2.13. To prove the resolution theorem by the induction method, we let p be the number of positive components of $\mathbf{x} \in P$. When $p = 0$, $\mathbf{x} = \mathbf{0}$ is obviously an extreme point of P. Assume that the theorem holds for $p = 0, 1, \ldots, k$, and $\mathbf{x}$ has $k + 1$ positive components.

If $\mathbf{x}$ is an extreme point, then there is nothing to prove. If $\mathbf{x}$ is not an extreme point, we let $\mathbf{x}^T = (\overline{\mathbf{x}}^T \mid \mathbf{0})$, where $\overline{\mathbf{x}}^T = (x_1, \ldots, x_{k+1}) > \mathbf{0}$ and $\mathbf{A} = [\overline{\mathbf{A}} \mid \mathbf{N}]$. Then Theorem 2.1 shows that the columns of $\overline{\mathbf{A}}$ are linearly dependent, in other words, there exists a vector $\overline{\mathbf{w}} \in R^{k+1}$, $\overline{\mathbf{w}} \neq \mathbf{0}$ such that $\overline{\mathbf{A}}\overline{\mathbf{w}} = \mathbf{0}$. We define $\mathbf{w}^T = (\overline{\mathbf{w}}^T, \mathbf{0}) \in R^n$, then $\mathbf{w} \neq \mathbf{0}$ and $\mathbf{A}\mathbf{w} = \overline{\mathbf{A}}\overline{\mathbf{w}} = \mathbf{0}$. There are three possibilities: $\mathbf{w} \geq \mathbf{0}$, $\mathbf{w} < \mathbf{0}$, and $\mathbf{w}$ has both positive and negative components.

For $\mathbf{w} \geq \mathbf{0}$, consider $\mathbf{x}(\theta) = \mathbf{x} + \theta\mathbf{w}$ and pick θ^* to be the largest negative value of θ such that $\mathbf{x}^* = \mathbf{x}(\theta^*)$ has at least one more zero component than $\mathbf{x}$. Then follow the induction hypothesis to show the theorem holds. Similarly, show that in the remaining two cases, the theorem still holds.

2.14. For a linear programming problem with a nonempty feasible domain $P = \{\mathbf{x} \in R^n \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, prove that every extreme point of P is a vertex of P and the converse statement is also true.

3

The Revised Simplex Method

In Chapter 2 we have seen that if the optimal solution set of a linear programming problem is nonempty, then it contains at least one extreme point of the polyhedral set of the feasible domain. Thus an intuitive way to solve a linear programming problem is to traverse from one extreme point to a neighboring extreme point in a systematic fashion until we reach an optimal one. This is the basic idea of the simplex method and its variants. However, in doing so, as in any other iterative scheme, we have to resolve three important issues: (1) How do we start with an extreme point? (2) How do we move from one extreme point to a better neighboring extreme point in an "efficient" way? (3) When do we stop the process? This chapter addresses these issues for the simplex method with an emphasis on the so-called revised simplex method, which provides a computationally efficient implementation for linear programming.

3.1 ELEMENTS OF AN ITERATIVE SCHEME

The philosophy of solving an optimization problem via an iterative scheme is to start with a "rough" solution and successively improve the current solution until a desired goal is met. The simplex method, ellipsoid method, Karmarkar's projective scaling method, and the affine scaling method to be studied are all in this category. Basically, an iterative scheme consists of three major steps:

Step 1: Start from somewhere.
Step 2: Check if the goal is met.
Step 3: Move to a place closer to the goal.

The first step is to find a valid and yet convenient starting point. The choice of a starting point may affect the overall efficiency of an iterative scheme. It varies widely from one method to another. If a method is very sensitive to its starting point, it is certainly worth spending additional computational effort and time in finding a good starting point. Otherwise, we should spend minimum effort on it. Sometimes mathematical transformations are employed to transform a given problem into an equivalent form for a quick admissible starting point. Once the transformed problem is solved, its solution could then be used to obtain a solution to the original problem. In general, finding a starting point is not an easy task; it may take as much as half of the total computational effort. We shall study different starting mechanisms in later sections and chapters.

The second step of an iterative scheme is to check if we have reached our goal or not. For an optimization problem, this means testing for optimality of a solution. This test has to be carried out at each iteration for the current solution in hand. When the result turns out to be positive, the iterative scheme is terminated. Otherwise, we go to the third step for further improvement. The testing process usually requires a *stopping rule*, or *stopping criterion* for an iterative scheme. Once again, a computationally simple stopping rule is preferred for an efficient iterative scheme, since it is performed at each iteration.

If the stopping rule is met, we have achieved our goal. Otherwise, we proceed to make further improvement in getting closer to our goal. This is usually done by moving from a current solution to a better one. To do so we need two elements: (1) a *good direction* of movement, and (2) an appropriate *step length* along the good direction. A good direction should point to a better result, and the step length describes how far we should proceed along the direction. Needless to say, the efficiency of an iterative method depends strongly on the mechanism of finding a good direction and appropriate step-length. In general, the synthesis of the direction of movement and the associated step length calculation constitute the bulk of computation for an iterative scheme. Therefore special attention should be paid to this aspect to achieve speed and efficiency in practical implementations.

Bearing these ideas in mind, we shall study the guiding principles of the simplex method for solving linear programming problems. For computational efficiency, we focus on the *revised simplex method*, which is a systematic procedure for implementing the steps of the original simplex method in a smaller array.

3.2 BASICS OF THE SIMPLEX METHOD

Consider the following linear programming problem in its standard form:

$$\text{Minimize} \quad z = \mathbf{c}^T\mathbf{x} \tag{3.1a}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}; \qquad \mathbf{x} \geq \mathbf{0} \tag{3.1b}$$

where $\mathbf{A}$ is an $m \times n$ matrix with full row rank, $\mathbf{b}$ can always be adjusted to be an m–dimensional nonnegative vector, and $\mathbf{c}, \mathbf{x} \in R^n$.

The simplex method was first conceived in the summer of 1947 by G. B. Dantzig. Over the past four decades, although many variants of the simplex method been developed to improve its performance, the basic ideas have remained the same. We study the basic ideas in this section.

Considering the fundamental theorem of linear programming, we know that if the feasible domain $P = \{\mathbf{x} \in R^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is nonempty, then the minimum objective value $z = \mathbf{c}^T\mathbf{x}$ over P either is unbounded or is attainable at an extreme point of P. This motivates the simplex method to restrict its iterations to the extreme points of P only. It starts with an extreme point of P, checks for optimality, and then moves to another extreme point with improved objective value if the current extreme point is not optimal. Owing to the correspondence between extreme points and basic feasible solutions as described in Corollary 2.2, the simplex method can be described in terms of basic feasible solutions in an iterative scheme:

Step 1: Find a basic feasible solution.

Step 2: Check if current basic feasible solution is optimal. If it is optimal, stop. Otherwise, go to next step.

Step 3: Move to a basic feasible solution with improved objective value, then return to Step 2.

For Step 1, two commonly used mechanisms of finding a starting basic feasible solution are the *two-phase method* and the *big-M method*. We shall introduce these two mechanisms in Section 3.4. Once a starting point is obtained, in Step 2 it is checked whether the current solution achieves the optimum. A stopping rule called *nonnegative reduced costs* will be introduced in Section 3.3 for this purpose. If the objective cost function can be further reduced, the stopping rule will be violated and the simplex method proceeds to Step 3 to find an improved basic feasible solution. Under the assumption of nondegeneracy, from Chapter 2, we know that each basic feasible solution has $n - m$ adjacent basic feasible solutions, which can be reached by moving along corresponding *edge directions* from the current solution with appropriate step lengths. The simplex method chooses an edge direction that leads to an adjacent basic feasible solution with improved objective value. This is the so-called *pivoting process*, which will be discussed in Section 3.3.

3.3 ALGEBRA OF THE SIMPLEX METHOD

In order to introduce the simplex method in algebraic terms, we standardize some notations here. For a given basic feasible solution $\mathbf{x}^*$, we can always denote it by

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{x}_B^* \\ \hline \mathbf{x}_N^* \end{bmatrix}$$

where the elements of the vector $\mathbf{x}_B^*$ represent the basic variables and the elements of vector $\mathbf{x}_N^*$ represent nonbasic variables. Needless to say, $\mathbf{x}_B^* \geq \mathbf{0}$ and $\mathbf{x}_N^* = \mathbf{0}$ for the basic feasible solution. Corresponding to the basic variables $\mathbf{x}_B^*$ and nonbasic variables $\mathbf{x}_N^*$, we partition $\mathbf{A}$ and $\mathbf{c}$ as

$$\mathbf{A} = [\mathbf{B}|\mathbf{N}] \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} \mathbf{c}_B \\ \hline \mathbf{c}_N \end{bmatrix} \tag{3.2}$$

where $\mathbf{B}$ is an $m \times m$ nonsingular matrix that is referred to as the *basis* and $\mathbf{N}$ is referred to as *nonbasis* with dimensionality of $m \times (n - m)$.

Once a basis $\mathbf{B}$ is known, every feasible solution $\mathbf{x} \in P$ can be rearranged in a corresponding order as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \hline \mathbf{x}_N \end{bmatrix}$$

with both $\mathbf{x}_B$ and $\mathbf{x}_N$ being nonnegative. Hence the linear programming problem defined by (3.1) becomes

$$\text{Minimize} \quad z = \mathbf{c}_B{}^T\mathbf{x}_B + \mathbf{c}_N{}^T\mathbf{x}_N \tag{3.3a}$$

$$\text{subject to} \quad \mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N = \mathbf{b}; \quad \mathbf{x}_B \geq \mathbf{0}; \quad \mathbf{x}_N \geq \mathbf{0} \tag{3.3b}$$

The equation in (3.3b) implies that

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \tag{3.4}$$

Substituting (3.4) back into (3.3a) results in

$$\begin{aligned} z &= \mathbf{c}_B{}^T\left(\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N\right) + \mathbf{c}_N{}^T\mathbf{x}_N \\ &= \mathbf{c}_B{}^T\mathbf{B}^{-1}\mathbf{b} + \left(\mathbf{c}_N{}^T - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{N}\right)\mathbf{x}_N \\ &= \mathbf{c}_B{}^T\mathbf{B}^{-1}\mathbf{b} + \mathbf{r}^T\begin{bmatrix} \mathbf{x}_B \\ \hline \mathbf{x}_N \end{bmatrix} \end{aligned} \tag{3.5}$$

where

$$\mathbf{r} = \begin{bmatrix} \mathbf{0} \\ \hline \mathbf{c}_N - (\mathbf{B}^{-1}\mathbf{N})^T\mathbf{c}_B \end{bmatrix} \tag{3.6}$$

Note that $\mathbf{r}$ is an n−dimensional column vector. Its first m components, corresponding to the basic variables, are set to be zero and the remaining $n - m$ components correspond to nonbasic variables. Also note that the objective value of z^* at current basic feasible solution $\mathbf{x}^*$ is $\mathbf{c}_B{}^T\mathbf{B}^{-1}\mathbf{b}$, since $\mathbf{x}_B^* = \mathbf{B}^{-1}\mathbf{b}$ and $\mathbf{x}_N^* = \mathbf{0}$ at this point. Consequently, Equation (3.5) becomes

$$z - z^* = \mathbf{r}^T\begin{bmatrix} \mathbf{x}_B \\ \hline \mathbf{x}_N \end{bmatrix} \quad \text{for each } \mathbf{x} \in P \tag{3.7}$$

Now it is apparent that if $\mathbf{r} \geq \mathbf{0}$, i.e., every component of $\mathbf{c}_N{}^T - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{N}$ (or equivalently, of $\left(\mathbf{c}_N - (\mathbf{B}^{-1}\mathbf{N})^T\mathbf{c}_B\right)^T$) is nonnegative, then $z - z^* \geq 0$ for each feasible solution $\mathbf{x} \epsilon P$, since

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \hline \mathbf{x}_N \end{bmatrix} \geq \mathbf{0}$$

In this case the current basic feasible solution $\mathbf{x}^*$ is an optimal solution. On the other hand, if any component of $\mathbf{r}$ is negative, its corresponding element of $\mathbf{x}_N$ may be increased from zero to some positive value (or equivalently a nonbasic variable is brought into the basis) to gain a reduction in the objective value z. Hence vector $\mathbf{r}$ is named the *reduced cost vector,* which consists of *reduced costs.* Summarizing previous discussions, we have derived the following result:

Theorem 3.1. If

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{x}_B^* \\ \hline \mathbf{x}_N^* \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix} \geq \mathbf{0}$$

is a basic feasible solution with nonnegative reduced costs vector $\mathbf{r}$, given by Equation (3.6), then $\mathbf{x}^*$ is an optimal solution to the linear programming problem (3.1).

Moreover, we have developed a stopping rule based on the appearance of nonnegative reduced costs.

3.3.1 Stopping the Simplex Method—Checking for Optimality

Let $\mathbf{x}^*$ be a current basic feasible solution with $\mathbf{B}$ being its corresponding basis, $\mathbf{N}$ the nonbasis, $\tilde{\mathbf{B}}$ the index set of basic variables in $\mathbf{x}^*$, and $\tilde{\mathbf{N}}$ the index set of nonbasic variables. Moreover, for each nonbasic variable x_q $(q \in \tilde{\mathbf{N}})$, let c_q be the cost coefficient associated with it and $\mathbf{N}_q$ the column in $\mathbf{N}$ that corresponds to x_q. Then Theorem 3.1 says that if

$$r_q = c_q - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{N}_q \geq 0 \qquad \text{for each} \quad q \in \tilde{\mathbf{N}} \tag{3.8}$$

then we can terminate the simplex method with an optimal solution $\mathbf{x}^*$. Otherwise, we have to move to another basic feasible solution for some potential improvement in the objective value.

Note that $\mathbf{N}_q = \mathbf{A}_q$ for each q in $\tilde{\mathbf{N}}$, since they represent the same columns.

3.3.2 Iterations of the Simplex Method—Moving for Improvement

After taking care of Step 2, we now focus on the process of moving to a basic feasible solution with improved objective value. The process includes finding a good moving direction *(direction of translation)* and an appropriate step length.

Direction of Translation. A *direction of translation* is a vector $\mathbf{d} \in R^n$ along which we propose to translate our current basic feasible solution. Since the idea of the simplex method is to hop from a current extreme point to an adjacent extreme point of P, we consider only those directions which point from a current extreme point to its adjacent neighbors. In other words, such a direction must be along an edge of P. Hence they are *edge directions.*

Consider the correspondence relation between the extreme points and the basic feasible solutions of P. We see that, under the assumption of nondegeneracy, each basic feasible solution (extreme point) of P has exactly $n - m$ (the number of nonbasic variables) adjacent neighbors in P. An adjacent basic feasible solution is obtained by introducing a nonbasic variable (increase its value from zero to positive) to replace one basic variable (reduce its value from positive to zero). The interrelationship is described by Equation (3.4). To be more specific, if a nonbasic variable x_q is being considered, the remaining nonbasic variables are kept with zero value and Equation (3.4) becomes

$$\mathbf{x}_B = \mathbf{B}^{-1}(\mathbf{b} - x_q\mathbf{A}_q) \tag{3.9}$$

where $\mathbf{A}_q$ is the column corresponding to x_q in $\mathbf{A}$. Hence we know that the edge direction corresponding to increasing x_q is given by

$$\mathbf{d}^q = \left(\frac{-\mathbf{B}^{-1}\mathbf{A}_q}{\mathbf{e}_q}\right) \quad \text{for} \quad q \in \tilde{\mathbf{N}} \tag{3.10}$$

where $\mathbf{e}_q$ is an $(n-m)$-dimensional vector with 1 at the position corresponding to x_q and 0 corresponding to other nonbasic variables. Note that $\mathbf{d}^q \in R^n$, and moving along this direction will increase x_q, keep other nonbasic variables at zero, and change the values of current basic variables according to Equation (3.9). Also note that, since $\mathbf{A}_q$ and $\mathbf{N}_q$ represent the same column of matrix $\mathbf{A}$,

$$\mathbf{A}\mathbf{d}^q = [\mathbf{B}|\mathbf{N}]\left(\frac{-\mathbf{B}^{-1}\mathbf{A}_q}{\mathbf{e}_q}\right) = \mathbf{A}_q - \mathbf{N}_q = \mathbf{0} \tag{3.11}$$

Therefore, under the assumption of nondegeneracy, the edge direction $\mathbf{d}^q$ is a *feasible direction,* because for current basic feasible solution $\mathbf{x}$ with a sufficient small scalar $\alpha > 0$,

$$\mathbf{A}(\mathbf{x} + \alpha\mathbf{d}^q) = \mathbf{A}\mathbf{x} + \alpha\mathbf{A}\mathbf{d}^q = \mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.12a}$$

and

$$\mathbf{x} + \alpha\mathbf{d}^q \geq \mathbf{0} \tag{3.12b}$$

However, for a degenerate basic feasible solution, since the value of some basic variables is zero, (3.12b) could be violated for any positive α. In this case, we have an *infeasible edge direction.* Any amount of translation along an infeasible direction causes infeasibility, which mandates the step length α to be zero. This happens because a degenerated basic feasible solution is *overdetermined* by more than n hyperplanes passing through it, and some edge directions lead to the region outside of P. Figure 3.1 illustrates this situation. In the figure, $\mathbf{x}$ is the current basic feasible solution, which is the intersection of three lines in a two-dimensional plane. Hence it is overdetermined. It is clear to see that $\mathbf{d}^1$ is a feasible edge direction, but $\mathbf{d}^2$ is not. More details of degeneracy will be discussed in Section 3.5.

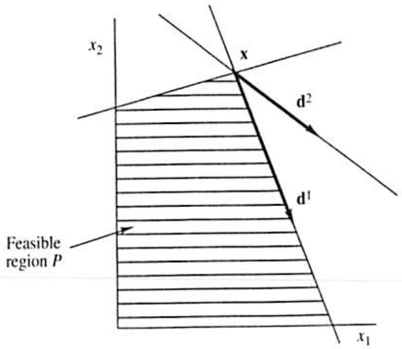

Figure 3.1

Now, for current basic feasible solution $\mathbf{x}^*$, suppose that $\mathbf{d}^q$ given by (3.10) is a feasible edge direction, our task is to determine if it is a *good direction of translation,* i.e., a direction which leads to an improvement in the objective value. This means we expect

$$\mathbf{c}^T(\mathbf{x}^* + \alpha\mathbf{d}^q) < \mathbf{c}^T\mathbf{x}^* \quad \text{for} \quad \alpha > 0. \tag{3.13}$$

Consequently, we require

$$\mathbf{c}^T\mathbf{d}^q = \left[\mathbf{c}_B^T|\mathbf{c}_N^T\right]\left[\frac{-\mathbf{B}^{-1}\mathbf{A}_q}{\mathbf{e}_q}\right] = c_q - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{A}_q < 0 \tag{3.14}$$

Note again that $\mathbf{A}_q$ and $\mathbf{N}_q$ are the same column vector, and Equation (3.14) actually requires the reduced cost $r_q < 0$ to assure the corresponding edge direction is a good direction of translation.

Summarizing our findings, we have the following theorem.

Theorem 3.2. Let

$$\mathbf{x}^* = \left[\frac{\mathbf{B}^{-1}\mathbf{b}}{\mathbf{0}}\right]$$

be a basic feasible solution to the linear programming problem defined by (3.1) with basis $\mathbf{B}$. If the reduced cost $r_q < 0$, for some nonbasic variable x_q, then the edge direction $\mathbf{d}^q$ given by (3.10) leads to an improvement in the objective value.

Note that when $\mathbf{x}^*$ is nondegenerate, each edge direction is a feasible direction, therefore a positive step length α can be chosen to translate current basic feasible solution along the direction $\mathbf{d}^q$ to a distinct adjacent neighbor with improved objective value.

However, when $\mathbf{x}^*$ is degenerate, $\mathbf{d}^q$ may become an infeasible edge direction that forces a step length $\alpha = 0$. In this case, no actual translation happens, and we stay at the same extreme point with two different representations in terms of basic variables.

Also note that for a feasible edge direction $\mathbf{d}^q$ with $r_q < 0$, if $\mathbf{d}^q \geq \mathbf{0}$, then $\mathbf{x}^* + \alpha\mathbf{d}^q$ is always feasible as long as $\alpha > 0$. Therefore, as α approaches infinity, the given linear programming problem becomes unbounded below and we have the following result.

Theorem 3.3. Let $\mathbf{x}^*$ be a basic feasible solution to the linear programming problem defined by (3.1). If there is a feasible edge direction $\mathbf{d}^q \geq \mathbf{0}$ with a reduced cost $r_q < 0$, for some nonbasic variable x_q, then the linear programming problem is unbounded below.

For a current basic feasible solution, it is possible to have more than one nonbasic variable with negative reduced cost. Among the corresponding good edge directions, in theory we can choose an arbitrary one as a direction of translation. Different rules for selecting a nonbasic variable to enter the basis result in variants of the simplex method. The two most commonly used rules are the *smallest index rule*, which picks the smallest index q for which $r_q < 0$, and the *largest reduction rule* which picks the index q with the most negative value of r_q. Although the second rule looks better in cost reduction as far as the current iteration is concerned, there is no evidence showing that it is an overall better choice.

Step Length. Once a good edge direction $\mathbf{d}^q$ given by (3.10) is selected as the direction of translation at the current basic feasible solution $\mathbf{x}^*$, we have to determine an appropriate *step length* $\alpha \geq 0$ such that $\mathbf{x}^* + \alpha\mathbf{d}^q$ becomes a new basic feasible solution by bringing in the nonbasic variable x_q and dropping out a basic variable to form a new basis. By Theorem 3.3, we know that if $\mathbf{d}^q \geq \mathbf{0}$ is a feasible direction, then the given linear programming problem is unbounded below. In case $\mathbf{d}^q$ has negative components, since $\mathbf{A}\mathbf{d}^q = \mathbf{0}$ has been verified before, in order to keep $\mathbf{x}^* + \alpha\mathbf{d}^q \geq \mathbf{0}$, we need to choose α according to the following formula:

$$\alpha = \underset{j \in \tilde{\mathbf{B}}}{\text{Minimum}} \left[-\frac{x_j^*}{d_j^q} \,\middle|\, d_j^q < 0 \right] \tag{3.15}$$

where x_j^* is the jth element of $\mathbf{x}^*$, $\tilde{\mathbf{B}}$ is the index set of basic variables, and d_j^q is the component in $\mathbf{d}^q$ corresponding to the basic variable x_j^*.

This formula is referred to in the literature as the *minimum ratio test*. It determines which basic variable will become nonbasic (with zero value) as the nonbasic variable x_q is introduced to the new basis. It is not difficult to show that under the assumption of nondegeneracy, there is a unique basic variable x_p ($p \in \tilde{\mathbf{B}}$) which provides a positive step length α leading to a distinct basic feasible solution. Also note that at a degenerate point, the step length obtained by (3.15) may become zero. In this case, although in theory we have changed our basis, we actually stay at the same extreme point of P. This process of changing basis is sometimes called the *pivoting process*. By *pivot-in* we mean a nonbasic variable entering the new basis, and *pivot-out* a basic variable leaving the current basis.

Just like the pivot-in process, there may be several basic variables achieving the minimum ratio at the same time. Among these candidates for pivot-out, different variants of the simplex method pick different candidates. But no evidence supports a particular variant of the simplex method for all cases.

The following theorem summarizes the discussions in this subsection:

Theorem 3.4. Let

$$\mathbf{x}^* = \left[\frac{\mathbf{B}^{-1}\mathbf{b}}{\mathbf{0}}\right]$$

be a basic feasible solution to the linear programming problem defined by (3.1) with basis $\mathbf{B}$. If a reduced cost $r_q < 0$ is found for some nonbasic variable x_q, then the edge direction $\mathbf{d}^q$ given by (3.10) together with the step length α determined by (3.15) lead to another basic feasible solution whose objective value is no worse than that of the current one.

Note that if $\mathbf{d}^q$ is a feasible direction and $\alpha > 0$, then the adjacent basic feasible solution obtained in Theorem 3.4 represents a distinct extreme point with improved objective value. However, when $\alpha = 0$, we stay at the same extreme point with the same objective value. Moreover, based on Theorems 3.1–3.4, we can sketch the key steps of the simplex algorithm as follows:

Step 1: Find a basic feasible solution $\mathbf{x}$ with a basis matrix $\mathbf{B}$ and nonbasis matrix $\mathbf{N}$.

Step 2: Compute the reduced cost r_q for each nonbasic variable x_q according to Equation (3.8). If $r_q \geq 0$ for each nonbasic variable, then stop. The current basic feasible solution is optimal. Otherwise, go to Step 3.

Step 3: Compute the direction of movement $\mathbf{d}^q$ with $r_q < 0$ according to (3.10). If $\mathbf{d}^q \geq \mathbf{0}$, then the linear programming problem is unbounded. Otherwise, compute the step length α according to (3.15), update the current basic feasible solution by $\mathbf{x} \leftarrow \mathbf{x} + \alpha\mathbf{d}^q$ and update the corresponding basis matrix. Go to Step 2.

The following example illustrates this procedure.

Example 3.1

$$\begin{array}{ll}
\text{Minimize} & -x_1 - x_2 - x_3 \\
\text{subject to} & 2x_1 \qquad\qquad + x_4 \qquad\qquad = 1 \\
& \qquad 2x_2 \qquad\qquad + x_5 \qquad = 1 \\
& \qquad\qquad 2x_3 \qquad\qquad + x_6 = 1 \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{array}$$

Note that

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

$$\mathbf{b} = [1 \quad 1 \quad 1]^T$$

and

$$\mathbf{c} = [-1 \quad -1 \quad -1 \quad 0 \quad 0 \quad 0]^T$$

Step 1: Let us pick x_4, x_5 and x_6 as basic variables, then

$$\begin{bmatrix} \mathbf{x}_B \\ \hline \mathbf{x}_N \end{bmatrix} = [x_4 \quad x_5 \quad x_6 \,|\, x_1 \quad x_2 \quad x_3]^T = [1 \quad 1 \quad 1 \quad | \quad 0 \quad 0 \quad 0]^T$$

is a basic feasible solution. If we consider x_4, x_5 and x_6 as slack variables, and draw a three dimensional graph based on the coordinates of x_1, x_2, x_3, this solution corresponds to the vertex (0,0,0) of the polyhedron $P = \{\mathbf{x} \in R^3 \,|\, 0 \leq x_i \leq 1/2, i = 1, 2, 3\}$.

In this case,

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{N} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$\tilde{\mathbf{B}} = \{4, 5, 6\} \qquad \tilde{\mathbf{N}} = \{1, 2, 3\}$$

$$\mathbf{c}_B = [0 \quad 0 \quad 0]^T, \quad \text{and} \quad \mathbf{c}_N = [-1 \quad -1 \quad -1]^T$$

Step 2: Following Equation (3.8), the reduced cost vector $\mathbf{r} = [0 \quad 0 \quad 0 \quad -1 \quad -1 \quad -1]^T$. Since $r_1 = r_2 = r_3 = -1 < 0$, the current solution is not optimal, and we go to Step 3.

Step 3: Let us pick a nonbasic variable with negative reduced cost, say $q = 1$. This means x_1 is entering the basis. According to (3.10), $\mathbf{d}^1 = [-2 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]^T$, and we use (3.15) to determine the step length. The result shows that x_4 is leaving the basis and $\alpha = 1/2$. Hence the new basic feasible solution is given by

$$\begin{bmatrix} \mathbf{x}_B \\ \hline \mathbf{x}_N \end{bmatrix} + \alpha \mathbf{d}^1 = [1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0]^T + 1/2 \times [-2 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]^T$$

$$= [0 \quad 1 \quad 1 \quad 1/2 \quad 0 \quad 0]^T$$

Note that the new basic variables are x_5, x_6, x_1 and nonbasic variables are x_2, x_3, x_4. Moreover,

$$\mathbf{x}_B = [x_5 \quad x_6 \quad x_1]^T = [1 \quad 1 \quad 1/2]^T$$

$$\mathbf{x}_N = [x_2 \quad x_3 \quad x_4]^T = [0 \quad 0 \quad 0]^T$$

the new basis matrix

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and the corresponding nonbasis

$$\mathbf{N} = \begin{bmatrix} 0 & 0 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

By now we have completed one simplex iteration. It is easy to verify that the new solution corresponds to the vertex (1/2, 0, 0) with a reduced objective value $-1/2$. If we go back to Step 2 for two more iterations, we can reach an optimal solution

$$[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6]^T = [1/2 \quad 1/2 \quad 1/2 \quad 0 \quad 0 \quad 0]^T$$

with an objective value $-3/2$.

As we discussed before, under the assumption of nondegeneracy, at each iteration the step length α in Step 3 is always positive. This leads the simplex method to a distinct extreme point with lower objective value after each iteration. Hence the simplex method is not going to revisit any extreme point that has been visited before. Since the total number of extreme points of the feasible domain P is finite, we have the following theorem:

Theorem 3.5. Under the assumption of nondegeneracy, the revised simplex method terminates in a finite number of iterations.

In the presence of degeneracy, the simplex method may be trapped into an endless loop without termination. This phenomenon is called *cycling*, and we shall study it in Section 3.5.

3.4 STARTING THE SIMPLEX METHOD

For some linear programming problems, it is easy to find a starting basic feasible solution. But this task can be as difficult as finding an optimal solution from a given basic feasible solution. In this section we introduce two commonly used starting mechanisms.

3.4.1 Two-Phase Method

Consider the linear programming problem defined by (3.1). Without loss of generality, we can assume $\mathbf{b} \geq \mathbf{0}$. Remember that we have n variables and m constraints. We let $\mathbf{x}^a = (x_1{}^a, x_2{}^a, \ldots, x_m{}^a)^T \in R^m$ be an m-dimensional vector of *artificial variables*, and consider an associated *Phase I* problem:

$$\text{Minimize} \quad z = \sum_{i=1}^{m} x_i{}^a \tag{3.16a}$$

$$\text{subject to} \quad \mathbf{Ax} + \mathbf{x}^a = \mathbf{b};\quad \mathbf{x} \geq \mathbf{0};\quad \mathbf{x}^a \geq \mathbf{0} \tag{3.16b}$$

Note that the Phase I problem defined by (3.16) has $n + m$ variables and m constraints. Since $\mathbf{b} \geq \mathbf{0}$ is assumed,

$$\left[\begin{array}{c}\mathbf{x}\\ \hline \mathbf{x}^a\end{array}\right] = \left[\begin{array}{c}\mathbf{0}\\ \hline \mathbf{b}\end{array}\right]$$

is a basic feasible solution to the Phase I problem. Also note that the Phase I problem is always bounded below by 0, since $\mathbf{x}^a \geq \mathbf{0}$ is required. Therefore, applying the simplex method (with a cycling prevention mechanism to be discussed later) to a Phase I problem always results in an optimal solution

$$\left[\begin{array}{c}\mathbf{x}^0\\ \hline \mathbf{x}^{a*}\end{array}\right]$$

There are two possible cases:

Case 1 $\mathbf{x}^{a*} \neq \mathbf{0}$. If $\mathbf{x}^{a*} \neq \mathbf{0}$, then the original problem is infeasible, since if the original problem has a feasible solution $\mathbf{x}$, then

$$\left[\begin{array}{c}\mathbf{x}\\ \hline \mathbf{0}\end{array}\right]$$

is feasible to the Phase I problem with zero objective value. This violates the optimality of the solution

$$\left[\begin{array}{c}\mathbf{x}^0\\ \hline \mathbf{x}^{a*}\end{array}\right]$$

Case 2 $\mathbf{x}^{a*} = \mathbf{0}$. In this case, if the current basis does not contain any artificial variable in it, then $\mathbf{x}^0$ forms a starting basic feasible solution to the original problem. In particular, if it is nondegenerate, $\mathbf{x}^0$ has exactly m positive elements in it to form the basis. On the other hand, if it is degenerate with at least one artificial variable remaining in the basis, say $x_i{}^a = 0$ is the kth basic variable in current basis, then we denote $\mathbf{e}_k$ to be an $m-$dimensional vector with its kth element being equal to 1 and the rest being equal to 0 and consider the value $\mathbf{e}_k^T\mathbf{B}^{-1}\mathbf{A}_q$ for each nonbasic variable x_q associated with the current optimal solution.

There are two possibilities:

1. If $\mathbf{e}_k^T\mathbf{B}^{-1}\mathbf{A}_q \neq 0$ for a nonbasic variable x_q, then we can bring $x_q = 0$ to the current basis as a basic variable to replace $x_i{}^a$. In this case, the optimal solution to the Phase I problem provides a starting basis without any artificial variable in it for the original linear programming problem.

2. If $\mathbf{e}_k^T\mathbf{B}^{-1}\mathbf{A}_q = 0$ for every nonbasic variable x_q, then we know the kth row of the constraint set $\mathbf{Ax} = \mathbf{b}$ is redundant. In this case we can remove that redundant row from the original constraints and restart the Phase I problem.

Validation of these two cases is left to the reader as an exercise.

3.4.2 Big-*M* Method

Unlike the two-phase method, the big-M method imposes a large penalty $M > 0$ for each artificial variable and solves the following linear programming problem:

$$\text{Minimize} \quad z = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} M x_i{}^a \tag{3.17a}$$

$$\text{subject to} \quad \mathbf{Ax} + \mathbf{x}^a = \mathbf{b};\quad \mathbf{x} \geq \mathbf{0},\quad \mathbf{x}^a \geq \mathbf{0} \tag{3.17b}$$

Note that

$$\left[\begin{array}{c}\mathbf{x}\\ \hline \mathbf{x}^a\end{array}\right] = \left[\begin{array}{c}\mathbf{0}\\ \hline \mathbf{b}\end{array}\right]$$

is a starting basic feasible solution and M can be thought of as the penalty to be paid for $\mathbf{x}^a \neq \mathbf{0}$. In theory, when M is chosen to be large enough, the artificial variables will not appear in the final solution. In reality, we may raise a fundamental issue, namely how big should M be? This is a very important issue in implementation. Consider the following simple example:

Example 3.2

$$\text{Minimize} \quad x_1$$

$$\text{subject to} \quad \epsilon x_1 - x_2 - x_3 = \epsilon \qquad (\epsilon > 0)$$

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

The associated big-M problem can be stated as follows:

$$\text{Mimimize} \quad x_1 + Mx_4$$

$$\text{subject to} \quad \epsilon x_1 - x_2 - x_3 + x_4 = \epsilon$$

$$x_1, x_2, x_3, x_4 \geq 0$$

It is clear that $\bar{\mathbf{x}}^T = (1, 0, 0, 0)$ and $\hat{\mathbf{x}}^T = (0, 0, 0, \epsilon)$ are two basic feasible solutions to the big-M problem with objective value of 1 and ϵM, respectively. Since $\bar{\mathbf{x}}$ corresponds to a basic feasible solution to the original problem but $\hat{\mathbf{x}}$ does not, we have to make sure that $1 < \epsilon M$, or $M > 1/\epsilon$ for any given $\epsilon > 0$. Consequently we see the difficulty of choosing M for a general implementation.

When the simplex method is applied to solve the big-M problem (3.17) with sufficient large $M > 0$, since the problem is feasible, we either arrive at an optimal solution to the big-M problem or conclude that the big-M problem is unbounded below. But what can we say about the original problem in each case?

Case 1 The big-M problem has a finite optimum at $(\mathbf{x}^*, \mathbf{x}^{a*})$. In this case, if $\mathbf{x}^{a*} = \mathbf{0}$, then $\mathbf{x}^*$ is an optimal solution to the original linear programming problem. The reason is that, for each feasible solution $\mathbf{x}$ of the original linear program,

$$\begin{bmatrix} \mathbf{x} \\ \hline \mathbf{0} \end{bmatrix}$$

is a feasible solution to the big-M problem. Hence we know

$$\mathbf{c}^T\mathbf{x} = \mathbf{c}^T\mathbf{x} + M \times 0 \geq \mathbf{c}^T\mathbf{x}^* + M\sum_{i=1}^{m} x_i^{a*} = \mathbf{c}^T\mathbf{x}^*$$

On the other hand, if $\mathbf{x}^{a*} \neq \mathbf{0}$, then we can conclude that the original linear programming problem has no feasible solution. To show this case, we assume that the original problem has a feasible solution $\mathbf{x}$. Then

$$\begin{bmatrix} \mathbf{x} \\ \hline \mathbf{0} \end{bmatrix}$$

is a feasible solution to the big-M problem and

$$\mathbf{c}^T\mathbf{x} = \mathbf{c}^T\mathbf{x} + M \times 0 \geq \mathbf{c}^T\mathbf{x}^* + M\sum_{i=1}^{m} x_i^{a*}$$

But this inequality is impossible, since M is sufficiently large and at least one $x_i^{a*} > 0$. Therefore $\mathbf{x}$ could not be a feasible solution to the original problem.

Case 2 The big-M problem is unbounded below. Similar to Case 1, we can see that if all artificial variables are equal to zero, then the original problem is also unbounded below. Otherwise, if at least one artificial variable is positive, then the original problem is infeasible.

3.5 DEGENERACY AND CYCLING

We have seen that the step length α can assume zero value, if one or more of the basic variables involved in the minimum ratio test turn out to be zero. In this case, the current basic feasible solution is degenerate, and the new basic feasible solution stays at the same extreme point although the new basis is different. In other words, although it appears that geometrically we are stagnant at an extreme point of the feasible domain P, algebraically we are not. Thus technically we can proceed with the simplex iterations even if $\alpha = 0$, but the real danger is that at some point we might return to the old basis. This phenomenon is called *cycling*. Note that for a degenerate basic feasible solution $\mathbf{x}$ with $p(< m)$ positive components, we may have up to

$$C((n-p), (n-m)) = \frac{(n-p)!}{(n-m)!(m-p)!}$$

different bases corresponding to the same extreme point $\mathbf{x}$. The following example given by E. M. L. Beale in 1955 shows that the simplex method could be trapped into cycling problem if the largest reduction rule is used for entering basis and the minimum ratio test with smallest index rule as tie-breaker is used for leaving basis.

Example 3.3

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

In the exercise, it can be verified that the optimal solution is given by $x_1 = 3/4, x_2 = x_3 = 0, x_4 = 1, x_5 = 0, x_6 = 1, x_7 = 0$ with an optimal objective value $-5/4$. However, if we start with a basis $\{x_1, x_2, x_3\}$ and follow the previously mentioned rules for pivoting, then the successive new bases are $\{x_4, x_2, x_3\}$, $\{x_4, x_5, x_3\}$, $\{x_6, x_5, x_3\}$, $\{x_6, x_7, x_3\}$, $\{x_1, x_7, x_3\}$, and return to $\{x_1, x_2, x_3\}$. If the same sequence of pivots is repeated again and again, the simplex method will cycle forever among these bases without reaching the optimal solution.

Another interesting point is that a degenerate basic feasible solution can be optimal even if some of the reduced costs are negative, since the corresponding edge directions may be infeasible. This phenomenon is caused by the *overdetermined system* associated with a degenerate basic feasible solution.

To be more precise, for a nondegenerate basic feasible solution $\mathbf{x}$, it must lie in the intersection of n linearly independent hyperplanes defined by the system

$$\overline{\mathbf{M}}\mathbf{x} = \begin{bmatrix} \mathbf{b} \\ \hline \mathbf{0} \end{bmatrix} \tag{3.18}$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \hline \mathbf{x}_N \end{bmatrix} \quad \text{and} \quad \overline{\mathbf{M}} = \left[\begin{array}{c|c} \mathbf{B} & \mathbf{N} \\ \hline \mathbf{0} & \mathbf{I} \end{array}\right] \tag{3.19}$$

As a matter of fact, since $\mathbf{B}$ is nonsingular, $\overline{\mathbf{M}}$ is also nonsingular and $\mathbf{x}$ is uniquely determined by

$$\mathbf{x} = \overline{\mathbf{M}}^{-1}\begin{bmatrix} \mathbf{b} \\ \hline \mathbf{0} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{B}^{-1} & -\mathbf{B}^{-1}\mathbf{N} \\ \hline \mathbf{0} & \mathbf{I} \end{array}\right]\begin{bmatrix} \mathbf{b} \\ \hline \mathbf{0} \end{bmatrix} \tag{3.20}$$

Nevertheless, for a degenerate basic feasible solution $\mathbf{x} \in R^n$, it satisfies not only the n equations in (3.18), but also at least one more linear equation

$$x_p = 0$$

for some basic variable x_p. Hence it is *overdetermined* by more than n linear equations. Therefore, some edge directions lead to infeasibility, as shown in Figure 3.1.

Note that the matrix $\overline{\mathbf{M}}$ is called the *fundamental matrix* and each edge direction $\mathbf{d}^q$ is a column of $\overline{\mathbf{M}}^{-1}$ which corresponds to a nonbasic variable x_q.

3.6 PREVENTING CYCLING

Having looked at the trap of cycling, we need some means to prevent it from happening. As we have seen, cycling can occur only when degeneracy is encountered. Since it is intuitively clear that degenerate basic feasible solutions can be eliminated by slightly perturbing the constraint parameters (resulting in only a slightly perturbed optimal value and optimal solutions), it should not be surprising that cycling can be prevented. Among several methods available for the prevention of cycling, two most commonly used are the *lexicographic rule* proposed by G. B. Dantzig, A. Orden, and P. Wolfe in 1955, and the *Bland's rule* due to R. G. Bland in 1977.

Observe that in the absence of degeneracy the objective values at each iteration of the simplex method form a strictly decreasing monotone sequence that guarantees no basis will be repeated. When degeneracy is involved, the sequence is no longer strictly decreasing. To prevent revisiting the same basis, we need to incorporate another index to keep some strictly monotone property for cycling prevention.

3.6.1 Lexicographic Rule

Basically, the lexicographic rule is used to select a leaving variable from the current basis. It ensures no cycles by the fact that while the objective value $\mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b}$ may remain constant in the presence of degeneracy, the vector $\left[\mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b}|\mathbf{c}_B^T\mathbf{B}^{-1}\right]^T$ can be kept lexicographically monotone decreasing.

In this rule, we first use the minimum ratio test (3.15) to decide pivot-out candidates. If the test generates a unique index, then the corresponding variable leaves the basis. In case there is a tie among several indices, we restrict ourselves to these indices and conduct another minimum ratio test with the value of x_j^* being replaced by its corresponding element in the vector $\mathbf{B}^{-1}\mathbf{A}_{p1}$, where $\mathbf{A}_{p1}$ is the column in $\mathbf{A}$ corresponding to the basic variable x_{p1} with the smallest index. If the tie is still unbroken, we conduct further minimum ratio tests on those still tied indices by using $\mathbf{B}^{-1}\mathbf{A}_{p2}$, where $\mathbf{A}_{p2}$ is the column in $\mathbf{A}$ corresponding to the basic variable x_{p2} with the second smallest index, and so forth. In the exercise, we show that when or before all the m columns of basic variables are used, the tie must be broken, and the unique index leads to a lexicographically monotone-decreasing sequence of $\left[\mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b}|\mathbf{c}_B^T\mathbf{B}^{-1}\right]^T$.

3.6.2 Bland's Rule

Bland's rule is very simple. It specifies the choice of both the entering and leaving variables. In this rule, variables are first ordered in sequence, then

1. Among all nonbasic variables with negative reduced costs, choose the one with the smallest index to enter the basis.
2. When there is a tie in the minimum ratio test, choose the basic variable with the smallest index to leave the basis.

Bland's rule actually creates the following monotone property (to be proved in an exercise): If a variable x_q enters the basis, then it cannot leave the basis until some other variable with a larger index, which was nonbasic when x_q enters, also enters the basis. This monotone property prevents cycling, because in a cycle any variable that enters the basis must also leave the basis, which implies that there is some largest indexed variable that enters and leaves the basis. This certainly contradicts the monotone property.

3.7 THE REVISED SIMPLEX METHOD

In this section we introduce the revised simplex method, which is a computationally efficient implementation of the simplex approach. It does not need the *simplex tableau* used in the original simplex method, but it computes all pertinent information in a systematic and space saving-manner.

Consider the sketched simplex iterative scheme. From the implementation point of view, the most laborious computation is spent on calculating $\mathbf{B}^{-1}$ at each iteration. Once this inverse matrix is known, other computational work becomes simple. However, a straightforward implementation of inverting an $m \times m$ matrix may require $O(m^3)$ elementary operations. Moreover, it is undesirable to compute the inverse matrix explicitly because of the numerical-stability and error-propagation problem (due to round-off and truncation errors resulting from finite word length of computers). In the revised simplex method, instead of inverting the basis matrix $\mathbf{B}$ directly, we get around the problem by solving an equivalent system of simultaneous equations.

A step-by-step computation procedure of the revised simplex algorithm can be described as follows:

Let

$$\mathbf{x} = \left[\frac{\mathbf{x}_B}{\mathbf{0}}\right]$$

be a current basic feasible solution (which may be obtained by the two-phase method), A_{j_i} be the j_ith column of $\mathbf{A}$, $\mathbf{B} = [A_{j_1} A_{j_2} A_{j_3} \ldots A_{j_m}]$ be the basis, and $\tilde{\mathbf{B}} = \{j_1, j_2, j_3, \ldots, j_m\}$ be the index set of the basic variables.

Step 1: Compute the "simplex multipliers," $\mathbf{w}$, by solving the system

$$\mathbf{B}^T\mathbf{w} = \mathbf{c}_B \tag{3.21}$$

where $\mathbf{c}_B$ is cost vector corresponding to the basic variables, i.e., $\mathbf{c}_B = (c_{j_1}, c_{j_2}, c_{j_3}, \ldots, c_{j_m})^T$.

Step 2: Compute the reduced costs

$$r_j = c_j - \mathbf{w}^T A_j \qquad \forall j \notin \tilde{\mathbf{B}}. \tag{3.22}$$

Step 3 (check for optimality): If $r_j \geq 0 \quad \forall j \notin \tilde{\mathbf{B}}$, then STOP. The current solution is OPTIMAL.

Step 4 (enter the basis): Choose $q \notin \tilde{\mathbf{B}}$ such that $r_q < 0$.

Step 5 (edge direction): Compute $\mathbf{d}^q$ by solving the system

$$\mathbf{B}\mathbf{d} = -\mathbf{A}_q \tag{3.23}$$

and set $\mathbf{d}^q = \begin{bmatrix} \mathbf{d} \\ \mathbf{e}_q \end{bmatrix}$.

Step 6 (check for unboundedness): If $\mathbf{d} \geq \mathbf{0}$, then STOP. The problem is unbounded below.

Step 7 (leave the basis and step-length): Find an index j_p and step-length α according to

$$\alpha = \frac{-x_{j_p}}{d_{j_p}} = \min_{1 \leq i \leq m} \left\{ \frac{-x_{j_i}}{d_{j_i}} \,\middle|\, d_{j_i} < 0 \right\} \tag{3.24}$$

Step 8 (update): Set

$$x_q \leftarrow \alpha \tag{3.25}$$

$$x_{j_i} \leftarrow x_{j_i} + \alpha d_{j_i} \quad \text{for} \quad 1 \leq i \leq m \tag{3.26}$$

$$\mathbf{B} \leftarrow \mathbf{B} + [\mathbf{A}_q - \mathbf{A}_{j_p}]\mathbf{e}_p^T \tag{3.27}$$

$$\tilde{\mathbf{B}} \leftarrow \tilde{\mathbf{B}} \cup \{q\} \backslash \{j_p\} \tag{3.28}$$

Go to Step 1.

Note that $\mathbf{B}^{-1}$ is implicitly calculated in both (3.21) and (3.23). One can use the well-known *Gauss Jordan elimination* method for solving systems of equations. However, a more popular implementation used in most modern computer packages is the *LU factorization* method, since it is more efficient, accurate, and numerically stable. This method is particularly preferred when the problem is sparse and in large scale.

The basic idea of LU factorization method is to *triangularize* the matrix $\mathbf{B}$ as a product of a *lower triangular matrix* $\mathbf{L}$ and an *upper triangular matrix* $\mathbf{U}$. In this way, solving (3.21) becomes solving

$$(\mathbf{L}\mathbf{U})^T\mathbf{w} = \mathbf{U}^T\mathbf{L}^T\mathbf{w} = \mathbf{c}_B \tag{3.29}$$

Since $\mathbf{U}^T$ is a lower triangular matrix, we can first denote $\mathbf{L}^T\mathbf{w}$ by $\mathbf{y}$ and solve

$$\mathbf{U}^T\mathbf{y} = \mathbf{c}_B \tag{3.30}$$

by the *forward solve process,* in which we obtain the first element of $\mathbf{y}$ directly from the first equation in (3.30), and then substitute it into the second equation in (3.30) for the second element of $\mathbf{y}$, and so on. Once $\mathbf{y}$ is obtained, then $\mathbf{w}$ can be obtained by solving

$$\mathbf{L}^T\mathbf{w} = \mathbf{y}. \tag{3.31}$$

This time, since $\mathbf{L}^T$ is upper triangular, $\mathbf{w}$ can be easily solved by the *backward solve process,* in which we obtain the last element of $\mathbf{w}$ directly from the last equation in (3.31), and then substitute it into the second last equation in (3.31) for the second last element of $\mathbf{w}$, and so on. Similar techniques work for solving (3.23), too. For more serious implementation, one is encouraged to know more about how to obtain the L and U factors, how to update them, and how to use scaling techniques for numerical accuracy. These will be covered in Chapter 10.

Also note that in (3.27), $\mathbf{e}_p$ is an m-vector with one at its pth element and zero everywhere else. The right-hand side of (3.27) uses matrix operations to replace the column $\mathbf{A}_{j_p}$ by $\mathbf{A}_q$ in the basis $\mathbf{B}$. The readers can easily check this out. As to (3.28), it simply means that we add index q and drop j_p in the index set $\tilde{\mathbf{B}}$.

Given below are two examples. One has degenerate basic feasible solutions, which illustrate how the revised simplex method works.

Example 3.4 *Nondegenerate*

Consider the following linear programming problem:

$$\begin{aligned} \text{Minimize} \quad & -3x_1 - 2x_2 \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 40 \\ & 2x_1 + x_2 + x_4 = 60 \\ & x_1, x_2, x_3, x_4, \geq 0 \end{aligned}$$

Note that this problem and Example 2.1 have the same feasible domain but different objective function. The graph of its feasible domain can be referred to Figure 2.2. To solve this problem, we first note that

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 40 \\ 60 \end{bmatrix}, \quad \mathbf{c}^T = [-3 \quad -2 \quad 0 \quad 0]$$

An easy starting basic feasible solution can be identified at $\mathbf{x}^0 = [0 \quad 0 \quad 40 \quad 60]^T$ with the basis matrix

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

index set $\tilde{\mathbf{B}} = \{3, 4\}$ and $\mathbf{c}_B = [0 \quad 0]^T$. This solution corresponds to the extreme point $(0, 0)$ in Figure 2.2, and the objective value at this point is zero. The revised simplex method proceeds as follows.

Step 1: Compute the simplex multipliers **w** by solving

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T \mathbf{w} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which implies that $\mathbf{w} = [0 \quad 0]^T$.

Step 2: Compute the reduced costs

$$r_1 = c_1 - \mathbf{w}^T A_1 = -3 - [0 \quad 0]^T [1 \quad 2] = -3$$

$$r_2 = c_2 - \mathbf{w}^T A_2 = -2 - [0 \quad 0]^T [1 \quad 1] = -2$$

Step 3 (check for optimality): Since $r_i < 0$, for $i = 1, 2$, the current solution is not optimal.

Step 4 (enter the basis): We may choose $q = 1$ here, since $r_1 < 0$. (Actually, we may choose $q = 2$, too.)

Step 5 (edge direction): Since $\mathbf{A}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, we compute **d** by solving

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{d} = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

which results in $\mathbf{d} = [-1 \quad -2]^T$. Note that the edge direction is given by $\mathbf{d}^1 = [1 \quad 0 \quad -1 \quad -2]^T$.

Step 6 (check for unboundedness): Since $\mathbf{d} < \mathbf{0}$, no unboundedness has been detected, and we proceed further.

Step 7 (leave the basis and step length): The minimum ratio test shows that

$$\alpha = \frac{-x_4}{d_4} = \min_{3 \le i \le 4} \left\{ \frac{-x_i}{d_i} \right\} = \min \left\{ \frac{-40}{-1}, \frac{-60}{-2} \right\} = 30$$

Hence x_4 will leave the basis.

Step 8 (update): Set

$$x_1 \leftarrow \alpha = 30 \quad \text{(enter the basis)}$$

x_2 stays at zero

$$x_3 \leftarrow 40 + (30)(-1) = 10$$

$$x_4 \leftarrow 60 + (30)(-2) = 0 \quad \text{(leave the basis)}$$

$$\mathbf{B} \leftarrow \mathbf{B} + [A_1 - A_4][0 \quad 1] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \left[\begin{pmatrix} 1 \\ 2 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] [0 \quad 1] = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$$

$$\tilde{\mathbf{B}} \leftarrow \tilde{\mathbf{B}} \cup \{1\} \backslash \{4\} = \{3, 1\}$$

This finishes one iteration of the revised simplex method. The new solution $(x_1, x_2) = (30, 0)$ is a vertex of the polytope, which is adjacent to the original

---

vertex $(x_1, x_2) = (0, 0)$ but has a smaller objective value (-90 as opposed to 0). On the next iteration, the revised simplex method will step to the vertex whose first two coordinates are (20, 20), and this one will turn out to be optimal. Its objective value is -100.

Now we add one more constraint, $x_1 \le 30$, to make the previous example degenerate at (30, 0) and consider the following problem:

Example 3.5 *Degenerate*

$$\begin{array}{ll} \text{Minimize} & -3x_1 - 2x_2 \\ \text{subject to} & x_1 + x_2 + x_3 \quad\quad = 40 \\ & 2x_1 + x_2 + x_4 \quad\quad = 60 \\ & x_1 + x_5 \quad\quad\quad\quad = 30 \\ & x_1, x_2, x_3, x_4, x_5 \ge 0 \end{array}$$

Figure 3.2 is a two-dimensional graph of the feasible domain P. Note that the extreme point (30, 0) is now "overdetermined" as the intersection of three lines, instead of two.

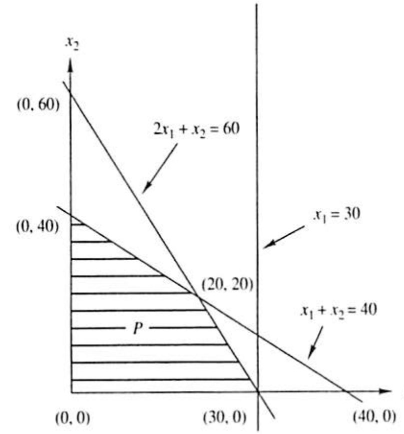

Figure 3.2

For the revised simplex method, we have

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 40 \\ 60 \\ 30 \end{bmatrix}, \quad \text{and} \quad \mathbf{c}^T = [-3 \quad -2 \quad 0 \quad 0 \quad 0]$$

As in Example 3.4, we start it from the extreme point (0, 0) which corresponds to the basic feasible solution $\mathbf{x}^0 = [0 \quad 0 \quad 40 \quad 60 \quad 30]^T$ with the basis matrix

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and the index set $\tilde{\mathbf{B}} = \{3, 4, 5\}$.

Carrying out one iteration of the revised simplex method as we did in Example 3.4, the new solution becomes $\mathbf{x}^1 = [30 \quad 0 \quad 10 \quad 0 \quad 0]^T$ which is a degenerate solution obtained from the negative reduced cost of r_1 and the minimum ratio test with $\alpha = 30$ and a tie between x_4 and x_5. This means x_1 is entering the basis and either x_4 or x_5 is leaving the basis. Let us choose x_5 to leave the basis. Therefore our current basis becomes

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

with an index set $\tilde{\mathbf{B}} = \{3, 4, 1\}$.

Continue one more iteration, it can be easily checked that $\mathbf{w} = [0 \quad 0 \quad -3]^T$, $r_2 < 0$, $\mathbf{d} = [-1 \quad -1 \quad 0]^T$, and $\alpha = \min\left[\frac{-10}{-1}, \frac{0}{-1}\right] = 0$. Therefore we know that x_2 enters the basis, x_4 leaves the basis, and the step length is zero. Updating related information, we have reached a new basic feasible solution $\mathbf{x}^2 = [30 \quad 0 \quad 10 \quad 0 \quad 0]^T$ with a new basis index $\tilde{\mathbf{B}} = \{3, 2, 1\}$. Note that $\mathbf{x}^1 = \mathbf{x}^2$. This means we actually stay at the same extreme point (30, 0) in Figure 3.2 owing to the zero step length.

For one more iteration, it can be checked out that x_5 is entering the basis to replace x_3, and the current basic feasible solution $\mathbf{x}^3 = [20 \quad 20 \quad 0 \quad 0 \quad 10]^T$ which is an optimal solution.

Note that in the previous example the revised simplex method avoids cycling problem even without any cycling prevention mechanism. But this is not true in general.

3.8 CONCLUDING REMARKS

In this chapter, we have seen the basic concepts behind the simplex method and have developed a computationally attractive procedure to implement the revised simplex method. In the next chapter we will study the duality theory of linear programming and develop two more implementation schemes for the simplex method, namely the dual simplex method and the primal dual simplex method. The performance of the simplex method will be discussed in Chapter 5.

REFERENCES FOR FURTHER READING

3.1. Bartels, R. H., and Golub, G. H., "The simplex method for linear programming using LU decomposition," *Communications of the ACM* 12, 26–268 (1969).

3.2. Bazaraa, M. S., Jarvis, J. J., and Sherali, H. D., *Linear Programming and Network Flows* (2d ed.), John Wiley, New York (1990).

3.3. Beale, E. M. L., "Cycling in the dual simplex algorithm," *Naval Research Logistics Quarterly* 2, 269–276 (1955).

3.4. Bland, R. G, "New finite pivoting rules for the simplex method," *Mathematics of Operations Research* 2, 103–107 (1977).

3.5. Chvatal, V., *Linear Programming*, Freeman, San Francisco (1983).

3.6. Dantzig, G. B., *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ (1963).

3.7. Gill, P. E., Murray, W., and Wright, M. H., *Numerical Linear Algebra and Optimization*, Vol. 1, Addison-Wesley, Redwood City, CA (1991).

3.8. Goldfarb, D, and Todd, M. J., "Linear programming," in *Optimization*, Handbook in Operations Research and Management Science, ed. by Nemhauser, G. L. and Rinnooy Kan, A. H. G., Vol. 1, 73–170, Elsevier-North Holland, Amsterdam (1989).

3.9. Golub, G. H., and Van Loan, C. F., *Matrix Computations*, Johns Hopkins University Press, Baltimore (1983).

3.10. Kotiah, T., and Slater, N., "On two-server Poisson queues with two types of customers," *Operations Research* 21, 597–603 (1973).

3.11. Luenberger, D. G., *Introduction to Linear and Nonlinear Programming* (2d ed.), Addison-Wesley, Reading, MA (1973).

EXERCISES

3.1. Consider a linear programming problem with its feasible domain $P = \{\mathbf{x} \in R^n \mid \mathbf{A}\mathbf{x} \le \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}$. If $\mathbf{b} \ge \mathbf{0}$, suggest an easy way to find a starting basic feasible solution.

3.2. Consider a linear programming problem in its standard form with $P = \{\mathbf{x} \in R^n \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}$.

(a) Let $\mathbf{d} \in R^n$. Show that a necessary condition for $\mathbf{d}$ to be a feasible direction is that $\mathbf{A}\mathbf{d} = \mathbf{0}$.

(b) Suppose $\mathbf{x} = (x_1, \dots, x_n)^T \in P$ with $x_i > 0$ when $d_i \ne 0$. Show that there exists a scalar $\alpha > 0$ such that $\mathbf{x} + \alpha\mathbf{d} \ge \mathbf{0}$.

3.3. Consider the following linear programming problem:

$$\begin{array}{ll} \text{Minimize} & -2x_1 - x_2 + x_3 + x_4 + 2x_5 \\ \text{subject to} & -2x_1 + x_2 + x_3 + x_4 + x_5 = 12 \\ & -x_1 + 2x_2 + x_4 - x_5 \qquad = 5 \\ & x_1 - 3x_2 + x_3 + 4x_5 \qquad = 11 \\ & x_1, x_2, x_3, x_4, x_5 \qquad \ge 0 \end{array}$$

Let x_3, x_4, and x_5 be basic variables and x_1, x_2 nonbasic.

(a) Write down $\mathbf{B}$, $\mathbf{N}$ and $\overline{\mathbf{M}}$, calculate $\mathbf{B}^{-1}$, $\overline{\mathbf{M}}^{-1}$ and the corresponding basic feasible solution.

(b) Applying the Gaussian elimination method to the three constraints, you can express basic variables x_3, x_4, and x_5 in terms of nonbasic variables x_1 and x_2. Now reformulate the linear programming problem in terms of two nonbasic variables.

(c) Draw a two-dimensional graph for the reformulated linear programming problem and explain why you can represent the feasible domain P of the original linear program by a two-dimensional graph, although $P \subset R^5$.

(d) Mark the basic feasible solution of (a) on the two-dimensional graph.

(e) Calculate $\mathbf{B}^{-1}\mathbf{A}$ and $\mathbf{B}^{-1}\mathbf{b}$ and compare the results with those in (b). What is your conclusion and explanation?

(f) Go back to (a) and write down the direction vectors $\mathbf{d}^1$ and $\mathbf{d}^2$. Compute the corresponding reduced costs.

(g) Between $\mathbf{d}^1$ and $\mathbf{d}^2$, which direction leads to a potential reduction in the objective value? This is your direction of translation. How far can we proceed along that direction without violating the nonnegativity constraints? This is your step length.

(h) Take the direction of translation and step length of (g) to move to a new solution. Show that the new solution is not only a basic feasible solution but also an adjacent extreme point of the previous one.

(i) Update all data in (a). Is the new solution optimal? Why?

(j) Redo (b) and (c) in terms of the new basic and nonbasic variables. Notice that the new graph is different from the previous two-dimensional graph. Why?

(k) Summarize what you have learned from (a) to (j).

(l) In general, if a given linear programming problem in its standard form has n variables and $n-2$ nonredundant constraints, you can always have a two-dimensional graphic representation for it. Why?

3.4. Complete the simplex iterations for Example 3.1.

3.5. In Case 2 of the two-phase method, prove the following statement is true: "If $\mathbf{e}_k^T\mathbf{B}^{-1}\mathbf{A}_q = 0$ for every nonbasic variable x_q, then the kth row of the constraint set $\mathbf{A}\mathbf{x} = \mathbf{b}$ must be redundant."

3.6. In Case 1 of the big-M method, verify that the inequality

$$\mathbf{c}^T\mathbf{x} = \mathbf{c}^T\mathbf{x} + M \times 0 \geq \mathbf{c}^T\mathbf{x}^* + M\sum_{i=1}^{m} x_i^{a*}$$

is impossible in that situation.

3.7. Solve the following linear programming problem by both the two-phase method and the big-M method.

$$\begin{array}{ll} \text{Minimize} & 3x_1 - 3x_2 + x_3 \\ \text{subject to} & x_1 + 2x_2 - x_3 \quad \geq 5 \\ & -3x_1 - x_2 + x_3 \leq 4 \\ & x_1, x_2, x_3 \quad \geq 0. \end{array}$$

3.8. Show that for a degenerate basic feasible solution $\mathbf{x}$ with $p(< m)$ positive components, we may have up to

$$C((n-p), (n-m)) = \frac{(n-p)!}{(n-m)!(m-p)!}$$

different bases corresponding to the same extreme point $\mathbf{x}$.

3.9. Show that $\overline{\mathbf{M}}^{-1}$ in Equation (3.20) is correct.

3.10. Following the pivoting rules of Example 3.3 to show the cycling behavior of the simplex method. Then solve the problem again with the help of the lexicographic rule or Bland's rule to eliminate the cycling problem.

3.11. Give a complete proof of the fact that Bland's rule prevents cycling as sketched in Section 3.6.2.

3.12. Solve the system of equations $\mathbf{B}\mathbf{x} = \mathbf{y}$ with

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 4 & 8 & 10 \\ 2 & 8 & 29 & 22 \\ 1 & 10 & 22 & 42 \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} 2 \\ 6 \\ 16 \\ 33 \end{bmatrix}$$

(a) Show that $\mathbf{B}$ can be factorized as the product of $\mathbf{L}$ and $\mathbf{U}$ where

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 2 & 4 & 3 & 0 \\ 1 & 5 & 0 & 4 \end{bmatrix} \quad \text{and} \quad \mathbf{U} = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 1 & 2 & 4 & 5 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

(b) Define $\mathbf{w} = \mathbf{U}\mathbf{x}$, then solve $\mathbf{L}\mathbf{w} = \mathbf{y}$ for $\mathbf{w}$ by the forward solve process.

(c) Solve $\mathbf{U}\mathbf{x} = \mathbf{w}$ for $\mathbf{x}$ by the backward solve process.

3.13. Complete the simplex iterations of Example 3.4.

3.14. Complete the simplex iterations of Example 3.5.

3.15. Draw a detailed flow chart of the revised simplex method with the two-phase method for computer implementation.

3.16. Develop computer codes based on the flow chart of the last exercise and test the following problems:

(a)

$$\begin{array}{ll} \text{Minimize} & x_1 + x_2 + x_3 - 3x_4 + 6x_5 + 4x_6 \\ \text{subject to} & x_1 + x_2 + 3x_4 - x_5 + 2x_6 = 6 \\ & x_2 + x_3 - x_4 + 4x_5 + x_6 = 3 \\ & x_1 + x_3 - 2x_4 + x_5 + 5x_6 = 5 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{array}$$

(b)

$$\begin{array}{ll} \text{Minimize} & -x_4 + 7x_5 + x_6 + 2x_7 \\ \text{subject to} & x_1 + x_4 + x_5 + x_6 + x_7 = 1 \\ & x_2 + (1/2)x_4 - (11/2)x_5 - (5/2)x_6 + 9x_7 = 0 \\ & x_3 + (1/2)x_4 - (3/2)x_5 - (1/2)x_6 + x_7 = 0 \\ & x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0 \end{array}$$