Chapter 3
Simplex Method

The simplex method is an efficient and widely used LP problem solver. Since proposed by George B. Dantzig in 1947, it has been dominating this area for more than 60 years.

The basic idea behind the simplex method is quite simple. In geometric words, it moves from a vertex to an adjacent vertex, while improving the objective value, until reaching an optimal vertex. Such doing is based on Theorem 2.3.2, guaranteeing the existence of a basic optimal solution if an optimal solution exists. It seems to be natural to hunt for an optimal solution among vertices in the feasible region, as it usually involves infinitely many point but only finitely many vertices (no more than C_n^m). So, such a strategy shrinks the hunting scope from the whole feasible region to a finite subset.

The idea may be traced back to as early as Fourier (1823). It was materialized algebraically by Dantzig (1951a). In this chapter, the simplex method will be presented in a tableau form first, then it is revised to a more applicable version. Discussed will also be related topics, such as how to get the method started, finiteness problem and finite pivot rules, and computational complexity. The last section will comment on features of the method.

3.1 Simplex Tableau

We begin with introduction of the so-called "simplex tableau" for problem (1.10). In Sect. 1.6, we already obtained the canonical form (1.11) of its constraint system, without touching the objective function at all. Now we put the objective function in the equation form

$$x_1 + 2x_2 - x_4 + x_5 - f = 0,$$

where f is called *objective variable*, at the bottom of the constraint system. The according tableau is

x_1	x_2	x_3	x_4	x_5	f	RHS
1			2/11	−4/11		7/11
	1		15/11	−19/11		14/11
		1	1/11	9/11		9/11
1	2		−1	1	−1	

Then we eliminate the nonzero entries, corresponding to pivot columns (associated with variables x_1, x_2, x_3), in its bottom (objective) row. To do so, add -1 times of the first row to the bottom row first:

x_1	x_2	x_3	x_4	x_5	f	RHS
1			2/11	−4/11		7/11
	1		15/11	−19/11		14/11
		1	1/11	9/11		9/11
	2		−13/11	15/11	−1	−7/11

then add -2 times of row 2 to the bottom row:

x_1	x_2	x_3	x_4	x_5	f	RHS
1			2/11	−4/11		7/11
	1		15/11	−19/11		14/11
		1	1/11	9/11		9/11
			−43/11	53/11	−1	−35/11

where the north-west corner is the unit matrix, corresponding to zero entries in the bottom row. Thereby, the tableau offers not only a basic solution

$$x_1 = 7/11, \quad x_2 = 14/11, \quad x_3 = 9/11, \quad x_4 = x_5 = 0, \qquad (3.1)$$

but also a reduced form of the objective function over the feasible region. Note that the solution is a basic feasible solution, associated with the objective value $35/11$, which is equal to the opposite number of the south-east corner entry.

The same tableau may be obtained otherwise by putting coefficients of the constraint system and of the objective function together to form an initial tableau, then applying the relevant Gauss-Jordan elimination.

Such a tableau is called *simplex tableau*, whose general form is as shown by Table 3.1[1].

The associated terms coincide with those the same named for the canonical form of the system $Ax = b$ (Sect. 1.6):

---

[1]It is always possible to arrange the unit matrix at the north-west corner of the simplex tableau by column exchanges. Practically, however, this is not needed, and the matrix corresponding to basic variables is usually a permutation matrix.

Table 3.1 Simplex tableau

x_{j_1}	x_{j_2}	$\cdots$	x_{j_m}	$x_{j_{m+1}}$	$\cdots$	x_{j_n}	f	RHS
1				$\bar{a}_{1\,j_{m+1}}$	$\cdots$	$\bar{a}_{1\,j_n}$		$\bar{b}_1$
	1			$\bar{a}_{2\,j_{m+1}}$	$\cdots$	$\bar{a}_{2\,j_n}$		$\bar{b}_2$
		$\ddots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
			1	$\bar{a}_{m\,j_{m+1}}$	$\cdots$	$\bar{a}_{m\,j_n}$		$\bar{b}_m$
				$\bar{z}_{j_{m+1}}$	$\cdots$	$\bar{z}_{j_n}$	-1	$-\bar{f}$

Variables (components) corresponding to the unit matrix are *basic variables (components)*, and the rest are *nonbasic variables (components)*. The basic and nonbasic index sets

$$B = \{j_1, \ldots, j_m\}, \quad \text{and} \quad N = A \backslash B.$$

are *basis* and *nonbasis*, respectively. The sets of basic and nonbasic variables are also called basis and nonbasis. The importance of the simplex tableau lies in that it gives a *basic solution* $\bar{x}_B = \bar{b}$; $\bar{x}_N = 0$. If $\bar{x}_B \geq 0$, the solution and tableaus are *basic feasible solution* and *feasible (simplex) tableau*, respectively. If the objective function attains the minimum value over the feasible region, the solution and tableau are said to be *basic optimal solution and optimal (simplex) tableau*.

In addition, $\bar{z}_N$ in the simplex tableau is termed *reduced costs (coefficients)*. The *opposite* number of the south-east corner entry gives the according objective value $\bar{f}$.

Throughout this book, it is stipulated that the bottom row of a simplex tableau always corresponds to the objective function. It will be seen that the f column does not change in solution process by the simplex method, and hence can be omitted. However, it is indispensable in the context of the "reduced simplex method", presented in Chap. 15.

3.2 Simplex Method: Tableau Form

In the previous section, a simplex tableau of the LP problem (1.10) together with the associated basic feasible solution (3.1) were obtained. But it can no not be asserted that the solution is optimal, since the reduced cost of variable x_4 is negative. As the value of x_4 increases from 0 while the value of x_5 fixed at 0, in fact, the objective function would decrease further, reaching a lower value than the current.

The new value of x_4 should be as large as possible, so that the associated objective value becomes as low as possible, subject to maintaining nonnegativity of corresponding values of x_1, x_2, x_3, satisfying

$$\begin{cases} x_1 = & 7/11 - (2/11)x_4 \geq 0, \\ x_2 = 14/11 - (15/11)x_4 \geq 0, \\ x_3 = & 9/11 - (1/11)x_4 \geq 0, \end{cases} \tag{3.2}$$

The preceding set of inequalities are equivalent to

$$\begin{cases} x_4 \le 7/2 \\ x_4 \le 14/15 \\ x_4 \le 9 \end{cases}$$

whose solution set is

$$x_4 \le \min\{7/2, 14/15, 9\} = 14/15.$$

Thereby, $\bar{x}_4 = 14/15$ is the largest possible value taken by x_4. Substituting it to (3.2) gives the new feasible solution

$$\bar{x} = (7/15, 0, 11/15, 14/15, 0)^T, \tag{3.3}$$

corresponding to objective value $\bar{f} = -7/15$ lower than $35/11$.

The according new simplex tableau is obtained by taking entry $15/11$ at row 2 and column 4 as the pivot. To this end, firstly multiply row 2 by $11/15$ to turn the pivot to 1, leading to

x_1	x_2	x_3	x_4	x_5	f	RHS
1			2/11	−4/11		7/11
	11/15		1	−19/15		14/15
		1	1/11	9/11		9/11
			−43/11	53/11	−1	−35/11

Then add $-2/11$, $-1/11$ and $43/11$ times of row 2 to rows 1, 3 and 4, respectively, giving the new simplex tableau

x_1	x_2	x_3	x_4	x_5	f	RHS
1	−2/15			−2/15		7/15
	11/15		1	−19/15		14/15
	−1/15	1		14/15		11/15
	43/15			−2/15	−1	7/15

which clearly corresponds to the new basic feasible solution (3.3).

As the reduced cost, associated with variable x_5, in the objective line is negative, still it cannot be asserted that the new solution is optimal. Similarly as in the previous step, we consider the following set of inequalities to determine the new value of x_5 that can be increased to and an associated pivot:

$$\begin{cases} x_1 = 7/15 + (2/15)x_5 \ge 0 \\ x_4 = 14/15 + (19/15)x_5 \ge 0 \\ x_3 = 11/15 - (14/15)x_5 \ge 0 \end{cases}$$

Since coefficients of x_5 in the first two inequalities are positive, the according values of x_1 and x_4 remain nonnegative as x_5 increases from 0, while x_2 is fixed on zero. It is therefore only needed to consider the third inequality, associated with the negative coefficient of x_5. Setting $x_3 = 0$ in the third equation gives $x_5 = 11/14$, leading to the basic feasible solution

$$\bar{x} = (4/7, 0, 0, 27/14, 11/14)^T, \qquad (3.4)$$

associated with objective value $\bar{f} = -4/7$ lower than $-7/15$.

   To obtain the associated simplex tableau, it is only needed to enter x_5 to and drop x_3 from the basis by taking the entry $14/15$ at row 3 and column 5 as the pivot. Multiply row 3 by $15/14$ gives

x_1	x_2	x_3	x_4	x_5	f	RHS
1	$-2/15$			$-2/15$		$7/15$
	$11/15$		1	$-19/15$		$14/15$
	$-1/14$	$15/14$		1		$11/14$
	$43/15$			$-2/15$	-1	$7/15$

Then add $2/15$, $19/15$ and $2/15$ times of row 3 to rows 1, 2 and 4, respectively, leading to

x_1	x_2	x_3	x_4	x_5	f	RHS
1	$-1/7$	$1/7$				$4/7$
	$9/14$	$19/14$	1			$27/14$
	$-1/14$	$15/14$		1		$11/14$
	$20/7$	$1/7$			-1	$4/7$

where reduced costs in the bottom row are all nonnegative. As will be proved a little later, it is now can be asserted that the corresponding basic feasible solution is optimal, which is just (3.4), and we are done.

   Now turn to the general standard LP problem (1.7). Following the preceding example, we describe an iteration by determining a pivot, and then updating the tableau by relevant elementary transformations.

   Assume at the current iteration that we are faced with the feasible Tableau 3.1, the right-hand side of which gives the basic feasible solution

$$\bar{x}_B = \bar{b} \geq 0, \qquad \bar{x}_N = 0, \qquad (3.5)$$

associated with the objective value $\bar{f}$ equal to the opposite number of the south-east corner entry of the tableau.

Lemma 3.2.1. *If reduced costs are all nonnegative, the feasible simplex tableau is optimal, giving a basic optimal solution.*

Proof. The simplex tableau results from a series of elementary transformations, and hence equivalent to the original problem. Its bottom row represents equality

$$f = \bar{f} + \bar{z}_N^T x_N. \tag{3.6}$$

Assume that $\tilde{x}$ is any feasible solution, associated with objective value $\tilde{f}$. Substituting it to (3.6) leads to

$$\tilde{f} = \bar{f} + \bar{z}_N \tilde{x}_N \geq \bar{f},$$

where the inequality is from $\bar{z}_N \geq 0$ and $\tilde{x} \geq 0$. Therefore, $\bar{x}$ is a basic optimal solution.   □

The reduced costs are often called *check numbers*, as their sign can be used to jude the optimality of a simplex tableau. Usually, there are negative check numbers in the tableau.

Lemma 3.2.2. *Assuming that $\bar{z}_q < 0$ holds for some $q \in N$, and that*

$$\bar{a}_{i,q} \leq 0, \qquad i = 1, \dots, m, \tag{3.7}$$

then the LP problem is (lower) unbounded.

Proof. The simplex tableau is associated with the constraint system

$$x_{j_i} = \bar{b}_i - \sum_{j \in N} \bar{a}_{i\,j} x_j, \qquad i = 1, \dots, m.$$

Setting $x_j = 0$, $j \in N$, $j \neq q$ in the preceding and combining the result with the nonnegative constrains gives

$$x_{j_i} = \bar{b}_i - \bar{a}_{i\,q} x_q \geq 0, \qquad i = 1, \dots, m. \tag{3.8}$$

It is known from (3.7) that the set (3.8) of inequalities hold for all $x_q = \alpha \geq 0$, associated with feasible value

$$\hat{f} = \bar{f} + \alpha \bar{z}_q, \tag{3.9}$$

which, since $\bar{z}_q < 0$, can be arbitrarily low as α increases. Therefore, the problem is lower unbounded.   □

If (3.7) does not hold, then the value that the nonbasic variable x_q takes on will be restricted by the set (3.8) of inequalities. It is not difficulty to verify that the following rule gives the largest possible value α of x_q subject to (3.8).

Rule 3.2.1 (Row pivot rule) Determine a row index p and stepsize α such that

$$\alpha = \bar{b}_p / \bar{a}_{pq} = \min\{\bar{b}_i / \bar{a}_{i\,q} \mid \bar{a}_{i\,q} > 0, \ i = 1, \dots, m\} \geq 0. \tag{3.10}$$

which is often called *minimum-ratio test*.

Setting $x_q = \alpha$ in the equation part of (3.8) gives a new basic feasible solution, i.e.,

$$\hat{x}_{j_i} = \bar{b}_i - \alpha \bar{a}_{i\,q}, \quad i = 1, \ldots, m; \qquad \hat{x}_j = 0, \quad j \in N, \ j \neq q; \qquad \hat{x}_q = \alpha. \tag{3.11}$$

Taking $\bar{a}_{p\,q}$ as the pivot, the according simplex tableau is obtained by multiplying row p by $1/\bar{a}_{p\,q}$ to convert the pivot to 1, adding $-\bar{a}_{i\,q}$ times of row p to rows $i = 1, \ldots, m, \ i \neq p$, and adding $-\bar{z}_q$ times of row p to the objective row. Finally, (B, N) is updated by exchanging j_p and q, and an iteration is complete.

It is seen from (3.9) that the associated objective value decreases strictly if $\alpha > 0$; no real decrement is made if $\alpha = 0$.

Definition 3.2.1. If some components of $\bar{b}$ is equal to zero, then the associated basic feasible solution (or tableau) is degenerate.

A LP problems is said to be nondegenerate if all basic solutions are nondegenerate.

In degeneracy case, the stepsize α defined by (3.10) could vanish, and hence the objective function remains unchanged (see (3.9)). That is to say, the associated "new solution" (3.11) is actually the same as the old although the basis is changed.

In general, there are multiple choices for q, as any q with negative $\bar{z}_q$ is eligible to be chosen. Dantzig's original *minimum reduced cost rule* is as follows.

Rule 3.2.2 (Column pivot rule) Select a column index q such that

$$q \in \arg \min_{j \in N} \bar{z}_j. \tag{3.12}$$

Thus, this rule selects the column with the most negative reduced cost as the pivot column.[2] For unit increment of the nonbasic variable x_q, this choice leads to the largest amount of decrease in the objective value.

The overall steps are summarized to the following algorithm (Dantzig 1947).

Algorithm 3.2.1 (Simplex algorithm: tableau form). Initial: a feasible simplex tableau of the form Table 3.1. This algorithm solves the standard LP problem (1.7).

1. Determine a pivot column index $q \in \arg \min_{j \in N} \bar{z}_j$.
2. Stop if $\bar{z}_q \geq 0$.
3. Stop if $I = \{i = 1, \ldots, m \mid \bar{a}_{i\,q} > 0\} = \emptyset$.
4. Determine a pivot row index $p \in \arg \min_{i \in I} \bar{b}_i / \bar{a}_{iq}$.
5. Convert $\bar{a}_{p\,q}$ to 1, and eliminate the other nonzeros in the column by elementary transformations.
6. Go to Step 1.

---

[2]Any choice is eligible if there is a tie when the number of the most negative reduced costs is more than one. Similarly below.

Theorem 3.2.1. *Under the nondegeneracy assumption, the simplex algorithm terminates either at*

(i) *Step 2, generating a basic optimal solution; or at*
(ii) *Step 3, detecting lower unboundedness of the problem.*

Proof. Note that there are infinitely many basic feasible solutions. If is clear that Algorithm 3.2.1 generates a sequence of basic feasible solutions, while the associated objective value decreases, due to the nonnegative stepssize α. Under the nondegeneracy assumption, the stepsize α is positive, and hence the objective value decreases strictly in each iteration. In the solution process, therefore, any basic solution can only appear once at most. So, infiniteness of the solution process implies that there are infinitely many basic feasible solutions, as is a contradiction. Therefore, Algorithm 3.2.1 terminates.

The meanings of the exits of the Algorithm 3.2.1 comes from Lemmas 3.2.1 and 3.2.2.   □

It should be aware that the nondegeneracy assumption is beyond reality at all. As practical problems are almost always degenerate, termination of the simplex Algorithm 3.2.1 is actually not guaranteed. In other words, the possibility is not ruled out that indices enter and leave the basis infinitely many times. In fact, few instances that cannot be solved by the simplex algorithm had been constructed (we will handle this topic in Sects. 3.6 and 3.7). Even so, the possibility for not terminating is very rare, so as dose not matter to broad applications of the simplex algorithm.

A simplex tableau is nothing but a concise expression of a standard LP problem. As they represent problems equivalent to the original problem itself, all the tableaus created by the simplex algorithm are viewed as equivalent. Recursive formulas between a simplex tableau and its predecessor are listed below:

1. The objective row

$$
\begin{aligned}
\beta &= -\bar{z}_q/\bar{a}_{pq}, \\
\hat{f} &= \bar{f} - \beta\bar{b}_p, \\
\hat{z}_j &= \bar{z}_j + \beta\bar{a}_{pj}, \qquad j \in N, \\
\hat{z}_{j_i} &= \begin{cases} \beta & i = p, \\ 0 & i = 1,\ldots,m,\ i \neq p. \end{cases}
\end{aligned}
\tag{3.13}
$$

2. The right-hand side

$$
\begin{aligned}
\alpha &= \bar{b}_p/\bar{a}_{pq}, \\
\hat{b}_i &= \begin{cases} \bar{b}_i - \alpha\bar{a}_{iq} & i = 1,\ldots,m,\ i \neq p, \\ \alpha & i = p. \end{cases}
\end{aligned}
\tag{3.14}
$$

3. Entries of the constraint matrix

$$\hat{a}_{t,j} = \begin{cases} 0, & t = 1,\ldots,m, \ t \neq p; \ j = q. \\ 1, & t = p; \ j = q, \\ \bar{a}_{t\,j} - (\bar{a}_{p\,j}/\bar{a}_{p\,q})\bar{a}_{t\,q}, & t = 1,\ldots,m, \ t \neq p; j \in N, \ j \neq q, \\ \bar{a}_{p\,j}/\bar{a}_{p\,q}, & t = p; \ j \in N, \ j \neq q. \end{cases}$$

(3.15)

$$\hat{a}_{t\,j_i} = \begin{cases} 0, & t = 1,\ldots,m; \ i = 1,\ldots,m, \ i \neq p; \ i \neq t, \\ 1, & t = i = 1,\ldots,m; \ i \neq p, \\ -\bar{a}_{t\,q}/\bar{a}_{p\,q}, & t = 1,\ldots,m, \ t \neq p; \ i = p, \\ 1/\bar{a}_{p\,q}, & t = i = p. \end{cases}$$

Example 3.2.1. Solve the following problem by Algorithm 3.2.1:

$$\begin{aligned} \min \quad & f = -4x_1 - 3x_2 - 5x_3, \\ \text{s.t.} \quad & 2x_1 + x_2 + 3x_3 \quad\quad + x_5 \quad\quad\quad\quad = 15, \\ & x_1 + x_2 + x_3 + x_4 \quad\quad\quad\quad\quad = 12, \\ & -2x_1 + x_2 - 3x_3 \quad\quad\quad\quad + x_7 = \ \ 3, \\ & 2x_1 + x_2 \quad\quad\quad\quad\quad +x_6 \quad\quad = \ \ 9, \\ & \quad\quad\quad\quad\quad x_j \geq 0, \ j = 1,\ldots,7. \end{aligned}$$

Answer   Initial: the following feasible simplex tableau can be directly obtained from the problem:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
2	1	3*		1			15
1	1	1	1				12
−2	1	−3				1	3
2	1				1		9
−4	−3	−5					

Iteration 1:

1. $\min\{-4, -3, -5\} = -5 < 0, \ q = 3$.
3. $I = \{1, 2\} \neq \emptyset$.
4. $\min\{15/3, \ 12/1\} = 15/3, \ p = 1$.
5. Take 3 in row 1 and column 3 as the pivot (marked by "*", the same below).

Multiply row 1 by $1/3$, then add $-1, 3, 5$ times of row 1 to rows 2,3,5, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
2/3	1/3	1		1/3			5
1/3	2/3		1	$-1/3$			7
		2*		1		1	18
2	1				1		9
$-2/3$	$-4/3$			5/3			25

Iteration 2:

1. $\min\{-2/3, -4/3, 5/3\} = -4/3 < 0$, $q = 2$.
3. $I = \{1, 2, 3, 4\} \neq \emptyset$.
4. $\min\{5/(1/3), 7/(2/3), 18/2, 9/1\} = 9/1$, $p = 3$.
5. Multiply row 3 by $1/2$, then add $-1/3, -2/3, -1, 4/3$ times of row 3 to rows 1,2,4,5, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
2/3		1		1/6		$-1/6$	2
1/3			1	$-2/3$		$-1/3$	1
	1			1/2		1/2	9
2*				$-1/2$	1	$-1/2$	0
$-2/3$				7/3		2/3	37

Iteration 3:

1. $\min\{-2/3, 7/3, 2/3\} = -2/3 < 0$, $q = 1$.
3. $I = \{1, 2, 4\} \neq \emptyset$.
4. $\min\{2/(2/3), 1/(1/3), 0/2\} = 0$, $p = 4$.
5. Multiply row 4 by $1/2$, then add $-2/3, -1/3, 2/3$ times of row 4 to rows 1,2,5, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
		1		1/3	$-1/3$		2
			1	$-7/12$	$-1/6$	$-1/4$	1
	1			1/2		1/2	9
1				$-1/4$	1/2	$-1/4$	
				13/6	1/3	1/2	37

Now all reduced costs in the preceding tableau are nonnegative, and hence the basic optimal solution and associated objective value are, respectively,

$$\bar{x} = (0, 9, 2, 1, 0, 0, 0)^T, \qquad \bar{f} = -37.$$

It is seen from the preceding example that the tableau in the second iteration already attained the basic optimal solution, but an additional iteration was performed, due to the existence of a negative reduced cost. This occurred because of degeneracy leading to a zero stepsize. So, the condition that reduced costs are all nonnegative is sufficient but not necessary for optimality.

3.3   Start-Up of the Simplex Method

Algorithm 3.2.1 must start from a feasible simplex tableau. In Example 3.2.1, there is a feasible simplex tableau available, as is not the case in general. A so-called *Phase-I* procedure is usually carried out to provide an initial feasible simplex tableau (if any), then Algorithm 3.2.1 is used to achieve optimality or detect unboundedness of the problem. Thus, the simplex algorithm described in the previous section is actually a "Phase-II" procedure. A standard LP problem is usually solved by the two procedures in succession, referred to as *two-phase simplex method*. In this section, a classical Phase-I procedure using artificial variables will be presented first; described is then a closely related start-up method, the so-called "big M".

Assume that all components of the right-hand side are nonnegative, i.e.,

$$b_i \geq 0, \ i = 1, \ldots, m.$$

If not so, multiply each constraint equation with negative right-hand side by -1 before hand. Then construct an auxiliary problem as follows.

For each $i = 1, \ldots, m$, introduce a nonnegative *artificial variable* x_{n+i} to the ith equation, and take the sum of all artificial variables as the auxiliary objective function, i.e.,

$$f' = \sum_{i=1}^{m} x_{n+i}.$$

Using the constraint system, we eliminate all artificial variables from the auxiliary objective, resulting in

$$
\begin{aligned}
\min \quad & f' = \sum_{i=1}^{m} b_i - \left(\sum_{i=1}^{m} a_{i1}\right)x_1 - \cdots - \left(\sum_{i=1}^{m} a_{in}\right)x_n, \\
\text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + x_{n+1} && = b_1, \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \qquad + x_{n+2} && = b_2, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \qquad\qquad\quad + x_{n+m} = b_m, \\
& \qquad\qquad\qquad\qquad x_j \geq 0, \quad j = 1, \ldots, n+m.
\end{aligned}
\tag{3.16}
$$

Clearly, there is an available feasible simplex tableau to the preceding auxiliary program, corresponding to the basic feasible solution

$$x' = (\underbrace{0, \ldots, 0}_{n}, \underbrace{b_1, \ldots, b_m}_{m})^T \geq 0.$$

Thereby, the program can be solved by Algorithm 3.2.1.

Regarding the outcome, we have the following.

Theorem 3.3.1. *The auxiliary program has an optimal solution, associated with a nonnegative optimal value:*

(i) *If the optimal value is strictly greater than zero, the original problem is infeasible;*

(ii) *If the optimal value is equal to zero, the first n components of the optimal solution to the auxiliary program form a feasible solution to the original problem.*

Proof. Clearly, there exists a feasible solution to problem (3.16). Since artificial components of all feasible solutions are nonnegative, all feasible values of the auxiliary program are nonnegative too. Therefore, there exists an optimal solution, associated with a nonnegative objective value:

(i) If the optimal value is strictly greater than zero, it can be asserted that the original problem is infeasible, because if it had a feasible solution $\bar{x}_j \geq 0$, $j = 1, \ldots, n$, then

$$\bar{x}_1, \ldots, \bar{x}_n, \bar{x}_{n+1} = \cdots = \bar{x}_{n+m} = 0$$

clearly satisfied constraints of (3.16), and hence was a feasible solution to (3.16), corresponding to auxiliary objective value zero, as is a contradiction.

(ii) If the optimal value is zero, then artificial components of the optimal solution are 0. From substituting it to the constraints of (3.16), it is therefore seen that its first n components just satisfy the constraints of the original problem, and hence constitute a feasible solution to the latter.   □

Corollary 3.3.1. *The original problem is feasible if and only if the optimal value of the auxiliary program vanishes.*

Once a feasible solution to the original problem is obtained by the preceding approach, a feasible simplex tableau can be yielded from the optimal auxiliary tableau by the "following-up steps" below. These steps come from the fact that setting all the artificial variables to zero in the system, corresponding to the auxiliary optimal tableau, leads to a system equivalent to the original one.

Following-up steps:

(A) Delete columns, associated to all nonbasic artificial variables (which can be deleted once the corresponding artificial variable leaves the basis).
(B) Go to step D if there is no basic artificial variable.
(C) Delete the row, associated to a basic artificial variable, if all its nonbasic entries are zero (see the Note below); otherwise, take a nonzero entry of it as pivot to let the artificial variable become nonbasic, and then delete the associated column. This is repeated until no artificial variable is basic.
(D) Cover the auxiliary objective row by the original costs, and then eliminate all basic entries of this row, giving a feasible simplex tableau to the original problem.

Note: In step C, the row is deleted because substituting 0 to the associated artificial variable turns the corresponding equation to an identity, as reflects dependence of the original constraint equations. So, the method can get rid of such dependency.

The preceding can be put into the following algorithm.

Algorithm 3.3.1 (Phase-1: artificial variable). This algorithm finds a feasible tableau.

1. Introduce artificial variables, and construct auxiliary program of form (3.16).
2. Call the simplex Algorithm 3.2.1.
3. If the optimal value of the auxiliary program is zero, create a feasible tableau via "Following-up steps".
4. The original problem is infeasible if the optimal value of the auxiliary program is strictly greater than zero.

Note that if a constraint matrix includes some columns of the unit matrix, such columns should be employed to reduce the number of artificial variables. The preceding discussions are still valid, though the auxiliary objective function involves artificial variables only.

Example 3.3.1. Find a feasible simplex tableau to the following problem:

$$\begin{aligned}
\min \quad f &= -x_1 + x_2 - 2x_3, \\
\text{s.t.} \quad x_1 - 3x_2 - 2x_3 + x_4 \qquad\qquad &= -4, \\
x_1 - x_2 + 4x_3 \qquad - x_5 \quad\;\; &= 2, \\
-3x_1 + x_2 + x_3 \qquad\qquad + x_6 &= 8, \\
x_j \geq 0, \, j = 1, \ldots, 6.
\end{aligned}$$

Answer   Construct auxiliary program: the first constraint equation is multiplied by -1 to turn its right-hand side to nonnegative; as the coefficients of x_6 give a unit vector $(0, 0, 1)^T$, only two artificial variables x_7, x_8 are introduced.

$$\min \quad f' = x_7 + x_8,$$
$$\text{s.t.} \quad -x_1 + 3x_2 + 2x_3 - x_4 \qquad\qquad\qquad + x_7 \qquad\quad = 4,$$
$$x_1 - x_2 + 4x_3 \qquad - x_5 \qquad\qquad + x_8 = 2,$$
$$-3x_1 + x_2 + x_3 \qquad\qquad + x_6 \qquad\qquad = 8,$$
$$x_j \geq 0, \ j = 1, \dots, 8.$$

Put the preceding auxiliary program into the following tableau:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	RHS
-1	3	2	-1			1		4
1	-1	4		-1			1	2
-3	1	1			1			8
						1	1	

Turn the preceding to a simplex tableau: eliminate nonzeros in x_7 and x_8 columns at the bottom (objective) row by adding -1 times of row 1 and of row 2 to that row:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	RHS
-1	3	2	-1			1		4
1	-1	4*		-1			1	2
-3	1	1			1			8
	-2	-6	1	1				-6

which is a feasible simplex tableau to the auxiliary program. Call Algorithm 3.2.1 to solve it:

Iteration 1:

1. $\min\{0, -2, -6, 1, 1\} = -6 < 0, \ q = 3.$
3. $I = \{1, 2, 3\} \neq \emptyset.$
4. $\min\{4/2, 2/4, 8/1\} = 1/2, \ p = 2.$
5. Multiply row 2 by $1/4$, and then add $-2, -1, 6$ times of row 2 to rows 1,3,4, respectively

(Erase x_8 column after artificial variable x_8 becomes nonbasic):

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
$-3/2$	$7/2$*		-1	$1/2$		1	3
$1/4$	$-1/4$	1		$-1/4$			$1/2$
$-13/4$	$5/4$			$1/4$	1		$15/2$
$3/2$	$-7/2$		1	$-1/2$			-3

Iteration 2:

1. $\min\{3/2, -7/2, 1, -1/2\} = -7/2 < 0, \ q = 2.$
3. $I = \{1, 3\} \neq \emptyset.$
4. $\min\{3/(7/2), (15/2)/(5/4)\} = 6/7, \ p = 1.$
5. Multiply row 1 by $2/7$, and then add $1/4, -5/4, 7/2$ times of row 1 to rows 2,3,4, respectively

(Erase x_7 column after artificial variable x_7 becomes nonbasic):
    Now, all the artificial variables become nonbasic, hence reached is the optimal objective value 0 of the auxiliary program.
    Covering the bottom row by original costs leads to

x_1	x_2	x_3	x_4	x_5	x_6	RHS
$-3/7$	1		$-2/7$	$1/7$		$6/7$
$1/7$		1	$-1/14$	$-3/14$		$5/7$
$-19/7$			$5/14$	$1/14$	1	$45/7$
-1	1	-2				

Adding -1 times of row 1 and 2 times of row 2 to the bottom row gives a feasible tableau of the original problem, i.e.,

x_1	x_2	x_3	x_4	x_5	x_6	RHS
$-3/7$	1		$-2/7$	$1/7$		$6/7$
$1/7$		1	$-1/14$	$-3/14$		$5/7$
$-19/7$			$5/14$	$1/14$	1	$45/7$
$-2/7$			$1/7$	$-4/7$		$4/7$

Thus, the preceding can be taken as an initial feasible tableau to get Algorithm 3.2.1 started to solve the original problem. Solving LP problems usually requires two phases, both of which are carried out using the simplex algorithm.
    On the other hand, it seems to be attractive to solve LP problems in a single phase, as leads to the following so-called *big-M method*.
    The according auxiliary program shares the same constraints as before with (3.16), while its objective function is the sum of the original objective function and M times of the sum of all the artificial variables, i.e.,

$$
\begin{aligned}
\min \quad & f' = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + M(x_{n+1} + x_{n+2} \cdots + x_{n+m}), \\
\text{s.t.} \quad & a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n + x_{n+1} && = b_1, \\
& a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n + x_{n+2} && = b_2, \\
& \qquad\qquad\qquad\qquad\qquad \vdots \\
& a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n + x_{n+m} && = b_m, \\
& x_j \geq 0, \quad j = 1, \ldots, n + m.
\end{aligned}
$$

Artificial variables in the objective function are eliminated using the constraint system. As a result, there will be a feasible simplex tableau to the auxiliary program, which can be taken as an initial one to get the simplex algorithm started.

The reason for using such an auxiliary objective function is as follows. Its artificial variable part may be regarded as a "penalty function", where M serves as a "penalty factor", is a sufficiently large positive number (far larger than the absolute value of any number involved in the computations). The big M inflicts penalty on possible increase of values of artificial variables, consequently forcing them minimized prior to the original objective.

It is difficult however to determine a suitable M in advance. Too large M could lead to bad numerical stability, while too small M degrades method's effect. It depends not only on the problem to be solved, but also the computer used. A practicable way is to take M as a parameter in the solution process.

To demonstrate, we again bring up Example 3.3.1. Its auxiliary program is of the following form:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	RHS
-1	3	2	-1			1		4
1	-1	4		-1			1	2
-3	1	1			1			8
-1	1	-2				M	M	

Add M times of row 1 and of row 2 to the objective row, giving

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	RHS
-1	3	2	-1			1		4
1	-1	4*		-1			1	2
-3	1	1			1			8
-1	$1-2M$	$-2-6M$	M	M				$-6M$

Thereby, we can get the simplex algorithm stated from the preceding tableau. In selection of a pivot column index q, however, it should be noted that M is so large that the sign of reduced costs depends upon coefficients of M only. In the preceding tableau, e.g., x_3 column is selected as the pivot column, as term M's coefficients are $0, -2, -6, 1, 1$ for costs of nonbasic variables x_1 through x_5 respectively, and

$$\min\{0, -2, -6, 1, 1\} = -6, \qquad q = 3.$$

Row 2 is selected as the pivot row by the minimum-ratio test below:

$$\min\{4/2, 2/4, 8/1\} = 1//2, \qquad p = 2.$$

Then elementary transformations are performed to make a corresponding basis change, completing the first iteration.

If the process continued, it can be found that sequences of iterates created by Big M method and the two-phase simplex method are actually the same. This is not surprising because, as mentioned previously, the big "penalty factor" M forces values of artificial variables vanishing first before pursuing optimality of the original problem–the two methods are essentially the same. Practically, however, the two-phase method is certainly preferable to the big M method, as it involves no any parameter, and easier to realize.

Nevertheless, the auxiliary programs, presented previously in this section, are usually found in textbooks only. If the number m of rows is large, the scale of the programs would become unacceptable large. A somehow practicable approach is to use an auxiliary program with a single artificial variable as follows.

Introducing artificial variable x_{n+1}, we consider the following auxiliary program instead:

$$
\begin{aligned}
\min \quad & f' = x_{n+1}, \\
\text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_1 x_{n+1} = b_1, \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + b_2 x_{n+1} = b_2, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + b_m x_{n+1} = b_m, \\
& \qquad\qquad\qquad\quad x_j \geq 0, \quad j = 1, \ldots, n+1,
\end{aligned} \tag{3.17}
$$

to which there is a feasible solution

$$
x' = (\underbrace{0, \ldots, 0}_{n}, 1)^T \geq 0.
$$

Results similar to Theorem 3.3.1 and Corollary 3.3.1 hold to the preceding auxiliary program. On the other hand, using the following auxiliary objective function leads to an analogue to the big M method:

$$
f' = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + M x_{n+1}.
$$

A drawback of such auxiliary programs seems to be lack of a explicit feasible simplex tableau. This will be seen not essential, however. In Sect. 13.2, we will present other Phase-I methods as well as a more practicable single artificial variable approach.

Now it is known that the answer to a LP problem must be one of the following three cases:

 (i) Infeasible problem: there exists no feasible solution;
 (ii) Unbounded problem: there exists a feasible solution but the feasible value is lower unbounded over the feasible region;
(iii) There exists an optimal basic solution.

In principle, a two-phase simplex method can be used to solve any LP problem, achieving an basic optimal solution, if any, or detecting infeasibility or unboundedness, otherwise.

3.4   Revised Simplex Tableau

Simplex tableau is not an unique tool to implement the simplex method. In fact, getting rig of the tableau can lead to a more compact variant of the simplex method. For this purpose, we will employ vectors or matrices more from now on.

The standard LP problem (1.8) may be represented by the following tableau:

x^T	f	RHS
A		b
c^T	-1	

Assume that through some elementary transformations, the preceding table becomes the simplex tableau Table 3.1, which may be succinctly put into

x_B^T	x_N^T	f	RHS
I	$\bar{N}$		$\bar{b}$
	$\bar{z}_N^T$	-1	$-\bar{f}$

$$(3.18)$$

Unless specified otherwise, thereafter the associated basic and nonbasis index sets are assumed to be

$$B = \{j_1, \cdots, j_m\}, \qquad N = A \backslash B = \{j_{m+1}, \cdots, j_n\}. \qquad (3.19)$$

Columns corresponding to B are said to be *basic*, and those to N *nonbasic*. Without confusion, B and N will also be used to respectively denote submatrices consisting of corresponding columns. The two submatrices are respectively called *basis matrix* and *nonbasis matrix*, or basis and nonbasis for short. It is clear that B is an invertible square matrix. The simplex tableau corresponds to the basic solution

$$\bar{x}_B = \bar{b}, \qquad \bar{x}_N = 0.$$

If $\bar{b} \geq 0$, $\bar{z}_N \geq 0$, then the tableau is an optimal (simplex) tableau, giving an basic optimal solution, and the according B and N are *optimal basis* and *optimal nonbasis*, respectively.

On the other hand, if $Ax = b$ is premultiplied by B^{-1}, and some transposition of terms is made, it follows that

$$x_B + B^{-1} N x_N = B^{-1} b.$$

Table 3.2 Equivalence between the associated quantities

Quantity	Simplex tableau	Relation	Revised simplex tableau
Objective row	$\bar{z}_N^T$	$=$	$c_N^T - c_B^T B^{-1} N$
Pivot column	$\bar{a}_q$	$=$	$B^{-1} a_q$
Right-hand side	$\bar{b}$	$=$	$B^{-1} b$
Pivot row	$e_p^T \bar{N}$	$=$	$e_p^T B^{-1} N$

Substituting the preceding to

$$c_B^T x_B + c_N^T x_N - f = 0$$

gives

$$(c_N^T - c_B^T B^{-1} N) x_N - f = -c_B^T B^{-1} b,$$

which can put in

x_B^T	x_N^T	f	RHS
I	$B^{-1} N$		$B^{-1} b$
	$c_N^T - c_B^T B^{-1} N$	-1	$-c_B^T B^{-1} b$

(3.20)

corresponding to basic solution $\bar{x}_B = B^{-1} b$, $\bar{x}_N = 0$ (hereafter $\bar{x}_B = B^{-1} b$ is often said basic solution for short). The preceding, representing a problem equivalent to the original, is called *revised simplex tableau*, compared to the simplex tableau (3.18).

For simplicity, x_B^T and f columns in the preceding two tableaus may be omitted, as they remain unchanged as basis changes.

Proposition 3.4.1. *Any simplex tableau and revised simplex tableau, corresponding to the same basis, are equivalent.*

Proof. Denote by (3.18) and (3.20) the two tableaus, having the same basis B. Since problems represented by them are equivalent, the corresponding entries of the two tableaus are equal. □

Based on the preceding Proposition, Table 3.2 gives equivalence correspondence between quantities, involved in simplex steps, of tableaus (3.18) and (3.20):

In conventional simplex context, each iteration corresponds to a basis B (or its inverse B^{-1}), with which any entry in a simplex tableau can be calculated from the original data (A, b, c). Thereby, Table 3.2 will be used as a tool to derive common simplex variants, such as the (revised) simplex algorithm in the next section and the dual (revised) simplex algorithm in Sect. 4.5.

Notations in this section will be employed throughout this book.

3.5 Simplex Method

A simplex tableau has to be calculated in each iteration by the tableau simplex Algorithm 3.2.1. But its $(m+1) \times (n+1)$ entries are not all useful in an iteration. In fact, only the objective row is needed for the selection of a pivot column, while the pivot column and right-hand side needed for the determination of a pivot row. Using B^{-1}, therefore, a variant without any simplex tableau can be derived by calculating the first three items in Table 3.2.

Let us consider updating B^{-1}. Assume that pivot column index q and row index p are already determined. Putting the nonbasic column a_q in place of B's pth column a_{j_p} gives the new basis below:

$$\hat{B} = (a_{j_1}, \ldots, a_{j_{p-1}}, a_q, a_{j_{p+1}}, \ldots, a_{j_m}). \tag{3.21}$$

It is now needed to compute $\hat{B}^{-1}$ to go on the next iteration.

Note that $\bar{a}_q = B^{-1}a_q$. Taking $\bar{a}_{pq}$ as the pivot, the according elementary transformations amount to premultiplying the first m rows of the tableau by $m \times m$ elementary matrix

$$E_p = \begin{pmatrix} 1 & & -\bar{a}_{1q}/\bar{a}_{pq} & & \\ & \ddots & \vdots & & \\ & & -\bar{a}_{p-1,q}/\bar{a}_{pq} & & \\ & & 1/\bar{a}_{pq} & & \\ & & -\bar{a}_{p+1,q}/\bar{a}_{pq} & & \\ & & \vdots & \ddots & \\ & & -\bar{a}_{mq}/\bar{a}_{pq} & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ p \\ \\ \\ \\ \end{matrix} \tag{3.22}$$

$$p$$

which may also be obtained by executing the same elementary transformations on the unit matrix. It is seen that such a matrix, which is the same as the unit matrix except for the pth column, is determined only by $\bar{a}_q$. Combining (3.21) and (3.22) gives

$$E_p B^{-1} \hat{B} = E_p(B^{-1}a_{j_1}, \ldots, B^{-1}a_{j_{p-1}}, B^{-1}a_q, B^{-1}a_{j_{p+1}}, \ldots, B^{-1}a_{j_m})$$

$$= E_p(e_1, \ldots, e_{p-1}, \bar{a}_q, e_{p+1}, \ldots, e_m)$$

$$= (E_p e_1, \ldots, E_p e_{p-1}, E_p \bar{a}_q, E_p e_{p+1}, \ldots, E_p e_m) = I,$$

from which the update of the basis' inverse follows, i.e.,

$$\hat{B}^{-1} = E_p B^{-1}. \tag{3.23}$$

Based on the preceding discussions and the equivalence between the simplex tableau and revised simplex tableau, we are able to revise Algorithm 3.2.1 to the following version (Dantzig and Orchard-Hays 1953):

Algorithm 3.5.1 (Simplex algorithm 1). Initial: (B, N), B^{-1}, $\bar{x}_B = B^{-1}b \geq 0$ and $\bar{f} = c_B^T \bar{x}_B$. This algorithm solves the standard LP problem (1.8).

1. Compute $\bar{z}_N = c_N - N^T \bar{y}$, $\bar{y} = B^{-T} c_B$.
2. Determine pivot column index $q \in \arg\min_{j \in N} \bar{z}_j$.
3. Stop if $\bar{z}_q \geq 0$ (optimality achieved).
4. Compute $\bar{a}_q = B^{-1} a_q$.
5. Stop if $\bar{a}_q \leq 0$ (unbounded problem).
6. Determine stepsize α and pivot row index p such that
   $\alpha = \bar{x}_{j_p} / \bar{a}_{pq} = \min\{\bar{x}_{j_i} / \bar{a}_{iq} \mid \bar{a}_{iq} > 0; \ i = 1, \ldots, m\}$.
7. Set $\bar{x}_q = \alpha$, and update $\bar{x}_B = \bar{x}_B - \alpha \bar{a}_q$, $\bar{f} = \bar{f} + \alpha \bar{z}_q$ if $\alpha \neq 0$.
8. Update B^{-1} by (3.23).
9. Update (B, N) by exchanging j_p and q.
10. Go to step 1.

The preceding, usually called *revised simplex algorithm*, will be referred to as *simplex algorithm 1*.

In step 1, vector $\bar{y}$ is calculated first, then it is used to compute reduced costs $\bar{z}_N$, as is referred to as *pricing*. $\bar{y}$ is called *simplex multipliers (vector)*, whose additional meanings will be clear later.

See Sect. 3.3 for how to provide an initial basic feasible solution (or basis). This topic will be handled further in Chap. 13.

Example 3.5.1. Solve the following problem by Algorithm 3.5.1:

$$\begin{aligned}
\min \quad & f = -4x_1 - 3x_2 - 5x_3, \\
\text{s.t.} \quad & 2x_1 + x_2 + 3x_3 \quad\quad\quad + x_5 \quad\quad\quad\quad\quad = 15, \\
& x_1 + x_2 + x_3 + x_4 \quad\quad\quad\quad\quad\quad = 12, \\
& x_2 - 3x_3 \quad\quad\quad\quad\quad\quad + x_7 = 3, \\
& 2x_1 + x_2 \quad\quad\quad\quad\quad + x_6 \quad\quad = 9, \\
& x_j \geq 0, \quad j = 1, \ldots, 7.
\end{aligned}$$

Answer   Initial: $B = \{5, 4, 7, 6\}$, $N = \{1, 2, 3\}$, $B^{-1} = I$, $\bar{x}_B = (15, 12, 3, 9)^T$, $f = 0$.

Iteration 1:

1. $\bar{y} = B^{-T} c_B = (0, 0, 0, 0)^T$, $\bar{z}_N = c_N - N^T \bar{y} = (-4, -3, -5)^T$.
2. $\min\{-4, -3, -5\} = -5 < 0$, $q = 3$.

4. $\bar{a}_3 = B^{-1}a_3 = (3, 1, -3, 0)^T$.

6. $\alpha = \min\{15/3, 12/1\} = 15/3 = 5, \ p = 1$.

7. $\bar{x}_B = (15, 12, 3, 9)^T - 5 \times (3, 1, -3, 0)^T = (0, 7, 18, 9)^T, \ x_3 = 5,$
   $f = 5 \times (-5) = -25$.

8. $B^{-1} = \begin{pmatrix} 1/3 & & & \\ -1/3 & 1 & & \\ 1 & & 1 & \\ 0 & & & 1 \end{pmatrix}$.

9. $B = \{3, 4, 7, 6\}, \ N = \{1, 2, 5\}, \ \bar{x}_B = (5, 7, 18, 9)^T$.

Iteration 2:

1. $\bar{y} = (-5/3, 0, 0, 0)^T, \ \bar{z}_N = (-4, -3, 0)^T - (-10/3, -5/3, -5/3)^T$
   $= (-2/3, -4/3, 5/3)^T$.

2. $\min\{-2/3, -4/3, 5/3\} = -4/3 < 0, \ q = 2$.

4. $\bar{a}_2 = (1/3, 2/3, 2, 1)^T$.

6. $\alpha = \min\{15, 21/2, 9, 9\} = 9, \ p = 3$.

7. $\bar{x}_B = (5, 7, 18, 9)^T - 9 \times (1/3, 2/3, 2, 1)^T = (2, 1, 0, 0)^T, \ x_2 = 9,$
   $f = -25 + 9 \times (-4/3) = -37$.

8. $B^{-1} = \begin{pmatrix} 1 & & -1/6 & \\ 0 & 1 & -1/3 & \\ 0 & & 1/2 & \\ 0 & & -1/2 & 1 \end{pmatrix} \begin{pmatrix} 1/3 & & & \\ -1/3 & 1 & & \\ 1 & & 1 & \\ 0 & & & 1 \end{pmatrix} = \begin{pmatrix} 1/6 & & -1/6 & \\ -2/3 & 1 & -1/3 & \\ 1/2 & & 1/2 & \\ -1/2 & & -1/2 & 1 \end{pmatrix}$.

9. $B = \{3, 4, 2, 6\}, \ N = \{1, 7, 5\}, \bar{x}_B = (2, 1, 9, 0)^T$.

Iteration 3:

1. $\bar{y} = (-7/3, 0, -2/3, 0)^T, \ \bar{z}_N = (-4, 0, 0)^T - (-10/3, -2/3, -7/3)^T$
   $= (-2/3, 2/3, 7/3)^T$.

2. $\min\{-2/3, 2/3, 7/3\} = -2/3, \ q = 1$.

4. $\bar{a}_1 = (2/3, 1/3, 0, 2)^T$.

6. $\alpha = \min\{2/(2/3), 1/(1/3), 0/2\} = 0, \ p = 4$.

7. $\bar{x}_B = (2, 1, 9, 0)^T, \ x_6 = 0, \ f = -37$.

8. $B^{-1} = \begin{pmatrix} 1 & & -1/3 & \\ 0 & 1 & -1/6 & \\ 0 & & 1 & 0 \\ 0 & & & 1/2 \end{pmatrix} \begin{pmatrix} 1/6 & & -1/6 & \\ -2/3 & 1 & -1/3 & \\ 1/2 & & 1/2 & \\ -1/2 & & -1/2 & 1 \end{pmatrix} = \begin{pmatrix} 1/3 & & 0 & -1/3 \\ -7/12 & 1 & -1/4 & -1/6 \\ 1/2 & & 1/2 & 0 \\ -1/4 & & -1/4 & 1/2 \end{pmatrix}$.

9. $B = \{3, 4, 2, 1\}, \ N = \{6, 7, 5\}, \bar{x}_B = (2, 1, 9, 0)^T$.

Iteration 4:

1. $\bar{y} = (-13/6, 0, -1/2, -1/3)^T, \ \bar{z}_N = (0, 0, 0)^T - (-1/3, -1/2, -13/6)^T =$
   $(1/3, 1/2, 13/6)^T \geq 0$.

2. The optimal basic solution and optimal value:

$$\bar{x} = (0, 9, 2, 1, 0, 0, 0)^T, \qquad \bar{f} = -37.$$

If some practicable pivot rule (Chap. 11) or pricing scheme (Sect. 25.3) is used in the simplex method, there will be a need for computing row p. In order not to increase the number of systems to be solved, modern LP codes are often based on the following variant, where the objective row is computed in recurrence (see (3.13)).

Algorithm 3.5.2 (Simplex algorithm 2). Initial: (B, N), B^{-1}, $\bar{x}_B = B^{-1}b \geq 0$, $\bar{z}_N = c_N - N^T B^{-T} c_B$ and $\bar{f} = c_B^T \bar{x}_B$. This algorithm solves the standard LP problem (1.8).

1. Determine pivot column index $q \in \arg\min_{j \in N} \bar{z}_j$.
2. Stop if $\bar{z}_q \geq 0$ (optimality achieved).
3. Compute $\bar{a}_q = B^{-1} a_q$.
4. Stop if $\bar{a}_q \leq 0$ (unbounded).
5. Determine stepsize α and pivot row index p such that
   $$\alpha = \bar{x}_{j_p}/\bar{a}_{pq} = \min\{\bar{x}_{j_i}/\bar{a}_{iq} \mid \bar{a}_{iq} > 0; \ i = 1, \ldots, m\}.$$
6. Set $\bar{x}_q = \alpha$, and update $\bar{x}_B = \bar{x}_B - \alpha \bar{a}_q$, $\bar{f} = \bar{f} + \alpha \bar{z}_q$ if $\alpha \neq 0$.
7. Compute $\sigma_N = N^T v$, where $v = B^{-T} e_p$.
8. Update by: $\bar{z}_N = \bar{z}_N + \beta \sigma_N$, $\bar{z}_{j_p} = \beta$, where $\beta = -\bar{z}_q/\bar{a}_{pq}$.
9. Update B^{-1} by (3.23).
10. Update (B, N) by exchanging j_p and q.
11. Go to step 1.

Although they are equivalent in theory, the revised Algorithms differ from the tableau algorithm numerically. For solving large-scale LP problems, they are certainly superior to the latter (especially when $m \ll n$, see Sect. 3.8). In fact, it serves as a basis for designing practicable simplex variants, though the formulation of the latter is simpler, providing a suitable tool for illustration.

Algorithm 3.5.1 was previously derived based on the equivalence of the simplex tableau and revised simplex tableau. It may be derived alternatively by taking a downhill edge, emanating from a current vertex, as a search direction to form a line search scheme, as follows.

Without loss of generality, let $B = \{1, \ldots, m\}$ and $N = \{m + 1, \ldots, n\}$ be respectively the basis and nonbasis, associated with basic feasible solution $\bar{x}$. Assume that a pivot column index $q \in N$ has been determined such that

$$\bar{z}_q = c_q - a_q^T B^{-T} c_B < 0.$$

Introduce vector

$$\Delta x = \begin{pmatrix} B^{-1} a_q \\ -e_{q-m} \end{pmatrix}, \qquad (3.24)$$

where e_{q-m} is the $(n-m)$-dimensional unit vector with the $(q-m)$th component 1. It is clear that

$$-c^T \Delta x = c_q - a_q^T B^{-T} c_B = \bar{z}_q < 0. \tag{3.25}$$

Therefore, $-\Delta x$ is a downhill with respect to $c^T x$. Taking it as search direction gives the following line search scheme:

$$\hat{x} = \bar{x} - \alpha \Delta x, \tag{3.26}$$

where $\alpha \geq 0$ is a stepsize to be determined.

Since $\bar{x}$ is feasible, it holds for any $\alpha \geq 0$ that

$$A\hat{x} = A\bar{x} - \alpha[B, N]\Delta x = A\bar{x} = b.$$

Therefore, what should do is to maximize α subject to $\hat{x}_B \geq 0$. When $B^{-1}a_q \not\leq 0$, such doing results in α and p such that

$$\alpha = \bar{x}_{j_p}/(B^{-1}a_{pq}) = \min\{\bar{x}_{j_i}/(B^{-1}a_{iq}) \mid B^{-1}a_{iq} > 0, \ i = 1, \ldots, m\}. \tag{3.27}$$

It is clear that the according new solution $\hat{x}$ is still feasible. In fact, it is verified that $\hat{x}$ is just the basic feasible solution, associated with the new basis resulting from the old by exchanging j_p and q.

The relation between the new and old basis matrices is

$$\hat{B} = B + (a_q - a_{j_p})e_p^T.$$

In view of that a_{j_p} is the pth column of B and that $B^{-1}a_{j_p} = e_p$ and $B^{-1}a_q = \bar{a}_q$ hold, it is not difficult to derive the following result from Sherman-Morrison formula (Golub and Van Loan 1989):

$$\hat{B}^{-1} = B^{-1} - \frac{B^{-1}(a_q - a_{j_p})e_p^T B^{-1}}{1 + e_p^T B^{-1}(a_q - a_{j_p})} = \left(I - \frac{(\bar{a}_q - e_p)e_p^T}{\bar{a}_{pq}}\right)B^{-1}, \tag{3.28}$$

which may serve as an update of B^{-1}. In fact, it is easily verified that the preceding and (3.23) are actually equivalent.

The search direction $-\Delta x$, defined by (3.24), can be further investigated geometrically. Regarding set

$$E = \{x \in \mathcal{R}^n \mid Ax = b, \ x \geq 0, \ x_j = 0, \ q \neq j \in N\}$$
$$= \{x \in \mathcal{R}^n \mid x_B = B^{-1}b - x_q(B^{-1}a_q) \geq 0, \ x_q \geq 0, \ x_j = 0, \ q \neq j \in N\}, \tag{3.29}$$

we have the following result.

Proposition 3.5.1. *Set E is a downhill edge, emanating from the current vertex $\bar{x}$, and $-\Delta x$ is its direction. If $B^{-1}a_q \leq 0$, then $-\Delta x$ is an (unbounded) extreme direction.*

Proof. It is clear that E is a half-line or edge, emanating from $\bar{x}$. By (3.29), (3.27) and (3.24), for any $x \in E \subset P$ it holds that

$$E = \{x \in \mathcal{R}^n \mid x_B = B^{-1}b - x_q(B^{-1}a_q),\ 0 \leq x_q \leq \alpha,\ x_j = 0,\ q \neq j \in N\}$$
$$= \{x \in \mathcal{R}^n \mid x = \bar{x} - x_q\Delta x,\ x_q \in [0, \alpha]\}.$$

By (3.25), it is known that the associated objective value satisfies

$$f = c^T x = c^T \bar{x} - x_q c^T \Delta x = c^T \bar{x} + x_q \bar{z}_q \leq c^T \bar{x}.$$

Note that (3.27) is well-defined when $B^{-1}a_q \not\leq 0$. If, in addition, $\bar{x}_{j_p} = 0$, then $\alpha = 0$, and hence E degenerates to vertex $\bar{x}$. If $\bar{x}_{j_p} > 0$, hence $\alpha > 0$, then the associated objective value strictly decreases with $x_q \in [0, \alpha]$. Therefore, $-\Delta x$ is a direction of the downhill edge E. When $B^{-1}a_q \leq 0$, it is clear that $\alpha = +\infty$ corresponds to the edge $E \in P$, and hence $-\Delta x$ is an extreme direction. $\square$

Note that edge E, defined by (3.29), could degenerate to the current vertex $\bar{x}$ if some component of $B^{-1}b$ vanishes and that the objective value is lower unbounded over the feasible region if $-\Delta x$ is an extreme direction.

3.6   Degeneracy and Cycling

It was seen that a zero stepsize leads to the same basic feasible solution, and hence the unchanged objective value. Thus, finiteness of the simple method is questionable (see, e.g., Ryan and Osborne 1988; Wolfe 1963). Soon after its emerging, in fact, the simplex method is found not to terminate in few cases. E.M.L. Beale (1955) and A.J. Hoffman (1953) offered such instances independently. The following is due to Beale.

Example 3.6.1. Solve the following problem by Algorithm 3.2.1:

$$
\begin{aligned}
\min\ & f = -3/4x_4 + 20x_5 - 1/2x_6 + 6x_7, \\
\text{s.t.}\ \ x_1\ \ & + 1/4x_4 - 8x_5 - x_6 + 9x_7 = 0, \\
x_2\ & + 1/2x_4 - 12x_5 - 1/2x_6 + 3x_7 = 0, \\
x_3\ & + x_6 = 1, \\
& x_j \geq 0, \quad j = 1, \dots, 7.
\end{aligned}
$$

Answer   Initial: the following feasible tableau is available from the preceding:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
1			1/4*	−8	−1	9	
	1		1/2	−12	−1/2	3	
		1			1		1
			−3/4	20	−1/2	6	

Iteration 1:

1. $\min\{-3/4, 20, 1, -1/2, 6\} = -3/4 < 0$, $q = 4$.
3. $I = \{1, 2\} \neq \emptyset$.
4. $\min\{0/(1/4), 0/(1/2)\} = 0$, $p = 1$.
5. Multiply row 1 by 4, and then add $-1/2, 3/4$ times of row 1 to rows 2,4, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
4			1	−32	−4	36	
−2	1			4*	3/2	−15	
		1			1		1
3				−4	−7/2	33	

Iteration 2:

1. $\min\{3, -4, -7/2, 33\} = -4 < 0$, $q = 5$.
3. $I = \{2\} \neq \emptyset$.
4. $\min\{0/4\} = 0$, $p = 2$.
5. Multiply row 2 by 1/4, and then add $32, 4$ times of row 2 to rows 1,4, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
−12	8		1		8*	−84	
−1/2	1/4			1	3/8	−15/4	
		1			1		1
1	1				−2	18	

Iteration 3:

1. $\min\{1, 1, -2, 18\} = -2 < 0$, $q = 6$.
3. $I = \{1, 2, 3\} \neq \emptyset$.
4. $\min\{0/8, 0/(3/8), 1/1\} = 0$, $p = 1$.
5. Multiply row 1 by 1/8, and then add $-3/8, -1, 2$ times of row 1 to rows 2,3,4, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
$-3/2$	1		1/8		1	$-21/2$	
1/16	$-1/8$		$-3/64$	1		3/16*	
3/2	-1	1	$-1/8$			21/2	1
-2	3		1/4			-3	

Iteration 4:

1. $\min\{-2, 3, 1/4, -3\} = -3 < 0$, $q = 7$.
3. $I = \{2, 3\} \neq \emptyset$.
4. $\min\{0/(3/16), 1/(21/2)\} = 0$, $p = 2$.
5. Multiply row 2 by $16/3$, and then add $21/2, -21/2, 3$ times of row 2 to rows 1,3,4, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
2*	-6		$-5/2$	56	1		
1/3	$-2/3$		$-1/4$	16/3		1	
-2	6	1	5/2	-56			1
-1	1		$-1/2$	16			

Iteration 5:

1. $\min\{-1, 1, -1/2, 16\} = -1 < 0$, $q = 1$.
3. $I = \{1, 2\} \neq \emptyset$.
4. $\min\{0/2, 0/(1/3)\} = 0$, $p = 1$.
5. Multiply row 1 by $1/2$, and then add $-1/3, 2, 1$ times of row 1 to rows 2,3,4, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
1	-3		$-5/4$	28	1/2		
	1/3*		1/6	-4	$-1/6$	1	
		1			1		1
	-2		$-7/4$	44	1/2		

Iteration 6:

1. $\min\{-2, -7/4, 44, 1/2\} = -2 < 0$, $q = 2$.
3. $I = \{2\} \neq \emptyset$.
4. $\min\{0/(1/3)\} = 0$, $p = 2$.

5. Multiply row 2 by 3, and then add 3, 2 times of row 2 to rows 1,4, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
1			1/4	-8	-1		
	1		1/2	-12	$-1/2$	3	
		1			1		1
			$-3/4$	20	$-1/2$	6	

It is seen that stepsizes are equally zero in all the six iterations, and the last tableau is the same as the first one, consequently. Therefore, continuing the process must generate the same sequence of tableaus, as is a phenomena called *cycling*. So, the simplex algorithm failed to solve Beale's problem. It is clear that such a hated infinite case to the simplex method could occur only when degeneracy presents.

At the early days of the simplex method, some scholars thought that degeneracy hardly happens in practice, and up to now the nondegeneracy is still frequently assumed in theory. However, it turns out that degeneracy almost always presents when the simplex method is applied to solving real-world LP problems. Even so, fortunately, cycling rarely occurs, except for few artificial instances, and the simplex method has achieved great success in practice.

The real problem caused by degeneracy seems to be *stalling*, as it degrades method's performance seriously when a large number of iterations stay at a vertex for too long a time before exiting it. It is especially a headache for highly degenerate problems, where vanished basic components occupy a large proportion, as leads to a huge number of iterations. But this hard problem is only with the simplex method using the conventional pivot rule, not with variants using rules, presented in Chap. 11.

3.7 Finite Pivot Rule

As was shown int the previous section, the finiteness of the simplex method is not guaranteed in general. An approach or pivot rule that turns the simplex method to a finite one is called *finite*.

Is there any finite approach or pivot rule?

The answer is positive. Charnes (1952) proposed a "perturbation approach" by adding a perturbation term to the right-hand side of the initial feasible simplex tableau, i.e.,

$$w = (\epsilon, \epsilon^2, \ldots, \epsilon^m)^T,$$

where $\epsilon > 0$ is a sufficiently small parameter (while still using Dantzig's original rule for pivot column selection).

Theorem 3.7.1. *The perturbation approach is finite.*

Proof. The perturbation term added to the right-hand side can be put in the form $w = Iw$. In any iteration, the right-hand side can be written

$$v \triangleq \bar{b} + Uw,$$

where U is a permutation, resulting from performing elementary transformations on I. Note that U and I have the same rank m, and every row of U is nonzero. Firstly, it holds that $\bar{b} \geq 0$, because if, otherwise, $\bar{b}_i < 0$ for some $i \in \{1, \ldots, m\}$, then it follows that $v_i < 0$, as contradicts to problem's feasibility. Further, it is clear that $v_i > 0$ holds for all row indices i, satisfying $\bar{b}_i > 0$; on the other hand, $v_i > 0$ also holds for all row index i, satisfying $\bar{b}_i = 0$, because the first nonzero of the ith row of U is positive (otherwise, it contradicts the feasibility). Therefore, $v > 0$ holds. Since each tableau corresponds to a nondegenerate basic feasible solution, there is no any possibility of cycling, hence the process terminates within finitely many iterations. Consequently, eliminating all parameter terms in the end tableau leads to the final tableau of the original problem.   □

The order of two vectors, determined by their first different components, is called lexicographic order. Equal vectors are regarded as equal in the lexicographic order. $(\lambda_1, \ldots, \lambda_t) \prec (\mu_1, \ldots, \mu_t)$ means that the former is less than the latter in lexicographic order, that is, for the smallest subscript i, satisfying $\lambda_i \neq \mu_i$, it holds that $\lambda_i < \mu_i$. Similarly, "$\succ$" is used to denote "greater than" in lexicographic order.

Once a pivot column index q is determined, the perturbation approach amounts to determining a pivot row index p by

$$p \in \arg\min\{(\bar{b}_i + u_{i\,1}\epsilon + u_{i\,2}\epsilon^2 + \cdots + u_{i\,m}\epsilon^m)/\bar{a}_{i\,q} \mid \bar{a}_{i\,q} > 0; \ i = 1, \ldots, m\}. \tag{3.30}$$

As ϵ is sufficiently small, the preceding is equivalent to the following so-called *lexicographic rule* (Dantzig et al. 1955):

$$p \in \arg min\{(\bar{b}_i, u_{i\,1}, u_{i\,2}, \ldots, u_{i\,m})/\bar{a}_{i\,q} \mid \bar{a}_{i\,q} > 0; \ i = 1, \ldots, m\}, \tag{3.31}$$

where $u_{i\,j}$ is the entry at the ith row and the jth column of U, and "min" is minimization in the sense of lexicographic order.

Among existing finite rules, Bland (1977) rule draws great attention due to its simplicity (also see Avis and Chvatal 1978).

Rule 3.7.1 (Bland rule) Among nonbasic variables, corresponding to negative reduced costs, select the smallest-indexed one to enter the basis. When there are multiple rows, attaining the same minimum-ratio, select the basic variable with the smallest index to leave the basis.

Theorem 3.7.2. *Bland rule is finite.*

Proof. Assume that cycling occurs with the simplex algorithm using Bland rule. If some variable leaves the basis in a circle, it must enter the basis gain. Denote by T the index set of such shuttling variables, and define

$$t = \max\{j \in T\}.$$

Note that the stepsize is always equal to 0 in each iteration in the circle, and hence leads to the same basic feasible solution; besides, the h-indexed component of the basic feasible solution is 0 for any $h \in T$.

Assume that x_t is selected to enter the basis for simplex tableau

$$\begin{array}{cc|c} \bar{A} & 0 & \bar{b} \\ \hline \bar{z}^T & -1 & \bar{\beta} \end{array} \tag{3.32}$$

thus $\bar{z}_t < 0$, and $\bar{z}_j \geq 0$ for any reduced cost's index $j < t$.

Assume that at another simplex tableau

$$\begin{array}{cc|c} \hat{A} & 0 & \hat{b} \\ \hline \hat{c}^T & -1 & \hat{\beta} \end{array} \tag{3.33}$$

basic variable x_t in row p leaves and nonbasic variable x_s enters the basis. Let $x_{j_1}, \ldots, x_{j_m}$ be basic variables ($x_{j_p} \equiv x_t$). It follows that $\hat{c}_s < 0$, and $\hat{c}_j \geq 0$ for any reduced cost's index $j < s$. Note that pivot is positive, i.e., $\hat{a}_{ps} > 0$; since $s \in T$, it holds that $s < t$.

Define v_k, $k = 1, \ldots, n, n+1$ as follows:

$$v_k = \begin{cases} 1, & k = s, \\ -\hat{a}_{is}, & k = j_i, \quad i = 1, \ldots, m, \\ \hat{c}_s, & k = n+1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.34}$$

Note that basic columns of $\hat{A}$ constitute a permutation. Nonbasic components of vector

$$v = (v_1, \ldots, v_n, v_{n+1})^T$$

are all 0, except for $v_s = 1$. For $i = 1, \ldots, m$, on the other hand, the basic entries in row i of $\hat{A}$, except for $\hat{a}_{ij_i} = 1$, are all zero; basic entries of $\hat{c}$ are all zero. Therefore it holds that

$$\begin{pmatrix} \hat{A} & 0 \\ \hat{c}^T & -1 \end{pmatrix} v = \begin{pmatrix} \hat{a}_{1,s} - \hat{a}_{1,s} \\ \vdots \\ \hat{a}_{m,s} - \hat{a}_{m,s} \\ \hat{c}_s - \hat{c}_s \end{pmatrix} = 0. \tag{3.35}$$

Since (3.32) can be obtained from (3.33) by premultiplying a series of elementary matrices, it follows that

$$\begin{pmatrix} \bar{A} & 0 \\ \bar{z}^T & -1 \end{pmatrix} v = 0, \tag{3.36}$$

where the last equality is

$$\sum_{k=1}^{n} \bar{z}_k v_k - v_{n+1} = 0,$$

hence

$$\sum_{k=1}^{n} \bar{z}_k v_k = v_{n+1} = \hat{c}_s < 0.$$

Therefore, there exists some index $h < n + 1$ such that

$$\bar{z}_h v_h < 0, \tag{3.37}$$

giving $\bar{z}_h \neq 0$ and $v_h \neq 0$.

On the other hand, it is known by $v_h \neq 0$ and the definition of v that $h \in \{j_1, \ldots, j_m, s\}$. Thus, there are only following three cases arising:

(i) $h = s$. $v_h = 1$ in this case. Since x_t is an entering variable for simplex tableau (3.32) and $h = s < t$, hence $\bar{z}_h > 0$, it follows that $\bar{z}_h v_h = \bar{z}_h > 0$, contradicting (3.37).

(ii) $h = j_p = t$. In this case, from $\bar{z}_h = \bar{z}_t < 0$ and $v_h = -\hat{a}_{ps} < 0$, it follows that $\bar{z}_h v_h > 0$, contradicting (3.37).

(iii) $h = j_i \neq j_p$ or $h \neq t$. Now x_h is a nonbasic variable of simplex tableau (3.32) (otherwise, $\bar{z}_h = 0$); it is also a basic index of simplex tableau (3.33), hence $h \in T$. It follows that

$$\hat{b}_i = 0, \qquad h < t, \tag{3.38}$$

and hence $\bar{z}_h > 0$. Further, it holds that

$$v_h = -\hat{a}_{i,s} > 0,$$

since, otherwise, $v_h \neq 0$ gives $\hat{a}_{i,s} > 0$, from which and (3.38) it follows that x_h, rather than x_t, were selected to leave the basis for simplex tableau (3.33), as a contradiction to (3.37). Therefore, Bland rule is finite. $\square$

Chang (1979), Terlaky (1985) and Wang (1987) independently proposed a so-called "criss-cross" finite variant of Bland rule, which is embedded in a somehow different context, compared with the simplex method (see Chap. 18).

Unfortunately, it turns out that these finite rules are very slow in practice, not be mentioned in the same breath with the conventional rule. This is not surprising, however. For example, Rule 3.7.1 gives nonbasic variables with small index priority to enter the basis, while we all know that basic variables of an optimal solution are not necessarily small indexed.

Rule 3.7.1 actually uses a priority order, coinciding with decreasing indices, for selection of an entering variable. It is clear that the "ideal" order, if any, should enter the basic variables of an optimal solution to the basis. According to the heuristic Proposition 2.5.1, inequality constraints with small pivoting-indices should be satisfied as equations by an optimal solution, therefore the corresponding variables would be better to be nonbasic (zero-valued). In other words, variables with large pivoting-indices should have the priority to enter the basis (stipulation: among variables with equal pivoting-indices, select one with the largest index). Thus, we have the following variant of Bland rule (Pan 1990, 1992c).

Rule 3.7.2. Among nonbasic variables, corresponding to negative reduced costs, select the largest pivoting-indexed one to enter the basis. When there are multiple rows, attaining the same minimum-ratio, select the largest pivoting-indexed basic variable to leave the basis. When multiple variables correspond to the same largest pivoting-index, take the largest indexed one.

Theorem 3.7.3. *Rule 3.7.2 is finite.*

Proof. This rule is equivalent to Rule 3.7.1 if variables are re-given indices in accordance with their pivoting-indices.   □

Preliminary computational experiments with small test problems showed that performance of Rule 3.7.2 is much better than Bland's Rule 3.7.1. It might be the best among known finite rules. However, it is still inferior to the conventional rule, as requiring more iterations than the latter in general (Pan 1990).

Bland's Rule can be easily generalized to the following finite rule.

Rule 3.7.3. Given any order for variables. Among nonbasic variables, corresponding to negative reduced costs, select one the smallest in this order to enter the basis. When there are multiple rows attaining the same minimum-ratio, select the basic variable smallest in the order to leave the basis.

In Example 3.6.1 (Beale problem), we have seen that cycling occurred with the simplex algorithm. The situation will be different if Rule 3.7.2 is used in the place of the conventional rule.

Example 3.7.1. Solve Beale problem by Algorithm 3.2.1 using Rule 3.7.2:

$$\min \quad f = -3/4x_4 + 20x_5 - 1/2x_6 + 6x_7,$$
$$\text{s.t.} \quad x_1 \qquad + 1/4x_4 - \quad 8x_5 - \qquad x_6 + 9x_7 = 0,$$
$$x_2 \qquad + 1/2x_4 - 12x_5 - 1/2x_6 + 3x_7 = 0,$$
$$x_3 \qquad \qquad + \qquad x_6 \qquad = 1,$$
$$x_j \geq 0, \quad j = 1, \dots, 7.$$

Answer As the coefficient matrix includes a unit matrix, it is easy to transform the constraints to "$\geq$" type of inequalities:

$$\min \quad f = -3/4x_4 + 20x_5 - 1/2x_6 + 6x_7,$$
$$\text{s.t.} \quad -1/4x_4 + \quad 8x_5 + \qquad x_6 - 9x_7 \quad \geq \quad 0,$$
$$-1/2x_4 + 12x_5 + 1/2x_6 - 3x_7 \quad \geq \quad 0,$$
$$- \qquad x_6 \qquad \qquad \geq -1,$$
$$x_j \geq 0, \quad j = 4, \dots, 7.$$

Note that the first three constraints correspond to the original variables x_1, x_2, x_3, respectively; pivoting-indices of constraints may be regarded as those for the associated variables.

The gradient of the objective function is $c = (-3/4, 20, -1/2, 6)^T$. The gradient of the first constraint is $a_1 = (-1/4, 8, 1, -9)^T$. The pivoting-index of this constraint (or corresponding variable x_1) is $\alpha_1 = -a_1^T c / \|a_1\| = -8.74$. Similarly, calculate all pivoting-indices and put them in the following table in decreasing order:

Variable	Constraint				α_i
x_4	x_4			≥ 0	0.75
x_6		x_6		≥ 0	0.50
x_3		$-x_6$		≥ -1	-0.50
x_7			x_7	≥ 0	-6.00
x_1	$-1/4x_4 + \quad 8x_5 +$	$x_6 - 9x_7$		≥ 0	-8.74
x_5	x_5			≥ 0	-20.00
x_2	$-1/2x_4 + 12x_5 + 1/2x_6 - 3x_7$			≥ 0	-114.78

Now call Algorithm 3.2.1 with Rule 3.7.2.
Initial: The following feasible simplex tableau is obtained directly from the preceding problem:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
1			1/4*	−8	−1	9	
	1		1/2	−12	−1/2	3	
		1			1		1
			−3/4	20	−1/2	6	

Iteration 1:

1. Among nonbasic variables x_4 ($\alpha_4 = 0.75$) and x_6 ($\alpha_6 = 0.50$) with negative reduced costs, select the largest pivoting-indexed x_4 to enter the basis, $q = 4$.
3. $I = \{1, 2\} \neq \emptyset$.
4. $\min\{0/(1/4), 0/(1/2)\} = 0$. Among basic variables in rows 1 and 2, select the largest pivoting-indexed x_1 ($\alpha_1 = -8.74 > -114.78 = \alpha_2$) to leave the basis, $p = 1$.
5. Multiply row 1 by 4, and then add $-1/2, 3/4$ times of row 1 to rows 2,4, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
4			1	-32	-4	36	
-2	1			4	$3/2^*$	-15	
		1			1		1
3				-4	$-7/2$	33	

Iteration 2:

1. Among nonbasic variables x_5 ($\alpha_5 = -20.00$) and x_6 ($\alpha_6 = 0.50$) with negative reduced costs, select the largest-pivoting-indexed x_6 to enter the basis, $q = 6$.
3. $I = \{2, 3\} \neq \emptyset$.
4. $\min\{0/(3/2), 1/1\} = 0$, only x_2 is eligible for leaving the basis, $p = 2$.
5. Multiply row 2 by $2/3$, and then add $4, -1, 7/2$ times of row 2 to rows 1,3,4, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
$-4/3$	$8/3$		1	$-64/3$		-4	
$-4/3$	$2/3$			$8/3$	1	-10	
$4/3$	$-2/3$	1		$-8/3$		10^*	1
$-5/3$	$7/3$			$16/3$		-2	

Iteration 3:

1. Among nonbasic variables x_1 ($\alpha_1 = -8.74$) and x_7 ($\alpha_7 = -6.00$) with negative reduced costs, select the largest-pivoting-indexed x_7 to enter the basis, $q = 7$.
3. $I = \{3\} \neq \emptyset$.
4. Only x_3 is eligible for leaving the basis, $p = 3$.
5. Multiply row 3 by $1/10$, and then add $4, 10, 2$ times of row 3 to rows 1,2,4, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
$-4/5$	$12/5$	$2/5$	1	$-112/5$			$2/5$
		1			1		1
$2/15^*$	$-1/15$	$1/10$		$-4/15$		1	$1/10$
$-7/5$	$11/5$	$1/5$		$24/5$			$1/5$

Iteration 4:

1. Only nonbasic variable x_1 is eligible for entering the basis, $q = 1$.
3. $I = \{3\} \neq \emptyset$.
4. Only x_7 is eligible for leaving the basis, $p = 3$.
5. Multiply row 3 by $15/2$, and then add $4/5, 7/5$ times of row 3 to rows 1,4, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
	2	$2/5$	1	-24		6	1
		1			1		1
1	$-1/2$	$3/4$		-2		$15/2$	$3/4$
	$3/2$	$5/4$		2		$21/2$	$5/4$

All reduced costs are now nonnegative. The optimal solution and optimal value are

$$\bar{x} = (3/4, 0, 0, 1, 0, 1, 0)^T, \qquad \bar{f} = -5/4.$$

Thus, Beale problem is solved without cycling.

3.8 Computational Complexity

The evaluation of an algorithm is concerned with the amount of required arithmetics and storages, numerical stability and degree of difficulty for programming. In this section, we will discuss the simplex method's computational complexity, including time complexity (estimate of the number of required four basic arithmetics and comparisons), and storage complexity (estimate of the number of memory locations).

Either time or storage complexity is closely related to the scale of the problem handled: the larger the problem is, the higher the complexity. Therefore, analyzing complexity must be done with fixed problem's size. As for a standard LP problem, it is convenient to use m and n to determine its size roughly. Further, problem's

size also depends on concrete values of (A, b, c), as can be characterized by total number L, called *input length* of binary digits of input data. For reaching a certain solution precision, the amount of arithmetics is a function of m, n, L. If the number of arithmetics required by solving some type of problems is bounded above by some function $\tau f(m, n, L)$, the algorithm is said to have order $O(f(m, n, L))$ of (time) complexity, where $\tau > 0$ is a constant and $f(m, n, L)$ is *complexity function*. If $f(m, n, L)$ is a polynomial in m, n and L, it is said to be of polynomial time complexity. Usually, such algorithms are regarded as "good" ones, and the lower the order of the polynomial is, the better the algorithm. On the other hand, if $f(m, n, L)$ is exponential in m, n or L, it is said to be of exponential time complexity. Such algorithms are regarded as "bad", as they can fail to solve larger problems by consuming unacceptable amount of time. Note, however, that such complexity is the worst case complexity, as the amount of arithmetics never exceeds $\tau f(m, n, L)$.

The following table lists numbers of arithmetics per iteration and storage locations, required by tableau simplex Algorithm 3.2.1 vs. simplex Algorithm 3.5.1:

	Algorithm 3.2.1	Algorithm 3.5.1
Multiplications	$(m + 1)(n - m + 1)$	$m(n + 2m) + 2m + 1$
Additions	$m(n - m + 1)$	$m(n + 2m) - 2m + 1$
Storages	$(m + 1)(n + 1)$	$(m + 1)(m + 2)$

In the preceding table, the amount of storage locations required by Algorithm 3.2.1 excludes that for original data (A, b, c), though these data should be stored, in practice. In fact, both algorithms have to restart from scratch periodically after a certain number of iterations (see Sect. 5.1), let alone Algorithm 3.5.1 utilizes a part of them in each iteration. Therefore, storage requirement of the tableau simplex algorithm is significantly high, relative to that of the revised version, especially when $n \gg m$.

As iterative algorithms, their time complexity depends on the required number of iterations, as well as that of arithmetics per iteration. As they are equivalent theoretically, the two algorithms would require the same iterations in solving any standard LP problem, if rounding errors are neglected. Thus, we only compare the amount of arithmetics, mainly multiplications, in a single iteration. It is seen from the table that Algorithm 3.5.1 is much superior to the tableau version if $m \ll n$. In fact, the latter is not applied in practice, but only seen in textbooks. Note that all listed in the table is for dense computations. As for sparse computations, the former and its variants are even very much superior to the latter (see Chap. 5).

In addition, it is seen that the numbers of arithmetics per iteration are polynomial functions in m, n. Therefore, required iterations are a key to their time complexity. Note that each iteration corresponds to a basis, and the number of bases is no more than C_n^m. If $n \geq 2m$, then $C_n^m \geq (n/m)^m \geq 2^m$, as indicates that the required number of iterations could attain an exponent order. Indeed, Klee and Minty (1972) offered an example, indicating that the simplex method using the conventional pivot rule passes through all the 2^m vertices. Thus, the conventional rule is not polynomial,

in the sense that it does not turn the simplex method to a polynomial-time one. Moreover, it turns out that Bland's Rule 3.7.1, the "most improvement rule", and many other subsequent rules, like the steepest edge rule, are all not polynomial (Chap. 11). Actually, it has not been clear whether there exists a polynomial rule though such possibility seems to be very low, if any.

Computational experiences indicate that the conventional simplex algorithm is slow for solving certain type of LP problems, such as some hard large-scale problems or problems with combinatorial constraints; e.g., with zero or one coefficients or those from "Krawchuk polynomials" (Schrijver 1986, p. 141; also see Klee 1965). Nevertheless, its average efficiency is quite high. For solving small or medium LP problems, in particular, it usually requires iterations no more than $4m$ to $6m$ (including Phase-1).

The fact that the non-polynomial-time simplex algorithm and its variants perform very well in practice reveals that the worst case complexity is of limited reference value, even could be misguiding. In fact, the worst case hardly happens in practice, and complexity under some probability sense would be closer to reality. In this aspect, Borgwardt (1982a,b) showed that an average complexity of the simplex algorithm is polynomial. Specifically, for LP problem

$$\min \quad c^T x,$$
$$\text{s.t.} \quad Ax \le b,$$

where $A \in \mathcal{R}^{m \times n}$, $b > 0$, and components of c are random under certain assumptions, he proved that the mathematical expectation of iterations, required by the simplex algorithm using a special pivot rule, is

$$O(n^3 m^{1/(n-1)}).$$

Using a different probability model and pivot rule, Smale (1983a,b) proved that average complexity of the simplex algorithm when solving

$$\min \quad c^T x,$$
$$\text{s.t.} \quad Ax \le b, \qquad x \ge 0,$$

is bounded above by

$$O((\log n)^{m^2+m}),$$

which is not polynomial, but still better than Borgwardt's result when $m \ll n$. Combining Borgwardt's pivot rule and a generalized Smale's probability model, Haimovich (1996) proved that the average complexity of iterations required is linear polynomial. These theoretical results coincide with real situation.

Finally, we stress that algorithms' evaluation is basically a practical issue. In a word, practice is the unique touchstone: the value and vitality of an algorithm lie on its performance only.

3.9   On Features of the Simplex Method

In this final section of the chapter, we focuss on some features of the simplex method.

It is interesting that the method's prefix "simplex" came from a chat between G.B. Dantzig and T. Motzkin (Dantzig 1991) in the early days of LP. The latter indicated that the m columns of the basis matrix and the entering column just form a "simplex" in the m-dimensional space. Thus, each iteration in Dantzig's method may be viewed as a movement from a simplex to an adjacent simplex. Dantzig accepted his suggestion by consorting with the "simplex".

Accordingly, the simplex method is pivotal and basis-based, as is closely related to the linear structure of the LP model. Each iteration of it is characterized by a basis: once a basis is determined, so is done the corresponding basic feasible solution. If optimality cannot be asserted, a pivot is selected to make a basis change to improve the solution, or unboundedness of the problem is detected. Consequently, computational work per iteration, involved in the simplex method, is much less than that required by the interior-point method (Chap. 9).

If an optimal basis is available, an LP problem can be handled by just solving a single system of linear equations. Even if this is not the case, a basis close to an optimal one is useful: less iterations are usually required starting from a basis, yielded from a previously interrupted solution process. Such a so-called "warm start" features a source of main bonus of the simplex method. For instance, it is applied to sensitivity analysis and parametric programs (Chap. 6), the restarting tactic used in implementation (Chap. 5), as well as the decomposition principle (Sect. 25.6). In addition, the warm start is of great importance to the methodology for solving ILP problems.

It is noted that each iteration of the simplex method consists of a pivot selection and a basis change. Since it emerged, in fact, research on the method has not been beyond the scope of the two aspects. On one side, the pivot rule used in it is, no doubt, crucial to method's efficiency. As a result, new pivot rules were suggested from time to time, though Dantzig's original rule, because of its simplicity, had gained broad applications for a long time, as is a situation that has changed only about 20 years ago. More efficient rules will be presented in Chap. 11. On the other side, the computation related to pivot and basis change has been improved continually. Related results will be presented in later chapters, especially in Part II of this book.

As for concern whether an index enters and leaves the basis too many times, the following property seems to be favorable.

Proposition 3.9.1. *A leaving column in a simplex iteration does not enter the basis in the next iteration.*

Proof. Since an entering column corresponds to a negative reduced cost and the pivot determined is positive (see (3.10)), a leaving column corresponds to a positive reduced cost, after the associated elementary transformations carried out, and hence never enters the basis in the next iteration.   □

Nevertheless, it is not difficult to construct an instance, in which a column that just entered the basis leaves it immediately.

As was known, nonnegativity of nonbasic reduced costs is not a necessary condition for optimality. The following indicates that it is necessary if the nondegeneracy is ensured.

Proposition 3.9.2. *If a basic optimal solution is nondegenerate, reduced costs in the associated simplex tableau are all nonnegative.*

Proof. Assume that there are negative reduced costs in the simplex Tableau 3.1. Without loss of generality, assume $\bar{z}_q < 0$. If (3.7) holds, then unboundedness of the problem follows from Theorem 3.2.2, as contradicts the existence of an optimal solution; if, otherwise, (3.7) does not hold, it is known from the nondegeneracy assumption that $\alpha > 0$, and hence there is a feasible value strictly less than the optimal value, as is a contradiction. Therefore, reduced costs are all nonnegative.
                                                                              □

The preceding and Lemma 3.2.1 together imply that the condition of nonnegativity of nonbasic reduced costs is not only sufficient but also necessary to optimality under the nondegeneracy assumption. The following result concerns presence of multiple optimal solutions.

Proposition 3.9.3. *If reduced costs are all positive, there is a unique optimal solution to the LP problem. If a basic optimal solution is nondegenerate and there is a zero-valued reduced cost, then there are infinitely many optimal solutions; in the case when the feasible region is bounded, there are multiple basic optimal solutions.*

Proof. We prove the first half first. Assume that reduced costs in an optimal tableau are all positive, corresponding to the basic optimal solution $\bar{x}$. For any feasible solution $\hat{x} \geq 0$ different from $\bar{x}$, there is an index $s \in N$ such that $\hat{x}_s > 0$ (otherwise, the two are the same). Therefore, substituting $\hat{x}$ to (3.6) leads to

$$\hat{f} = \bar{f} + \sum_{j \in N} \bar{z}_j \hat{x}_j > \bar{f},$$

which implies that $\hat{x}$ is not optimal, as is a contradiction. Therefore, there is an unique optimal solution.

To prove the seconde half, assume that a tableau, say Table 3.1, gives a nondegenerate basic optimal solution and has zero reduced costs. Without loss of generality, assume $\bar{z}_q = 0$. If (3.7) holds, then inequalities of the right-hand side of (3.8) hold for any $x_q = \alpha > 0$, that is, there are infinitely many feasible solutions, corresponding to the same optimal value $-\bar{f}$ (see (3.9)); if the feasible region is bounded, then (3.7) does not hold, hence it is known from $\bar{b}_p > 0$ that the stepsize α, defined by (3.10), is positive. Thus, for any value of x_q in $[0, \alpha]$, a feasible solution can be determined by the equalities of (3.8), corresponding to optimal value $-\bar{f}$. Therefore, there are infinitely many optimal solutions. It is clear that entering x_q to and dropping x_{j_p} from the basis give a different basic optimal solution.   □

The last half of the proof actually describes an approach to obtain multiple basic optimal solutions by entering the nonbasic indices, corresponding to zero-valued reduced costs, to the basis. With this respect, an approach to intercepting for the optimal set will be described in Sect. 25.2.

As was mentioned in Sect. 2.4, the simplex method can be explained in terms of the active set method. In each iteration, in fact, a vertex is determined by $Ax = b$ and $x_j = 0$, $j \in N$, corresponding to n active constraints. Since it has zero basic components, a degenerate vertex is the intersection of superplanes, the number of which is greater than n. At first glance, this case seems rarely to occur in practice. Surprisingly, however, the situation is just the opposite: problems stemming from practice are almost all degenerate.

The simplex tableau is essentially the canonical form of $Ax = b$ (together with reduced costs), which may be initially created by the Gauss-Jordan elimination. Such a tableau was used to develop the simplex method previously, although the same can be done alternatively via the triangular form, involving an upper triangular submatrix rather than unit matrix. As it is associated with the Gauss elimination, in fact, the latter should be more relevant to implementation (see also the last paragraph of Sect. 1.6).

Finally, there are two issues that are not guaranteed by the simplex method.

As was well-known, the method is not a polynomial time one; even finiteness of it is, in presence of degeneracy, not guaranteed in theory. Practically, however, this might not be a serious problem, as the method performs well overall if implemented properly although some authors do not agree with this point (see, e.g., Kotiah and Steinberg 1978).

More seriously, the method in its very form is numerically unstable, because the selected pivot may be arbitrarily small in module (see, e.g., Chan 1985; Maros 2003b; Ogryczak 1988). Refer to Rule 3.2.1 used in steps 4 of Algorithm 3.2.1. The pivot $\bar{a}_{pq}$, selected by the minimum-ratio test, could be too small to carry out subsequent computations. Indeed, the simplex method in its very form can only solve few (even very small) LP problems.

Instead of Rule 3.2.1, the following rule may serve as a remedy for solving highly degenerate LP problems.

Rule 3.9.1 (Row rule) Define $I = \{i \mid \bar{a}_{iq} > 0, \ i = 1, \cdots, m\}$, $I_1 = \{i \mid \bar{b}_i = 0, \ i \in I\}$. Determine pivot row index p and stepsize α by

$$p \in \begin{cases} \arg\max\{\bar{a}_{iq} \mid i \in I_1\}, & \alpha = 0, & \text{if } I_1 \neq \emptyset, \\ \arg\min\{\bar{b}_i / \bar{a}_{iq} \mid i \in I\}, & \alpha = \bar{b}_p / \bar{a}_{pq}, & \text{otherwise.} \end{cases} \tag{3.39}$$

A more favorable and applicable remedy is Harris two-pass Rule 5.6.1 though somehow cumbersome (see also Greenberg 1978). Even so, the stability problem is still not overcome yet entirely, as is the source of many troubles encountered in practice. With this aspect, alternative methods presented in Chaps. 15 and 16 might be "terminators".

There are analogues to the preceding issues and remedies for various simplex variants, including the dual simplex method presented in the next chapter.