14.2 and 14.4)

Constraint boundary equation: The equation that gives the boundary of a $\leq$ or $\geq$ constraint by replacing the $\leq$ or $\geq$ by an =. (Section 14.6)

Corner point: A point that lies at a corner of the feasible region. From an algebraic viewpoint, if the problem has n decision variables, a corner point is a feasible solution that satisfies n constraint boundary equations simultaneously. (Section 14.1)

Edge of the feasible region: A line segment on the boundary of the feasible region that connects a pair of adjacent corner points. (Section 14.2)

Entering basic variable: The nonbasic variable that is converted into a basic variable during an iteration of the simplex method. (Section 14.7)

Enumeration-of-corner-points method: A method for finding an optimal solution by enumerating all the corner points and choosing the one with the best value of the objective function. (Section 14.1)

Gaussian elimination: The algebraic procedure used by the simplex method for finding a simultaneous solution (the current basic feasible solution) of a system of linear equations. (Section 14.7)

Graphical simplex method: A streamlined procedure for applying the simplex method graphically when the problem has just two decision variables. (Section 14.4)

Initialization step: The procedure for getting set up to start the iterations of an iterative algorithm. (Sections 14.3 and 14.7)

Interior point: A point inside the boundary of the feasible region. (Section 14.9)

Interior-point algorithm: An iterative algorithm that moves through the interior (inside the boundary) of the feasible region. (Section 14.9)

Iteration: A prescribed series of steps (e.g., to find the next trial solution) to be repeated over and over again by an iterative algorithm until the desired result (e.g., an optimal solution) has been obtained. (Sections 14.3 and 14.7)

Iterative algorithm: An algorithm that repeats iterations until the desired result has been obtained. (Section 14.3)

Leaving basic variable: The basic variable that is converted into a nonbasic variable during an iteration of the simplex method. (Section 14.7)

Minimum ratio rule: The rule used by the simplex method for determining the leaving basic variable for the current iteration. (Section 14.7)

Model management: The management of a large mathematical model being formulated, inputted, modified, and analyzed in a computer. (Section 14.8)

Model validation: The process of testing and refining a model to ensure its validity. (Section 14.8)

Modeling language: A computer language designed to expedite the formulation and generation of a mathematical model from exiisting databases. (Section 14.8)

Nonbasic variable: A decision variable or supplementary variable that the simplex method currently has set equal to zero. (Section 14.6)

Optimality test: The test used by an iterative algorithm to determine whether a trial solution is optimal. (Sections 14.3 and 14.7)

Postoptimality analysis: Analysis done after finding an optimal solution for the initial version of the model. (Section 14.1)

Proper form from Gaussian elimination: The form for the system of equations that the simplex method uses to display the current basic feasible solution. (Section 14.7)

Simplex method: A standard method for finding an optimal solution (and for providing information for what-if analysis) for a linear programming problem. (Section 14.1)

Simplex tableau: A special kind of table used for performing the tabular form of the simplex method. (Section 14.7)

Slack variable: A supplementary variable that gives the slack by which the right-hand side of a $\leq$ constraint exceeds the left-hand side. (Section 14.6)

Stopping rule: The rule used by an iterative algorithm to determine whether the desired result has been obtained. (Section 14.3)

Surplus variable: A supplementary variable that gives the surplus by which the left-hand side of a $\geq$ constraint exceeds the right-hand side. (Section 14.6)

Tabular form of the simplex method: A form of the simplex method that uses a series of tables called simplex tableaux to perform and record all the necessary steps. (Section 14.7)

What-if analysis: Analysis addressing questions about what would happen to the conclusions from a model if future conditions turn out to differ from what has been assumed in the model. (Section 14.1)

Learning Aids for This Chapter in Your MS Courseware

Chapter 14 Excel Files:

Wyndor Examples

Interior-Point Algorithm

Problems

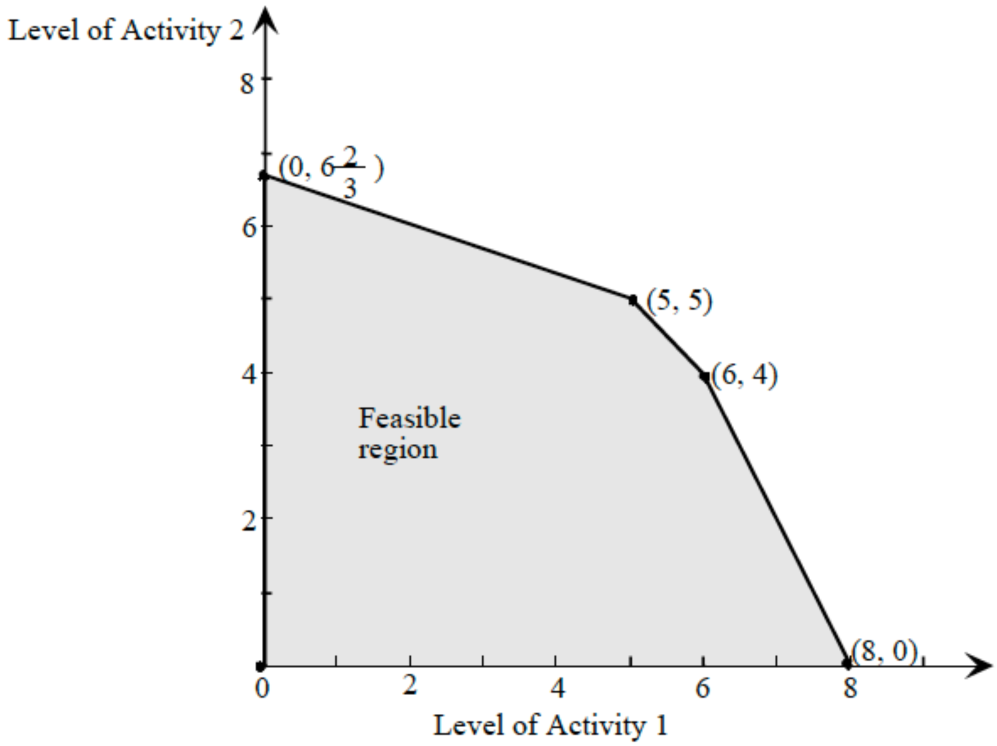
The symbols to the left of some of the problems (or their parts) have the following meaning:

E*: Use Excel.

R*: Use the routine listed above.

An asterisk on the problem number indicates that at least a partial answer is given at the end of all the problems.

14.1. A certain linear programming model involving two activities has the feasible region shown below.



The objective is to maximize the total profit from the two activities. The unit profit for activity 1 is \$1,000 and the unit profit for activity 2 is \$2,000. Apply the enumeration-of-corner-points method to find an optimal solution.

14.2. Reconsider the Profit & Gambit advertising-mix problem presented in Section 2.7. Refer to its feasible region —including the three corner points at (0, 9), (4, 3), and (8, 3) —shown in Figure 2.22. Use the enumeration-of-corner-points method to show that (4, 3) is the optimal solution.

14.3. Consider a resource-allocation problem having the following data:

Resource Usage Per Unit of Each Activity

Resource	Activity 1	Activity 2	Amount of Resource Available
A	1	0	5
B	0	1	4
C	2	1	9
D	3	4	21
Unit profit	$30	$20	

The objective is to determine the number of units of each activity to undertake so as to maximize the total profit.

E* *a*. Formulate a spreadsheet model for this problem.

b. Graph the feasible region. Use this graph to identify all the corner points.

E* *c*. Use the spreadsheet to determine the total profit for each of these corner points. Then use this information to identify an optimal solution.

E* *d*. Use Solver (which applies the simplex method) to confirm this optimal solution.

e. Use the graphical method to confirm this optimal solution.

14.4. Follow the instructions of Problem 14.3 for the following resource-allocation problem having the objective of maximizing the total profit from the two activities.

Resource Usage Per Unit of Each Activity

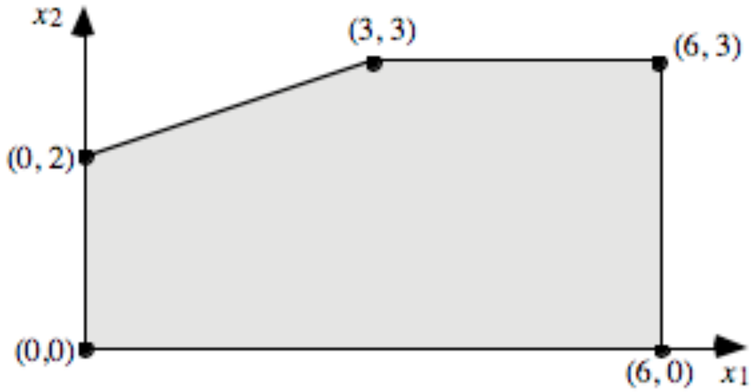
Resource	Activity 1	Activity 2	Amount of Resource Available
A	5	3	30
B	2	3	21
C	0	1	6
Unit profit	$300	$200	

14.5. Consider the five variations of the Wyndor problem given in Table 14.1.

a. Redraw the feasible region for this problem, as given in Figure 2.6. Then find and draw the *optimal objective function line* for each of these variations in order to verify that the optimal solution given in the table actually is optimal.

b. For each variation, verify that the optimal solution given in the table actually is optimal by applying the *enumeration-of-corner-points method*.

14.6. The shaded area in the following graph represents the feasible region of a linear programming problem whose objective function is to be maximized, where x_1 and x_2 represent the level of the two activities.



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.

14.7. Reconsider the preceding problem.

a. Use the graph of the feasible region to identify all the constraints for the linear programming model.

b. Construct a table like Table 14.3 that gives examples of objective functions that together give all the possibilities for multiple optimal solutions.

c. Suppose that the objective function is Profit $= -x_1 + 5x_2$. Use the enumeration-of-corner-points method to find the optimal solution.

d. Now suppose that the objective function is Profit $= -x_1 + 2x_2$. Use the enumeration-of-corner-points method to find all the optimal solutions.

14.8. Consider the following resource-allocation problem that has the objective of maximizing the total profit from the two activities.

Resource Usage Per Unit of Each Activity

Resource	Activity 1	Activity 2	Amount of Resource Available
A	5	4	20
B	6	9	30
C	2	5	15
Unit profit	$20 million	$30 million	

E* *a.* Formulate and solve a spreadsheet model for this problem. Also obtain the sensitivity report.

b. Explain why this sensitivity report indicates that the problem has other optimal solutions in addition to the one found in part *a*. Alter the spreadsheet slightly as needed to find another optimal solution with Solver.

c. Since the optimal solutions found in parts *a* and *b* are both corner points (Solver only considers corner points), describe how these solutions can be used to find other optimal solutions.

d. Use the graphical method to find all optimal solutions.

14.9. Follow the instructions of Problem 14.8 for the following resource-allocation problem, which again has the objective of maximizing the total profit from the two activities.

Resource Usage Per Unit of Each Activity

Resource	Activity 1	Activity 2	Amount of Resource Available
A	15	5	300
B	10	6	240
C	8	12	450
Unit profit	$500	$300	

14.10. Consider the following linear programming model in algebraic form, where x_1 and x_2 are the decision variables representing the levels of the two activities, and where the coefficients of these decision variables in the objective function (c_1 and c_2) have not yet been ascertained.

Maximize Profit = $c_1x_1 + c_2x_2$,

subject to

$$2x_1 + x_2 \leq 11$$

$$-x_1 + 2x_2 \leq 2$$

and

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

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

14.11. Consider the following cost-benefit-tradeoff problem.

Benefit Contribution Per Unit of Each Activity

	Activity		Minimum Acceptable
Benefit	1	2	Level
1	-2	1	1
2	1	-2	1
Unit cost	\$5,000	\$7,000	

E* *a.* Formulate a spreadsheet model for this problem.

E* *b.* Use Solver (which applies the simplex method) to find that the model has no feasible solutions.

c. Use the graphical method to demonstrate that the model has no feasible solutions.

14.12. The following linear programming model (in algebraic form) has one resource constraint, one benefit constraint, and two decision variables (A_1 and A_2) representing the levels of the two activities.

Maximize Profit = $90A_1 + 70A_2$,

subject to

Resource: $2A_1 + A_2 \leq 2$ (amount available)

Benefit: $A_1 - A_2 \geq 2$ (minimum acceptable level)

and

$$A_1 \geq 0, \quad A_2 \geq 0.$$

Follow the instructions of Problem 14.11 for this model.

14.13. Suppose that the following constraints have been provided for a linear programming model with decision variables x_1 and x_2.

$-x_1 + 3x_2 \leq 30$
$-3x_1 + x_2 \leq 30$

and

$x_1 \geq 0, x_2 \geq 0$

a. Demonstrate graphically that the feasible region is unbounded.

b. If the objective is to maximize Profit = $-x_1 + x_2$ 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 Profit = $x_1 - x_2$.

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?

E* *e.* Select an objective function for which this model has no optimal solution. Formulate the corresponding model on a spreadsheet. Then apply Solver. Explain the meaning of the message given by Solver.

14.14. Follow the instructions of Problem 14.13 when the constraints are the following:

$2x_1 - x_2 \leq 20$
$x_1 - 2x_2 \leq 20$

and

$x_1 \geq 0, x_2 \geq 0$

14.15. Consider the following algebraic form of a linear programming model with decision variables x_1 and x_2.

Maximize Profit = $x_1 + 2x_2$,

subject to

$x_1 \leq 2$

$x_2 \leq 2$

$x_1 + x_2 \leq 3$

and

$x_1 \geq 0, x_2 \geq 0$

a. Plot the feasible region and circle all the corner points.

b. For each corner point, identify the pair of constraint boundary equations that it satisfies.

c. For each corner point, use this pair of constraint boundary equations to solve algebraically for the values of x_1 and x_2 at the corner point.

d. For each corner point, identify its adjacent corner points.

e. For each pair of adjacent corner points, identify the constraint boundary they share by giving its equation.

14.16.* Reconsider Problem 2.6 about investing $6,000 in entrepreneurial ventures.

a. Use the graphical method to solve this problem. Circle all the corner points on the graph.

b. For each corner point, identify its adjacent corner points.

14.17. Consider the following algebraic form of a linear programming model with decision variables x_1 and x_2.

Maximize Profit = $3x_1 + 2x_2$,

subject to

$$2x_1 + x_2 \leq 6$$
$$x_1 + 2x_2 \leq 6$$

and

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

a. Use the graphical method to solve this model. Circle all the corner points on the graph.

b. Fill out a table like Table 14.5 for these corner points.

c. Also fill out a table like Table 14.6.

d. Use the enumeration-of-corner-points method to solve the model.

e. Describe graphically what the simplex method does step by step to solve the model.

14.18. Follow the instructions of Problem 14.17 for the following model:

Maximize Profit = $x_1 + 2x_2$,

subject to

$$x_1 + 3x_2 \leq 8$$
$$x_1 + x_2 \leq 4$$

and

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

14.19. Follow the instructions of Problem 14.17 for the following model:

Maximize Profit = $3x_1 + 2x_2$,

subject to

$$x_1 \leq 4$$
$$x_1 + 3x_2 \leq 15$$
$$2x_1 + x_2 \leq 10$$

and

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

14.20.* Describe graphically what the simplex method does step by step to solve the following model:

Maximize Profit = $10x_1 + 20x_2$,

subject to

$$-x_1 + 2x_2 \leq 15$$
$$x_1 + x_2 \leq 12$$
$$5x_1 + 3x_2 \leq 45$$

and

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

14.21. Describe graphically what the simplex method does step by step to solve the following model:

Maximize Profit = $2x_1 + 3x_2$,

subject to

$$-3x_1 + x_2 \leq 1$$
$$4x_1 + 2x_2 \leq 20$$
$$4x_1 - x_2 \leq 10$$
$$-x_1 + 2x_2 \leq 5$$

and

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

14.22. Describe graphically what the simplex method does step by step to solve the following model.

Minimize Cost = $15x_1 + 20x_2$,

subject to

$$x_1 + 2x_2 \geq 10$$
$$2x_1 - 3x_2 \leq 6$$
$$x_1 + x_2 \geq 6$$

and

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

14.23. Describe graphically what the simplex method does step by step to solve the following model.

Minimize Cost = $5x_1 + 7x_2$,

subject to

$$2x_1 + 3x_2 \geq 42$$
$$3x_1 + 4x_2 \geq 60$$
$$x_1 + x_2 \geq 18$$

and

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

14.24. Label each of the following statements about linear programming problems as True or False, and then justify your answer.

a. For minimization problems, if the objective function evaluated at a corner point is at least as small as its value at every adjacent corner point, then that solution is optimal.

b. Only corner points can be optimal, so the number of optimal solutions cannot exceed the number of corner points.

c. An optimal corner point may have an adjacent corner point that also is optimal (the same value of the objective function).

14.25. Each of the following statements is true under most circumstances, but not always. In each case, indicate when the statement will not be true and why.

a. The best corner point is an optimal solution.

b. An optimal solution is a corner point.

c. A corner point is the only optimal solution if none of its adjacent corner points are better (as measured by the value of the objective function).

14.26. The following statements give inaccurate paraphrases of the six solution concepts presented in Section 14.3. In each case, explain what is wrong with the statement.

a. The best corner point always is an optimal solution.

b. The iterative step of the simplex method checks whether the current corner point is optimal and, if not, moves to a new corner point.

c. Although any corner point can be chosen to be the initial corner point, the simplex method always chooses the origin.

d. When the simplex method is ready to choose a new corner point to move to from the current corner point, it only considers adjacent corner points because one of them is likely to be an optimal solution.

e. To choose the new corner point to move to from the current corner point, the simplex method identifies all the adjacent corner points and determines which one gives the largest rate of improvement in the value of the objective function.

14.27. Consider the following resource-allocation problem that has the objective of maximizing the total profit from the two activities.

Resource Usage Per Unit of Each Activity

	Activity		Amount of Resource
Resource	1	2	Available
A	3	1	15
B	1	2	10
Unit profit	$2 million	$1 million	

You are given that the corner points of the feasible region are (0, 0), (5, 0), (4, 3), and (0, 5).

E* *a.* Formulate a spreadsheet model for this problem.

E* *b.* Use the enumeration-of-corner-points method with this spreadsheet to solve the model.

E* *c.* Use Solver (which applies the simplex method) to confirm your answer for the optimal solution in part *b*.

d. Draw a graph of the feasible region and then identify the sequence of corner points examined by the simplex method in part *c* to solve the model.

14.28. Consider the following algebraic form of a linear programming model where x_1 and x_2 are the decision variables and Z is the value of the objective function.

Maximize $Z = 3x_1 + x_2$,

subject to

$x_1 + x_2 \leq 4$

and

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

a. Draw the feasible region and identify all the corner points.

b. Identify the sequence of corner points examined by the simplex method to reach the optimal solution.

R* *c.* Apply the routine for the interior-point algorithm in your MS Courseware to this problem to perform 10 iterations when starting from the initial trial solution $(x_1, x_2) = (1, 1)$. (Obtain these results by going to Textbook Spreadsheets for Chapter 14 on the book's website.)

14.29. Consider the following algebraic form of a linear programming model where x_1 and x_2 are the decision variables and Z is the value of the objective function.

Maximize $Z = x_1 + x_2$,

subject to

$x_1 + 2x_2 \leq 9$

$2x_1 + x_2 \leq 9$

and

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

a. Use the graphical method to solve the model.

b. Identify all the corner points of the feasible region.

c. Because x_1 and x_2 have the same coefficient in the objective function, there is a tie in the first iteration of the simplex method when choosing whether to increase x_1 or x_2 for moving away from the initial corner point at the origin. For each of these two possible choices, identify the sequence of corner points that then would be examined by the simplex method to reach the optimal solution.

R* *d.* Apply the routine for the interior-point algorithm in your MS Courseware to this problem to perform 10 iterations when starting from the initial trial solution $(x_1, x_2) = (1, 1)$. (Obtain these results by going to Textbook Spreadsheets for Chapter 14 on the book's website.)

R* *e.* Repeat part *d* when starting from the initial trial solution $(x_1, x_2) = (3, 1)$.

14.30. Consider the following algebraic form of a linear programming model where x_1, x_2, and x_3 are the decision variables and Z is the value of the objective function.

Maximize $Z = 2\,x_1 + 5\,x_2 + 7\,x_3$,

subject to

$x_1 + 2\,x_2 + 3\,x_3 = 6$

and

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

E* *a.* Use Solver (which applies the simplex method) to solve the model.

R* *b.* Apply the routine for the interior-point algorithm in your MS Courseware to this problem to perform 10 iterations when starting from the initial trial solution $(x_1, x_2, x_3) = (1, 1, 1)$. (Obtain these results by going to Textbook Spreadsheets for Chapter 14 on the book's website.)

14.31. Consider the following algebraic form of a linear programming model where x_1, x_2, x_3, and x_4 are the decision variables and Z is the value of the objective function.

Maximize $Z = 50\,x_1 + 60\,x_2 + 40\,x_3 + 30\,x_4$,

subject to

$$3\,x_1 + 5\,x_2 + 2\,x_3 + 3\,x_4 \le 130$$
$$4\,x_1 + 3\,x_2 + 5\,x_3 + x_4 \le 130$$
$$2\,x_1 + 6\,x_2 + 4\,x_3 + 7\,x_4 \le 190$$

and

$$x_1 \ge 0, x_2 \ge 0, x_3 \ge 0, x_4 \ge 0.$$

E* *a.* Use Solver (which applies the simplex method) to solve the model.

R* *b.* Apply the routine for the interior-point algorithm in your MS Courseware to this problem to perform 10 iterations when starting from the initial trial solution $(x_1, x_2, x_3, x_4) = (5, 6, 4, 3)$. (Obtain these results by going to Textbook Spreadheets for Chapter 14 on the book's website.)

14.32. Use the graphical simplex method to solve the linear programming model given in Problem 14.17.

14.33. Use the graphical simplex method to solve the linear programming model given in Problem 14.18.

14.34. Use the graphical simplex method to solve the linear programming model given in Problem 14.19.

14.35. Consider the following model:

Maximize $Z = 2x_1 + x_2$,

subject to

$$x_1 \le 2$$
$$x_2 \le 5$$

and

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

a. Plot the feasible region by hand. Then use this graph to apply the graphical simplex method.

b. Repeat part *a* but use the best rate of improvement criterion instead of the best adjacent corner point criterion.

14.36. Consider the following model:

Maximize Profit = $x_1 + 2\,x_2 + 3\,x_3$,

subject to

$x_1 \leq 1, \quad x_2 \leq 1, \quad x_3 \leq 1$

and

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

a. Draw a three-dimensional graph similar to Figure 14.16 that shows the feasible region.

b. Identify all the corner points.

c. Use the enumeration-of-corner-points method to find an optimal solution.

d. Outline the steps that the simplex method (with the best adjacent corner point criterion) would take to find an optimal solution.

14.37. Follow the instructions of Problem 14.36 for the following model:

Maximize Profit = $2\,x_1 + x_2 - x_3$,

subject to

$x_1 \leq 2, \quad x_2 \leq 3, \quad x_3 \leq 2$

$x_1 + x_2 \leq 4$

and

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

14.38. Consider the following functional constraints:

$$x_2 \leq 10$$

$$2\,x_1 + x_2 \leq 20$$

a. Define the slack variables for these constraints algebraically.

b. Which values of these slack variables cause the constraints to be satisfied?

c. Show these constraints after they have been converted into equations by incorporating the slack variables.

d. Assume that x_1 and have nonnegativity constraints and consider the corner point, $(x_1, x_2\,) = (10, 0)$. What are the values of the slack variables at this corner point? What are the equations of the constraint boundaries on which it lies? What is the corresponding basic feasible solution? Identify the nonbasic variables and the basic variables for this solution.

14.39. By introducing a slack variable s, a functional constraint in ≤ form has been converted into the following equation:

$$25\,x_1 + 40\,x_2 + 50\,x_3 + s = 500$$

a. What was the functional constraint before introducing the slack variable?

b. Which values of the slack variable cause the constraint to be satisfied?

c. A solution for (x_1, x_2, x_3) lies on the boundary of this constraint when the slack variable has which value?

14.40. Consider the following model:

Maximize Profit = $2x_1 + x_2$,

subject to

$$3x_1 + x_2 \leq 15$$
$$x_1 + 2x_2 \leq 10$$

and

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

You are given that its corner points are (0, 0), (5, 0), (4, 3), and (0, 5).

a. Use the enumeration-of-corner-points method to solve the model.

b. Use the information developed in part *a* to identify the path that the graphical simplex method would follow to solve the model.

c. Convert the functional constraints into equations by introducing slack variables.

d. Identify all the basic feasible solutions for the problem with slack variables. For each such solution, identify both the nonbasic variables and the basic variables.

e. What is the sequence of basic feasible solutions obtained by the simplex method when following the path identified in part *b*?

14.41. Consider the following functional constraints:

$$2x_1 + 3x_2 \geq 21$$
$$5x_1 + 3x_2 \geq 30$$

a. Define the surplus variables for these constraints algebraically.

b. Which values of these surplus variables cause the constraints to be satisfied?

c. Show these constraints after they have been converted into equations by incorporating the surplus variables.

d. Consider the corner point, (x_1, x_2) = (3, 5). What are the values of the surplus variables at this corner point? What are the equations of the constraint boundaries on which it lies? What is the corresponding basic feasible solution? Identify the nonbasic variables and the basic variables for this solution?

14.42. By introducing a surplus variable s, a functional constraint in ≥ form has been converted into the following equation:

$$20\,x_1 + 10\,x_2 - s = 100$$

a. What was the functional constraint before introducing the surplus variable?

b. Which values of the surplus variable cause the constraint to be satisfied?

c. A solution for (x_1, x_2) lies on the boundary of this constraint when the surplus variable has which value?

14.43. Consider the following model:

Minimize Cost = $2\,x_1 + x_2$,

subject to

$5\,x_1 + 3\,x_2 \geq 60$

$3\,x_1 + 2\,x_2 \leq 48$

$x_1 + x_2 \geq 15$

and

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

a. Starting with the corner point, $(x_1, x_2) = (16, 0)$, use the graphical simplex method to solve the model.

b. Convert the functional constraints into equations by introducing slack variables and surplus variables as needed.

c. What is the sequence of basic feasible solutions obtained by the simplex method when following the path identified in part *a*?

14.44. Consider the following model:

$$\text{Maximize} \quad \text{Profit} = x_1 + 3\,x_2,$$

subject to

$$x_1 \leq 7$$
$$x_2 \leq 2$$

and

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

a. Plot the feasible region by hand. Then use this graph to apply the graphical simplex method.

b. Repeat part *a* but use the best rate of improvement criterion instead of the best adjacent corner point criterion.

c. Introduce slack variables to convert the functional constraints into equations.

d. Fill out a table like Table 14.9 that shows both the geometric progression and algebraic progression of the simplex method when following the path identified in part *b*.

e. Flesh out the algebraic progression found in part *d* by applying the simplex method (in algebraic form) to solve the model.

14.45. Follow the instructions of Problem 14.44 for the model in Problem 14.15.

14.46. Consider the following model:

$$\text{Maximize} \quad \text{Profit} = 2x_1 + x_2,$$

subject to

$$x_1 + x_2 \leq 40$$
$$4x_1 + x_2 \leq 100$$

and

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

a. Use the graphical simplex method to solve the model.

b. Introduce slack variables to convert the functional constraints into equations.

c. Fill out a table like Table 14.9 that shows both the geometric progression and algebraic progression of the simplex method when following the path identified in part *a*.

d. Flesh out the algebraic progression found in part *c* by applying the simplex method (in algebraic form) to solve the model.

14.47. Follow the instructions of Problem 14.46 for the following model:

Maximize Profit = $2x_1 + 3x_2$,

subject to

$x_1 + 2x_2 \leq 30$

$x_1 + x_2 \leq 20$

and

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

14.48. Consider the following model:

Maximize Profit = $4x_1 + 3x_2 + 6x_3$,

subject to

$3x_1 + x_2 + 3x_3 \leq 30$

$2x_1 + 2x_2 + 3x_3 \leq 40$

and

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

a. Use the simplex method (in algebraic form) to solve this model.

b. Use Solver to verify the optimal solution you obtained in part *a*.

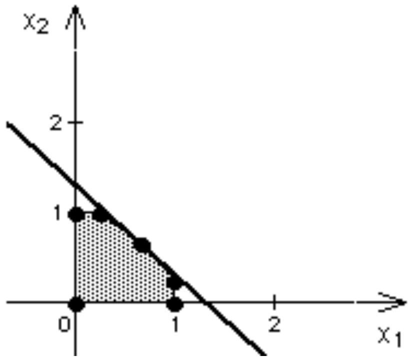
$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0.$

Cases

Any of the cases at the end of Chapter 3 can also be used in conjunction with this chapter. Although those cases do not focus on solution concepts for linear programming, they do require solving linear programming models of substantial size.

PARTIAL ANSWERS TO SELECTED PROBLEMS

14.16. *a.* Optimal Solution: (x_1, x_2) = (0.667, 0.667) and Profit = $6,000.



-Note: Corner points will be called A, B, C, D, E, and F going clockwise from (0,1).

b. -Corner Point A: F and B are adjacent.

B: A and C are adjacent.

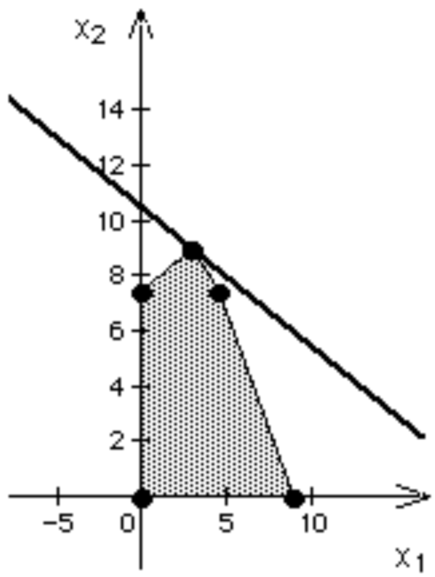
C: B and D are adjacent.

D: C and E are adjacent.

E: D and F are adjacent.

F: E and A are adjacent.

14.20.



Corner Point	Profit = $10x_1 + 20x_2$	Next Step
(0,0)	0	Check (0,7.5) and (9,0).
(0,7.5)	150	Move to (0,7.5).
(9,0)	90	Check (3,9).
(3,9)	210	Move to (3,9) Check (4.5,7.5).
(4.5,7.5)	195	Stop, (3,9) is optimal.