Advanced Topics in Linear Programming†

In this chapter, we discuss six advanced linear programming topics: the revised simplex method, the product form of the inverse, column generation, the Dantzig–Wolfe decomposition algorithm, the simplex method for upper-bounded variables, and Karmarkar's method for solving LPs. The techniques discussed are often utilized to solve large linear programming problems. The results of Section 6.2 play a key role throughout this chapter.

10.1 The Revised Simplex Algorithm

In Section 6.2, we demonstrated how to create an optimal tableau from an initial tableau, given an optimal set of basic variables. Actually, the results of Section 6.2 can be used to create a tableau corresponding to *any set of basic variables.* To show how to create a tableau for any set of basic variables BV, we first describe the following notation (assume the LP has m constraints):

BV = any set of basic variables (the first element of BV is the basic variable in the first constraint, the second variable in BV is the basic variable in the second constraint, and so on; thus, BV_j is the basic variable for constraint j in the desired tableau)

$\mathbf{b}$ = right-hand-side vector of the original tableau's constraints

$\mathbf{a}_j$ = column for x_j in the constraints of the original problem

B = $m \times m$ matrix whose jth column is the column for BV_j in the original constraints

c_j = coefficients of x_j in the objective function

$\mathbf{c}_{\text{BV}}$ = $1 \times m$ row vector whose jth element is the objective function coefficient for BV_j

$\mathbf{u}_i$ = $m \times 1$ column vector with ith element 1 and all other elements equal to zero

Summarizing the formulas of Section 6.2, we write:

$$B^{-1}\mathbf{a}_j = \text{column for } x_j \text{ in BV tableau} \tag{1}$$

$$\mathbf{c}_{\text{BV}}B^{-1}\mathbf{a}_j - c_j = \text{coefficient of } x_j \text{ in row } 0 \tag{2}$$

$$B^{-1}\mathbf{b} = \text{right-hand side of constraints in BV tableau} \tag{3}$$

$$\mathbf{c}_{\text{BV}}B^{-1}\mathbf{u}_i = \text{coefficient of slack variable } s_i \text{ in BV in row } 0 \tag{4}$$

†This chapter covers topics that may be omitted with no loss of continuity.

$$\mathbf{c}_{BV}B^{-1}(-\mathbf{u}_i) = \text{coefficient of excess variable } e_i \text{ in BV row 0} \tag{5}$$

$$M + \mathbf{c}_{BV}B^{-1}\mathbf{u}_i = \text{coefficient of artificial variable } a_i \text{ in BV row 0 (in a max problem)} \tag{6}$$

$$\mathbf{c}_{BV}B^{-1}\mathbf{b} = \text{right-hand side of BV row 0} \tag{7}$$

If we know BV, B^{-1}, and the original tableau, formulas (1)–(7) enable us to compute any part of the simplex tableau for any set of basic variables BV. This means that if a computer is programmed to perform the simplex algorithm, then all the computer needs to store on any pivot is the current set of basic variables, B^{-1}, and the initial tableau. Then (1)–(7) can be used to generate any portion of the simplex tableau. This idea is the basis of the revised simplex algorithm.

We illustrate the revised simplex algorithm by using it to solve the Dakota problem of Chapter 6. Recall that after adding slack variables s_1, s_2, and s_3, the initial tableau (tableau 0) for the Dakota problem is

$$\begin{aligned}
\max z &= 60x_1 + 30x_2 + 20x_3 \\
\text{s.t.} \quad & 8x_1 + 6x_2 + x_3 + s_1 = 48 \\
& 4x_1 + 2x_2 + 1.5x_3 + s_2 = 20 \\
& 2x_1 + 1.5x_2 + 0.5x_3 + s_3 = 8
\end{aligned}$$

No matter how many pivots have been completed, B^{-1} for the current tableau will simply be the 3×3 matrix whose jth column is the column for s_j in the current tableau. Thus, for the original tableau BV(0), the set of basic variables is given by

$$\text{BV}(0) = \{s_1, s_2, s_3\}$$
$$\text{NBV}(0) = \{x_1, x_2, x_3\}$$

We let B_i be the columns in the original LP that correspond to the basic variables for tableau i. Then

$$B_0^{-1} = B_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We can now determine which nonbasic variable should enter the basis by computing the coefficient of each nonbasic variable in the current row 0. This procedure is often referred to as **pricing out** the nonbasic variable. From (2)–(5), we see that we can't price out the nonbasic variables until we have determined $\mathbf{c}_{BV}B_0^{-1}$. Because $\mathbf{c}_{BV} = [0 \quad 0 \quad 0]$, we have

$$\mathbf{c}_{BV}B_0^{-1} = [0 \quad 0 \quad 0]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [0 \quad 0 \quad 0]$$

We now use (2) to price out each nonbasic variable:

$$\bar{c}_1 = [0 \quad 0 \quad 0]\begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix} - 60 = -60$$

$$\bar{c}_2 = [0 \quad 0 \quad 0]\begin{bmatrix} 6 \\ 2 \\ 1.5 \end{bmatrix} - 30 = -30$$

$$\begin{bmatrix} 1 \\ 1.5 \\ 0.5 \end{bmatrix} - 20 = -20 \qquad\qquad \bar{c}_3 = [0 \;\; 0$$

Because x_1 has the most negative coefficient in the current row 0, x_1 should enter the basis. To continue the simplex, all we need to know about the new tableau is the new set of basic variables, BV(1), and the corresponding B_1^{-1}. To determine BV(1), we find the row in which x_1 enters the basis. We compute the column for x_1 in the current tableau and the right-hand side of the current tableau.

From(1),

$$\text{Column for } x_1 \text{ in current tableau} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix}$$

From (3),

$$\text{Right-hand side of current tableau} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 48 \\ 20 \\ 8 \end{bmatrix} = \begin{bmatrix} 48 \\ 20 \\ 8 \end{bmatrix}$$

We now use the ratio test to determine the row in which x_1 should enter the basis. The appropriate ratios are row 1, $\frac{48}{8} = 6$; row 2, $\frac{20}{4} = 5$; and row 3, $\frac{8}{2} = 4$. Thus, x_1 should enter the basis in row 3. This means that our new tableau (tableau 1) will have BV(1) = $\{s_1, s_2, x_1\}$ and NBV(1) = $\{s_3, x_2, x_3\}$.

The new B^{-1} will be the columns of s_1, s_2, and s_3 in the new tableau. To determine the new B^{-1}, look at the column in tableau 0 for the entering variable x_1. From this column, we see that in going from tableau 0 to tableau 1, we must perform the following EROs:

1 Multiply row 3 of tableau 0 by $\frac{1}{2}$.

2 Replace row 1 of tableau 0 by -4(row 3 of tableau 0) + row 1 of tableau 0.

3 Replace row 2 of tableau 0 by -2(row 3 of tableau 0) + row 2 of tableau 0.

Applying these EROs to B_0^{-1} yields

$$B_1^{-1} = \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -2 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}$$

We can now price out all the nonbasic variables for the new tableau. First we compute

$$\mathbf{c}_{\text{BV}}B_1^{-1} = [0 \quad 0 \quad 60]\begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -2 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} = [0 \quad 0 \quad 30]$$

Then use (2) and (4) to price out tableau 1's nonbasic variables:

$$\bar{c}_2 = [0 \quad 0 \quad 30]\begin{bmatrix} 6 \\ 2 \\ 1.5 \end{bmatrix} - 30 = 15$$

$$\bar{c}_3 = [0 \quad 0 \quad 30]\begin{bmatrix}1\\1.5\\0.5\end{bmatrix} - 20 = -5$$

$$\text{Coefficient of } s_3 \text{ in row } 0 = [0 \quad 0 \quad 30]\begin{bmatrix}0\\0\\1\end{bmatrix} - 0 = 30$$

Because x_3 is the only variable with a negative coefficient in row 0 of tableau 1, we enter x_3 into the basis. To determine the new set of basic variables, BV(2), and the corresponding B_2^{-1}, we find the row in which x_3 enters the basis and compute

$$x_3 \text{ column in tableau } 1 = B_1^{-1}\mathbf{a}_3 = \begin{bmatrix}1 & 0 & -4\\0 & 1 & -2\\0 & 0 & 0.5\end{bmatrix}\begin{bmatrix}1\\1.5\\0.5\end{bmatrix} = \begin{bmatrix}-1\\0.5\\0.25\end{bmatrix}$$

$$\text{Right-hand side of tableau } 1 = B_1^{-1}\mathbf{b} = \begin{bmatrix}1 & 0 & -4\\0 & 1 & -2\\0 & 0 & 0.5\end{bmatrix}\begin{bmatrix}48\\20\\8\end{bmatrix} = \begin{bmatrix}16\\4\\4\end{bmatrix}$$

The appropriate ratios for determining where x_3 should enter the basis are row 1, none; row 2, $\frac{4}{0.5} = 8$; and row 3, $\frac{4}{0.25} = 16$. Hence, x_3 should enter the basis in row 2. Then tableau 2 will have BV(2) $= \{s_1, x_3, x_1\}$ and NBV(2) $= \{s_2, s_3, x_2\}$.

To compute B_2^{-1}, note that to make x_3 a basic variable in row 2, we must perform the following EROs on tableau 1:

1 Replace row 2 of tableau 1 by 2(row 2 of tableau 1).

2 Replace row 1 of tableau 1 by 2(row 2 of tableau 1) + row 1 of tableau 1.

3 Replace row 3 of tableau 1 by $-\frac{1}{2}$(row 2 of tableau 1) + row 3 of tableau 1.

Applying these EROs to B_1^{-1}, we obtain

$$B_2^{-1} = \begin{bmatrix}1 & 2 & -8\\0 & 2 & -4\\0 & -0.5 & 1.5\end{bmatrix}$$

We now price out the nonbasic variables in tableau 2. First we compute

$$\mathbf{c}_{\text{BV}}B_2^{-1} = [0 \quad 20 \quad 60]\begin{bmatrix}1 & 2 & -8\\0 & 2 & -4\\0 & -0.5 & 1.5\end{bmatrix} = [0 \quad 10 \quad 10]$$

Then we price out the nonbasic variables x_2, s_2, and s_3:

$$\bar{c}_2 = [0 \quad 10 \quad 10]\begin{bmatrix}6\\2\\1.5\end{bmatrix} - 30 = 5$$

$$\text{Coefficient of } s_2 \text{ in row } 0 = [0 \quad 10 \quad 10]\begin{bmatrix}0\\1\\0\end{bmatrix} - 0 = 10$$

$$\text{Coefficient of } s_3 \text{ in row } 0 = [0 \quad 10 \quad 10]\begin{bmatrix}0\\0\\1\end{bmatrix} - 0 = 10$$

Each nonbasic variable has a nonnegative coefficient in row 0, so tableau 2 is an optimal tableau. To find the optimal solution, we find the right-hand side of tableau 2. From (3), we obtain

$$\text{Right-hand side of tableau 2} = \begin{bmatrix}1 & 2 & -8\\0 & 2 & -4\\0 & -0.5 & 1.5\end{bmatrix}\begin{bmatrix}48\\20\\8\end{bmatrix} = \begin{bmatrix}24\\8\\2\end{bmatrix}$$

Because BV(2) $= \{s_1, x_3, x_1\}$, the optimal solution to the Dakota problem is

$$\begin{bmatrix}s_1\\x_3\\x_1\end{bmatrix} = \begin{bmatrix}24\\8\\2\end{bmatrix}$$

or $s_1 = 24, x_3 = 8, x_1 = 2, x_2 = s_2 = s_3 = 0$. The optimal z-value may be found from (7):

$$\mathbf{c}_{\mathrm{BV}}B_2^{-1}\mathbf{b} = [0 \quad 10 \quad 10]\begin{bmatrix}48\\20\\8\end{bmatrix} = 280$$

A summary of the revised simplex method (for a max problem) follows:

Step 0 Note the columns from which the current B^{-1} will be read. Initially, $B^{-1} = I$.

Step 1 For the current tableau, compute $\mathbf{c}_{\mathrm{BV}}B^{-1}$.

Step 2 Price out all nonbasic variables in the current tableau. If each nonbasic variable prices out to be nonnegative, then the current basis is optimal. If the current basis is not optimal, then enter into the basis the nonbasic variable with the most negative coefficient in row 0. Call this variable x_k.

Step 3 To determine the row in which x_k enters the basis, compute x_k's column in the current tableau ($B^{-1}\mathbf{a}_k$) and compute the right-hand side of the current tableau ($B^{-1}\mathbf{b}$). Then use the ratio test to determine the row in which x_k should enter the basis. We now know the set of basic variables (BV) for the new tableau.

Step 4 Use the column for x_k in the current tableau to determine the EROs needed to enter x_k into the basis. Perform these EROs on the current B^{-1}. This will yield the new B^{-1}. Return to step 1.

Most linear programming computer codes use some version of the revised simplex to solve LPs. Knowing the current tableau's B^{-1} and the initial tableau is all that is needed to obtain the next tableau, so the computational effort required to solve an LP by the revised simplex depends primarily on the size of B^{-1}. Suppose the LP being solved has m constraints and n variables. Then each B^{-1} will be an $m \times m$ matrix, and the effort required to solve an LP will depend primarily on the number of constraints (not the number of variables). This fact has important computational implications. For example, if we are solving an LP that has 500 constraints and 10 variables, the LP's dual will have 10 constraints and 500 variables. Then all the B^{-1}'s for the dual will be 10×10 matrices, and all the B^{-1}'s for the primal will be 500×500. Thus, it will be much easier to solve the dual than to solve the primal. In this situation, computation can be greatly reduced by solving the dual and reading the optimal primal solution from the SHADOW PRICE or DUAL VARIABLE section of a computer printout.

PROBLEMS

Group A

Use the revised simplex method to solve the following LPs:

1
$$
\begin{aligned}
\max z &= 3x_1 + x_2 + x_3\\
\text{s.t.} \quad x_1 + x_2 + x_3 &\le 6\\
2x_1 - x_3 &\le 4\\
x_2 + x_3 &\le 2\\
x_1, x_2, x_3 &\ge 0
\end{aligned}
$$

2
$$
\begin{aligned}
\max z &= 4x_1 + x_2\\
\text{s.t.} \quad x_1 + x_2 &\le 4\\
2x_1 + x_2 &\ge 6\\
3x_2 &\ge 6\\
x_1, x_2, x_3 &\ge 0
\end{aligned}
$$

Remember that B^{-1} is always found under the columns corresponding to the starting basis.)

3
$$
\begin{aligned}
\min z &= 3x_1 + x_2 - 3x_3\\
\text{s.t.} \quad x_1 - x_2 + x_3 &\le 4\\
x_1 + x_3 &\le 6\\
2x_2 - x_3 &\le 5\\
x_1, x_2, x_3 &\ge 0
\end{aligned}
$$

10.2 The Product Form of the Inverse

Much of the computation in the revised simplex algorithm is concerned with updating B^{-1} from one tableau to the next. In this section, we develop an efficient method to update B^{-1}.

Suppose we are solving an LP with m constraints. Assume that we have found that x_k should enter the basis, in row r. Let the column for x_k in the current tableau be

$$
\begin{bmatrix} \bar{a}_{1k} \\ \bar{a}_{2k} \\ \vdots \\ \bar{a}_{mk} \end{bmatrix}
$$

Define the $m \times m$ matrix E:

$$
\begin{array}{cc}
 & \text{(column } r\text{)} \\
E = \begin{bmatrix}
1 & 0 & \cdots & -\dfrac{\bar{a}_{1k}}{\bar{a}_{rk}} & \cdots & 0 & 0 \\
0 & 1 & \cdots & -\dfrac{\bar{a}_{2k}}{\bar{a}_{rk}} & \cdots & 0 & 0 \\
\vdots & \vdots & & \vdots & & \vdots & \vdots \\
0 & 0 & \cdots & \dfrac{1}{\bar{a}_{rk}} & \cdots & 0 & 0 \\
\vdots & \vdots & & \vdots & & \vdots & \vdots \\
0 & 0 & \cdots & -\dfrac{\bar{a}_{m-1,k}}{\bar{a}_{rk}} & \cdots & 1 & 0 \\
0 & 0 & \cdots & -\dfrac{\bar{a}_{mk}}{\bar{a}_{rk}} & \cdots & 0 & 1
\end{bmatrix} & \text{(row } r\text{)}
\end{array}
$$

In short, E is simply I_m with column r replaced by the column vector

$$\begin{bmatrix} -\dfrac{\bar{a}_{1k}}{\bar{a}_{rk}} \\ -\dfrac{\bar{a}_{2k}}{\bar{a}_{rk}} \\ \vdots \\ \dfrac{1}{\bar{a}_{rk}} \\ \vdots \\ -\dfrac{\bar{a}_{m-1,k}}{\bar{a}_{rk}} \\ -\dfrac{\bar{a}_{mk}}{\bar{a}_{rk}} \end{bmatrix}$$

DEFINITION ■ A matrix (such as E) that differs from the identity matrix in only one column is called an **elementary matrix.** ■

We now show that

$$B^{-1} \text{ for new tableau} = E(B^{-1} \text{ for current tableau}) \tag{8}$$

To see why this is true, note that the EROs used to go from the current tableau to the new tableau boil down to

$$\text{Row } r \text{ of new } B^{-1} = \left(\frac{1}{\bar{a}_{rk}}\right)(\text{row } r \text{ of current } B^{-1}) \tag{9}$$

and for $i \neq r$,

$$\begin{aligned} &\text{Row } i \text{ of new } B^{-1} \\ &= (\text{row } i \text{ of current } B^{-1}) - \left(\frac{\bar{a}_{ik}}{\bar{a}_{rk}}\right)(\text{row } r \text{ of current } B^{-1}) \end{aligned} \tag{10}$$

Recall from Section 2.1 that

$$\text{Row } i \text{ of } E(\text{current } B^{-1}) = (\text{row } i \text{ of } E)(\text{current } B^{-1}) \tag{11}$$

Combining (11) with the definition of E, we find that

$$\text{Row } r \text{ of } E(\text{current } B^{-1}) = \left(\frac{1}{\bar{a}_{rk}}\right)(\text{row } r \text{ of current } B^{-1})$$

and for $i \neq r$,

$$\begin{aligned} &\text{Row } i \text{ of } E(\text{current } B^{-1}) \\ &= (\text{row } i \text{ of current } B^{-1}) - \left(\frac{\bar{a}_{ik}}{\bar{a}_{rk}}\right)(\text{row } r \text{ of current } B^{-1}) \end{aligned}$$

Hence, (8) does agree with (9) and (10). Thus, we can use (8) to find the new B^{-1} from the current B^{-1}.

Define the initial tableau to be tableau 0, and let E_i be the elementary matrix E associated with the ith simplex tableau. Recall that $B_0^{-1} = I_m$. We now write

$$B_1^{-1} = E_0 B_0^{-1} = E_0$$

Then

$$B_2^{-1} = E_1 B_1^{-1} = E_1 E_0$$

and, in general,

$$B_k^{-1} = E_{k-1} E_{k-2} \cdots E_1 E_0 \tag{12}$$

Equation (12) is called the **product form of the inverse.** Most linear programming computer codes utilize the revised simplex method and compute successive B^{-1}'s by using the product form of the inverse.

EXAMPLE 1 Product Form of the Inverse

Use the product form of the inverse to compute B_1^{-1} and B_2^{-1} for the Dakota problem that was solved by the revised simplex in Section 10.1.

Solution Recall that in tableau 0, x_1 entered the basis in row 3. Hence, for tableau 0, $r = 3$ and $k = 1$. For tableau 0,

$$\begin{bmatrix} \bar{a}_{11} \\ \bar{a}_{21} \\ \bar{a}_{31} \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix}$$

Then

$$E_0 = \begin{bmatrix} 1 & 0 & -\frac{8}{2} \\ 0 & 1 & -\frac{4}{2} \\ 0 & 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -2 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}$$

$$B_1^{-1} = \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -2 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -2 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}$$

As we proceeded from tableau 1 to tableau 2, x_3 entered the basis in row 2. Hence, in computing E_1, we set $r = 2$ and $k = 3$. To compute E_1, we need to find the column for the entering variable (x_3) in tableau 1:

$$\begin{bmatrix} \bar{a}_{13} \\ \bar{a}_{23} \\ \bar{a}_{33} \end{bmatrix} = B_1^{-1}\mathbf{a}_3 = \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -2 \\ 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 1.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} -1 \\ 0.5 \\ 0.25 \end{bmatrix}$$

As before, x_3 enters the basis in row 2. Then

$$E_1 = \begin{bmatrix} 1 & -(-\frac{1}{0.5}) & 0 \\ 0 & \frac{1}{0.5} & 0 \\ 0 & -\frac{0.25}{0.50} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & -0.5 & 1 \end{bmatrix}$$

and (as before)

$$B_2^{-1} = E_1 B_1^{-1} = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & -0.5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -2 \\ 0 & 0 & 0.5 \end{bmatrix} = \begin{bmatrix} 1 & 2 & -8 \\ 0 & 2 & -4 \\ 0 & -0.5 & 1.5 \end{bmatrix}$$

In the next two sections, we use the product form of the inverse in our study of column generation and of the Dantzig–Wolfe decomposition algorithm.

PROBLEM

Group A

For the problems of Section 10.1, use the product form of the inverse to perform the revised simplex method.

10.3 Using Column Generation to Solve Large-Scale LPs

前序知识revised simplex algorithm

We have already seen that the revised simplex algorithm requires less computation than the simplex algorithm of Chapter 4. In this section, we discuss the method of column generation, devised by Gilmore and Gomory (1961). For LPs that have many variables, column generation can be used to increase the efficiency of the revised simplex algorithm. Column generation is also a very important component of the Dantzig–Wolfe decomposition algorithm, which is discussed in Section 10.4. To explain the idea of column generation, we solve a simple version of the classic *cutting stock problem.*

EXAMPLE 2 Odds and Evens

Woodco sells 3-ft, 5-ft, and 9-ft pieces of lumber. Woodco's customers demand 25 3-ft boards, 20 5-ft boards, and 15 9-ft boards. Woodco, who must meet its demands by cutting up 17-ft boards, wants to minimize the waste incurred. Formulate an LP to help Woodco accomplish its goal, and solve the LP by column generation.

Solution Woodco must decide how each 17-ft board should be cut. Hence, each decision corresponds to a way in which a 17-ft board can be cut. For example, one decision variable would correspond to a board being cut into three 5-ft boards, which would incur waste of $17 - 15 = 2$ ft. Many possible ways of cutting a board need not be considered. For example, it would be foolish to cut a board into one 9-ft and one 5-ft piece; we could just as easily cut the board into a 9-ft piece, a 5-ft piece, *and* a 3-ft piece. In general, any cutting pattern that leaves 3 ft or more of waste need not be considered because we could use the waste to obtain one or more 3-ft boards. Table 1 lists the sensible ways to cut a 17-ft board.

TABLE 1
Ways to Cut a Board in the Cutting Stock Problem

	Number of			
Combination	3-ft Boards	5-ft Boards	9-ft Boards	Waste (Feet)
1	5	0	0	2
2	4	1	0	0
3	2	2	0	1
4	2	0	1	2
5	1	1	1	0
6	0	3	0	2

We now define

$$x_i = \text{number of 17-ft boards cut according to combination } i$$

and formulate Woodco's LP:

$$\text{Woodco's waste} + \text{total customer demand} = \text{total length of board cut}$$

Because

$$\text{Total customer demand} = 25(3) + 20(5) + 15(9) = 310 \text{ ft}$$
$$\text{Total length of boards cut} = 17(x_1 + x_2 + x_3 + x_4 + x_5 + x_6)$$

we write

$$\text{Woodco's waste (in feet)} = 17x_1 + 17x_2 + 17x_3 + 17x_4 + 17x_5 + 17x_6 - 310$$

Then Woodco's objective function is to minimize

$$\min z = 17x_1 + 17x_2 + 17x_3 + 17x_4 + 17x_5 + 17x_6 - 310$$

This is equivalent to minimizing

$$17(x_1 + x_2 + x_3 + x_4 + x_5 + x_6)$$

which is equivalent to minimizing

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

Hence, Woodco's objective function is

$$\min z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \tag{13}$$

This means that Woodco can minimize its total waste by minimizing the number of 17-ft boards that are cut.

Woodco faces the following three constraints:

Constraint 1 At least 25 3-ft boards must be cut.

Constraint 2 A t least 20 5-ft boards must be cut.

Constraint 3 At least 15 9-ft boards must be cut.

Because the total number of 3-ft boards that are cut is given by $5x_1 + 4x_2 + 2x_3 + 2x_4 + x_5$, Constraint 1 becomes

$$5x_1 + 4x_2 + 2x_3 + 2x_4 + x_5 \geq 25 \tag{14}$$

Similarly, Constraint 2 becomes

$$x_2 + 2x_3 + x_5 + 3x_6 \geq 20 \tag{15}$$

and Constraint 3 becomes

$$x_4 + x_5 \geq 15 \tag{16}$$

Note that the coefficient of x_i in the constraint for k-ft boards is just the number of k-ft boards yielded if a board is cut according to combination i.

It is clear that the x_i should be required to assume integer values. Despite this fact, in problems with large demands, a near-optimal solution can be obtained by solving the cutting stock problem as an LP and then rounding all fractional variables upward. This procedure may not yield the best possible integer solution, but it usually yields a near-optimal integer solution. For this reason, we concentrate on the LP version of the cutting stock problem. Combining the sign restrictions with (13)–(16), we obtain the following LP:

$$\begin{aligned}
\min z &= x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
\text{s.t.} \quad & 5x_1 + 4x_2 + 2x_3 + 2x_4 + x_5 \geq 25 && \text{(3-ft constraint)} \\
& x_2 + 2x_3 + x_5 + 3x_6 \geq 20 && \text{(5-ft constraint)} \\
& x_4 + x_5 \geq 15 && \text{(9-ft constraint)} \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{aligned} \tag{17}$$

Note that x_1 only occurs in the 3-ft constraint (because combination 1 yields only 3-ft boards), and x_6 occurs in the 5-ft constraint (because combination 6 yields only 5-ft boards). This means that x_1 and x_6 can be used as starting basic variables for the 3-ft and 5-ft constraints. Unfortunately, none of combinations 1–6 yields only 9-ft boards, so the 9-ft constraint has no obvious basic variable. To avoid having to add an artificial variable to the 9-ft constraint, we define combination 7 to be the cutting combination that yields only one 9-ft board. Also, define x_7 to be the number of boards cut according to combination 7. Clearly, x_7 will be equal to zero in the optimal solution, but inserting x_7 in the starting basis allows us to avoid using the Big M or the two-phase simplex method. Note that the column for x_7 in the LP constraints will be

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

and a term x_7 will be added to the objective function. We can now use BV $= \{x_1, x_6, x_7\}$ as a starting basis for LP (17). If we let the tableau for this basis be tableau 0, then we have

$$B_0 = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_0^{-1} = \begin{bmatrix} \frac{1}{5} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then

$$\mathbf{c}_{\text{BV}} B_0^{-1} = [1 \quad 1 \quad 1] \begin{bmatrix} \frac{1}{5} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} = [\tfrac{1}{5} \quad \tfrac{1}{3} \quad 1]$$

If we now priced out each nonbasic variable it would tell us which variable should enter the basis. However, in a large-scale cutting stock problem, there may be thousands of variables, so pricing out each nonbasic variable would be an extremely tedious chore. This is the type of situation in which column generation comes into play. Because we are solving a minimization problem, we want to find a column that will price out positive (have a positive coefficient in row 0). In the cutting stock problem, each column, or variable, represents a combination for cutting up a board: A variable is specified by three numbers: a_3, a_5, and a_9, where a_i is the number of i-ft boards yielded by cutting one 17-ft board according to the given combination. For example, the variable x_2 is specified by $a_3 = 4$, $a_5 = 1$, and $a_9 = 0$. The idea of column generation is to search efficiently for a column that will price out favorably (positive in a min problem and negative in a max problem). For our current basis, a combination specified by a_3, a_5, and a_9 will price out as

$$\mathbf{c}_{\mathrm{BV}}B_0^{-1}\begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = \frac{1}{5}a_3 + \frac{1}{3}a_5 + a_9 - 1$$

Note that a_3, a_5, and a_9 must be chosen so they don't use more than 17 ft of wood. We also know that a_3, a_5, and a_9 must be nonnegative integers. In short, for any combination, a_3, a_5, and a_9 must satisfy

$$3a_3 + 5a_5 + 9a_9 \le 17 \quad (a_3 \ge 0,\ a_5 \ge 0,\ a_9 \ge 0;\ a_3, a_5, a_9 \text{ integer}) \tag{18}$$

We can now find the combination that prices out most favorably by solving the following knapsack problem:

$$\begin{aligned} \max z &= \tfrac{1}{5}a_3 + \tfrac{1}{3}a_5 + a_9 - 1 \\ \text{s.t.} \quad & 3a_3 + 5a_5 + 9a_9 \le 17 \\ & a_3, a_5, a_9 \ge 0;\ a_3, a_5, a_9 \text{ integer} \end{aligned} \tag{19}$$

Because (19) is a knapsack problem (without 0–1 restrictions on the variables), it can easily be solved by using the branch-and-bound procedure outlined in Section 9.5.

The resulting branch-and-bound tree is given in Figure 1. For example, to solve Problem 6 in Figure 1, we first set $a_5 = 1$ (because $a_5 \ge 1$ is necessary). Then we have 12 ft left in the knapsack, and we choose to make a_9 (the best item) as large as possible. Because $a_9 \ge 1$, we set $a_9 = 1$. This leaves 3 ft, so we set $a_3 = 1$ to fill the knapsack. From Figure 1, we find that the optimal solution to LP (19) is $z = \frac{8}{15}$, $a_3 = a_5 = a_9 = 1$. This corresponds to combination 5 and variable x_5. Hence, x_5 prices out $\frac{8}{15}$, and entering x_5

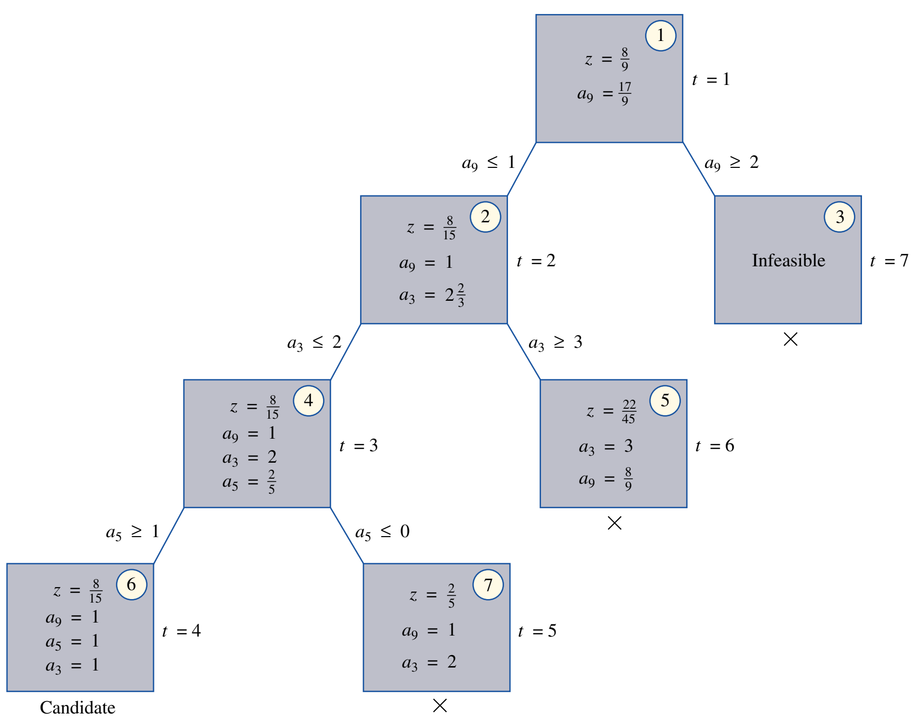


FIGURE 1
Branch-and-Bound Tree for IP (19)

into the basis will decrease Woodco's waste. To enter x_5 into the basis, we create the right-hand side of the current tableau and the x_5 column of the current tableau.

$$x_5 \text{ column in current tableau} = B_0^{-1}\begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}\frac{1}{5} & 0 & 0\\ 0 & \frac{1}{3} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}\frac{1}{5}\\ \frac{1}{3}\\ 1\end{bmatrix}$$

$$\text{Right-hand side of current tableau} = B_0^{-1}\mathbf{b} = \begin{bmatrix}\frac{1}{5} & 0 & 0\\ 0 & \frac{1}{3} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}25\\20\\15\end{bmatrix} = \begin{bmatrix}5\\ \frac{20}{3}\\ 15\end{bmatrix}$$

The ratio test indicates that x_5 should enter the basis in row 3. This yields BV(1) = $\{x_1, x_6, x_5\}$. Using the product form of the inverse, we obtain

$$B_1^{-1} = E_0 B_0^{-1} = \begin{bmatrix}1 & 0 & -\frac{1}{5}\\ 0 & 1 & -\frac{1}{3}\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\frac{1}{5} & 0 & 0\\ 0 & \frac{1}{3} & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$= \begin{bmatrix}\frac{1}{5} & 0 & -\frac{1}{5}\\ 0 & \frac{1}{3} & -\frac{1}{3}\\ 0 & 0 & 1\end{bmatrix}$$

Now

$$\mathbf{c}_{\mathrm{BV}}B_1^{-1} = [1 \quad 1 \quad 1]\begin{bmatrix}\frac{1}{5} & 0 & -\frac{1}{5}\\ 0 & \frac{1}{3} & -\frac{1}{3}\\ 0 & 0 & 1\end{bmatrix} = [\tfrac{1}{5} \quad \tfrac{1}{3} \quad \tfrac{7}{15}]$$

With our new set of shadow prices ($\mathbf{c}_{\mathrm{BV}}B_1^{-1}$), we can again use column generation to determine whether there is any combination that should be entered into the basis. For the current set of shadow prices, a combination specified by a_3, a_5, and a_9 prices out to

$$\begin{bmatrix}\frac{1}{5} & \frac{1}{3} & \frac{7}{15}\end{bmatrix}\begin{bmatrix}a_3\\ a_5\\ a_9\end{bmatrix} - 1 = \frac{1}{5}a_3 + \frac{1}{3}a_5 + \frac{7}{15}a_9 - 1$$

For the current tableau, the column generation procedure yields the following problem:

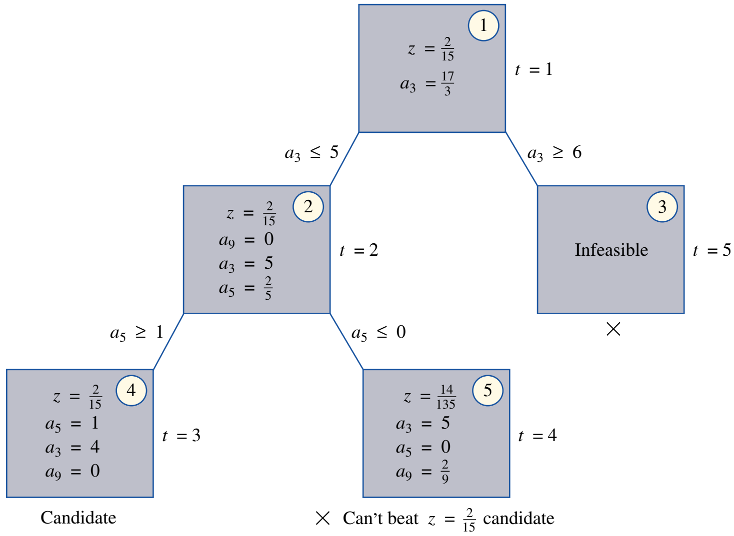
$$\begin{aligned} \max z &= \tfrac{1}{5}a_3 + \tfrac{1}{3}a_5 + \tfrac{7}{15}a_9 - 1\\ \text{s.t.} \quad & 3a_3 + 5a_5 + 9a_9 \le 17 \\ & a_3, a_5, a_9 \ge 0;\ a_3, a_5, a_9 \text{ integer}\end{aligned} \tag{20}$$

The branch-and-bound tree for (20) is given in Figure 2. We see that the combination with $a_3 = 4$, $a_5 = 1$, and $a_9 = 0$ (combination 2) will price out better than any other (it will have a row 0 coefficient of $\frac{2}{15}$). Combination 2 prices out most favorably, so we now enter x_2 into the basis. The column for x_2 in the current tableau is

$$B_1^{-1}\begin{bmatrix}4\\1\\0\end{bmatrix} = \begin{bmatrix}\frac{1}{5} & 0 & -\frac{1}{5}\\ 0 & \frac{1}{3} & -\frac{1}{3}\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}4\\1\\0\end{bmatrix} = \begin{bmatrix}\frac{4}{5}\\ \frac{1}{3}\\ 0\end{bmatrix}$$

The right-hand side of the current tableau is

FIGURE 2
Branch-and-Bound Tree for IP (20)



$$B_1^{-1}\mathbf{b} = \begin{bmatrix} \frac{1}{5} & 0 & -\frac{1}{5} \\ 0 & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 25 \\ 20 \\ 15 \end{bmatrix} = \begin{bmatrix} 2 \\ \frac{5}{3} \\ 15 \end{bmatrix}$$

The ratio test indicates the x_2 should enter the basis in row 1. Hence, BV(2) $= \{x_2, x_6, x_5\}$. Using the product form of the inverse, we find that

$$E_1 = \begin{bmatrix} \frac{5}{4} & 0 & 0 \\ -\frac{5}{12} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then

$$B_2^{-1} = E_1 B_1^{-1} = \begin{bmatrix} \frac{5}{4} & 0 & 0 \\ -\frac{5}{12} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{5} & 0 & -\frac{1}{5} \\ 0 & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & 0 & -\frac{1}{4} \\ -\frac{1}{12} & \frac{1}{3} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix}$$

The new set of shadow prices is given by

$$\mathbf{c}_{\text{BV}} B_2^{-1} = [1 \quad 1 \quad 1] \begin{bmatrix} \frac{1}{4} & 0 & -\frac{1}{4} \\ -\frac{1}{12} & \frac{1}{3} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix} = [\tfrac{1}{6} \quad \tfrac{1}{3} \quad \tfrac{1}{2}]$$

For this set of shadow prices, a combination specified by a_3, a_5, and a_9 will price out to $\frac{1}{6}a_3 + \frac{1}{3}a_5 + \frac{1}{2}a_9 - 1$. Thus, the column-generation procedure requires us to solve the following problem:

$$\begin{aligned} &\max z = \tfrac{1}{6}a_3 + \tfrac{1}{3}a_5 + \tfrac{1}{2}a_9 - 1 \\ &\text{s.t.} \quad 3a_3 + 5a_5 + 9a_9 \le 17 \\ &a_3, a_5, a_9 \ge 0;\ a_3, a_5, a_9 \text{ integer} \end{aligned} \tag{21}$$

The branch-and-bound tree for IP (21) is left as an exercise (see Problem 1 at the end of this section). The optimal z-value for (21) is found to be $z = 0$. This means that no combination can price out favorably. Hence, our current basic solution must be an optimal solution. To find the values of the basic variables in the optimal solution, we find the right-hand side of the current tableau:

$$B_2^{-1}\mathbf{b} = \begin{bmatrix} \frac{1}{4} & 0 & -\frac{1}{4} \\ -\frac{1}{12} & \frac{1}{3} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 25 \\ 20 \\ 15 \end{bmatrix} = \begin{bmatrix} \frac{5}{2} \\ \frac{5}{6} \\ 15 \end{bmatrix}$$

Therefore, the optimal solution to Woodco's cutting stock problem is given by $x_2 = \frac{5}{2}$, $x_6 = \frac{5}{6}$, $x_5 = 15$. If desired, we could obtain a "reasonable" integer solution by rounding x_2 and x_6 upward. This yields the integer solution $x_2 = 3$, $x_6 = 1$, $x_5 = 15$.

If we have a starting bfs for a cutting stock problem, we need not list all possible ways in which a board may be cut. At each iteration, a good combination (one that will improve the z-value when entered into the basis) is generated by solving a branch-and-bound problem. The fact that we don't have to list all the ways a board can be cut is very helpful; a cutting stock problem that was solved in Gilmore and Gomory (1961) for which customers demanded boards of 40 different lengths involved more than 100 million possible ways a board could be cut. At the last stage of the column-generation procedure for this problem, solving a single branch-and-bound problem indicated that none of the 100 million (nonbasic) ways

10.4 The Dantzig–Wolfe Decomposition Algorithm

In many LPs, the constraints and variables may be decomposed in the following manner:

Constraints in set 1 only involve variables in Variable set 1.
Constraints in set 2 only involve variables in Variable set 2.

⋮

Constraints in set k only involve variables in Variable set k.

Constraints in set $k + 1$ may involve any variable. The constraints in set $k + 1$ are referred to as the **central constraints.** LPs that can be decomposed in this fashion can often be solved efficiently by the Dantzig–Wolfe decomposition algorithm.

EXAMPLE 3 Decomposition

Steelco manufactures two types of steel (steel 1 and steel 2) at two locations (plants 1 and 2). Three resources are needed to manufacture a ton of steel: iron, coal, and blast furnace time. The two plants have different types of furnaces, so the resources needed to manufacture a ton of steel depend on the location (see Table 2). Each plant has its own coal mine. Each day, 12 tons of coal are available at plant 1 and 15 tons at plant 2. Coal cannot be shipped between plants. Each day, plant 1 has 10 hours of blast furnace time available, and plant 2 has 4 hours available. Iron ore is mined in a mine located midway between the two plants; 80 tons of iron are available each day. Each ton of steel 1 can be sold for \$170/ton, and each ton of steel 2 can be sold for \$160/ton. All steel that is sold is shipped to a single customer. It costs \$80 to ship a ton of steel from plant 1, and \$100

TABLE 2

Resource Requirements for Steelco

Product (1 Ton)	Iron Required (Tons)	Coal Required (Tons)	Blast Furnace Time Requested (Hours)
Steel 1 at plant 1	8	3	2
Steel 2 at plant 1	6	1	1
Steel 1 at plant 2	7	3	1
Steel 2 at plant 2	5	2	1

ferred to as the **central constraints.** LPs that can be decomposed in this fashion can often be solved efficiently by the Dantzig–Wolfe decomposition algorithm.

Steelco manufactures two types of steel (steel 1 and steel 2) at two locations (plants 1 and 2). Three resources are needed to manufacture a ton of steel: iron, coal, and blast furnace time. The two plants have different types of furnaces, so the resources needed to manufacture a ton of steel depend on the location (see Table 2). Each plant has its own coal mine. Each day, 12 tons of coal are available at plant 1 and 15 tons at plant 2. Coal cannot be shipped between plants. Each day, plant 1 has 10 hours of blast furnace time available, and plant 2 has 4 hours available. Iron ore is mined in a mine located midway between the two plants; 80 tons of iron are available each day. Each ton of steel 1 can be sold for \$170/ton, and each ton of steel 2 can be sold for \$160/ton. All steel that is sold is shipped to a single customer. It costs \$80 to ship a ton of steel from plant 1, and \$100 a ton from plant 2. Assuming that the only variable cost is the shipping cost, formulate and solve an LP to maximize Steelco's revenues less shipping costs.

Solution Define

$$x_1 = \text{tons of steel 1 produced daily at plant 1}$$
$$x_2 = \text{tons of steel 2 produced daily at plant 1}$$
$$x_3 = \text{tons of steel 1 produced daily at plant 2}$$
$$x_4 = \text{tons of steel 2 produced daily at plant 2}$$

Steelco's revenue is given by $170(x_1 + x_3) + 160(x_2 + x_4)$, and Steelco's shipping cost is $80(x_1 + x_2) + 100(x_3 + x_4)$. Therefore, Steelco wants to maximize

$$\begin{aligned} z &= (170 - 80)x_1 + (160 - 80)x_2 + (170 - 100)x_3 + (160 - 100)x_4 \\ &= 90x_1 + 80x_2 + 70x_3 + 60x_4 \end{aligned}$$

Steelco faces the following five constraints:

Constraint 1 At plant 1, no more than 12 tons of coal can be used daily.

Constraint 2 At plant 1, no more than 10 hours of blast furnace time can be used daily.

Constraint 3 At plant 2, no more than 15 tons of coal can be used daily.

Constraint 4 At plant 2, no more than 4 hours of blast furnace time can be used daily.

Constraint 5 At most, 80 tons of iron ore can be used daily.

Constraints 1–5 lead to the following five LP constraints:

$$3x_1 + x_2 \le 12 \qquad \text{(Plant 1 coal constraint)} \tag{23}$$

$$2x_1 + x_2 \le 10 \qquad \text{(Plant 1 furnace constraint)} \qquad (24)$$
$$3x_3 + 2x_4 \le 15 \qquad \text{(Plant 2 coal constraint)} \qquad (25)$$
$$x_3 + x_4 \le 4 \qquad \text{(Plant 2 furnace constraint)} \qquad (26)$$
$$8x_1 + 6x_2 + 7x_3 + 5x_4 \le 80 \qquad \text{(Iron ore constraint)} \qquad (27)$$

We also need the sign restrictions $x_i \ge 0$. Putting it all together, we write Steelco's LP as

$$\begin{aligned}
\max z &= 90x_1 + 80x_2 + 70x_3 + 60x_4 \\
\text{s.t.} \quad & 3x_1 + x_2 \le 12 && \text{(Plant 1 coal constraint)} && (22)\\
& 2x_1 + x_2 \le 10 && \text{(Plant 1 furnace constraint)} && (23)\\
& 3x_3 + 2x_4 \le 15 && \text{(Plant 2 coal constraint)} && (24)\\
& x_3 + x_4 \le 4 && \text{(Plant 2 furnace constraint)} && (25)\\
& 8x_1 + 6x_2 + 7x_3 + 5x_4 \le 80 && \text{(Iron ore constraint)} && (26)\\
& x_1, x_2, x_3, x_4 \ge 0
\end{aligned}$$

Using our definition of decomposition, we may decompose the Steelco LP in the following manner:

Variable set 1 x_1 and x_2 (plant 1 variables).

Variable set 2 x_3 and x_4 (plant 2 variables).

Constraint 1 (22) and (23) (plant 1 constraints).

THEOREM 1

Constraint 2 (24) and (25) (plant 2 constraints).

Constraint 3 (26).

Constraint set 1 and Variable set 1 involve activities at plant 1 and do not involve x_3 and x_4 (which represent plant 2 activities). Constraint set 2 and Variable set 2 involve activities at plant 2 and do not involve x_1 and x_2 (plant 1 activities). Constraint set 3 may be thought of as a centralized constraint that interrelates the two sets of variables. (Solution to be continued.)

Problems in which several plants manufacture several products can easily be decomposed along the lines of Example 3.

To efficiently solve LPs that decompose along the lines of Example 3, Dantzig and Wolfe developed the Dantzig–Wolfe decomposition algorithm. To simplify our discussion of this algorithm, we assume we are solving an LP in which each subproblem has a bounded feasible region.† The decomposition algorithm depends on the results in Theorem 1.

Suppose the feasible region for an LP is bounded and the extreme points (or basic feasible solutions) of the LP's feasible region are $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_k$. Then any point $\mathbf{x}$ in the LP's feasible region may be written as a linear combination of $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_k$. In other words, there exist weights $\mu_1, \mu_2, \ldots, \mu_k$ satisfying

$$\mathbf{x} = \mu_1\mathbf{P}_1 + \mu_2\mathbf{P}_2 + \cdots + \mu_k\mathbf{P}_k \qquad (27)$$

†See Bradley, Hax, and Magnanti (1977) for a discussion of decomposition that includes the case where at least one subproblem has an unbounded feasible region.

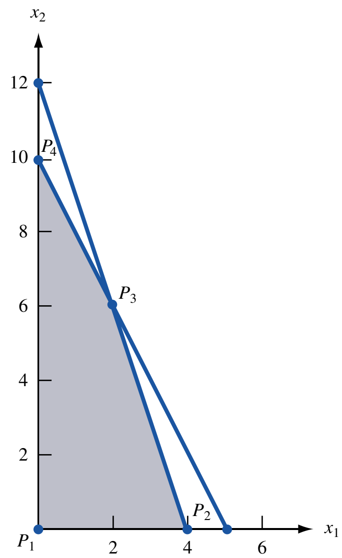


FIGURE 3
Feasible Region for Constraint Set 1

Moreover, the weights $\mu_1, \mu_2, \ldots, \mu_k$ in (27) may be chosen such that

$$\mu_1 + \mu_2 + \cdots + \mu_k = 1 \quad \text{and} \quad \mu_i \geq 0 \quad \text{for } i = 1, 2, \ldots, k \tag{28}$$

Any linear combination of vectors for which the weights satisfy (28) is called a **convex combination.** Thus, Theorem 1 states that if an LP's feasible region is bounded, then any point within may be written as a convex combination of the extreme points of the LP's feasible region.

We illustrate Theorem 1 by showing how it applies to the LPs defined by Constraint set 1 and Constraint set 2 of Example 3. To begin, we look at the feasible region defined by the sign restrictions $x_1 \geq 0$ and $x_2 \geq 0$ and Constraint set 1 (consisting of (22) and (23)). This feasible region is the interior and the boundary of the shaded quadrilateral $P_1P_2P_3P_4$ in Figure 3. The extreme points are $\mathbf{P}_1 = [0 \quad 0]$, $\mathbf{P}_2 = [4 \quad 0]$, $\mathbf{P}_3 = [2 \quad 6]$, and $\mathbf{P}_4 = [0 \quad 10]$. For this feasible region, Theorem 1 states that any point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

in the feasible region for Constraint set 1 may be written as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mu_1 \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \mu_2 \begin{bmatrix} 4 \\ 0 \end{bmatrix} + \mu_3 \begin{bmatrix} 2 \\ 6 \end{bmatrix} + \mu_4 \begin{bmatrix} 0 \\ 10 \end{bmatrix} = \begin{bmatrix} 4\mu_2 + 2\mu_3 \\ 6\mu_3 + 10\mu_4 \end{bmatrix}$$

where $\mu_i \geq 0 (i = 1, 2, 3,4)$ and $\mu_1 + \mu_2 + \mu_3 + \mu_4 = 1$. For example, the point

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

is in the feasible region $P_1P_2P_3P_4$. A glance at Figure 3 shows that

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

may be written as a linear combination of $\mathbf{P}_1$,$\mathbf{P}_2$, and $\mathbf{P}_3$. A little algebra shows that

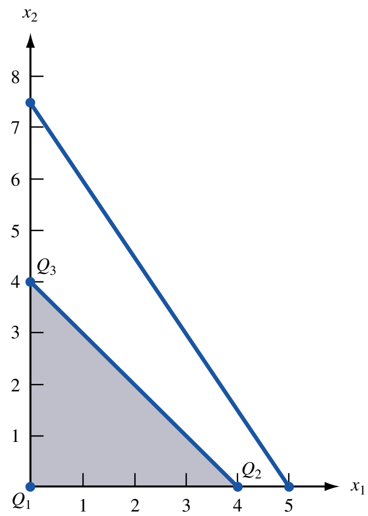


FIGURE 4
Feasible Region for Constraint Set 2

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} 4 \\ 0 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} 2 \\ 6 \end{bmatrix}$$

As another illustration of Theorem 1, consider the feasible region defined by the sign restrictions $x_3 \geq 0$ and $x_4 \geq 0$ and Constraint set 2 [(24) and (25)]. The feasible region for this LP is the shaded area $Q_1Q_2Q_3$ in Figure 4. The extreme points are $\mathbf{Q}_1 = (0, 0)$, $\mathbf{Q}_2 = (4, 0)$, and $\mathbf{Q}_3 = (0, 4)$. Theorem 1 tells us that any point

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

that is in the feasible region for Constraint set 2 may be written as

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \mu_1\begin{bmatrix} 0 \\ 0 \end{bmatrix} + \mu_2\begin{bmatrix} 4 \\ 0 \end{bmatrix} + \mu_3\begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

where $\mu_i \geq 0$ and $\mu_1 + \mu_2 + \mu_3 = 1$. For example, the feasible point

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

may be written as

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 4 \\ 0 \end{bmatrix} + \frac{1}{4}\begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

For our purposes, it is not important to know how to determine the set of weights corresponding to a particular feasible point. The decomposition algorithm does not require us to be able to find the weights for an arbitrary point.

To explain the basic ideas of the decomposition algorithm, we assume that the set of variables has been decomposed into set 1 and set 2. The reader should have no trouble generalizing to a situation where the set of variables is decomposed into more than two sets of variables.

The Dantzig–Wolfe decomposition algorithm proceeds as follows:

Step 1 Let the variables in Variable set 1 be $x_1, x_2, \ldots, x_{n_1}$. Express the variables as a convex combination (see Theorem 1) of the extreme points of the feasible region for Constraint set 1 (the constraints that only involve the variables in Variable set 1). If we let $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_k$ be the extreme points of this feasible region, then any point

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_1} \end{bmatrix}$$

in the feasible region for Constraint set 1 may be written in the form

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_1} \end{bmatrix} = \mu_1\mathbf{P}_1 + \mu_2\mathbf{P}_2 + \cdots + \mu_k\mathbf{P}_k \tag{29}$$

where $\mu_1 + \mu_2 + \cdots + \mu_k = 1$ and $\mu_i \geq 0$ $(i = 1, 2, \ldots, k)$.

Step 2 Express the variables in Variable set 2, $x_{n_1+1}, x_{n_1+2}, \ldots, x_n$, as a convex combination of the extreme points of Constraint set 2's feasible region. If we let the extreme points of the feasible region be $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_m$, then any point in Constraint set 2's feasible region may be written as

$$\begin{bmatrix} x_{n_1+1} \\ x_{n_1+2} \\ \vdots \\ x_n \end{bmatrix} = \lambda_1\mathbf{Q}_1 + \lambda_2\mathbf{Q}_2 + \cdots + \lambda_m\mathbf{Q}_m \tag{30}$$

where $\lambda_i \geq 0$ $(i = 1, 2, \ldots, m)$ and $\lambda_1 + \lambda_2 + \cdots + \lambda_m = 1$.

Step 3 Using (29) and (30), express the LP's objective function and centralized constraints in terms of the μ_i's and the λ_i's. After adding the constraints (called convexity constraints) $\mu_1 + \mu_2 + \cdots + \mu_k = 1$ and $\lambda_1 + \lambda_2 + \cdots + \lambda_m = 1$ and the sign restrictions

$\mu_i \geq 0$ $(i = 1, 2, \ldots, k)$ and $\lambda_i \geq 0$ $(i = 1, 2, \ldots, m)$, we obtain the following LP, which is referred to as the **restricted master:**

$$\begin{array}{ll}
\text{max (or min)} & [\text{objective function in terms of } \mu_i\text{'s and } \lambda_i\text{'s}] \\
\text{s.t.} & [\text{central constraints in terms of } \mu_i\text{'s and } \lambda_i\text{'s}] \\
 & \mu_1 + \mu_2 + \cdots + \mu_k = 1 \qquad \text{(Convexity constraints)} \\
 & \lambda_1 + \lambda_2 + \cdots + \lambda_m = 1 \\
 & \mu_i \geq 0 \quad (i = 1, 2, \ldots, k) \qquad \text{(Sign restrictions)} \\
 & \lambda_i \geq 0 \quad (i = 1, 2, \ldots, m)
\end{array}$$

In many large-scale LPs, the restricted master may have millions of variables (corresponding to the many basic feasible solutions of extreme points for each constraint set). Fortunately, however, we rarely have to write down the entire restricted master; all we need is to generate the column in the restricted master that corresponds to a specific μ_i or λ_i.

Step 4 Assume that a basic feasible solution for the restricted master is readily available.†

†If this is not the case, then the two-phase simplex method must be used. See Bradley, Hax, and Magnanti (1977) for details.

Then use the column generation method of Section 10.3 to solve the restricted master.

Step 5 Substitute the optimal values of the μ_i's and λ_i's found in step 4 into (29) and (30). This will yield the optimal values of $x_1, x_2, \ldots, x_n$.

Solution **Example 3 (Continued)** For Example 3, we have already seen that

$$\text{Variable set } 1 = \{x_1, x_2\} \tag{22}$$

$$\text{Constraint set } 1 = \begin{cases} 3x_1 + x_2 \le 12 \\ 2x_1 + x_2 \le 10 \end{cases} \tag{23}$$

We have also seen that the feasible region for Constraint set 1 has four extreme points, and any feasible point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

for Constraint set 1 may be written as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mu_1 \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \mu_2 \begin{bmatrix} 4 \\ 0 \end{bmatrix} + \mu_3 \begin{bmatrix} 2 \\ 6 \end{bmatrix} + \mu_4 \begin{bmatrix} 0 \\ 10 \end{bmatrix} = \begin{bmatrix} 4\mu_2 + 2\mu_3 \\ 6\mu_3 + 10\mu_4 \end{bmatrix} \tag{29$'$}$$

where $\mu_1 + \mu_2 + \mu_3 + \mu_4 = 1$ and $\mu_i \ge 0$.

$$\text{Variable set } 2 = x_3 \text{ and } x_4 \tag{24}$$

$$\text{Constraint set } 2 = \begin{cases} 3x_3 + 2x_4 \le 15 \\ x_3 + x_4 \le 4 \end{cases} \tag{25}$$

Any point

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

in the feasible region for Constraint set 2 may be written as

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \lambda_1 \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 4 \\ 0 \end{bmatrix} + \lambda_3 \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 4\lambda_2 \\ 4\lambda_3 \end{bmatrix} \tag{30$'$}$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and $\lambda_i \ge 0$ $(i = 1, 2, 3)$.

We now obtain the restricted master by substituting (29′) and (30′) into the objective function and the centralized constraint. The objective function for (21) becomes

$$\begin{aligned} 90x_1 + 80x_2 + 70x_3 + 60x_4 &= 90(4\mu_2 + 2\mu_3) + 80(6\mu_3 + 10\mu_4) + 70(4\lambda_2) + 60(4\lambda_3) \\ &= 360\mu_2 + 660\mu_3 + 800\mu_4 + 280\lambda_2 + 240\lambda_3 \end{aligned}$$

The centralized constraint becomes

$$8(4\mu_2 + 2\mu_3) + 6(6\mu_3 + 10\mu_4) + 7(4\lambda_2) + 5(4\lambda_3) \le 80$$

or

$$32\mu_2 + 52\mu_3 + 60\mu_4 + 28\lambda_2 + 20\lambda_3 \le 80$$

After adding a slack variable s_1 to this constraint and writing down the convexity constraints and the sign restrictions, we obtain the following restricted master program:

$$\begin{aligned} \max z &= 360\mu_2 + 660\mu_3 + 800\mu_4 + 280\lambda_2 + 240\lambda_3 \\ \text{s.t.} \quad & 32\mu_2 + 52\mu_3 + 60\mu_4 + 28\lambda_2 + 20\lambda_3 + s_1 = 80 \end{aligned}$$

$$\mu_1 + \mu_2 + \mu_3 + \mu_4 = 1$$
$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$
$$\mu_i, \lambda_i \geq 0$$

There is a more insightful way to obtain the column for a variable in the restricted master. Recall that each variable in the restricted master corresponds to an extreme point for the feasible region of Constraint set 1 or Constraint set 2. As an example, let's focus on how to find the column in the restricted master for a variable μ_i, which corresponds to an extreme point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

for Constraint set 1. Because x_1 and x_2 correspond to activity at plant 1, we may consider any specification of x_1 and x_2 as a "proposal" from plant 1. For example, the point

$$\begin{bmatrix} 2 \\ 6 \end{bmatrix}$$

corresponds to plant 1 proposing to produce 2 tons of type 1 steel and 6 tons of type 2 steel. Then the weight μ_i may be thought of as a fraction of the proposal corresponding to extreme point $\mathbf{P}_i$ that is included in the actual production schedule. For example, because

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix} = \frac{1}{3}\mathbf{P}_1 + \frac{1}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_3$$

we may think of

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

as consisting of one-third of plant 1 proposal $\mathbf{P}_1$, one-third of plant 1 proposal $\mathbf{P}_2$, and one-third of plant 1 proposal $\mathbf{P}_3$.

We can now describe an easy method to determine the column for any variable in the restricted master. Suppose we want to determine the column for the extreme point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

corresponding to the weight μ_i. If we include a fraction μ_i of the extreme point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

what will this contribute to the objective function? If $\mu_i = 1$, then

$$\mu_i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

will contribute $90x_1 + 80x_2$ to the objective function. By the Proportionality Assumption, if we use a fraction μ_i of the extreme point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

then it will contribute $\mu_i(90x_1 + 80x_2)$ to the objective function. Similarly, if $\mu_i = 1$, then

$$\mu_i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

will contribute $8x_1 + 6x_2$ of iron usage. Thus, for an arbitrary value of μ_i,

$$\mu_i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

will contribute an amount $\mu_i(8x_1+6x_2)$ to the left-hand side of the iron ore usage constraint.

To be more specific, let's use the reasoning we have just described to determine the column in the restricted master for the weight μ_3 corresponding to the extreme point

$$\begin{bmatrix} 2 \\ 6 \end{bmatrix}$$

Our logic shows that the left-hand side of the objective function involving μ_3 is μ_3 $[90(2) + 80(6)] = 660\mu_3$. Similarly, the term involving μ_3 on the left-hand side of the iron ore constraint will be $\mu_3[8(2) + 6(6)] = 52\mu_3$. Also, μ_3 will have a coefficient of 1 in the first convexity constraint and a zero coefficient in the other convexity constraint. (If the reader understood how we obtained the μ_3 column, there should be little trouble with what follows; readers who are confused should reread the last two pages before continuing.)

We now solve the restricted master by using the revised simplex method and column generation. We refer to our initial tableau as tableau 0. Then $\text{BV}(0) = \{s_1, \mu_1, \lambda_1\}$. Also,

$$B_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \text{so} \qquad B_0^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Because s_1, μ_1, and λ_1 don't appear in the objective function of the restricted master, we have $\mathbf{c}_{\text{BV}} = [0 \quad 0 \quad 0]$, and the tableau 0 shadow prices are given by

$$\mathbf{c}_{\text{BV}}B_0^{-1} = [0 \quad 0 \quad 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [0 \quad 0 \quad 0]$$

We now apply the idea of column generation in two stages. First, we determine whether there is any weight μ_i associated with Constraint set 1 that prices out favorably (because we are solving a max problem, a negative coefficient in row 0 is favorable). A weight μ_i associated with an extreme point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

of Constraint set 1 will have the following column in the restricted master:

$$\text{Objective function coefficient for } \mu_i = 90x_1 + 80x_2$$

$$\text{Column in constraints for } \mu_i = \begin{bmatrix} 8x_1 + 6x_2 \\ 1 \\ 0 \end{bmatrix}$$

From this information, we see that in tableau 0, the column for the weight μ_i corresponding to

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

will price out to

$$\mathbf{c}_{BV}B_0^{-1}\begin{bmatrix} 8x_1 + 6x_2 \\ 1 \\ 0 \end{bmatrix} - (90x_1 + 80x_2) = -90x_1 - 80x_2$$

Since

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

must satisfy Constraint set 1 (or the plant 1 constraints), the weight μ_i that prices out most negatively will be the weight associated with the extreme point that is the optimal solution to the following LP:

Tableau 0
Plant 1 Subproblem

$$\begin{aligned} \min z &= -90x_1 - 80x_2 \\ \text{s.t.} \quad 3x_1 + x_2 &\le 12 \\ 2x_1 + x_2 &\le 10 \\ x_1, x_2 &\ge 0 \end{aligned}$$

Solving the plant 1 subproblem graphically, we obtain the solution $z = -800$, $x_1 = 0$, $x_2 = 10$. This means that the weight μ_i associated with the extreme point

$$\begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

will price out most negatively. Recall that

$$\mathbf{P}_4 = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

This means that μ_4 will price out with a coefficient of -800 in the restricted master.

We now look at the weights associated with Constraint set 2 and try to determine the weight λ_i that will price out most negatively. The λ_i corresponding to an extreme point

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

of Constraint set 2 will have the following column in the restricted master:

$$\text{Objective function coefficient for } \lambda_i = 70x_3 + 60x_4$$

$$\text{Column in constraints for } \lambda_i = \begin{bmatrix} 7x_3 + 5x_4 \\ 0 \\ 1 \end{bmatrix}$$

This means that the λ_i corresponding to the extreme point

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

will price out to

$$\mathbf{c}_{BV}B_0^{-1}\begin{bmatrix} 7x_3 + 5x_4 \\ 0 \\ 1 \end{bmatrix} - (70x_3 + 60x_4) = -70x_3 - 60x_4$$

Note that

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

must satisfy Constraint set 2. Thus, the extreme point whose weight λ_i prices out most favorably will be the solution to the following LP:

Tableau 0
Plant 2 Subproblem

$$\begin{aligned} \min z &= -70x_3 - 60x_4 \\ \text{s.t.} \quad 3x_3 + 2x_4 &\le 15 \\ x_3 + x_4 &\le 4 \\ x_3, x_4 &\ge 0 \end{aligned}$$

The optimal solution to this LP is $z = -280$, $x_3 = 4$, $x_4 = 0$. Because

$$\begin{bmatrix} 4 \\ 0 \end{bmatrix} = \mathbf{Q}_2$$

λ_2 prices out the most negatively of all the λ_i's. But μ_4 prices out more negatively than λ_2, so we enter μ_4 into the basis (by using the revised simplex procedure). To do this, we need to find the column for μ_4 in tableau 0 and also find the right-hand side of tableau 0. The column for μ_4 in tableau 0 is

$$B_0^{-1}\begin{bmatrix} 8(0) + 6(10) \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 60 \\ 1 \\ 0 \end{bmatrix}$$

and the right-hand side of tableau 0 is

$$B_0^{-1}\mathbf{b} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 80 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 80 \\ 1 \\ 1 \end{bmatrix}$$

The ratio test now indicates that μ_4 should enter the basis in the second constraint. Then $\text{BV}(1) = \{s_1, \mu_4, \lambda_1\}$. Because

$$E_0 = \begin{bmatrix} 1 & -60 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_1^{-1} = E_0 B_0^{-1} = \begin{bmatrix} 1 & -60 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The objective function coefficient for μ_4 is $90(0) + 80(10) = 800$, so the new set of shadow prices may be found from

$$\mathbf{c}_{\text{BV}}B_1^{-1} = [0 \quad 800 \quad 0]\begin{bmatrix} 1 & -60 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [0 \quad 800 \quad 0]$$

We now try to find the weight that prices out most negatively in the current tableau. As before, we solve the current tableau's plant 1 and plant 2 subproblems. Also, as before, a weight μ_i that corresponds to a Constraint 1 extreme point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

will price out to

$$\mathbf{c}_{\mathrm{BV}}B_1^{-1}\begin{bmatrix} 8x_1 + 6x_2 \\ 1 \\ 0 \end{bmatrix} - (90x_1 + 80x_2)$$

$$= [0 \quad 800 \quad 0]\begin{bmatrix} 8x_1 + 6x_2 \\ 1 \\ 0 \end{bmatrix} - (90x_1 + 80x_2) = 800 - 90x_1 - 80x_2$$

Because

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

must satisfy Constraint set 1, the μ_i that prices out most favorably will correspond to the point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

that solves the following LP:

Tableau 1
Plant 1 Subproblem

$$\begin{aligned} \min z &= 800 - 90x_1 - 80x_2 \\ \text{s.t.} \quad & 3x_1 + x_2 \le 12 \\ & 2x_1 + x_2 \le 10 \\ & x_1, x_2 \ge 0 \end{aligned}$$

The optimal solution to this LP is $z = 0$, $x_1 = 0$, $x_2 = 10$. This means that no μ_i can price out favorably. We now solve the plant 2 subproblem in an effort to find a λ_i that prices out favorably. A λ_i corresponding to an extreme point

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

of Constraint set 2 will price out to

$$\mathbf{c}_{\mathrm{BV}}B_1^{-1}\begin{bmatrix} 7x_3 + 5x_4 \\ 0 \\ 1 \end{bmatrix} - (70x_3 + 60x_4) = -70x_3 - 60x_4$$

Because

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

must satisfy the plant 2 constraints, the λ_i that will price out most negatively will correspond to the extreme point

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

that solves the plant 2 subproblem for tableau 1:

Tableau 1
Plant 2 Subproblem

$$\begin{aligned} \min z &= -70x_3 - 60x_4 \\ \text{s.t.} \quad & 3x_3 + 2x_4 \le 15 \end{aligned}$$

$$x_3 + \quad x_4 \le 4$$
$$x_3, x_4 \ge 0$$

The optimal solution to this LP is $x_3 = 4$, $x_4 = 0$, $z = -280$. This means that the λ_i corresponding to

$$\begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

prices out to -280. Because

$$\begin{bmatrix} 4 \\ 0 \end{bmatrix} = \mathbf{Q}_2$$

λ_2 prices out to -280. No μ_i has priced out negatively, so the best we can do is to enter λ_2 into the basis. To enter λ_2 into the basis, we need the column for λ_2 in tableau 1 and the right-hand side for tableau 1. The column for λ_2 in tableau 1 is given by

$$\mathbf{c}_{\mathrm{BV}}B_2^{-1} = [280 \quad 800 \quad 0]\begin{bmatrix} \quad & \quad & \quad \end{bmatrix} = [10 \quad 200 \quad 0]$$

By solving the plant 1 subproblem for tableau 2, we can determine whether any μ_i prices out favorably. The μ_i corresponding to

$$\begin{bmatrix} \quad \end{bmatrix}$$

prices out to

$$\mathbf{c}_{\mathrm{BV}}B_2^{-1}\begin{bmatrix} \quad \end{bmatrix} - (90x_1 + 80x_2)$$

$$= [10 \quad 200 \quad 0]\begin{bmatrix} \quad \end{bmatrix} - (90x_1 + 80x_2) = 200 - 10x_1 - 20x_2$$

Thus, we have

Tableau 2
Plant 1 Subproblem

$$\begin{aligned} \min z &= 200 - 10x_1 - 20x_2 \\ \text{s.t.} \quad 3x_1 + x_2 &\le 12 \\ 2x_1 + x_2 &\le 10 \\ x_1, x_2 &\ge 0 \end{aligned}$$

The optimal solution to this LP is $z = 0$, $x_1 = 0$, $x_2 = 10$. As before, this means that no μ_i can price out favorably.

To determine whether the λ_i corresponding to the extreme point

$$\begin{matrix} \frac{1}{28} & -\frac{60}{28} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{28} & \frac{60}{28} & 1 \end{matrix}$$

$$\begin{bmatrix} \quad \end{bmatrix}$$

out favorably. The μ_i corresponding to

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

prices out to

$$\mathbf{c}_{\mathrm{BV}}B_2^{-1}\begin{bmatrix} 8x_1 + 6x_2 \\ 1 \\ 0 \end{bmatrix} - (90x_1 + 80x_2)$$

$$= [10 \quad 200 \quad 0]\begin{bmatrix} 8x_1 + 6x_2 \\ 1 \\ 0 \end{bmatrix} - (90x_1 + 80x_2) = 200 - 10x_1 - 20x_2$$

Thus, we have

Tableau 2
Plant 1 Subproblem

$$\begin{aligned} \min z &= 200 - 10x_1 - 20x_2 \\ \text{s.t.} \quad & 3x_1 + x_2 \le 12 \\ & 2x_1 + x_2 \le 10 \\ & x_1, x_2 \ge 0 \end{aligned}$$

The optimal solution to this LP is $z = 0$, $x_1 = 0$, $x_2 = 10$. As before, this means that no μ_i can price out favorably.

To determine whether the λ_i corresponding to the extreme point

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

should be entered into the basis, observe that it prices out to

$$[10 \quad 200 \quad 0]\begin{bmatrix} 7x_3 + 5x_4 \\ 0 \\ 1 \end{bmatrix} - (70x_3 + 60x_4) = -10x_4$$

Because

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

must satisfy Constraint set 2, the λ_i that prices out most favorably will be the λ_i associated with the point

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

that solves the following LP:

Tableau 2
Plant 2 Subproblem

$$\begin{aligned} \min z &= -10x_4 \\ \text{s.t.} \quad & 3x_3 + 2x_4 \le 15 \\ & x_3 + x_4 \le 4 \\ & x_3, x_4 \ge 0 \end{aligned}$$

This LP has the solution $z = -40$, $x_3 = 0$, $x_4 = 4$. Thus, the λ_i corresponding to

$$\begin{bmatrix} 0 \\ 4 \end{bmatrix} = \mathbf{Q}_3$$

should enter the basis, and λ_3 should be entered into the basis. The λ_3 column in tableau 2 is

$$B_2^{-1}\begin{bmatrix} 7(0) + 5(4) \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{28} & -\frac{60}{28} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{28} & \frac{60}{28} & 1 \end{bmatrix}\begin{bmatrix} 20 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{20}{28} \\ 0 \\ \frac{8}{28} \end{bmatrix}$$

Tableau 2's right-hand side is

$$B_2^{-1}\mathbf{b} = \begin{bmatrix} \frac{1}{28} & -\frac{60}{28} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{28} & \frac{60}{28} & 1 \end{bmatrix}\begin{bmatrix} 80 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{20}{28} \\ 1 \\ \frac{8}{28} \end{bmatrix}$$

The ratio test indicates that λ_3 should enter the basis in Constraint 1 or Constraint 3; we arbitrarily choose Constraint 1. Thus, BV(3) $= \{\lambda_3, \mu_4, \lambda_1\}$. Because

$$E_2 = \begin{bmatrix} \frac{28}{20} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{2}{5} & 0 & 1 \end{bmatrix}$$

$$B_3^{-1} = E_2B_2^{-1} = \begin{bmatrix} \frac{28}{20} & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{2}{5} & 0 & 1 \end{bmatrix}\begin{bmatrix} \frac{1}{28} & -\frac{60}{28} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{28} & \frac{60}{28} & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{20} & -3 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{20} & 3 & 1 \end{bmatrix}$$

λ_3 corresponds to

$$\begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

so the coefficient of λ_3 in the objective function of the restricted master is $70x_3 + 60x_4 = 70(0) + 60(4) = 240$. The μ_4 and λ_1 coefficients in the objective function have already been found to be 800 and 0, respectively, so we have $\mathbf{c}_{\text{BV}} = [240 \quad 800 \quad 0]$, and the new set of shadow prices is given by

$$\mathbf{c}_{\text{BV}}B_3^{-1} = [240 \quad 800 \quad 0]\begin{bmatrix} \frac{1}{20} & -3 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{20} & 3 & 1 \end{bmatrix} = [12 \quad 80 \quad 0]$$

With these shadow prices, the μ_i corresponding to the extreme point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

will price out to

$$[12 \quad 80 \quad 0]\begin{bmatrix} 8x_1 + 6x_2 \\ 1 \\ 0 \end{bmatrix} - (90x_1 + 80x_2) = 80 + 6x_1 - 8x_2$$

Then we have

Tableau 3
Plant 1 Subproblem

$$\min z = 80 + 6x_1 - 8x_2$$
$$\text{s.t.} \quad 3x_1 + x_2 \le 12$$

$$
\begin{aligned}
2x_1 + x_2 &\le 10 \\
x_1, x_2 &\ge 0
\end{aligned}
$$

The optimal solution to this LP is $z = 0$, $x_1 = 0$, $x_2 = 10$. Again, this means that no μ_i prices out favorably.

Using the new shadow prices, we now determine whether any λ_i will price out favorably. If no λ_i prices out favorably, then we will have found an optimal tableau. The λ_i corresponding to

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

will price out to

$$[12 \quad 80 \quad 0]\begin{bmatrix} 7x_3 + 5x_4 \\ 0 \\ 1 \end{bmatrix} - (70x_3 + 60x_4) = 14x_3$$

Then we have

Tableau 3
Plant 2 Subproblem

$$
\begin{aligned}
\min z &= 14x_3 \\
\text{s.t.} \quad 3x_3 + 2x_4 &\le 15 \\
x_3 + x_4 &\le 4 \\
x_3, x_4 &\ge 0
\end{aligned}
$$

The optimal solution to this LP is $z = 0$, $x_3 = x_4 = 0$. This means that no λ_i can price out favorably. Because no μ_i or λ_i prices out favorably for tableau 3, tableau 3 must be an optimal tableau for the restricted master. Recall that BV(3) $= \{\lambda_3, \mu_4, \lambda_1\}$. Thus,

$$\begin{bmatrix} \lambda_3 \\ \mu_4 \\ \lambda_1 \end{bmatrix} = B_3^{-1} = \begin{bmatrix} \frac{1}{20} & -3 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{20} & 3 & 1 \end{bmatrix}\begin{bmatrix} 80 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

Thus the optimal solution to the restricted master is $\lambda_3 = 1$, $\mu_4 = 1$, $\lambda_1 = 0$, and all other weights equal 0.

We can now use the representation of the Constraint set 1 feasible region as a convex combination of its extreme points to determine that the optimal value of

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

is given by

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0\mathbf{P}_1 + 0\mathbf{P}_2 + 0\mathbf{P}_3 + \mathbf{P}_4 = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

Similarly, we can use the representation of the Constraint set 2 feasible region as a convex combination of its extreme points to determine that the optimal value of

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

is given by

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = 0\mathbf{Q}_1 + 0\mathbf{Q}_2 + \mathbf{Q}_3 = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

Then the optimal solution to Steelco's problem is $x_2 = 10$, $x_4 = 4$, $x_1 = x_3 = 0$, $z = 1040$. Thus, Steelco can maximize its net profit by manufacturing 10 tons of steel 2 at plant 1 and 4 tons of steel 2 at plant 2.

REMARKS **1** If there are k sets of variables, then the restricted master will contain the central constraints and k convexity constraints (one convexity constraint for each set of variables). For each tableau, there will also be k subproblems that must be solved (one for the weights associated with the extreme points of the constraint set corresponding to each set of variables). After solving these subproblems, use the revised simplex algorithm to enter into the basis the weight that prices out most favorably.

2 A major virtue of decomposition is that solving several relatively small LPs is often much easier than solving one large LP. For example, consider an analog of Example 3 in which there are five plants and each plant has 50 constraints. Also suppose that there are 40 central constraints. Then the master problem will involve a 45×45 B^{-1}, and each subproblem will involve a 50×50 B^{-1}. Solving the original LP would involve a 290×290 B^{-1}. Clearly, storing a 290×290 matrix requires more computer memory than storing five 50×50 matrices and a 45×45 matrix. This illustrates how decomposition greatly reduces storage requirements.

3 Decomposition has an interesting economic interpretation. What is the meaning of the shadow prices for the restricted master of Example 3? For each tableau, the shadow price for the central constraint (reflecting the limited amount of iron ore) is the amount by which an extra unit of iron would increase profits. It can be shown that for any tableau, the shadow price for the plant i ($i = 1, 2$) convexity constraint is the profit obtained from the current mix of extreme points being used at plant i less the value of the centralized resource (calculated via the centralized shadow price) required by the current mix of extreme points that is being used at plant i. For example, in tableau 3, the shadow price for the plant 1 convexity constraint is 80. Currently, plant 1 is utilizing the mix $x_1 = 0$ and $x_2 = 10$. This mix yields a profit of $80(10) = \$800$, and it uses $6(10) = 60$ tons of iron worth $60(12) = \$720$. Thus, the plant 1 convexity constraint has a shadow price of $800 - 720 = \$80$. This means that if Δ of the plant 1 weight were taken away, profits would be reduced by 80Δ.

We can now give an economic interpretation of the pricing-out procedure that we use to generate our subproblems. If we are at tableau 3, what are the benefits and costs if we try to introduce the μ_i associated with the extreme point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

into the basis? Recall that for tableau 3, the iron shadow price is 12 and the plant 1 convexity constraint has a shadow price of 80. In determining whether μ_i should enter the basis, we must balance

Increased profits for μ_i = profits earned by $\mu_i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$= 90(\mu_i x_1) + 80(\mu_i x_2)$

against the costs incurred if μ_i is entered into the basis.

If we enter μ_i into the basis, we incur two costs: first, \$12 for each ton of iron used. This amounts to a cost of $12[8(\mu_i x_1) + 6(\mu_i x_2)]$. By entering μ_i into the basis, we are also diverting a fraction μ_i of the available plant 1 weights away from the current mix. This incurs an opportunity cost of $80\mu_i$. Hence,

Increase in cost from entering μ_i into basis $= 96\mu_i x_1 + 72\mu_i x_2 + 80\mu_i$

This means that entering μ_i into the basis can increase profits if and only if

$$90\mu_i x_1 + 80\mu_i x_2 > 96\mu_i x_1 + 72\mu_i x_2 + 80\mu_i$$

Canceling the μ_i's from both sides, we see that μ_i will price out favorably if

$$90x_1 + 80x_2 > 96x_1 + 72x_2 + 80 \qquad \text{or} \qquad 0 > 80 + 6x_1 - 8x_2$$

Thus, the best μ_i will be the μ_i associated with the extreme point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

that minimizes $80 + 6x_1 - 8x_2$. This is indeed the objective function for the plant 1 tableau 3 subproblem.

This discussion shows that the Dantzig–Wolfe decomposition algorithm combines centralized information (from the shadow prices of the centralized constraints) with local information (the shadow price of each plant's convexity constraint) in an effort to determine which weights should be entered into the basis (or equivalently, which extreme points from each plant should be used).

PROBLEMS

Group A

Use the Dantzig–Wolfe decomposition algorithm to solve the following problems:

1

$$\begin{aligned} \max z = 7x_1 + 5x_2 + 3x_3 & \\ \text{s.t.} \quad x_1 + 2x_2 + x_3 &\le 10 \\ x_2 + x_3 &\le 5 \\ x_1 &\le 3 \\ 2x_2 + x_3 &\le 8 \\ x_1, x_2, x_3 &\ge 0 \end{aligned}$$

2

$$\begin{aligned} \max z = 4x_1 + 2x_2 + 3x_3 + 4x_4 + 2x_5 & \\ \text{s.t.} \quad x_1 + 2x_2 + \ \ x_3 &\le 8 \\ x_1 + 2x_2 + 2x_3 &\le 8 \\ x_4 + x_5 &\le 3 \\ x_1, x_2, x_3, x_4, x_5 &\ge 0 \end{aligned}$$

3

$$\begin{aligned} \max z = 3x_1 + 6x_2 + 5x_3 & \\ \text{s.t.} \quad x_1 + 2x_2 + \ \ x_3 &\le 4 \\ 2x_1 + 3x_2 + 2x_3 &\le 6 \\ x_1 + \ \ x_2 &\le 2 \\ 2x_1 + \ \ x_2 &\le 3 \\ x_1, x_2, x_3 &\ge 0 \end{aligned}$$

(*Hint:* There is no law against having only one set of variables and one subproblem.)

4 Give an economic interpretation to explain why λ_3 priced out favorably in the plant 2 tableau 2 subproblem.

5 Give an example to show why Theorem 1 does not hold for an LP with an unbounded feasible region.

10.5 The Simplex Method for Upper-Bounded Variables

Often, LPs contain many constraints of the form $x_i \le u_i$ (where u_i is a constant). For example, in a production-scheduling problem, there may be many constraints of the type

EXAMPLE 4 Simplex with Upper Bounds 1

$x_i \le u_i$, where

$$x_i = \text{period } i \text{ production}$$
$$u_i = \text{period } i \text{ production capacity}$$

Because a constraint of the form $x_i \le u_i$ provides an upper bound on x_i, it is called an **upper-bound constraint.** Because $x_i \le u_i$ is a legal LP constraint, we can clearly use the ordinary simplex method to solve an LP that has upper-bound constraints. However, if an LP contains several upper-bound constraints, then the procedure described in this section (called the simplex method for upper-bounded variables) is much more efficient than the ordinary simplex algorithm.

To efficiently solve an LP with upper-bound constraints, we allow the variable x_i to be nonbasic if $x_i = 0$ (the usual criterion for a nonbasic variable) or if $x_i = u_i$. To accomplish this, we use the following gimmick: For each variable x_i that has an upper-bound constraint $x_i \le u_i$, we define a new variable x_i' by the relationship $x_i + x_i' = u_i$, or $x_i = u_i - x_i'$. Note that if $x_i = 0$, then $x_i' = u_i$, whereas if $x_i = u_i$, then $x_i' = 0$. Whenever we want x_i to equal its upper bound of u_i, we simply replace x_i by $u_i - x_i'$. This is called an **upper-bound substitution.**

We are now ready to describe the simplex method for upper-bounded variables. We assume that a basic solution is available and that we are solving a max problem. As usual, at each iteration, we choose to increase the variable x_i that has the most negative coefficient in row 0. Three possible occurrences, or bottlenecks, can restrict the amount by which we increase x_i:

Bottleneck 1 x_i cannot exceed its upper bound of u_i.

Bottleneck 2 x_i increases to a point where it causes one of the current basic variables to become negative. The smallest value of x_i that will cause one of the current basic variables to become negative may be found by expressing each basic variable in terms of x_i (recall that we used this idea in Chapter 4, in discussing the simplex algorithm).

Bottleneck 3 x_i increases to a point where it causes one of the current basic variables to exceed its upper bound. As in bottleneck 2, the smallest value of x_i for which this bottleneck occurs can be found by expressing each basic variable in terms of x_i.

Let BN_k ($k = 1, 2, 3$) be the value of x_i where bottleneck k occurs. Then x_i can be increased only to a value of $\min\{BN_1, BN_2, BN_3\}$. The smallest of BN_1, BN_2, and BN_3 is called the winning bottleneck. If the winning bottleneck is BN_1, then we make an upper-bound substitution on x_i by replacing x_i by $u_i - x_i'$. If the winning bottleneck is BN_2, then we enter x_i into the basis in the row corresponding to the basic variable that caused BN_2

TABLE 3
Initial Tableau for Example 5

	Basic Variable
$z - 4x_1 - 2x_2 - 3x_3 = 0$	$z = 0$
$2x_1 + x_2 + x_3 + s_1 = 10$	$s_1 = 10$
$x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_3 + s_2 = 6$	$s_2 = 6$
$2x_1 + 2x_2 + 4x_3 + s_3 = 20$	$s_3 = 20$

TABLE 4

Replacing x_1 by $4 - x_1'$

	Basic Variable
$z + 4x_1' - 2x_2 - 3x_3 = 16$	$z = 16$
$-2x_1' + x_2 + x_3 + s_1 = 2$	$s_1 = 2$
$-x_1' + \frac{1}{2}x_2 + \frac{1}{2}x_3 + s_2 = 2$	$s_2 = 2$
$-2x_1' + 2x_2 + 4x_3 + s_3 = 12$	$s_3 = 12$

TABLE 5

Replacing x_3 by $1 - x_3'$

	Basic Variable
$z + 4x_1' - 2x_2 + 3x_3' = 19$	$z = 19$
$-2x_1' + x_2 - x_3' + s_1 = 1$	$s_1 = 1$
$-x_1' + \frac{1}{2}x_2 - \frac{1}{2}x_3' + s_2 = \frac{3}{2}$	$s_2 = \frac{3}{2}$
$-2x_1' + 2x_2 - 4x_3' + s_3 = 8$	$s_3 = 8$

TABLE 6

Optimal Tableau for Example 4

	Basic Variable
$z + x_3' + 2s_1 = 21$	$z = 21$
$-2x_1' + x_2 - x_3' + s_1 = 1$	$x_2 = 1$
$-\frac{1}{2}s_1 + s_2 = 1$	$s_2 = 1$
$2x_1' - 2x_3' - 2s_1 + s_3 = 6$	$s_3 = 6$

to occur. If the winning bottleneck is BN_3, then we make an upper-bound substitution of the variable x_j (by replacing x_j by $u_j - x_j'$) that reaches its upper bound when $x_i = BN_3$. Then we enter x_i into the basis in the row for which x_j was a basic variable.

After following this procedure, we examine the new row 0. If each variable has a nonnegative coefficient in row 0, then we have obtained an optimal tableau. Otherwise, we try to increase the variable with the most negative coefficient in row 0. Our procedure ensures (through BN_1 and BN_3) that no upper-bound constraint is ever violated and (through BN_2) that all of the nonnegativity constraints are satisfied.

EXAMPLE 5

Solve the following LP:

$$\begin{aligned}
\max z &= 4x_1 + 2x_2 + 3x_3 \\
\text{s.t.} \quad & 2x_1 + x_2 + x_3 \le 10 \\
& x_1 + \tfrac{1}{2}x_2 + \tfrac{1}{2}x_3 \le 6 \\
& 2x_1 + 2x_2 + 4x_3 \le 20 \\
& x_1 \le 4 \\
& x_2 \le 3 \\
& x_3 \le 1
\end{aligned}$$

TABLE 7
Initial Tableau for Example 6

	Basic Variable
$z \quad - 6x_3 = 0$	$z = 0$
$x_1 \quad - \quad x_3 = 6$	$x_1 = 6$
$x_2 + 2x_3 = 8$	$x_2 = 8$

TABLE 8
Replacing x_1 by $8 - x_1'$

	Basic Variable
$z \quad - 6x_3 = 0$	$z = 0$
$x_1' \quad + \quad x_3 = 2$	$x_1' = 6$
$x_2 + 2x_3 = 8$	$x_2 = 8$

TABLE 9
Optimal Tableau for Example 5

	Basic Variable
$z + 6x_1' = 12$	$z = 12$
$x_1' \quad + x_3 = 2$	$x_3 = 2$
$- 2x_1' + x_2 = 4$	$x_2 = 4$

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

Solution The initial tableau for this problem is given in Table 3. Because x_1 has the most negative coefficient in row 0, we try to increase x_1 as much as we can. The three bottlenecks for x_1 are computed as follows: x_1 cannot exceed its upper bound of 4, so $BN_1 = 4$. To compute BN_2, we solve for the current set of basic variables in terms of x_1:

$$\begin{aligned} s_1 &= 10 - 2x_1 \qquad (s_1 \geq 0 \text{ iff } x_1 \leq 5) \\ s_2 &= 6 - x_1 \qquad (s_2 \geq 0 \text{ iff } x_1 \leq 6) \\ s_3 &= 20 - 2x_1 \qquad (s_3 \geq 0 \text{ iff } x_1 \leq 10) \end{aligned}$$

Hence, $BN_2 = \min\{5, 6, 10\} = 5$. The current basic variables ($\{s_1, s_2, s_3\}$) have no upper bounds, so there is no value of BN_3. Then the winning bottleneck is $\min\{4, 5\} = 4 = BN_1$. Thus, we must make an upper-bound substitution on x_1 by replacing x_1 by $4 - x_1'$. The resulting tableau is Table 4.

Because x_3 has the most negative coefficient in row 0, we try to increase x_3 as much as possible. The x_3 bottlenecks are computed as follows: x_3 cannot exceed its upper bound of 1, so $BN_1 = 1$. For BN_2, we solve for the current set of basic variables in terms of x_3:

$$\begin{aligned} s_1 &= 2 - x_3 \qquad (s_1 \geq 0 \text{ iff } x_3 \leq 2) \\ s_2 &= 2 - \tfrac{1}{2}x_3 \qquad (s_2 \geq 0 \text{ iff } x_3 \leq 4) \\ s_3 &= 12 - 4x_3 \qquad (s_3 \geq 0 \text{ iff } x_3 \leq 3) \end{aligned}$$

Thus, $BN_2 = \min\{2, 4, 3\} = 2$. Because s_1, s_2, and s_3 do not have an upper bound, there is no BN_3. The winning bottleneck is $\min\{1, 2\} = BN_1 = 1$, so we make an upper-bound substitution on x_3 by replacing x_3 by $1 - x_3'$. The resulting tableau is Table 5.

Because x_2 now has the most negative coefficient in row 0, we try to increase x_2. The computation of the bottlenecks follows: For BN_1, x_2 cannot exceed its upper bound of 3, so $BN_1 = 3$. For BN_2,

$$\begin{aligned} s_1 &= 1 - x_2 \qquad (s_1 \geq 0 \text{ iff } x_2 \leq 1) \\ s_2 &= \tfrac{3}{2} - \tfrac{1}{2}x_2 \qquad (s_2 \geq 0 \text{ iff } x_2 \leq 3) \\ s_3 &= 8 - 2x_2 \qquad (s_3 \geq 0 \text{ iff } x_2 \leq 4) \end{aligned}$$

Thus, $BN_2 = \min\{1, 3, 4\} = 1$. Note that BN_2 occurs because s_1 is forced to zero. None of the basic variables in the current set has an upper-bound constraint, so there is no BN_3. The winning bottleneck is $\min\{3, 1\} = 1 = BN_2$, so x_2 will enter the basis in the row in which s_1 was a basic variable (row 1). After the pivot is performed, the new tableau is Table 6. Because each variable has a nonnegative coefficient in row 0, this is an optimal tableau. Thus, the optimal solution to the LP is $z = 21$, $s_2 = 1$, $x_2 = 1$, $s_3 = 6$, $x_1' = 0$, $s_1 = 0$, $x_3' = 0$. Because $x_1' = 4 - x_1$ and $x_3' = 1 - x_3$, we also have $x_1 = 4$ and $x_3 = 1$.

Solve the following LP:

$$\max z = 6x_3$$

$$\text{s.t.} \quad x_1 - x_3 = 6$$

$$x_2 + 2x_3 = 8$$

$$x_1 \le 8,\ x_2 \le 10,\ x_3 \le 5;\ x_1, x_2, x_3 \ge 0$$

Solution After putting the objective function in our standard row 0 format, we obtain the tableau in Table 7. Fortunately, the basic feasible solution $z = 0$, $x_1 = 6$, $x_2 = 8$, $x_3 = 0$ is readily apparent. We can now proceed with the simplex method for upper-bounded variables. Because x_3 has the most negative coefficient in row 0, we try to increase x_3. Because x_3 cannot exceed its upper bound of 5, $BN_1 = 5$. To compute BN_2,

$$x_1 = 6 + x_3 \qquad (x_1 \ge 0 \text{ iff } x_3 \ge -6)$$

$$x_2 = 8 - 2x_3 \qquad (x_2 \ge 0 \text{ iff } x_3 \le 4)$$

Thus, all the current basic variables will remain nonnegative as long as $x_3 \le 4$. Hence, $BN_2 = 4$. For BN_3, note that $x_1 \le 8$ will hold iff $6 + x_3 \le 8$, or $x_3 \le 2$. Also, $x_2 \le 10$ will hold iff $8 - 2x_3 \le 10$, or $x_3 \ge -1$. Thus, for $x_3 \le 2$, each basic variable remains less than or equal to its upper bound, so $BN_3 = 2$. Note that BN_3 occurs when the basic variable x_1 attains its upper bound. The winning bottleneck is $\min\{5, 4, 2\} = 2 = BN_3$, so the largest that we can make x_3 is 2, and the bottleneck occurs because x_1 attains its upper bound of 8. Thus, we make an upper-bound substitution on x_1 by replacing x_1 by $8 - x_1'$. The resulting tableau is

$$z - 6x_3 = 0$$

$$-x_1' - x_3 = -2$$

$$x_2 + 2x_3 = 8$$

After rewriting $-x_1' - x_3 = -2$ as $x_1' + x_3 = 2$, we obtain the tableau in Table 8.

Because x_1, the variable that caused BN_3, was basic in row 1, we now make x_3 a basic variable in row 1. After the pivot, we obtain the tableau in Table 9, which is optima. Thus, the optimal solution to the LP is $z = 12$, $x_3 = 2$, $x_2 = 4$, $x_1' = 0$. Because $x_1' = 0$, $x_1 = 8 - x_1' = 8$.

To illustrate the efficiencies obtained by using the simplex algorithm with upper bounds, suppose we are solving an LP (call it LP 1) with 100 variables, each having an upper-bound constraint, with five other constraints. If we were to solve LP 1 by the revised simplex method, the B^{-1} for each tableau would be a 105×105 matrix. If we were to use the simplex method for upper-bounded variables, however, the B^{-1} for each tableau would be only a 5×5 matrix. Although the computation of the winning bottleneck in each iteration is more complicated than the ordinary ratio test, solving LP 1 by the simplex method for upper-bounded variables would still be much more efficient than by the ordinary revised simplex.

PROBLEMS

Use the upper-bounded simplex algorithm to solve the following LPs:

Group A

1

$$\begin{aligned} \max z = 4x_1 + 3x_2 + 5x_3 & \\ \text{s.t.} \quad 2x_1 + 2x_2 + x_3 + x_4 &\le 9 \\ 4x_1 - x_2 - x_3 + x_5 &\le 6 \\ 2x_2 + x_3 &\le 5 \\ x_1 &\le 2 \\ x_2 &\le 3 \\ x_3 &\le 4 \\ x_4 &\le 5 \\ x_5 &\le 7 \\ x_1, x_2, x_3, x_4, x_5 &\ge 0 \end{aligned}$$

2

$$\begin{aligned} \min z = -4x_1 - 9x_2 & \\ \text{s.t.} \quad 3x_1 + 5x_2 &\le 6 \\ 5x_1 + 6x_2 &\le 10 \\ 2x_1 - 3x_2 &\le 4 \\ x_1 &\le 2 \\ x_2 &\le 1 \\ x_1, x_2 &\ge 0 \end{aligned}$$

3

$$\begin{aligned} \max z = 4x_1 + 3x_2 & \\ \text{s.t.} \quad 2x_1 - x_2 &\le 1 \\ x_1 + 6x_2 &\le 6 \\ x_2 &\le 5 \\ x_1, x_2 &\ge 0 \end{aligned}$$

4 Suppose an LP contained lower-bound constraints of the following form: $x_j \ge L_j$. Suggest an algorithm that could be used to solve such a problem efficiently.

10.6 Karmarkar's Method for Solving LPs

As discussed in Section 4.13, Karmarkar's method for solving LPs is a polynomial time algorithm. This is in contrast to the simplex algorithm, an exponential time algorithm. Unlike the ellipsoid method (another polynomial time algorithm), Karmarkar's method appears to solve many LPs faster than does the simplex algorithm. In this section, we give a description of the basic concepts underlying Karmarkar's method. Note that several versions of Karmarkar's method are computationally more efficient than the version we de-

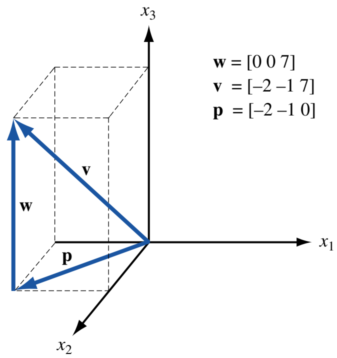


FIGURE 5
Projection of $[-2 \quad -1 \quad 7]$ onto $x_3 = 0$

scribe; our goal is simply to introduce the reader to the exciting ideas used in Karmarkar's method. For a more detailed description of Karmarkar's method, see Hooker (1986), Parker and Rardin (1988), and Murty (1989).

Karmarkar's method is applied to an LP in the following form:

$$\begin{aligned} \min z &= \mathbf{cx} \\ \text{s.t.} \quad A\mathbf{x} &= \mathbf{0} \\ x_1 + x_2 + \cdots + x_n &= 1 \\ \mathbf{x} &\geq 0 \end{aligned} \tag{31}$$

In (31), $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T$, A is an $m \times n$ matrix, $\mathbf{c} = [c_1 \quad c_2 \quad \cdots \quad c_n]$ and $\mathbf{0}$ is an n-dimensional column vector of zeros. The LP must also satisfy

$$[\tfrac{1}{n} \quad \tfrac{1}{n} \quad \cdots \quad \tfrac{1}{n}]^T \qquad \text{is feasible} \tag{32}$$

$$\text{Optimal } z\text{-value} = 0 \tag{33}$$

Although it may seem unlikely that an LP would have the form (31) and satisfy (32)–(33),

LEMMA 1

it is easy to show that any LP may be put in a form such that (31)–(33) are satisfied. We will demonstrate this at the end of this section.

The following three concepts play a key role in Karmarkar's method:

1 Projection of a vector onto the set of $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{0}$

2 Karmarkar's centering transformation

3 Karmarkar's potential function

We now discuss the first two concepts, leaving a discussion of Karmarkar's potential function to the end of the section. Before discussing the ideas just listed, we need a definition.

DEFINITION ■ The ***n*-dimensional unit simplex** S is the set of points $[x_1 \quad x_2 \quad \cdots \quad x_n]^T$ satisfying $x_1 + x_2 + \cdots + x_n = 1$ and $x_j \geq 0$, $j = 1, 2, \ldots, n$. ■

Projection

Suppose we are given a point $\mathbf{x}^0$ that is feasible for (31), and we want to move from $\mathbf{x}^0$ to another feasible point (call it $\mathbf{x}^1$) that, for some fixed vector $\mathbf{v}$, will have a larger value of $\mathbf{vx}$. Suppose that we find $\mathbf{x}^1$ by moving away from $\mathbf{x}^0$ in a direction $\mathbf{d} = [d_1 \quad d_2 \quad \cdots \quad d_n]$. For $\mathbf{x}^1$ to be feasible, $\mathbf{d}$ must satisfy $A\mathbf{d} = \mathbf{0}$ and $d_1 + d_2 + \cdots + d_n = 0$. If we choose the direction $\mathbf{d}$ that solves the optimization problem

$$\begin{aligned} &\max\ \mathbf{vd} \\ &\text{s.t.} \quad A\mathbf{d} = \mathbf{0} \\ &d_1 + d_2 + \cdots + d_n = 0 \\ &\|\mathbf{d}\| = 1 \end{aligned}$$

then we will be moving in the "feasible" direction that maximizes the increase in $\mathbf{vx}$ per unit of length moved. The direction $\mathbf{d}$ that solves this optimization problem is given by the **projection** of $\mathbf{v}$ onto the set of $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T$ satisfying $A\mathbf{x} = \mathbf{0}$ and $x_1 + x_2 + \cdots + x_n = 0$. The projection of $\mathbf{v}$ onto the set of $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{0}$ and $x_1 + x_2 + \cdots + x_n = 0$ is given by $[I - B^T(BB^T)^{-1}B]\mathbf{v}$, where B is the $(m + 1) \times n$ matrix whose first m rows are A and whose last row is a vector of 1's.

Geometrically, what does it mean to project a vector $\mathbf{v}$ onto the set of $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{0}$? It can be shown that any vector $\mathbf{v}$ may be written (uniquely) in the form $\mathbf{v} = \mathbf{p} + \mathbf{w}$, where $\mathbf{p}$ satisfies $A\mathbf{p} = \mathbf{0}$ and $\mathbf{w}$ is perpendicular to all vectors $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{0}$. Then $\mathbf{p}$ is the projection of $\mathbf{v}$ onto the set of $\mathbf{x}$ satisfying $A\mathbf{x} = 0$. An example of this idea is given in Figure 5, where $\mathbf{v} = [-2 \quad -1 \quad 7]$ is projected onto the set of three-dimensional vectors satisfying $x_3 = 0$ (the x_1–x_2-plane). In this case, we decompose $\mathbf{v}$ as $\mathbf{v} = [-2 \quad -1 \quad 0] + [0 \quad 0 \quad 7]$. Thus, $\mathbf{p} = [-2 \quad -1 \quad 0]$. It is easy to show that $\mathbf{p}$ is the vector in the set of $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{0}$ that is "closest" to $\mathbf{v}$. This is apparent from Figure 5.

Karmarkar's Centering Transformation

Given a feasible point (in (31)) $\mathbf{x}^k = [x_1^k \quad x_2^k \quad \cdots \quad x_n^k]$ in S having $x_j^k > 0, j = 1, 2, \ldots, n$, we write the **centering transformation** associated with the point $\mathbf{x}^k$ as $f([x_1 \quad x_2 \quad \cdots \quad x_n] | \mathbf{x}^k)$. If $\mathbf{x}^k$ is a point in S, then $f([x_1 \quad x_2 \quad \cdots \quad x_n] | \mathbf{x}^k)$ transforms a point $[x_1 \quad x_2 \quad \cdots \quad x_n]^T$ in S into a point $[y_1 \quad y_2 \quad \cdots \quad y_n]^T$ in S, where

$$y_j = \frac{\dfrac{x_j}{x_j^k}}{\displaystyle\sum_{r=1}^{r=n} \frac{x_r}{x_r^k}} \tag{34}$$

Let $\text{Diag}(\mathbf{x}^k)$ be the $n \times n$ matrix with all off-diagonal entries equal to 0 and $\text{Diag}(\mathbf{x}^k)_{ii} = x_i^k$. The centering transformation specified by (34) can be shown to have the properties listed in Lemma 1.

Karmarkar's centering transformation has the following properties:

$$f(\mathbf{x}^k \mid \mathbf{x}^k) = [\tfrac{1}{n} \quad \tfrac{1}{n} \quad \cdots \quad \tfrac{1}{n}]^T \tag{35}$$

$$\text{For } \mathbf{x} \neq \mathbf{x}', \qquad f(\mathbf{x} \mid \mathbf{x}^k) \neq f(\mathbf{x}' \mid \mathbf{x}^k) \tag{36}$$

$$f(\mathbf{x} \mid \mathbf{x}^k) \in S \tag{37}$$

For any point $[y_1 \quad y_2 \quad \cdots \quad y_n]^T$ in S, there is a unique point **(38)**

$[x_1 \quad x_2 \quad \cdots \quad x_n]^T$ in S satisfying

$$f([x_1 \quad x_2 \quad \cdots \quad x_n]^T \mid \mathbf{x}^k) = [y_1 \quad y_2 \quad \cdots \quad y_n]^T \tag{38$'$}$$

The point $[x_1 \quad x_2 \quad \cdots \quad x_n]^T$ is given by

$$x_j = \frac{x_j^k y_j}{\sum_{r=1}^{r=n} x_r^k y_r}$$

If $[x_1 \quad x_2 \quad \cdots \quad x_n]^T$ and $[y_1 \quad y_2 \quad \cdots \quad y_n]^T$ satisfy (38′), we write $f^{-1}([y_1 \quad y_2 \quad \cdots \quad y_n]^T \mid \mathbf{x}^k) = [x_1 \quad x_2 \quad \cdots \quad x_n]^T$.

$$\text{A point } \mathbf{x} \text{ in } S \text{ will satisfy } A\mathbf{x} = \mathbf{0} \qquad \text{if} \qquad A[\text{Diag}(\mathbf{x}^k)]f(\mathbf{x} \mid \mathbf{x}^k) = 0 \tag{39}$$

(See Problem 5 for a proof of Lemma 1.)

To illustrate the centering transformation, consider the following LP:

$$\begin{aligned} \min z &= x_1 + 3x_2 - 3x_3 \\ \text{s.t.} \quad & x_2 - x_3 = 0 \\ & x_1 + x_2 + x_3 = 1 \\ & x_i \geq 0 \end{aligned} \tag{40}$$

This LP is of the form (31); the point $[\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3}]^T$ is feasible, and the LP's optimal z-value is 0. The feasible point $[\frac{1}{4} \quad \frac{3}{8} \quad \frac{3}{8}]$ yields the following transformation:

$$f([x_1 \quad x_2 \quad x_3] \mid [\tfrac{1}{4} \quad \tfrac{3}{8} \quad \tfrac{3}{8}]) = \left[\frac{4x_1}{4x_1 + \frac{8x_2}{3} + \frac{8x_3}{3}} \quad \frac{\frac{8x_2}{3}}{4x_1 + \frac{8x_2}{3} + \frac{8x_3}{3}} \quad \frac{\frac{8x_3}{3}}{4x_1 + \frac{8x_2}{3} + \frac{8x_3}{3}}\right]$$

For example,

$$f([\tfrac{1}{3} \quad \tfrac{1}{3} \quad \tfrac{1}{3}] \mid [\tfrac{1}{4} \quad \tfrac{3}{8} \quad \tfrac{3}{8}]) = [\tfrac{12}{28} \quad \tfrac{8}{28} \quad \tfrac{8}{28}]$$

We now refer to the variables $x_1, x_2, \ldots, x_n$ as being the *original* space and the variables $y_1, y_2, \ldots, y_n$ as being the *transformed* space. The unit simplex involving variables $y_1, y_2, \ldots, y_n$ will be called the transformed unit simplex. We now discuss the intuitive meaning of (35)–(39). Equation (35) implies that $f(\cdot \mid \mathbf{x}^k)$ maps $\mathbf{x}^k$ into the "center" of the transformed unit simplex. Equations (36)–(37) imply that any point in S is transformed into a point in the transformed unit simplex, and no two points in S can yield the same point in the transformed unit simplex (that is, f is a one-to-one mapping). Equation (38) implies that for any point $\mathbf{y}$ in the transformed unit simplex, there is a point $\mathbf{x}$ in S that is transformed into $\mathbf{y}$. The formula for the $\mathbf{x}$ that is transformed into $\mathbf{y}$ is also given. Thus, (36)–(38) imply that f is a one-to-one and an onto mapping from S to S. Finally, (39) states that feasible points in the original problem correspond to points $\mathbf{y}$ in the transformed unit simplex that satisfy $A[\text{Diag}(\mathbf{x}^k)]\mathbf{y} = 0$.

Description and Example of Karmarkar's Method

We assume that we will be satisfied with a feasible point having an optimal z-value $< \epsilon$ (for some small ϵ). Karmarkar's method proceeds as follows:

Step 1 Begin at the feasible point $\mathbf{x}^0 = [\frac{1}{n} \quad \frac{1}{n} \quad \cdots \quad \frac{1}{n}]^T$ and set $k = 0$.

Step 2 Stop if $\mathbf{cx}^k < \epsilon$. If not, go to step 3.

Step 3 Find the new point $\mathbf{y}^{k+1} = [y_1^{k+1} \quad y_2^{k+1} \quad \cdots \quad y_n^{k+1}]^T$ in the transformed unit simplex given by

$$\mathbf{y}^{k+1} = [\tfrac{1}{n} \quad \tfrac{1}{n} \quad \cdots \quad \tfrac{1}{n}]^T - \frac{\theta(I - P^T(PP^T)^{-1}P)[\text{Diag}(\mathbf{x}^k)]\mathbf{c}^T}{\|\mathbf{c}_p\|\sqrt{n(n-1)}}$$

Here, $\|\mathbf{c}_p\|$ = the length of $(I - P^T(PP^T)^{-1}P)[\text{Diag}(\mathbf{x}^k)]\mathbf{c}^T$, P is the $(m + 1) \times n$ matrix whose first m rows are $A[\text{Diag}(\mathbf{x}^k)]$ and whose last row is a vector of 1's, and $0 < \theta < 1$ is chosen to ensure convergence of the algorithm. $\theta = \frac{1}{4}$ is known to ensure convergence.

Now obtain a new point $\mathbf{x}^{k+1}$ in the original space by using the centering transformation to determine the point corresponding to $\mathbf{y}^{k+1}$. That is, $\mathbf{x}^{k+1} = f^{-1}(\mathbf{y}^{k+1} \mid \mathbf{x}^k)$. Increase k by 1 and return to step 2.

REMARKS

1 In step 3, we move from the "center" of the transformed unit simplex in a direction opposite to the projection of $\text{Diag}(\mathbf{x}^k)\mathbf{c}^T$ onto the transformation of the feasible region (the set of $\mathbf{y}$ satisfying $A[\text{Diag}(\mathbf{x}^k)]\mathbf{y} = \mathbf{0}$). From our discussion of the projection, this ensures that we maintain feasibility (in the transformed space) and move in a direction that maximizes the rate of decrease of $[\text{Diag}(\mathbf{x}^k)]\mathbf{c}^T$.
2 By moving a distance

$$\frac{\theta}{\qquad}$$

from the center of the transformed unit simplex, we ensure that $\mathbf{y}^{k+1}$ will remain in the interior of the transformed unit simplex.
3 When we use the inverse of Karmarkar's centering transformation to transform $\mathbf{y}^{k+1}$ back into $\mathbf{x}^{k+1}$, the definition of projection and (39) imply that $\mathbf{x}^{k+1}$ will be feasible for the original LP (see Problem 6).
4 Why do we project $[\text{Diag}(\mathbf{x}^k)]\mathbf{c}^T$ rather than $\mathbf{c}^T$ onto the transformed feasible region? The answer to this question must await our discussion of Karmarkar's potential function. Problem 7 provides another explanation of why we project $[\text{Diag}(\mathbf{x}^k)]\mathbf{c}^T$ rather than $\mathbf{c}^T$.

We now work out the first iteration of Karmarkar's method when applied to (40), choosing $\epsilon = 0.10$.

First Iteration of Karmarkar's Method

Step 1 $\mathbf{x}^0 = [\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3}]^T$ and $k = 0$.

Step 2 $\mathbf{x}^0$ yields $z = \frac{1}{3} > 0.10$, so we must proceed to step 3.

Step 3

$$A = [0 \quad 1 \quad -1], \qquad \text{Diag}(\mathbf{x}^k) = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$$

$$A[\text{Diag}(\mathbf{x}^k)] = [0 \quad \tfrac{1}{3} \quad -\tfrac{1}{3}], \qquad P = \begin{bmatrix} 0 & \frac{1}{3} & -\frac{1}{3} \\ 1 & 1 & 1 \end{bmatrix}$$

$$PP^T = \begin{bmatrix} \frac{2}{9} & 0 \\ 0 & 3 \end{bmatrix}, \qquad (PP^T)^{-1} = \begin{bmatrix} \frac{9}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$$

$$(I - P^T(PP^T)^{-1}P) = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{1}{6} & \frac{1}{6} \\ -\frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{bmatrix}, \qquad \mathbf{c} = [1 \quad 3 \quad -3]$$

$$[\text{Diag } \mathbf{x}^k]\mathbf{c}^T = \begin{bmatrix} \frac{1}{3} \\ 1 \\ -1 \end{bmatrix}$$

$$(I - P^T(PP^T)^{-1}P)[\text{Diag } \mathbf{x}^k]\mathbf{c}^T = [\tfrac{2}{9} \quad -\tfrac{1}{9} \quad -\tfrac{1}{9}]$$

Now, (using $\theta = 0.25$), we obtain

$$\mathbf{y}^1 = [\tfrac{1}{3} \quad \tfrac{1}{3} \quad \tfrac{1}{3}]^T - \frac{0.25[\frac{2}{9} \quad -\frac{1}{9} \quad -\frac{1}{9}]^T}{\sqrt{3(2)}\|[\frac{2}{9} \quad -\frac{1}{9} \quad -\frac{1}{9}]\|}$$

Because

$$\|[\tfrac{2}{9} \quad -\tfrac{1}{9} \quad -\tfrac{1}{9}]\|^T = \sqrt{(\tfrac{2}{9})^2 + (-\tfrac{1}{9})^2 + (-\tfrac{1}{9})^2} = \frac{\sqrt{6}}{9}$$

we obtain

$$\mathbf{y}^1 = [\tfrac{1}{3} \quad \tfrac{1}{3} \quad \tfrac{1}{3}]^T - [\tfrac{6}{72} \quad -\tfrac{3}{72} \quad -\tfrac{3}{72}]^T = [\tfrac{1}{4} \quad \tfrac{3}{8} \quad \tfrac{3}{8}]^T$$

Using (38′), we now obtain $\mathbf{x}^1 = [x_1^1 \quad x_2^1 \quad x_3^1]^T$ from

$$x_1^1 = \frac{\frac{1}{3}(\frac{1}{4})}{} = \frac{1}{4}$$

$$x_2^1 = \frac{\frac{1}{3}(\frac{3}{8})}{} = \frac{3}{8}$$

$$x_3^1 = \frac{\frac{1}{3}(\frac{3}{8})}{} = \frac{3}{8}$$

Thus, $\mathbf{x}^1 = [\frac{1}{4} \quad \frac{3}{8} \quad \frac{3}{8}]^T$. It will always be the case (see Problem 3) that $\mathbf{x}^1 = \mathbf{y}^1$, but for $k > 1$, $\mathbf{x}^k$ need not equal $\mathbf{y}^k$. Note that for $\mathbf{x}^1$, we have $z = \frac{1}{4} + 3(\frac{3}{8}) - 3(\frac{3}{8}) = \frac{1}{4} < \frac{1}{3}$ (the z-value for $\mathbf{x}^0$).

Potential Function

Because we are projecting $[\text{Diag}(\mathbf{x}^k)]\mathbf{c}^T$ rather than $\mathbf{c}^T$, we cannot be sure that each iteration of Karmarkar's method will decrease z. In fact, it is possible for $\mathbf{cx}^{k+1} > \mathbf{cx}^k$ to occur. To explain why Karmarkar projects $[\text{Diag}(\mathbf{x}^k)]\mathbf{c}^T$, we need to discuss Karmarkar's potential function. For $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T$, we define the potential function $f(\mathbf{x})$ by

$$f(\mathbf{x}) = \sum_{j=1}^{j=n} \ln\left(\frac{\mathbf{cx}^T}{x_j}\right)$$

Karmarkar showed that if we project $[\text{Diag}(\mathbf{x}^k)]\mathbf{c}^T$ (not $\mathbf{c}^T$) onto the feasible region in the transformed space, then for some $\delta > 0$, it will be true that for $k = 0, 1, 2, \ldots,$

$$f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) \geq \delta \tag{41}$$

Inequality (41) states that each iteration of Karmarkar's method decreases the potential function by an amount bounded away from 0. Karmarkar shows that if the potential function evaluated at $\mathbf{x}^k$ is small enough, then $\mathbf{z} = \mathbf{cx}^k$ will be near 0. Because $f(\mathbf{x}^k)$ is decreased by at least δ per iteration, it follows that by choosing k sufficiently large, we can ensure that the z-value for $\mathbf{x}^k$ is less than ϵ.

Putting an LP in Standard Form for Karmarkar's Method

We now show how to convert any LP to the form defined by (31)–(33). To illustrate, we show how to transform the following LP

$$\begin{aligned} \max z &= 3x_1 + x_2 \\ \text{s.t.} \quad 2x_1 - x_2 &\leq 2 \\ x_1 + 2x_2 &\leq 5 \\ x_1, x_2 &\geq 0 \end{aligned} \tag{42}$$

into the form defined by (31)–(33).

We begin by finding the dual of (42).

$$\begin{aligned} \min w &= 2y_1 + 5y_2 \\ \text{s.t.} \quad 2y_1 + y_2 &\geq 3 \\ -y_1 + 2y_2 &\geq 1 \\ y_1, y_2 &\geq 0 \end{aligned} \tag{42$'$}$$

From the Dual Theorem (Theorem 1 of Chapter 6), we know that if (x_1, x_2) is feasible in (42), (y_1, y_2) is feasible in (42$'$), and the z-value for (x_1, x_2) in (42) equals the w-value for (y_1, y_2) in (42$'$), then (x_1, x_2) is optimal for (42). This means that any feasible solution to the following set of constraints will yield the optimal solution to (42):

$$\begin{aligned} 3x_1 + x_2 - 2y_1 - 5y_2 &= 0 \\ 2x_1 - x_2 &\leq 2 \\ x_1 + 2x_2 &\leq 5 \\ 2y_1 + y_2 &\geq 3 \\ -y_1 + 2y_2 &\geq 1 \\ \text{All variables} &\geq 0 \end{aligned} \tag{43}$$

Inserting slack and excess variables into (43) yields

$$\begin{aligned} 3x_1 + x_2 - 2y_1 - 5y_2 &= 0 \\ 2x_1 - x_2 + s_1 &= 2 \\ x_1 + 2x_2 + s_2 &= 5 \\ 2y_1 + y_2 - e_1 &= 3 \\ -y_1 + 2y_2 - e_2 &= 1 \\ \text{All variables} &\geq 0 \end{aligned} \tag{44}$$

We now find a number M such that any feasible solution to (44) will satisfy

$$\text{sum of all variables in (44)} \leq M \tag{45}$$

and add constraint (45) to (44). Being conservative, we can see that any values of the variables that yield an optimal primal solution to (42) and an optimal dual solution to (42′) will have no variable exceeding 10. This would yield $M = 10(8) = 80$. We then add a slack variable (dummy variable d_1) to (45). Our new goal is then to find a feasible solution to

$$
\begin{aligned}
3x_1 + x_2 - 2y_1 - 5y_2 &= 0\\
2x_1 - x_2 + s_1 &= 2\\
x_1 + 2x_2 + s_2 &= 5\\
2y_1 + y_2 - e_1 &= 3 \qquad \textbf{(46)}\\
-y_1 + 2y_2 - e_2 &= 1\\
x_1 + x_2 + y_1 + y_2 + s_1 + s_2 + e_1 + e_2 + d_1 &= 80
\end{aligned}
$$

All variables ≥ 0

We now define a new dummy variable d_2; $d_2 = 1$. We can use this new variable to "homogenize" the constraints in (46), which have nonzero right-hand sides. To do this, we add the appropriate multiple of the constraint $d_2 = 1$ to each constraint in (46) (except the last constraint) having a nonzero right-hand side. For example we add $-2(d_2 = 1)$ to the constraint $2x_1 - x_2 + s_1 = 2$. We also replace the last constraint in (46) by the following two constraints:

(a) Add $d_2 = 1$ to the last constraint

SUMMARY The Revised Simplex Method and the Product Form of the Inverse

Step 0 Note the columns from which the current B^{-1} will be read. Initially $B^{-1} = I$.

Step 1 For the current tableau, compute $\mathbf{c}_{\text{BV}}B^{-1}$.

Step 2 Price out all nonbasic variables in the current tableau. If (for a max problem) each nonbasic variable prices out nonnegative, the current basis is optimal. If the current basis is not optimal, enter into the basis the nonbasic variable with the most negative coefficient in row 0. Call this variable x_k.

Step 3 To determine the row in which x_k enters the basis, compute x_k's column in the current tableau ($B^{-1}\mathbf{a}_k$) and compute the right-hand side of the current tableau ($B^{-1}\mathbf{b}$). Then use the ratio test to determine the row in which x_k should enter the basis. We now know the set of basic variables (BV) for the new tableau.

Step 4 Use the column for x_k in the current tableau to determine the EROs needed to enter x_k into the basis. Perform these EROs on the current B^{-1} to yield the new B^{-1}. Return to step 1.

Alternatively, we may use the product form of the inverse to update B^{-1}. Suppose we have found that x_k should enter the basis in row r. Let the column for x_k in the current tableau be

$$
\begin{bmatrix} \bar{a}_{1k} \\ \bar{a}_{2k} \\ \vdots \\ \bar{a}_{mk} \end{bmatrix}
$$

Define the $m \times m$ matrix E by

$$E = \begin{array}{cc} & \text{(column } r\text{)} \\ & \left[\begin{array}{cccccc} 1 & 0 & \cdots & -\dfrac{\bar{a}_{1k}}{\bar{a}_{rk}} & \cdots & 0 \quad 0 \\ 0 & 1 & \cdots & -\dfrac{\bar{a}_{2k}}{\bar{a}_{rk}} & \cdots & 0 \quad 0 \\ \vdots & \vdots & & \vdots & & \vdots \quad \vdots \\ 0 & 0 & \cdots & \dfrac{1}{\bar{a}_{rk}} & \cdots & 0 \quad 0 \\ \vdots & \vdots & & \vdots & & \vdots \quad \vdots \\ 0 & 0 & \cdots & -\dfrac{\bar{a}_{m-1,k}}{\bar{a}_{rk}} & \cdots & 1 \quad 0 \\ 0 & 0 & \cdots & -\dfrac{\bar{a}_{mk}}{\bar{a}_{rk}} & \cdots & 0 \quad 1 \end{array}\right] \end{array} \begin{array}{c} \\ \\ \\ \\ \text{(row } r\text{)} \\ \\ \\ \\ \end{array}$$

Then

$$B^{-1} \text{ for new tableau} = E(B^{-1} \text{ for current tableau})$$

Return to step 1.

Column Generation

When an LP has many variables, it is very time-consuming to price out each nonbasic variable individually. The column generation approach lets us determine the nonbasic variable that prices out most favorably by solving a subproblem (such as the branch-and-bound problems in the cutting stock problem).

Dantzig–Wolfe Decomposition Method

In many LPs, the constraints and variables may be decomposed in the following manner:

Constraints in set 1 only involve variables in Variable set 1.

Constraints in set 2 only involve variables in Variable set 2.

$\vdots$

Constraints in set k only involve variables in Variable set k.

Constraints in set $k + 1$ may involve any variable. The constraints in set $k + 1$ are referred to as the **central constraints.**

LPs that can be decomposed in this fashion can often be efficiently solved by the Dantzig–Wolfe decomposition algorithm. The following explanation assumes that $k = 2$.

Step 1 Let the variables in Variable set 1 be $x_1, x_2, \ldots, x_{n_1}$. Express the variables in Variable set 1 as a convex combination of the extreme points of the feasible region for Constraint set 1 (the constraints that involve only the variables in Variable set 1). If we let $\mathbf{P}_1$, $\mathbf{P}_2, \ldots, \mathbf{P}_k$ be the extreme points of this feasible region, then any point

Group B

5 Prove Lemma 1.

6 Show that the point $\mathbf{x}^k$ in Karmarkar's method is feasible for the original LP.

7 Given a point $\mathbf{y}^k$ in Karmarkar's method, express the LP's original objective function as a function of $\mathbf{y}^k$. Use the answer to this question to give a reason why $[\text{Diag}(\mathbf{x}^k)]\mathbf{c}^T$ is projected, rather than $\mathbf{c}^T$.

SUMMARY

The Revised Simplex Method and the Product Form of the Inverse

Step 0 Note the columns from which the current B^{-1} will be read. Initially $B^{-1} = I$.

Step 1 For the current tableau, compute $\mathbf{c}_{\mathrm{BV}}B^{-1}$.

Step 2 Price out all nonbasic variables in the current tableau. If (for a max problem) each nonbasic variable prices out nonnegative, the current basis is optimal. If the current basis is not optimal, enter into the basis the nonbasic variable with the most negative coefficient in row 0. Call this variable x_k.

Step 3 To determine the row in which x_k enters the basis, compute x_k's column in the current tableau ($B^{-1}\mathbf{a}_k$) and compute the right-hand side of the current tableau ($B^{-1}\mathbf{b}$). Then use the ratio test to determine the row in which x_k should enter the basis. We now know the set of basic variables (BV) for the new tableau.

Step 4 Use the column for x_k in the current tableau to determine the EROs needed to enter x_k into the basis. Perform these EROs on the current B^{-1} to yield the new B^{-1}. Return to step 1.

Alternatively, we may use the product form of the inverse to update B^{-1}. Suppose we have found that x_k should enter the basis in row r. Let the column for x_k in the current tableau be

$$\begin{bmatrix} \bar{a}_{1k} \\ \bar{a}_{2k} \\ \vdots \\ \bar{a}_{mk} \end{bmatrix}$$

Define the $m \times m$ matrix E by

$$E = \begin{array}{cccccccl} & & & \text{(column } r) & & & & \\ 1 & 0 & \cdots & -\dfrac{\bar{a}_{1k}}{\bar{a}_{rk}} & \cdots & 0 & 0 & \\ 0 & 1 & \cdots & -\dfrac{\bar{a}_{2k}}{\bar{a}_{rk}} & \cdots & 0 & 0 & \\ \vdots & \vdots & & \vdots & & \vdots & \vdots & \\ 0 & 0 & \cdots & \dfrac{1}{\bar{a}_{rk}} & \cdots & 0 & 0 & \text{(row } r) \\ \vdots & \vdots & & \vdots & & \vdots & \vdots & \\ 0 & 0 & \cdots & -\dfrac{\bar{a}_{m-1,k}}{\bar{a}_{rk}} & \cdots & 1 & 0 & \\ 0 & 0 & \cdots & -\dfrac{\bar{a}_{mk}}{\bar{a}_{rk}} & \cdots & 0 & 1 & \end{array}$$

Then

$$B^{-1} \text{ for new tableau} = E(B^{-1} \text{ for current tableau})$$

Return to step 1.

Column Generation

When an LP has many variables, it is very time-consuming to price out each nonbasic variable individually. The column generation approach lets us determine the nonbasic variable that prices out most favorably by solving a subproblem (such as the branch-and-bound problems in the cutting stock problem).

Dantzig–Wolfe Decomposition Method

In many LPs, the constraints and variables may be decomposed in the following manner:

Constraints in set 1 only involve variables in Variable set 1.

Constraints in set 2 only involve variables in Variable set 2.

$$\vdots$$

Constraints in set k only involve variables in Variable set k.

Constraints in set $k + 1$ may involve any variable. The constraints in set $k + 1$ are referred to as the **central constraints.**

LPs that can be decomposed in this fashion can often be efficiently solved by the Dantzig–Wolfe decomposition algorithm. The following explanation assumes that $k = 2$.

Step 1 Let the variables in Variable set 1 be $x_1, x_2, \ldots, x_{n_1}$. Express the variables in Variable set 1 as a convex combination of the extreme points of the feasible region for Constraint set 1 (the constraints that involve only the variables in Variable set 1). If we let $\mathbf{P}_1$, $\mathbf{P}_2, \ldots, \mathbf{P}_k$ be the extreme points of this feasible region, then any point

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_1} \end{bmatrix}$$

in the feasible region for Constraint set 1 may be written in the form

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_1} \end{bmatrix} = \mu_1\mathbf{P}_1 + \mu_2\mathbf{P}_2 + \cdots + \mu_k\mathbf{P}_k \tag{29}$$

where $\mu_1 + \mu_2 + \cdots + \mu_k = 1$ and $\mu_i \geq 0$ $(i = 1, 2, \ldots, k)$.

Step 2 Express the variables in Variable set 2, $x_{n_1+1}, x_{n_1+2}, \ldots, x_n$, as a convex combination of the extreme points of Constraint set 2's feasible region. If we let the extreme points of the feasible region be $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_m$, then any point in Constraint set 2's feasible region may be written as

$$\begin{bmatrix} x_{n_1+1} \\ x_{n_1+2} \\ \vdots \\ x_n \end{bmatrix} = \lambda_1 \mathbf{Q}_1 + \lambda_2 \mathbf{Q}_2 + \cdots + \lambda_m \mathbf{Q}_m \quad (30)$$

where $\lambda_i \geq 0$ ($i = 1, 2, \ldots, m$) and $\lambda_1 + \lambda_2 + \cdots + \lambda_m = 1$.

Step 3 Using (29) and (30), express the LP's objective function and centralized constraints in terms of the μ_i's and the λ_i's. After adding the constraints (called convexity constraints), $\mu_1 + \mu_2 + \cdots + \mu_k = 1$ and $\lambda_1 + \lambda_2 + \cdots + \lambda_m = 1$ and the sign restrictions $\mu_i \geq 0$ ($i = 1, 2, \ldots, k$) and $\lambda_i \geq 0$ ($i = 1, 2, \ldots, m$), we obtain the following LP, which is referred to as the **restricted master:**

TABLE 10

	Hours	
Plant	Product 1	Product 2
1	2	3
2	3	4

TABLE 11

	Profit per Product ($)	
Plant	Product 1	Product 2
1	8	6
2	10	8

max (or min) [objective function in terms of μ_i's and λ_i's]

s.t. [central constraints in terms of μ_i's and λ_i's]

$\mu_1 + \mu_2 + \cdots + \mu_k = 1$ (Convexity constraints)

$\lambda_1 + \lambda_2 + \cdots + \lambda_m = 1$

$\mu_i \geq 0$ ($i = 1, 2, \ldots, k$) (Sign restrictions)

$\lambda_i \geq 0$ ($i = 1, 2, \ldots, m$)

Step 4 Assume that a basic feasible solution for the restricted master is readily available. Then use the column generation method of Section 10.3 to determine whether there is any μ_i or λ_i that can improve the z-value for the restricted master. If so, use the revised simplex method to enter that variable into the basis. Otherwise, the current tableau is optimal for the restricted master. If the current tableau is not optimal, continue with column generation until an optimal solution is found.

Step 5 Substitute the optimal values of the μ_i's and λ_i's