[V]

CHAPTER 3

Degeneracy

In the previous chapter, we discussed what it means when the ratios computed to calculate the leaving variable are all nonpositive (the problem is unbounded). In this chapter, we take up the more delicate issue of what happens when some of the ratios are infinite (i.e., their denominators vanish).

1. Definition of Degeneracy

$\bar{b}_i = 0$

We say that a dictionary is *degenerate* if $\bar{b}_i$ vanishes for some $i \in \mathcal{B}$. A degenerate dictionary could cause difficulties for the simplex method, but it might not. For example, the dictionary we were discussing at the end of the last chapter,

$$
\begin{array}{rcrrrrrl}
\zeta & = & 5 & + & 1 & x_3 & - & 1\ x_1 \\
\hline
x_2 & = & 5 & + & 2 & x_3 & - & 3\ x_1 \\
x_4 & = & 7 & & & & - & 4\ x_1 \\
x_5 & = & 0 & & & & & x_1\ ,
\end{array}
$$

is degenerate, but it was clear that the problem was unbounded and therefore no more pivots were required. Furthermore, had the coefficient of x_3 in the equation for x_2 been -2 instead of 2, then the simplex method would have picked x_2 for the leaving variable and no difficulties would have been encountered.

Problems arise, however, when a degenerate dictionary produces degenerate pivots. We say that a pivot is a *degenerate pivot* if one of the ratios in the calculation of the leaving variable is $+\infty$; i.e., if the numerator is positive and the denominator vanishes. To see what happens, let us look at a few examples.

2. Two Examples of Degenerate Problems

Here is an example of a degenerate dictionary in which the pivot is also degenerate:

$$
\begin{array}{rcrrrrrrr}
\zeta & = & 3 & - & 0.5\ x_1 & + & 2\ x_2 & - & 1.5\ w_1 \\
\hline
x_3 & = & 1 & - & 0.5\ x_1 & & & - & 0.5\ w_1 \\
w_2 & = & 0 & & x_1 & - & x_2 & + & w_1\ .
\end{array}
\tag{3.1}
$$

$\leftarrow 0 \le x_2 \le 0$

R. J. Vanderbei, *Linear Programming*, International Series in Operations Research & Management Science 285,
https://doi.org/10.1007/978-3-030-39415-8_3

For this dictionary, the entering variable is x_2 and the ratios computed to determine the leaving variable are 0 and $+\infty$. Hence, the leaving variable is w_2, and the fact that the ratio is infinite means that as soon as x_2 is increased from zero to a positive value, w_2 will go negative. Therefore, x_2 cannot really increase. Nonetheless, it can be reclassified from nonbasic to basic (with w_2 going the other way). Let us look at the result of this degenerate pivot:

$$
\begin{array}{rcrrrrrrrr}
\zeta & = & 3 & + & 1.5 & x_1 & - & 2 & w_2 & + & 0.5 & w_1 \\
\hline
x_3 & = & 1 & - & 0.5 & x_1 & & & & - & 0.5 & w_1 \\
x_2 & = & & & & x_1 & - & & w_2 & + & & w_1 .
\end{array} \tag{3.2}
$$

$x_1 \leq 2$

Note that $\bar{\zeta}$ remains unchanged at 3. Hence, this degenerate pivot has not produced any increase in the objective function value. Furthermore, the values of the variables have not even changed: both before and after this degenerate pivot, they are

$$(x_1, x_2, x_3, w_1, w_2) = (0, 0, 1, 0, 0).$$

But we are now representing this solution in a new way, and perhaps the next pivot will make an improvement, or if not the next pivot perhaps the one after that. Let's see what happens for the problem at hand. The entering variable for the next iteration is x_1 and the leaving variable is x_3, producing a nondegenerate pivot that leads to

$$
\begin{array}{rcrrrrrrrr}
\zeta & = & 6 & - & 3 & x_3 & - & 2 & w_2 & - & w_1 \\
\hline
x_1 & = & 2 & - & 2 & x_3 & & & & - & w_1 \\
x_2 & = & 2 & - & 2 & x_3 & - & & w_2 & & .
\end{array}
$$

These two pivots illustrate what typically happens. When one reaches a degenerate dictionary, it is usual that one or more of the subsequent pivots will be degenerate but that eventually a nondegenerate pivot will lead us away from these degenerate dictionaries. While it is typical for some pivot to "break away" from the degeneracy, the real danger is that the simplex method will make a sequence of degenerate pivots and eventually return to a dictionary that has appeared before, in which case the simplex method enters an infinite loop and never finds an optimal solution. This behavior is called *cycling*.

Unfortunately, under certain pivoting rules, cycling is possible. In fact, it is possible even when using one of the most popular pivoting rules:

- Choose the entering variable as the one with the largest coefficient in the ζ-row of the dictionary.

- When two or more variables compete for leaving the basis, pick an x-variable over a slack variable and, if there is a choice, use the variable with the smallest subscript. In other words, reading left to right, pick the first leaving-variable candidate from the list:

$$x_1,\ x_2,\ \ldots,\ x_n,\ w_1,\ w_2,\ \ldots,\ w_m.$$

However, it is hard to find examples of cycling in which m and n are small. In fact, it has been shown that if a problem has an optimal solution but cycles off-optimum, then the problem must involve dictionaries with at least four (non-slack) variables and two constraints. Here is an example that cycles:

0

$$\begin{array}{rcl}
\zeta &=& + x_1 - 2x_2 - 2x_4 \\ \hline
w_1 &=& - 0.5x_1 + 3.5x_2 + 2x_3 - 4x_4 \\
w_2 &=& - 0.5x_1 + x_2 + 0.5x_3 - 0.5x_4 \\
w_3 &=& 1 - x_1 .
\end{array}$$

$x_1 \leq 0$
$x_1 \leq 0$
$x_1 \leq 1$

And here is the sequence of dictionaries the above pivot rules produce. For the first pivot, x_1 enters and w_1 leaves bringing us to:

1

$$\begin{array}{rcl}
\zeta &=& - 2w_1 + 5x_2 + 4x_3 - 10x_4 \\ \hline
x_1 &=& - 2w_1 + 7x_2 + 4x_3 - 8x_4 \\
w_2 &=& w_1 - 2.5x_2 - 1.5x_3 + 3.5x_4 \\
w_3 &=& 1 + 2w_1 - 7x_2 - 4x_3 + 8x_4 .
\end{array}$$

$x_2 \leq 0$

For the second iteration, x_2 enters and w_2 leaves bringing us to:

2

$$\begin{array}{rcl}
\zeta &=&   - 2w_2 + x_3 - 3x_4 \\ \hline
x_1 &=&   0.8w_1 - 2.8w_2 - 0.2x_3 + 1.8x_4 \\
x_2 &=&   0.4w_1 - 0.4w_2 - 0.6x_3 + 1.4x_4 \\
w_3 &=& 1 - 0.8w_1 + 2.8w_2 + 0.2x_3 - 1.8x_4 .
\end{array}$$

$x_3 \leq 0$

For the third iteration, x_3 enters and x_1 leaves:

3

$$\begin{array}{rcl}
\zeta &=& + 4w_1 - 16w_2 - 5x_1 + 6x_4 \\ \hline
x_3 &=&   4w_1 - 14w_2 - 5x_1 + 9x_4 \\
x_2 &=& - 2w_1 + 8w_2 + 3x_1 - 4x_4 \\
w_3 &=& 1 - x_1 .
\end{array}$$

$x_4 \leq 0$

For the fourth iteration, x_4 enters and x_2 leaves:

4

$$
\begin{array}{rcrrrrrrrr}
\zeta & = & & + 1 w_1 & - & 4 w_2 & - & 0.5 x_1 & - & 1.5 x_2 \\
\hline
x_3 & = & & - 0.5 w_1 & + & 4 w_2 & + & 1.75 x_1 & - & 2.25 x_2 \\
x_4 & = & & - 0.5 w_1 & + & 2 w_2 & + & 0.75 x_1 & - & 0.25 x_2 \\
w_3 & = & 1 & & & & - & x_1 & & .
\end{array}
$$

In the fifth iteration, w_1 enters and x_3 leaves:

5

$$
\begin{array}{rcrrrrrrrr}
\zeta & = & & - 2 x_3 & + & 4 w_2 & + & 3 x_1 & - & 6 x_2 \\
\hline
w_1 & = & & - 2 x_3 & + & 8 w_2 & + & 3.5 x_1 & - & 4.5 x_2 \\
x_4 & = & & x_3 & - & 2 w_2 & - & x_1 & + & 2 x_2 \\
w_3 & = & 1 & & & & - & x_1 & & .
\end{array}
$$

Lastly, for the sixth iteration, w_2 enters and x_4 leaves:

6

$$
\begin{array}{rcrrrrrrrr}
\zeta & = & & & - & 2 x_4 & + & 1 x_1 & - & 2 x_2 \\
\hline
w_1 & = & & 2 x_3 & - & 4 x_4 & - & 0.5 x_1 & + & 3.5 x_2 \\
w_2 & = & & 0.5 x_3 & - & 0.5 x_4 & - & 0.5 x_1 & + & x_2 \\
w_3 & = & 1 & & & & - & x_1 & & .
\end{array}
$$

Note that we have come back to the original dictionary, and so from here on the simplex method simply cycles through these six dictionaries and never makes any further progress toward an optimal solution. As bad as cycling is, the following theorem tells us that nothing worse can happen:

THEOREM 3.1. *If the simplex method fails to terminate, then it must cycle.*

PROOF. A dictionary is completely determined by specifying which variables are basic and which are nonbasic. There are only

$$\binom{n+m}{m} = \frac{(n+m)!}{n!m!}$$

different possibilities. This number is big, but it is finite. If the simplex method fails to terminate, it must visit some of these dictionaries more than once. Hence, the algorithm cycles. $\square$

Note that, if the simplex method cycles, then all the pivots within the cycle must be degenerate. This is easy to see, since the objective function value never decreases.

Hence, it follows that all the pivots within the cycle must have the same objective function value, i.e., all of the these pivots must be degenerate.

In practice, degeneracy is common because a zero right-hand side value crops up frequently in real-world problems. Cycling is not as common, but it can happen and therefore must be addressed. Computer implementations of the simplex method in which numbers are represented as integers or as simple rational numbers are at risk of cycling and one of the techniques described in the following sections must be used to avoid the problem. But, most implementations of the simplex method are written with floating point numbers (the computer approximation to a full set of real numbers). With floating point computation there is inevitable round-off error. Hence, a zero appearing as a right-hand side value generally shows up not as an exact zero but rather as a very small number. The result is that the dictionary appears to be slightly off from actually being degenerate and therefore cycling is usually avoided.

3. The Perturbation/Lexicographic Method

As we have seen, there is not just one algorithm called the simplex method. Instead, the simplex method is a whole family of related algorithms from which we can pick a specific instance by specifying what we have been referring to as pivoting rules. We have also seen that, using a very natural pivoting rule, the simplex method can fail to converge to an optimal solution by occasionally cycling indefinitely through a sequence of degenerate pivots associated with a nonoptimal solution.

So this raises a natural question: are there pivoting rules for which the simplex method will, with certainty, either reach an optimal solution or prove that no such solution exists? The answer to this question is yes, and we shall present two choices of such pivoting rules.

The first method is based on the observation that degeneracy is sort of an accident. That is, a dictionary is degenerate if one or more of the b_i's vanish. Our examples have generally used small integers for the data, and in this case it does not seem too surprising that sometimes cancellations occur and we end up with a degenerate dictionary. But each right-hand side could in fact be any real number, and in the world of real numbers the occurrence of any specific number, such as zero, seems to be quite unlikely. So how about perturbing a given problem by adding small random perturbations independently to each of the right-hand sides? If these perturbations are small enough, we can think of them as insignificant and hence not really changing the problem. If they are chosen independently, then the probability of an exact cancellation is zero.

Such random perturbation schemes are used in some implementations, but what we have in mind as we discuss perturbation methods is something a little bit different. Instead of using independent identically distributed random perturbations, let us consider using a fixed perturbation for each constraint, with the perturbation getting much smaller on each succeeding constraint. Indeed, we introduce a small positive number

ϵ_1 for the first constraint and then a much smaller positive number ϵ_2 for the second constraint, etc. We write this as

$$0 < \epsilon_m \ll \cdots \ll \epsilon_2 \ll \epsilon_1 \ll \text{ all other data.}$$

The idea is that each ϵ_i acts on an entirely different scale from all the other ϵ_i's and the data for the problem. What we mean by this is that no linear combination of the ϵ_i's using coefficients that might arise in the course of the simplex method can ever produce a number whose size is of the same order as the data in the problem. Similarly, each of the "lower down" ϵ_i's can never "escalate" to a higher level. Hence, cancellations can only occur on a given scale. Of course, this complete isolation of scales can never be truly achieved in the real numbers, so instead of actually introducing specific values for the ϵ_i's, we simply treat them as abstract symbols having these scale properties.

To illustrate what we mean, let us look at a specific example. Consider the following degenerate dictionary:

$$\begin{array}{rcrrrr}
\zeta & = & & +\ 6\ x_1 & +\ 4\ x_2 \\
\hline
w_1 & = & & +\ 9\ x_1 & +\ 4\ x_2 \\
w_2 & = & & -\ 4\ x_1 & -\ 2\ x_2 \\
w_3 & = & 1 & & -\ \ x_2 & .
\end{array}$$

The first step is to introduce symbolic parameters

$$0 < \epsilon_3 \ll \epsilon_2 \ll \epsilon_1$$

to get a perturbed problem:

$$\begin{array}{rcllllrr}
\zeta & = & & & & & +\ 6\ x_1 & +\ 4\ x_2 \\
\hline
w_1 & = & 0 & +\ \epsilon_1 & & & +\ 9\ x_1 & +\ 4\ x_2 \\
w_2 & = & 0 & & +\ \epsilon_2 & & -\ 4\ x_1 & -\ 2\ x_2 \\
w_3 & = & 1 & & & +\ \epsilon_3 & & -\ \ x_2 & .
\end{array}$$

This dictionary is not degenerate. The entering variable is x_1 and the leaving variable is unambiguously w_2. The next dictionary is

$$\begin{array}{rcllllrr}
\zeta & = & & & 1.5\ \epsilon_2 & & -\ 1.5\ w_2 & +\ 1\ x_2 \\
\hline
w_1 & = & 0 & +\ \epsilon_1 & +\ 2.25\ \epsilon_2 & & -\ 2.25\ w_2 & -\ 0.5\ x_2 \\
x_1 & = & 0 & & +\ 0.25\ \epsilon_2 & & -\ 0.25\ w_2 & -\ 0.5\ x_2 \\
w_3 & = & 1 & & & +\ \epsilon_3 & & -\ \ x_2 & .
\end{array}$$

For the next pivot, the entering variable is x_2 and, using the fact that $\epsilon_2 \ll \epsilon_1$, we see that the leaving variable is x_1. The new dictionary is

$$\begin{array}{rcrrrrrrr}
\zeta & = & & & 2\,\epsilon_2 & & -\,2\,w_2 & -\,2\,x_1 \\
\hline
w_1 & = & 0 & +\,\epsilon_1 & +\,2\,\epsilon_2 & & -\,2\,w_2 & +\,x_1 \\
x_2 & = & 0 & & +\,0.5\,\epsilon_2 & & -\,0.5\,w_2 & -\,2\,x_1 \\
w_3 & = & 1 & & -\,0.5\,\epsilon_2 & +\,\epsilon_3 & +\,0.5\,w_2 & +\,2\,x_1\,.
\end{array}$$

This last dictionary is optimal. At this point, we simply drop the symbolic ϵ_i parameters and get an optimal dictionary for the unperturbed problem:

$$\begin{array}{rcrrr}
\zeta & = & & -\,2\,w_2 & -\,2\,x_1 \\
\hline
w_1 & = & 0 & -\,2\,w_2 & +\,x_1 \\
x_2 & = & 0 & -\,0.5\,w_2 & -\,2\,x_1 \\
w_3 & = & 1 & +\,0.5\,w_2 & +\,2\,x_1\,.
\end{array}$$

When treating the ϵ_i's as symbols, the method is called the *lexicographic method*. Note that the lexicographic method does not affect the choice of entering variable but does amount to a precise prescription for the choice of leaving variable.

It turns out that the lexicographic method produces a variant of the simplex method that never cycles:

THEOREM 3.2. *The simplex method always terminates provided that the leaving variable is selected by the lexicographic rule.*

PROOF. It suffices to show that no degenerate dictionary is ever produced. As we have discussed before, the ϵ_i's operate on different scales and hence cannot cancel with each other. Therefore, we can think of the ϵ_i's as a collection of independent variables. Extracting the ϵ terms from the first dictionary, we see that we start with the following pattern:

$$\begin{array}{cccc}
\epsilon_1 & & & \\
& \epsilon_2 & & \\
& & \ddots & \\
& & & \epsilon_m\,.
\end{array}$$

And, after several pivots, the ϵ terms will form a system of linear combinations, say,

$$\begin{array}{ccccc} r_{11}\epsilon_1 & + & r_{12}\epsilon_2 & \dots & + & r_{1m}\epsilon_m \\ r_{21}\epsilon_1 & + & r_{22}\epsilon_2 & \dots & + & r_{2m}\epsilon_m \\ \vdots & & \vdots & \ddots & & \vdots \\ r_{m1}\epsilon_1 & + & r_{m2}\epsilon_2 & \dots & + & r_{mm}\epsilon_m. \end{array}$$

Since this system of linear combinations is obtained from the original system by pivot operations and, since pivot operations are reversible, it follows that the rank of the two systems must be the same. Since the original system had rank m, we see that every subsequent system must have rank m. This means that there must be at least one nonzero r_{ij} in every row i, which of course implies that none of the rows can be degenerate. Hence, no dictionary can be degenerate. □

4. Bland's Rule

The second pivoting rule we consider is called *Bland's rule*. It stipulates that both the entering and the leaving variable be selected from their respective sets of choices by choosing the variable x_k with the smallest index k.

THEOREM 3.3. *The simplex method always terminates provided that both the entering and the leaving variable are chosen according to Bland's rule.*

The proof may look rather involved, but the reader who spends the time to understand it will find the underlying elegance most rewarding.

PROOF. It suffices to show that such a variant of the simplex method never cycles. We prove this by assuming that cycling does occur and then showing that this assumption leads to a contradiction. So let us assume that cycling does occur. Without loss of generality, we may assume that it happens from the beginning. Let $D_0, D_1, \dots, D_{k-1}$ denote the dictionaries through which the method cycles. That is, the simplex method produces the following sequence of dictionaries:

$$D_0, D_1, \dots, D_{k-1}, D_0, D_1, \dots.$$

We say that a variable is fickle if it is in some basis and not in some other basis. Let x_t be the fickle variable having the largest index and let D denote a dictionary in $D_0, D_1, \dots, D_{k-1}$ in which x_t leaves the basis. Again, without loss of generality we may assume that $D = D_0$. Let x_s denote the corresponding entering variable. Suppose that D is recorded as follows:

$$\begin{aligned} \zeta &= v + \sum_{j \in \mathcal{N}} c_j x_j \\ x_i &= b_i - \sum_{j \in \mathcal{N}} a_{ij} x_j \qquad i \in \mathcal{B}. \end{aligned}$$

[C]

whenever $j \in N$, gives the objective function a numerical value of at most z^*; hence the current solution is optimal. On the other hand, if there is more than one candidate for entering the basis, then any of these candidates may serve. (In hand calculations involving small problems, it is customary to choose the candidate x_j that has the largest coefficient $\bar{c}_j$. In most computer implementations of the simplex method, however, this practice is abandoned. More on this subject in Chapter 7.)

Finding the leaving variable. The leaving variable is *that basic variable whose nonnegativity imposes the most stringent upper bound on the increase of the entering variable.* Again, this rule is ambiguous in the sense that it may provide more than one candidate for leaving the basis, or no candidate at all. The latter alternative is illustrated on the dictionary

$$\begin{array}{rcl} x_2 &=& 5 + 2x_3 - x_4 - 3x_1 \\ x_5 &=& 7 - 3x_4 - 4x_1 \\ \hline z &=& 5 + x_3 - x_4 - x_1. \end{array}$$

The entering variable is x_3, but neither of the two basic variables x_2, x_5 imposes an upper bound on its increase. Therefore, we can make x_3 as large as we wish (maintaining $x_1 = x_4 = 0$) and still retain feasibility: setting $x_3 = t$ for any positive t, we obtain a feasible solution with $x_1 = 0$, $x_2 = 5 + 2t$, $x_4 = 0$, $x_5 = 7$, and $z = 5 + t$. Since t can be made arbitrarily large, z can be made arbitrarily large. We conclude that the problem is *unbounded*: for every number M, there is a feasible solution $x_1, x_2, \ldots, x_5$ such that $x_3 - x_4 - x_1 > M$. The same conclusion can be reached in general: if there is no candidate for leaving the basis, then we can make the value of the entering variable, and therefore also the value of the objective function, as large as we wish. In that case, the problem is unbounded. On the other hand, if there is more than one candidate for leaving the basis, then any of these candidates may serve. Once the entering and leaving variables have been selected, pivoting is a straightforward matter.

Degeneracy. The presence of more than one candidate for leaving the basis has interesting consequences. For illustration, consider the dictionary

$$\begin{array}{rcl} x_4 &=& 1 - 2x_3 \\ x_5 &=& 3 - 2x_1 + 4x_2 - 6x_3 \\ x_6 &=& 2 + x_1 - 3x_2 - 4x_3 \\ \hline z &=& 2x_1 - x_2 + 8x_3. \end{array}$$

Having chosen x_3 to enter the basis, we find that each of the three basic variables x_4, x_5, x_6 limits the increase of x_3 to $\frac{1}{2}$. Hence each of these three variables is a candidate for leaving the basis. We arbitrarily choose x_4. Pivoting as usual, we obtain the dictionary

$$
\begin{array}{rcrrrr}
x_3 & = & 0.5 & & & -\ 0.5x_4 \\
x_5 & = & & -\ 2x_1 & +\ 4x_2 & +\ \ 3x_4 \\
x_6 & = & & x_1 & -\ 3x_2 & +\ \ 2x_4 \\
\hline
z & = & 4 & +\ 2x_1 & -\ \ x_2 & -\ \ 4x_4.
\end{array}
$$

This dictionary differs from all the dictionaries we have encountered so far in one important respect: along with the nonbasic variables, the basic variables x_5 and x_6 have value zero in the associated solution. Basic solutions with one or more basic variables at zero are called *degenerate*.

Although harmless in its own right, degeneracy may have annoying side effects. These are illustrated on the next iteration in our example. There, x_1 enters the basis and x_5 leaves; because of degeneracy, the constraint $x_5 \geq 0$ limits the increment of x_1 to *zero*. Hence the value of x_1 will remain unchanged, and so will the values of the remaining variables and the value of the objective function z. This is annoying, for the motivation behind the simplex method is a desire to increase the value of z in each iteration. In this particular iteration, that desire remains unfulfilled: pivoting changes the dictionary into

$$
\begin{array}{rcrrrr}
x_1 & = & & 2x_2 & +\ 1.5x_4 & -\ 0.5x_5 \\
x_3 & = & 0.5 & & -\ 0.5x_4 & \\
x_6 & = & & -\ \ x_2 & +\ 3.5x_4 & -\ 0.5x_5 \\
\hline
z & = & 4 & +\ 3x_2 & -\ \ \ x_4 & -\ \ \ x_5
\end{array}
$$

but it does not affect the associated solution at all. Simplex iterations that do not change the basic solution are called *degenerate*. (As the reader may verify, the next iteration is degenerate again, but the one after that turns out to be nondegenerate and brings us to the optimal solution.)

In a sense, degeneracy is something of an accident: a basic variable may vanish only if the results of successive pivot operations just happen to cancel each other out. And yet degeneracy abounds in LP problems arising from practical applications. It has been said that nearly all such problems yield degenerate basic feasible solutions at some stage of the simplex method. Whenever that happens, the simplex method may stall by going through a few (and sometimes quite a few) degenerate iterations in a row. Typically, such a block of degenerate iterations ends with a breakthrough represented by a nondegenerate iteration; an example of the atypical case is presented next.

Termination: Cycling

Can the simplex method go through an endless sequence of iterations without ever finding an optimal solution? Yes, it can. To justify this claim, let us consider the initial dictionary

$$
\begin{aligned}
x_5 &= \quad -0.5x_1 + 5.5x_2 + 2.5x_3 - 9x_4 \\
x_6 &= \quad -0.5x_1 + 1.5x_2 + 0.5x_3 - x_4 \\
x_7 &= 1 - x_1 \\
\hline
z &= \quad 10x_1 - 57x_2 - 9x_3 - 24x_4
\end{aligned}
$$

and let us agree on the following:

(i) The entering variable will always be the nonbasic variable that has the largest coefficient in the z-row of the dictionary.

(ii) If two or more basic variables compete for leaving the basis, then the candidate with the smallest subscript will be made to leave.

Now the sequence of dictionaries constructed in the first six iterations goes as follows.

After the first iteration:

$$
\begin{aligned}
x_1 &= \quad 11x_2 + 5x_3 - 18x_4 - 2x_5 \\
x_6 &= \quad -4x_2 - 2x_3 + 8x_4 + x_5 \\
x_7 &= 1 - 11x_2 - 5x_3 + 18x_4 + 2x_5 \\
\hline
z &= \quad 53x_2 + 41x_3 - 204x_4 - 20x_5.
\end{aligned}
$$

After the second iteration:

$$
\begin{aligned}
x_2 &= \quad -0.5x_3 + 2x_4 + 0.25x_5 - 0.25x_6 \\
x_1 &= \quad -0.5x_3 + 4x_4 + 0.75x_5 - 2.75x_6 \\
x_7 &= 1 + 0.5x_3 - 4x_4 - 0.75x_5 - 13.25x_6 \\
\hline
z &= \quad 14.5x_3 - 98x_4 - 6.75x_5 - 13.25x_6.
\end{aligned}
$$

After the third iteration:

$$
\begin{aligned}
x_3 &= \quad 8x_4 + 1.5x_5 - 5.5x_6 - 2x_1 \\
x_2 &= \quad -2x_4 - 0.5x_5 + 2.5x_6 + x_1 \\
x_7 &= 1 \qquad\qquad\qquad\qquad - x_1 \\
\hline
z &= \quad 18x_4 + 15x_5 - 93x_6 - 29x_1.
\end{aligned}
$$

After the fourth iteration:

$$
\begin{aligned}
x_4 &= \quad -0.25x_5 + 1.25x_6 + 0.5x_1 - 0.5x_2 \\
x_3 &= \quad -0.5x_5 + 4.5x_6 + 2x_1 - 4x_2 \\
x_7 &= 1 \qquad\qquad\qquad - x_1 \\
\hline
z &= \quad 10.5x_5 - 70.5x_6 - 20x_1 - 9x_2.
\end{aligned}
$$

After the fifth iteration:

$$
\begin{array}{rcl}
x_5 &=& 9x_6 + 4x_1 - 8x_2 - 2x_3 \\
x_4 &=& -x_6 - 0.5x_1 + 1.5x_2 + 0.5x_3 \\
x_7 &=& 1 - x_1 \\
\hline
z &=& 24x_6 + 22x_1 - 93x_2 - 21x_3.
\end{array}
$$

After the sixth iteration:

$$
\begin{array}{rcl}
x_6 &=& -0.5x_1 + 1.5x_2 + 0.5x_3 - x_4 \\
x_5 &=& -0.5x_1 + 5.5x_2 + 2.5x_3 - 9x_4 \\
x_7 &=& 1 - x_1 \\
\hline
z &=& 10x_1 - 57x_2 - 9x_3 - 24x_4.
\end{array}
$$

Since the dictionary constructed after the sixth iteration is identical with the initial dictionary, the method will go through the same six iterations again and again without ever finding the optimal solution (which, as we shall see later, has $z = 1$). This phenomenon is known as cycling. More precisely, we say that the simplex method *cycles* if one dictionary appears in two different iterations (and so the sequence of iterations leading from the dictionary to itself can be repeated over and over without end). Note that cycling can occur only in the presence of degeneracy: since the value of the objective function increases with each nondegenerate iteration and remains unchanged after each degenerate one, all the iterations in the sequence leading from a dictionary to itself must be degenerate. Cycling is one reason why the simplex method may fail to terminate; the following theorem shows that it is the only reason.

THEOREM 3.1. If the simplex method fails to terminate, then it must cycle.

PROOF. To begin, note that there are only finitely many ways of choosing m basic variables from all the $n + m$ variables. Thus, if the simplex method fails to terminate, then some basis must appear in two different iterations. Now it only remains to be proved that any two dictionaries with the same basis must be identical. (This fact becomes trivial as soon as one describes dictionaries in terms of matrices, as we shall do in Chapter 7. Nevertheless, we can and shall present an easy proof from scratch right now.) Consider two dictionaries