

Chapter 13

Simplex Phase-I Method

The tableau simplex algorithm requires a feasible simplex tableau to get itself started, whereas the revised simplex algorithm starts from a feasible basis or its inverse. The task of a Phase-I method is to provide such a starting point to Phase-II for achieving optimality. Such a two-Phase methodology has been successful in practice, compared with one-Phase, like the big-M.

Not have been used in practice, the artificial-variable method, presented in Sect. 3.3, is usually seen in textbooks only, because it is so rigid that there is no much choice for an initial basis, as the basis has to involve all the artificial variables, let alone expanding problem's scale. Moreover, some cumbersome treatment has to be carried out subsequently after zero optimal value of the auxiliary program is attained but there are still artificial variables remaining basic.

This chapter will focus on practicable Phase-I methods, which are artificial variable free, or involves a single artificial variable only (for further references, see Pan 1994b; Pan and Li 2003; Pan et al. 2004).

For simplicity of exposition, the conventional pivot rule will be utilized in this chapter. Nevertheless, it is preferable to use the nested largest-distance rule instead, although all rules presented in Chap. 11 are applicable for Phase-I.

13.1 Infeasibility-Sum Method

The auxiliary objective function of this method only involves variables, associated with negative components of the current basic solution. The method is widely used in practice, as it is artificial variable free, and performs remarkably.

Assume that an initial simplex tableau of form (3.18) is available, associated with basic solution

$$\bar{x}_B = \bar{b}, \quad \bar{x}_N = 0. \tag{13.1}$$

Introduce notation

$$I = \{i = 1, \dots, m \mid \bar{b}_i < 0\}, \quad \bar{I} = \{1, \dots, m\} \setminus I. \quad (13.2)$$

Assume that $I \neq \emptyset$. Construct the following auxiliary program:

$$\begin{aligned} \min \quad & -\sum_{i \in I} x_{j_i}, \\ \text{s.t.} \quad & x_B = \bar{b} - \bar{N}x_N, \\ & x_{j_i} \geq 0, \quad i \in \bar{I}; \quad x_j \geq 0, \quad j \in N, \end{aligned} \quad (13.3)$$

where the objective function, called “infeasibility-sum”, is the negative sum of variables, associated with negative components of the basic solution. Note that the constraint system of the auxiliary program is the same as that of the original problem, except the nonnegativity restriction is only imposed on variables, associated with nonnegative components of the basic solution.

It is clear that solution (13.1) is feasible to the auxiliary program. The associated feasible simplex tableau can be obtained by eliminating nonzero entries in the objective row by relevant elementary transformations. Consequently, the reduced objective function is

$$-\sum_{i \in I} x_{j_i} = w_0 + \bar{z}_N^T x_N, \quad (13.4)$$

where

$$w_0 = -\sum_{i \in I} \bar{b}_i > 0, \quad \bar{z}_j = \sum_{i \in I} \bar{a}_{i,j}, \quad j \in N.$$

The positiveness of objective value w_0 comes from (13.2).

Theorem 13.1.1 (Infeasibility test). *If reduced costs of an auxiliary simplex tableau are all nonnegative, the original problem is infeasible.*

Proof. Assume that $\bar{z}_N \geq 0$, and $\bar{x} \geq 0$ is a feasible solution to the original problem. Substitute \bar{x} to (13.4) results in a system, whose left-hand side is clearly less than or equal to zero, and, by $w_0 > 0$ and $\bar{z}_N \geq 0$, the right-hand side is strictly greater than zero, as is a contradiction. Therefore, the statement is valid. \square

If there is a negative reduced cost, then any existing column rule applies. Assume that a column index q was determined such that

$$\bar{z}_q = \sum_{i \in I} \bar{a}_{i,q} < 0. \quad (13.5)$$

As there is no nonnegativity restriction on current infeasible basic variables, the auxiliary program could be unbounded even if the original problem itself bounded.

In fact, it is the case when $\bar{I} = \emptyset$ or the following condition holds:

$$\{i \in \bar{I} \mid \bar{a}_{iq} > 0\} = \emptyset. \quad (13.6)$$

Thus, the conventional row rule is not applicable. To overcome this difficulty, it is modified as follows.

Rule 13.1.1 (Auxiliary row rule) Determine a row index p such that

$$\alpha = \bar{b}_p / \bar{a}_{pq} = \begin{cases} \min\{\bar{b}_i / \bar{a}_{iq} \mid \bar{a}_{iq} > 0, i \in \bar{I}\}, & \text{if } \{i \in \bar{I} \mid \bar{a}_{iq} > 0\} \neq \emptyset, \\ \max\{\bar{b}_i / \bar{a}_{iq} \mid \bar{a}_{iq} < 0, i \in I\}, & \text{otherwise} \end{cases} \quad (13.7)$$

It will be seen a little later the bonus, yielding from use of “max” in the second expression of (13.7).

Lemma 13.1.2. *Assume that the selected column index q satisfies (13.5). Then rule 13.1.1 determines a row index p such that the simplex tableau, resulting from the according basis change, is feasible to the auxiliary program.*

Proof. First, it can be asserted that there is a row index $t \in I$ such that $\bar{a}_{tq} < 0$; because otherwise from

$$\bar{a}_{iq} \geq 0, \quad \forall i \in I,$$

it follows that

$$\sum_{i \in I} \bar{a}_{iq} \geq 0,$$

which contradicts the assumption (13.5). Therefore, the second expression of (13.7) is always well-defined, hence so is the row rule.

Take \bar{a}_{pq} as the pivot. Executing according elementary transformations leads to a new simplex tableau, whose right-hand side can be expressed in terms of its predecessor’s entries (see (3.14)), i.e.,

$$\begin{aligned} \hat{b}_i &= \bar{b}_i - \alpha \bar{a}_{iq}, \quad i \in \bar{I}, i \neq p, \\ \hat{b}_p &= \alpha, \end{aligned}$$

where α is determined by (13.7). It is easy to verify that these components are all nonnegative, hence the resulting simplex tableau is again feasible to the auxiliary program. \square

Theorem 13.1.3. *Assume feasibility of the original problem and nondegeneracy throughout solution process. The number of basic infeasible variables strictly decreases in finitely many iterations.*

Proof. It is seen from the proof of Lemma 13.1.2 that the second expression of (13.7) is still well-defined if “max” there replaced by “min”, i.e.,

$$\alpha_1 \triangleq \bar{b}_s / \bar{a}_{sq} = \min\{\bar{b}_i / \bar{a}_{iq} \mid \bar{a}_{iq} < 0, i \in I\} > 0. \quad (13.8)$$

If, in some iteration, the stepsize α is determined by the first expression of (13.7), and it holds that

$$\alpha \geq \alpha_1, \quad (13.9)$$

then the component, associated with row index s , of the new solution satisfies

$$\hat{x}_{j_s} = \bar{b}_s - \alpha \bar{a}_{sq} \geq 0,$$

which implies that a basic infeasible variable becomes feasible, at least.

If, in some iteration, the second expression of (13.7) is used, then, for the new solution, not only the basic components, associated with all row indices $i \in \bar{I}$, are feasible, but so are basic components, associated with row indices

$$s \in \{i \in I \mid \bar{a}_{iq} < 0\} \neq \emptyset.$$

This is of course favorable.

Assume that the number of basic infeasible variables does not strictly decreases forever. In each iteration, there is always a negative cost since the original problem is feasible (Theorem 13.1.1), and the stepsize α is determined by the first expression of (13.7), which does not satisfy (13.9). But the number of basis is finite, hence there are some bases appear infinitely, as cycling occurs. This is impossible because the objective value of the auxiliary program strictly decreases under the nondegeneracy assumption. Therefore the number of basic infeasible variables strictly decreases in finitely many iterations. \square

When, after some iteration, the number of basic infeasible variables strictly decreases, but does not vanish, a new auxiliary program is constructed, whose objective function (infeasibility-sum) involves less variables than its predecessor, and iterations are carried out, until infeasibility of the original problem is detected or a feasible simplex tableau is reached. Involving a series of infeasibility-sums as objective functions in general, this method is sometimes referred to as *method with piecewise sum of infeasibilities*.

The overall steps is put in the following algorithm.

Algorithm 13.1.1 (Tableau Phase-I: infeasibility-sum). Initial: simplex tableau of form (3.18). This algorithm finds a feasible simplex tableau by handling auxiliary programs of form (13.3).

1. Stop if $\bar{b}_i \geq 0, \forall i = 1, \dots, m$ (a feasible simplex tableau reached).
2. For all $j \in N$, compute $\bar{z}_j = \sum_{i \in I} \bar{a}_{i,j}$, where $I = \{i = 1, \dots, m \mid \bar{b}_i < 0\}$.

3. Stop if $\bar{z}_N \geq 0$ (the original problem is infeasible).
4. Form a simplex tableau to the auxiliary program (13.3).
5. Carry out a single iteration of simplex Algorithm 3.2.1, in which Rule 13.1.1 is used for row pivoting instead.
6. Go to step 1.

Example 13.1.1. Solve the following problem by Algorithm 13.1.1 in Phase-I:

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

Answer Phase-I. Multiply the second equation by -1 to turn the coefficient of x_5 to 1. Setting $I = \{1, 2\}$, construct auxiliary program

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

Its initial tableau is

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

Respectively add the first and the second row to the objective row to eliminate the nonzero basic entries, yielding simplex tableau

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

Iteration 1:

$q = 3; \min\{8/1\} = 8, p = 3$. So 1 in the x_3 column and the third row is the pivot. Add 2, 4, 5 times of row 3 to rows 1,2,4, respectively.

x_1	x_2	x_3	x_4	x_5	x_6	RHS
-5	-1		1		2	12
-13	5			1	4	30
-3	1	1			1	8
-18	4				6	42

The right-hand side of the preceding tableau is nonnegative, hence Phase-I is finished. To go to Phase-II, use coefficients of the original objective function to cover the bottom (objective) row.

x_1	x_2	x_3	x_4	x_5	x_6	RHS
-5	-1		1		2	12
-13	5			1	4	30
-3	1	1			1	8
-1	1	-2				

Add 2 times of row 3 to the bottom row to eliminate nonzero basic entries, obtaining feasible simplex tableau

x_1	x_2	x_3	x_4	x_5	x_6	RHS
-5	-1		1		2	12
-13	5			1	4	30
-3	1	1			1	8
-7	3				2	16

Carry out Phase-II from the preceding tableau. By the conventional rule, x_1 column is selected as the pivot column. All components of this column are nonpositive, as detects unboundedness of the original problem.

In practice, used is usually the following revised version of Algorithm 13.1.1.

Algorithm 13.1.2 (Phase-I: infeasibility-sum). Initial: (B, N) , B^{-1} , $\bar{x}_B = B^{-1}b$. This algorithm finds a basic feasible solution to the standard LP problem by handling auxiliary programs of form (13.3).

1. Stop if $\bar{x}_B \geq 0$ (producing a basic feasible solution).
2. Construct $c_B : c_{j_i} = \begin{cases} -1, & \text{If } \bar{x}_{j_i} < 0, \\ 0, & \text{If } \bar{x}_{j_i} \geq 0, \end{cases} \quad i = 1, \dots, m$.
3. Carry out a single iteration of Algorithm 3.5.1 (or 3.5.2), in which Rule 13.1.1 is used for row pivoting.
4. Stop if it terminates as step 3 (or step 2) (the original problem is infeasible).
5. Go to step 1.

13.2 Single-Artificial-Variable Method

Although the infeasible-sum method is artificial variable free, the row pivot rule used in it is cumbersome, compared with the normal row rule. This section describes a method, which uses the normal row rule and only involves a single artificial variable.

Let (B, N) be basis and nonbasis of the standard problem (1.8). There are two schemes for the method.

Scheme 1. The auxiliary program is based on the original data of the problem.

Given m -dimensional vector $h \geq 0$ such that $b - Bh \neq 0$. Introducing artificial variable x_{n+1} and normalized vector

$$a_{n+1} = (b - Bh) / \|b - Bh\|, \quad (13.10)$$

construct the following auxiliary program

$$\begin{aligned} \min \quad & x_{n+1}, \\ \text{s.t.} \quad & Ax + a_{n+1}x_{n+1} = b, \quad x, x_{n+1} \geq 0. \end{aligned} \quad (13.11)$$

It is easy to verify that

$$\bar{x}_B = h, \quad \bar{x}_N = 0, \quad \bar{x}_{n+1} = \|b - Bh\|$$

is a feasible solution to the program, though not necessarily a basic solution. For instance, setting $h = e > 0$ seems to be helpful to avoid zero stepsize, though according a_{n+1} would loose sparsity.

The constraint system of the auxiliary program is equivalent to

$$x_B = \bar{b} - \bar{a}_{n+1}x_{n+1} - \bar{N}x_N, \quad (13.12)$$

where

$$\bar{b} = B^{-1}b, \quad \bar{a}_{n+1} = B^{-1}a_{n+1}, \quad \bar{N} = B^{-1}N. \quad (13.13)$$

If $\bar{b} \geq 0$, then x_{n+1} is clearly allowed to be zero. Thus, erasing the artificial variable leads to a basic feasible solution to the original problem; otherwise, the following applies.

Proposition 13.2.1. *If $\bar{b} \not\geq 0$, row index set $\{i = 1, \dots, m \mid \bar{a}_{i,n+1} < 0, \bar{b}_i < 0\}$ is nonempty.*

Proof. We show that if $\bar{b}_i < 0$ for some $i \in \{1, \dots, m\}$, then $\bar{a}_{i,n+1} < 0$. Assume it is not the case, i.e.,

$$\bar{a}_{i,n+1} \geq 0.$$

Then from the preceding, (13.10) and (13.13) it is known that

$$0 \leq \bar{a}_{i,n+1} = e_i^T B^{-1} a_{n+1} = e_i^T B^{-1} (b - Bh) / \|b - Bh\| = (\bar{b}_i - h_i) / \|b - Bh\|,$$

hence $\bar{b}_i \geq h_i \geq 0$, as a contradiction. \square

Now assume $\bar{b} \not\geq 0$. If $\bar{x}_N = 0$ is fixed and the value of \bar{x}_{n+1} decreases from $\|b - Bh\|$, then \bar{x}_B changes from h accordingly. If $\bar{b}_i < 0$ for some $i \in \{1, \dots, m\}$, the according inequality constraint may block-up the value of \bar{x}_{n+1} from decreasing to zero. In fact, the smallest x_{n+1} value, satisfying inequalities

$$x_B = \bar{b} - \bar{a}_{n+1} x_{n+1} \geq 0, \quad x_{n+1} \geq 0,$$

is

$$\alpha = \bar{b}_r / \bar{a}_{r,n+1} = \min\{\bar{b}_i / \bar{a}_{i,n+1} \mid \bar{a}_{i,n+1} < 0, \bar{b}_i < 0, i = 1, \dots, m\} > 0. \quad (13.14)$$

As the value of \bar{x}_{n+1} decreases, the according value of \bar{x}_{j_r} hits 0 first. Let x_{j_r} leave and x_{n+1} enter the basis, then the resulting basis corresponds to a basic feasible solution to the auxiliary program, with its objective value strictly decreases. As it is lower bounded, solving the auxiliary program will render its optimal solution.

It is not difficult to show that a basic feasible solution or basis to the original problem can be obtained if the optimal value of the auxiliary program is equal to zero (x_{n+1} leaves the basis). The original problems is infeasible if the optimal value of the auxiliary program is positive.

Scheme 2. The auxiliary program is based on a canonical form of the constraint system.

Assume that the canonical form, associated with basis B , is available and that $\bar{b} = B^{-1}b \not\geq 0$. Giving m -dimensional vector $h \geq 0$, and defining

$$\bar{a}_{n+1} = (\bar{b} - h) / \|\bar{b} - h\|, \quad (13.15)$$

construct the following auxiliary program

$$\begin{aligned} \min \quad & x_{n+1}, \\ \text{s.t.} \quad & x_B = \bar{b} - \bar{a}_{n+1} x_{n+1} - \bar{N} x_N, \\ & x, x_{n+1} \geq 0. \end{aligned} \quad (13.16)$$

It is clear that there is a feasible solution to the program, i.e.,

$$\bar{x}_B = h, \quad \bar{x}_N = 0, \quad \bar{x}_{n+1} = \|\bar{b} - h\|.$$

Thus, determining pivot column index $q = n + 1$ and row index p satisfying (13.14), one can obtain a feasible basis by performing according basis change. It is noticed that Proposition 13.2.1 is still valid in this case.

As for how to determine h , it is only required that $h \geq 0$, in principle. With respect to practice, however, it is not proper for the associated $\|\bar{a}_{n+1}\|$ being too large. A simple choice is $h = e > 0$. Another choice is setting

$$\delta_i \geq 0, \quad i = 1, \dots, m.$$

and taking

$$h_i = \begin{cases} \delta_i, & \text{if } \bar{b}_i < \delta_i, \\ \bar{b}_i, & \text{otherwise,} \end{cases} \quad i = 1, \dots, m.$$

where $\delta_i \geq 0, i = 1, \dots, m$ are a set of constants, which are equal or close to zero. Consequently, (13.15) becomes

$$\bar{a}_{i,n+1} = \begin{cases} \bar{b}_i - \delta_i, & \text{if } \bar{b}_i < \delta_i \\ 0, & \text{otherwise,} \end{cases} \quad i = 1, \dots, m. \quad (13.17)$$

We prefer the latter because it is better for maintaining sparsity of \bar{a}_{n+1} .

More specifically, assume availability of a simplex tableau, say (3.18), of the original problem. Inserting the artificial x_{n+1} column, we obtain the auxiliary simplex tableau below:

x_B^T	x_N^T	x_{n+1}	RHS
I	\bar{N}	\bar{a}_{n+1}	\bar{b}

(13.18)

where the original objective row is excluded. Determine row index r by (13.14). Then drop x_{j_r} from and enter x_{n+1} to the basis by elementary transformations, so that x_{n+1} becomes the r th basic variable. Taking the r th row as the objective row, the auxiliary program can be solved by the simplex method. If the optimal value vanishes and x_{n+1} has left the basis, a feasible simplex tableau to the original problem is obtained. If the optimal value is nonzero, the original problem is infeasible.

The steps can be summarized to the following algorithm.

Algorithm 13.2.1 (Tableau Phase-I: single-artificial-variable). Initial: a simplex tableau of form (13.18). This algorithm finds a feasible simplex tableau to the standard LP problem.

1. Select a row index $r \in \arg \min\{\bar{b}_i \mid i = 1, \dots, m\}$.
2. Stop if $\bar{b}_r \geq 0$ (infeasible problem).
3. Determine row index $r \in \arg \min\{\bar{b}_i / \bar{a}_{i,n+1} \mid \bar{a}_{i,n+1} < 0, \bar{b}_i < 0, i = 1, \dots, m\}$.
4. Carry out elementary transformation to convert $\bar{a}_{r,n+1}$ to 1, and eliminate all other nonzeros of the column.

5. Determine column index $q \in \arg \max_{j \in N} \bar{a}_{rj}$.
6. Stop if $\bar{a}_{rj} \leq 0$.
7. Determine row index $p \in I = \arg \min \{\bar{b}_i / \bar{a}_{iq} \mid \bar{a}_{iq} > 0; i = 1, \dots, m\}$.
8. Set $p = r$ if $r \in I$.
9. Carry out elementary transformations to convert \bar{a}_{pq} to 1, and eliminate all other nonzeros of the column.
10. Stop if $p = r$ (feasibility achieved).
11. Go to step 5.

It is not difficult to transform Algorithm 13.2.1 to a revised version, as is omitted here. We will see that the single-variable auxiliary program can be handled in a relevant and favorable manner in the so-called “reduced simplex” context (Sect. 15.3).

Example 13.2.1. Find a feasible simplex tableau to the following problem by Algorithm 13.2.1:

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

Answer The constraint system is of a canonical form. Set $\delta_1 = \delta_2 = 0$. Introducing artificial variable x_7 , construct an auxiliary program of form (13.16):

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

Its initial tableau is (we put the objective row of the original problem at the bottom to turn to Phase-II conveniently when Phase-I is finished).

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

Iteration 1:

1. $\min\{-1, 2, -4\} = -4 < 0$.
3. $\max\{-1/-1, -4/-4\} = 1, r = 1$.

4. Multiply row 1 by -1 , then add 4 times of row 1 to row 3:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
1	-1	1		-1		1	1
-3	1	2			1		2
5^*	-7	1	1	-4			
1	3	-2	6				

5. $\max\{1, -1, 1, -1\} = 1 > 0$, $q = 1$.

7. $\min\{1/1, 0/5\} = 0$, $p = 3$.

8. $p \neq 1$.

9. Multiply row 3 by $1/5$, then add $-1, 3, -1$ 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
	$2/5$	$4/5$	$-1/5$	$-1/5$		1	1
	$-16/5$	$13/5$	$3/5$	$-12/5$	1		2
1	$-7/5$	$1/5^*$	$1/5$	$-4/5$			
	$22/5$	$-11/5$	$29/5$	$4/5$			

Iteration 2:

5. $\max\{2/5, 4/5, -1/5, 1/5\} = 4/5 > 0$, $q = 3$.

7. $\min\{1/(4/5), 2/(13/5), 0/(1/5)\} = 0$, $p = 3$.

8. $p \neq 1$.

9. Multiply row 3 by 5, then add $-4/5, -13/5, 11/5$ 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	6		-1	3		1	1
-13	15^*		-2	8	1		2
5	-7	1	1	-4			
11	-11		8	-8			

Iteration 3:

5. $\max\{-4, 6, -1, 3\} = 6 > 0$, $q = 2$.

7. $\min\{1/6, 2/15\} = 2/15$, $p = 2$.

8. $p \neq 1$.

9. Multiply row 2 by $1/15$, then add $-6, 7, 11$ 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
$6/5^*$			$-1/5$	$-1/5$	$-2/5$	1	$1/5$
$-13/15$	1		$-2/15$	$8/15$	$1/15$		$2/15$
$-16/15$		1	$1/15$	$-4/15$	$7/15$		$14/15$
$22/15$			$95/15$	$-32/15$	$11/15$		$22/15$

Iteration 4:

5. $\max\{5/6, -1/5, -1/5, -2/5\} = 5/6 > 0, q = 1.$
7. $\min\{(1/5)/(11/5)\} = 1/11, p = 1.$
8. $p = 1.$
9. Multiply row 1 by $5/6$, then add $13/15, 16/15, -6/5, -22/15$ 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			$-1/6$	$-1/6$	$-1/3$	$5/6$	$1/6$
	1		$-5/18$	$7/18$	$-2/9$	$13/18$	$5/18$
		1	$-1/9$	$-4/9$	$1/9$	$8/9$	$10/9$
			$61/9$	$-17/9$	$11/9$	$-11/9$	$11/9$

As the artificial variable x_7 has already left the basis, the resulting is a feasible simplex tableau to the original problem.

It is noted that δ_1 and δ_2 above were all set to 0, consequently the first 3 iterations made no progress due to degeneracy. Although parameters $\delta_i, i = 1, \dots, m$ are required to be nonnegative theoretically, therefore, they should be set to different positive values practically. Moreover, it seems to be relevant to set $\delta_i \ll 1$ such that $\|\bar{a}_{n+1}\|$ is not too large.

13.3 The Most-Obtuse-Angle Column Rule

Surprisingly enough, the conventional pivot rule used in the dual simplex algorithm was found to be useful, with favorable numerical results, for achieving primal feasibility (Pan 1994a); it is to say that the rule is itself applicable even when reduced costs are not nonnegative. This leads to a simple artificial-variable-free variant involving no reduced cost, as turns out to be a counterpart of the most-obtuse-angle row rule (Sect. 14.3).

Assume that a simplex tableau, say (3.18), is infeasible, and a pivot row index p has been selected by the conventional rule

$$p \in \arg \min\{\bar{b}_i \mid i = 1, \dots, m\}. \tag{13.19}$$

So it holds that $\bar{b}_p < 0.$

The key of the presented Phase-I method is to use the following column rule.

Rule 13.3.1 (Most-obtuse-angle column rule) Determine pivot column index

$$q \in \arg \min_{j \in N} \bar{a}_{pj}. \quad (13.20)$$

In case of $\bar{a}_{pq} < 0$, after the basis change, the basic infeasible variable x_{j_p} becomes nonbasic (hence feasible), and the p th component of the right-hand side of the tableau changes from negative to positive, that is, $\hat{b}_p = \bar{b}_p / \bar{a}_{pq} > 0$.

The steps are put in the following algorithm. A generalized version of it will be derived alternatively in Sect. 20.7.

Algorithm 13.3.1 (Tableau Phase-I: the most-obtuse-angle column rule). Initial: simplex tableau of form (3.18). This algorithm finds a feasible simplex tableau to the standard LP problem.

1. Select row index $p \in \arg \min\{\bar{b}_i \mid i = 1, \dots, m\}$.
2. Stop if $\bar{b}_p \geq 0$.
3. Determine column index $q \in \arg \min_{j \in N} \bar{a}_{pj}$.
4. Stop if $\bar{a}_{pq} \geq 0$.
5. Carry out elementary transformations to convert \bar{a}_{pq} to 1, and eliminate all other nonzeros of the column.
6. Go to step 1.

It is noted that only the right-hand side and the pivot row of the simplex tableau are used in each iteration of the preceding Algorithm.

Theorem 13.3.1. *Assume termination of Algorithm 13.3.1. It terminates either at*

- (i) *Step 2, obtaining a basic feasible solution; or at*
- (ii) *Step 4, detecting infeasibility of the problem.*

Proof. When it terminates at step 2, it is clear that $\bar{b} \geq 0$, hence a feasible simplex tableau is achieved. In case of termination at step 4, it holds that $\bar{b}_p < 0$ and that $\bar{a}_{pj} \geq 0, \forall j \in N$. By Lemma 3.3.1, there is no feasible solution to the problem. \square

The algorithm is not a monotone one, that is, the objective value would not monotonically change in solution process. Finiteness of it is hence not guaranteed even if nondegeneracy is assumed. Although Guerrero-Garcia and Santos-Palomo (2005) offer a cycling instance for this algorithm, it may be still expected that cycling hardly happens in practice.

Geometric meaning of the most-obtuse-angle column rule may be revealed by investigating in the dual space. Let (\bar{y}, \bar{z}) be the current dual basic solution. The $(m + n)$ -dimensional vector

$$d \triangleq \begin{pmatrix} h \\ \sigma_N \\ \sigma_B \end{pmatrix} = \begin{pmatrix} -B^{-T} \\ N^T B^{-T} \\ I \end{pmatrix} e_p \quad (13.21)$$

is an uphill direction. In fact, it is verified that d is in the null space of coefficient matrix of the dual constraint equalities, satisfying

$$A^T d + \sigma = 0,$$

and forms an acute angle with the dual objective gradient, i.e.,

$$(b^T, 0, 0)d = b^T h = -\bar{x}_{j_p} > 0.$$

The negative σ_q implies that d forms the largest possible obtuse angle with the gradient e_{m+q} of the dual nonnegative constraint $z_q \geq 0$. Therefore, if the uphill direction is close to the dual objective gradient, then e_{m+q} tends to form the most obtuse angle with the dual objective gradient. According to the heuristic characteristic of optimal basis (Sect. 2.5), it is favorable to let the dual constraint $z_q \geq 0$ be satisfied as equality, accordingly entering x_q to the basis.

We stress that the use of the conventional dual rule in step 1 is just for simplicity, does not mean itself the best choice. In fact, it should be much better to use rules, presented in Chap. 12 instead. From a geometric point of view, in fact, the most-obtuse-angle column rule should be best matched by the dual steepest-edge rule II.

Nevertheless, Rule 13.3.1 does not employ information associated with the objective function at all. Taking into account the extent to which the current vertex violates dual constraints, we suggest the following variant.

Rule 13.3.2 (Variant of most-obtuse-angle column rule) Given constant $0 < \tau \leq 1$. Select pivot column index

$$q \in \arg \min \{\bar{z}_j \mid \bar{a}_{pj} \leq \tau\theta, j \in N\}, \quad \theta = \min_{j \in N} \bar{a}_{pj} < 0$$

In order to widen the range of choice, τ should be close to 1, so that in the case when \bar{z}_N has negative components, the dual constraint inequality, that is violated the most, will be satisfied as equality. It might be suitable to set $\sigma = 0.95$, or so.

An advantage of this artificial-variable-free Phase-I method lie in its good numerical stability and remarkable simplicity. Even though there are no related numerical results available at present, it seems to be promising, as its dual counterpart performs remarkably for solving large and sparse problems in computational experiments (see Sect. 14.3).

Example 13.3.1. Find a feasible simplex tableau to the following problem by Algorithm 13.3.1:

$$\begin{aligned} \min \quad & f = 2x_1 - x_2, \\ \text{s.t.} \quad & 2x_1 - x_2 + x_3 = -2, \\ & x_1 + 2x_2 + x_4 = 3, \\ & -8x_1 + x_2 - x_4 + x_5 = -4, \\ & x_j \geq 0, \quad j = 1, \dots, 5. \end{aligned}$$

Answer Initial: Adding the second equality constraint to the third to eliminate x_4 . Such doing results in the following simplex tableau.

x_1	x_2	x_3	x_4	x_5	RHS
2	-1*	1			-2
1	2		1		3
-7	3			1	-1
2	-1				

Iteration 1:

- $\min\{-2, 3, -1\} = -2 < 0, p = 1$.
- $\min\{2, -1\} = -1 < 0, q = 2$.
- Multiply row 1 by -1 , then add $-2, -3, 1$ times of row 1 to rows 2,3,4, respectively:

x_1	x_2	x_3	x_4	x_5	RHS
-2	1	-1			2
5		2	1		-1
-1*		3		1	-7
		-1			2

Iteration 2:

- $\min\{2, -1, -7\} = -7 < 0, p = 3$.
- $\min\{-1, 3\} = -1 < 0, q = 1$.
- Add 2, 5 times of row 3 to rows 1,2, respectively:

x_1	x_2	x_3	x_4	x_5	RHS
	1	-7			16
		17	1		34
1		-3		-1	7
		-1			2

The right-hand side of the above tableau is nonnegative, hence the obtained is a feasible simplex tableau, from which Phase-II can get itself started.

In practice, usually used is the following revised version of the preceding algorithm.

Algorithm 13.3.2 (Phase-I: most-obtuse-angle column rule). Initial: (B, N) , $B^{-1}, \bar{x}_B = B^{-1}b$. This algorithm finds a basic feasible solution to the standard LP problem.

- Select pivot row index $p \in \arg \min\{\bar{x}_{j_i} \mid i = 1, \dots, m\}$.
- Stop if $\bar{x}_{j_p} \geq 0$ (feasibility achieved).

3. Compute $h = B^{-T}e_p$, $\sigma_N = N^T h$.
4. Determine pivot column index $q \in \arg \min_{j \in N} \sigma_j$.
5. Stop if $\sigma_q \geq 0$ (infeasible problem).
6. Compute $\bar{a}_q = B^{-1}a_q$ and $\alpha = \bar{x}_{j_p}/\sigma_q$.
7. Set $\bar{x}_q = \alpha$, and update: $\bar{x}_B = \bar{x}_B - \alpha \bar{a}_q$ 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.

13.4 Perturbation of Reduced Costs

As a result of not involving reduced costs, the feasible solution, generated by Algorithm 13.3.2, could be far from dual feasibility, and hence the number of iterations subsequently required by Phase-II would be large. In this aspect, the Phase-I method, presented in this section, seems to be advantageous (Pan 2000b,c).

It solves an auxiliary perturbed program. Let (3.18) be an infeasible simplex tableau and let $\delta_j \geq 0$, $j \in N$ be predetermined perturbation parameters. The auxiliary program results from perturbing reduced costs to nonnegative values. More precisely, introduce index set

$$J = \{j \in N \mid \bar{z}_j < \delta_j\}. \quad (13.22)$$

Replacing reduced costs \bar{z}_j in the simplex tableau by

$$\bar{z}'_j = \begin{cases} \delta_j, & \text{if } j \in J, \\ \bar{z}_j, & \text{if } j \in N \setminus J, \end{cases} \quad (13.23)$$

leads to a dual feasible tableau, associated with a perturbed program. As a result, the dual simplex algorithm can get itself started, until optimality of the perturbed program is achieved, or dual unboundedness of it is detected.

Theorem 13.4.1. *If the auxiliary perturbed program is dual unbounded, the original problem is infeasible.*

Proof. The perturbed program has the same constraints as the original problem. Unboundedness of the former implies that there is no feasible solution to former. and hence so to the latter. \square

Assume that optimality of the perturbed program is achieved. It might be well to assume that (3.18) is again the final tableau, associated with basis B . The right-hand side of it is now nonnegative. Consequently, computing $\hat{z}_N = c_N - N^T B^{-1} c_B$ (restoring reduced costs of the original problem) and covering \bar{z}_N with \hat{z}_N leads to a feasible simplex tableau to the original problem.

The overall steps are put into the following algorithm.

Algorithm 13.4.1 (Tableau Phase-I: reduced cost perturbation). Given perturbation parameters $\delta_j \geq 0$, $j \in N$. Initial: simplex tableau of form (3.18). This algorithm finds a feasible simplex tableau to the standard LP problem.

1. Perturb: $\bar{z}_j = \delta_j$, $\forall j \in \{j \in N \mid \bar{z}_j < \delta_j\}$.
2. Call the dual simplex Algorithm 4.4.1.
3. Stop if it terminates at step 3 (infeasible problem).
4. If it terminates at step 2, compute $\bar{z}_N = c_N - N^T B^{-1} c_B$.
5. Stop (feasibility achieved).

Note There exists a perturbation-free variant of the preceding algorithm: in each iteration, one use \bar{z}'_N determined by (13.23) in place of \bar{z}_N for minimum-ratio test, therefore saving computations for restoring reduced costs.

Algorithm 13.4.1 performed very well in preliminary computational experiments (Pan 2000b). A reason seems to be that not all perturbed reduced costs affect the final outcome. Sometimes, reduced costs restored in step 4 of Algorithm 13.4.1 are themselves nonnegative, and hence there is no need for carrying out Phase-II. In fact, it is verified that manipulations in step 2 amount to solving the following problem

$$\begin{aligned} \min \quad & b^T y, \\ \text{s.t.} \quad & A^T y \leq \hat{c}, \end{aligned}$$

where

$$\hat{c}_j = \begin{cases} a_j^T B^{-T} c_B + \delta_j, & \text{if } j \in J, \\ \bar{z}_j, & \text{if } j \in N \setminus J. \end{cases}$$

That is to say, perturbing $\bar{z}_j < \delta_j$ to $\bar{z}_j = \delta_j$ amounts to slackening dual constraint $a_j^T y \leq c_j$ to

$$a_j^T y \leq a_j^T B^{-T} c_B + \delta_j = c_j + (\delta_j - c_j + a_j^T B^{-T} c_B) = c_j + (\delta_j - \bar{z}_j).$$

If slackened constraints are all inactive for dual optimal solutions, or in other words, nonnegative constraints of primal variables, associated with perturbed reduced costs, are active for primal optimal solutions, the primal and dual optimal solutions will not be affected by such perturbations at all (for detailed analysis, see Pan 2000b).

Example 13.4.1. Solve the following problem using Algorithm 13.4.1 in Phase-I:

$$\begin{aligned} \min \quad & f = x_1 + x_2 - 3x_3, \\ \text{s.t.} \quad & -2x_1 - x_2 + 4x_3 + x_5 = -4, \\ & x_1 - 2x_2 + x_3 + x_6 = 5, \\ & -x_1 + 2x_3 + x_4 = -3, \\ & x_j \geq 0, \quad j = 1, \dots, 6. \end{aligned}$$

Answer**Phase-I.**

Initial: There is an available simplex tableau ($B = \{5, 6, 4\}$) to this problem. Perturb negative reduced cost -3 to 0 , so that it becomes dual feasible. As an illustration for implementation, we add the perturbed reduced costs at the bottom ($\delta_3 = 0$).

x_1	x_2	x_3	x_4	x_5	x_6	RHS
-2^*	-1	4		1		-4
1	-2	1			1	5
-1		2	1			-3
1	1	-3				
1	1					

Taking the bottom row as objective row, execute the dual simplex Algorithm 4.4.1. Note that the second (objective) bottom row does not take a part in pivoting.

Iteration 1:

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

x_1	x_2	x_3	x_4	x_5	x_6	RHS
1	$1/2$	-2		$-1/2$		2
	$-5/2$	3		$1/2$	1	3
	$1/2$		1	$-1/2^*$		-1
	$1/2$	-1		$1/2$		-2
	$1/2$	2		$1/2$		-2

Iteration 2:

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

x_1	x_2	x_3	x_4	x_5	x_6	RHS
1		-2	-1			3
	-2	3*	1		1	2
	-1		-2	1		2
	1	-1	1			-3
	1	2	1			-3

Optimality of the perturbed tableau is achieved. Deleting the perturbed (bottom) row gives a feasible simplex tableau of the original problem.

Phase-II.

Iteration 3: call the simplex Algorithm 3.2.1.

Select x_3 column as pivot column and the second row as pivot row. Multiply row 2 by $1/3$, then add $2, 1, -2$ times of row 2 to rows 1,4,5, respectively:

x_1	x_2	x_3	x_4	x_5	x_6	RHS
1	-4/3		-1/3		2/3	13/3
	-2/3	1	1/3		1/3	2/3
	-1		-2	1		2
	1/3		4/3		1/3	-7/3

The reduced costs are all nonnegative, optimality is hence achieved. Basic optimal solution and associated objective value:

$$\bar{x} = (13/3, 0, 2/3, 0, 2, 0)^T, \quad \bar{f} = 7/3.$$

Although the perturbation parameter was set to 0 in the preceding example, positive parameter value should be practically favorable for the sake of anti-degeneracy. It seems to be suitable to use

$$\delta_j \geq \epsilon > 0,$$

where ϵ is the feasibility tolerance ($\epsilon = 10^{-6}$, or so). Computational experiences indicate that the perturbation method is not sensitive to the magnitude of the parameters though exceeding 10^{-1} seems to be inadvisable.