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

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,

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

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's 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:

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

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

The original version of this chapter was revised. An erratum to this chapter can be found at DOI 10.1007/978-1-4614-7630-6_26

R.J. Vanderbei, *Linear Programming*, International Series in Operations Research & Management Science 196, DOI 10.1007/978-1-4614-7630-6_3,

value, w_2 will go negative. Therefore, x_2 can't really increase. Nonetheless, it can be reclassified from nonbasic to basic (with w_2 going the other way). Let's look at the result of this degenerate pivot:

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

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 haven't 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

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

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:

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

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:

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

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

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

For the third iteration, x_3 enters and x_1 leaves:

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

For the fourth iteration, x_4 enters and x_2 leaves:

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

In the fifth iteration, w_1 enters and x_3 leaves:

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

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

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

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

$$\left(\begin{array}{c} n+m \\ m \end{array} \right) = \frac{(n+m)!}{n!m!}$$

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. □

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 $\bar{b}_i$'s vanish. Our examples have generally used small integers for the data, and in this case it doesn't 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's look at a specific example. Consider the following degenerate dictionary:

$$\begin{array}{rl}
\zeta = & 6\,x_1 + 4\,x_2 \\
\hline
w_1 = 0 + & 9\,x_1 + 4\,x_2 \\
w_2 = 0 - & 4\,x_1 - 2\,x_2 \\
w_3 = 1 & - \quad x_2.
\end{array}$$

The first step is to introduce symbolic parameters

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

to get a perturbed problem:

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

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

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

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

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

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:

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

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've discussed before, the ϵ_i's operate on different scales and hence can't 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:

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

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

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

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's assume that cycling does occur. Without loss of generality, we may assume that it happens from the beginning. Let $D_0, D_1, \ldots, D_{k-1}$ denote the dictionaries through which the method cycles. That is, the simplex method produces the following sequence of dictionaries:

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

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 $D_0, D_1, \ldots, D_{k-1}$ 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:

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

Since x_s is the entering variable and x_t is the leaving variable, we have that $s \in \mathcal{N}$ and $t \in \mathcal{B}$.

Now let D^* be a dictionary in $D_1, D_2, \ldots, D_{k-1}$ in which x_t enters the basis. Suppose that D^* is recorded as follows:

$$\begin{aligned} \zeta &= v^* + \sum_{j \in \mathcal{N}^*} c_j^* x_j \\ x_i &= b_i^* - \sum_{j \in \mathcal{N}^*} a_{ij}^* x_j \qquad i \in \mathcal{B}^*. \end{aligned} \tag{3.3}$$

Since all the dictionaries are degenerate, we have that $v^* = v$, and therefore we can write the objective function in (3.3) as

$$\zeta = v + \sum_{j=1}^{n+m} c_j^* x_j, \tag{3.4}$$

where we have extended the notation c_j^* to all variables (both original and slack) by setting $c_j^* = 0$ for $j \in \mathcal{B}^*$.

Ignoring for the moment the possibility that some variables could go negative, consider the solutions obtained by letting x_s increase while holding all other variables in $\mathcal{N}$ at zero:

$$\begin{aligned} x_s &= y, \\ x_j &= 0, && j \in \mathcal{N} \setminus \{s\}, \\ x_i &= b_i - a_{is}y, && i \in \mathcal{B}. \end{aligned}$$

The objective function at this point is given by

$$\zeta = v + c_s y.$$

However, using (3.4), we see that it is also given by

$$\zeta = v + c_s^* y + \sum_{i \in \mathcal{B}} c_i^* (b_i - a_{is} y).$$

Equating these two expressions for ζ, we see that

$$\left(c_s - c_s^* + \sum_{i \in \mathcal{B}} c_i^* a_{is} \right) y = \sum_{i \in \mathcal{B}} c_i^* b_i.$$

Since this equation must be an identity for every y, it follows that the coefficient multiplying y must vanish (as must the right-hand side):

$$c_s - c_s^* + \sum_{i \in \mathcal{B}} c_i^* a_{is} = 0.$$

Now, the fact that x_s is the entering variable in D implies that

$$c_s > 0.$$

Recall that x_t is the fickle variable with the largest index. Since x_s is also fickle, we see that $s < t$. Since x_s is not the entering variable in D^* (as x_t is), we see that

$$c_s^* \leq 0.$$

From these last three displayed equations, we get

$$\sum_{i \in \mathcal{B}} c_i^* a_{is} < 0.$$

Hence, there must exist an index $r \in \mathcal{B}$ for which

$$c_r^* a_{rs} < 0. \tag{3.5}$$

Consequently, $c_r^* \neq 0$ and $r \in \mathcal{N}^*$. Hence, x_r is fickle and therefore $r \leq t$. In fact, $r < t$, since $c_t^* a_{ts} > 0$. To see that this product is positive, note that both its factors are positive: c_t^* is positive, since x_t is the entering variable in D^*, and a_{ts} is positive, since x_t is the leaving variable in D.

The fact that $r < t$ implies that $c_r^* \leq 0$ (otherwise, according to the smallest index criteria, r would be the entering variable for D^*). Hence, (3.5) implies that

$$a_{rs} > 0.$$

Now, since each of the dictionaries in the cycle describe the same solution, it follows that every fickle variable is zero in all these dictionaries (since it is clearly zero in

a dictionary in which it is nonbasic). In particular, $x_r = 0$. But in D, x_r is basic. Hence,

$$b_r = 0.$$

These last two displayed equations imply that x_r was a candidate to be the leaving variable in D, and since $r < t$, it should have been chosen over x_t. This is the contradiction we have been looking for. □

5. Fundamental Theorem of Linear Programming

Now that we have a Phase I algorithm and a variant of the simplex method that is guaranteed to terminate, we can summarize the main points of this chapter in the following theorem:

THEOREM 3.4. *For an arbitrary linear program in standard form, the following statements are true:*

(1) *If there is no optimal solution, then the problem is either infeasible or unbounded.*
(2) *If a feasible solution exists, then a basic feasible solution exists.*
(3) *If an optimal solution exists, then a basic optimal solution exists.*

PROOF. The Phase I algorithm either proves that the problem is infeasible or produces a basic feasible solution. The Phase II algorithm either discovers that the problem is unbounded or finds a basic optimal solution. These statements depend, of course, on applying a variant of the simplex method that does not cycle, which we now know to exist. □

6. Geometry

As we saw in the previous chapter, the set of feasible solutions for a problem in two dimensions is the intersection of a number of halfplanes, i.e., a polygon. In three dimensions, the situation is similar. Consider, for example, the following problem:

$$\begin{array}{lrcl} \text{maximize} & x_1 + 2x_2 + 3x_3 & & \\ \text{subject to} & x_1 \qquad + 2x_3 & \leq & 3 \\ & x_2 + 2x_3 & \leq & 2 \\ & x_1,\ x_2,\ x_3 & \geq & 0\,. \end{array} \tag{3.6}$$

The set of points satisfying $x_1 + 2x_3 = 3$ is a plane. The inequality $x_1 + 2x_3 \leq 3$ therefore consists of all points on one side of this plane; that is, it is a *halfspace*. The same is true for each of the other four inequalities. The feasible set consists of those points in space that satisfy all five inequalities, i.e., those points lying in the intersection of these halfspaces. This set is the *polyhedron* shown in Figure 3.1. This polyhedron is bordered by five *facets*, each facet being a portion of one of the planes that was defined by replacing a constraint inequality with an equation. For example, the "front" facet in the figure is a portion of the plane $x_1 + 2x_3 = 3$.

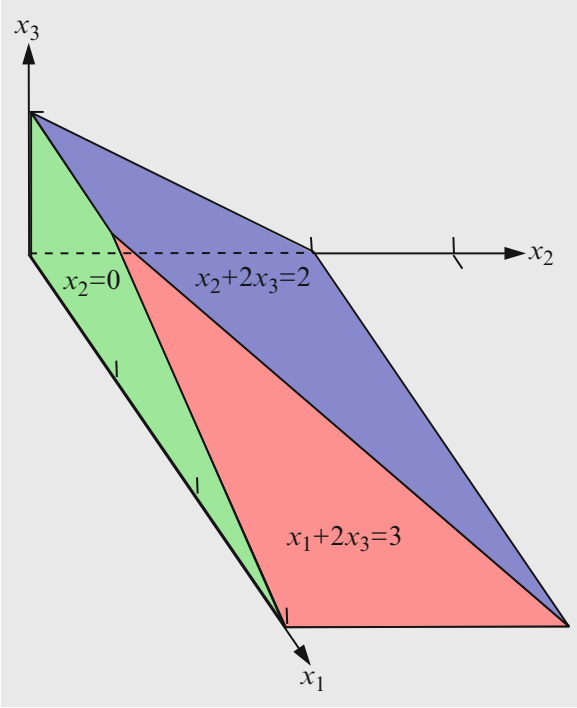

FIGURE 3.1. The set of feasible solutions for the problem given by (3.6).

The facets acquire a particularly simple description if we introduce slack variables into the problem:

$$\begin{aligned} w_1 &= 3 - x_1 \qquad - 2x_3 \\ w_2 &= 2 \qquad - x_2 - 2x_3 \ . \end{aligned}$$

Indeed, each facet corresponds precisely to some variable (either original or slack) vanishing. For instance, the front facet in the figure corresponds to $w_1 = 0$ whereas the "left" facet corresponds to $x_2 = 0$.

The correspondences can be continued. Indeed, each *edge* of the polyhedron corresponds to a pair of variables vanishing. For example, the edge lying at the interface of the left and the front facets in the figure corresponds to both $w_1 = 0$ and $x_2 = 0$.

Going further yet, each *vertex* of the polyhedron corresponds to three variables vanishing. For instance, the vertex whose coordinates are $(1, 0, 1)$ corresponds to $w_1 = 0$, $x_2 = 0$, and $w_2 = 0$.

Now, let's think about applying the simplex method to this problem. Every basic feasible solution involves two basic variables and three nonbasic variables. Furthermore, the three nonbasic variables are, by definition, zero in the basic feasible solution. Therefore, for this example, the basic feasible solutions stand in

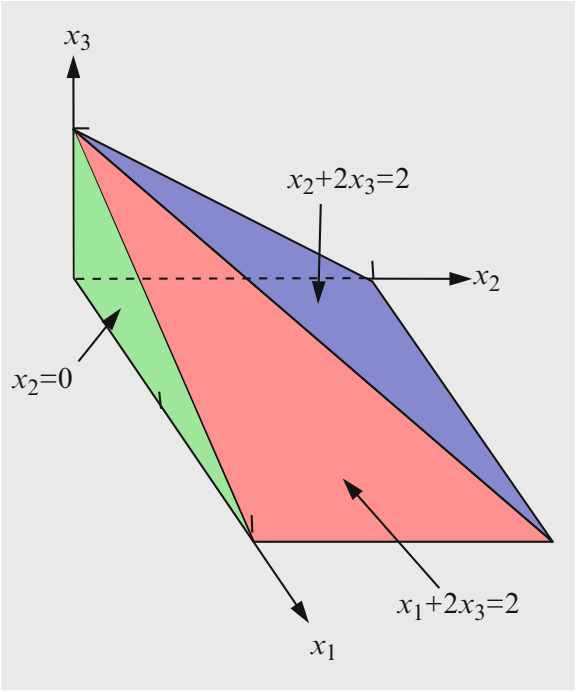

FIGURE 3.2. The set of feasible solutions for the (degenerate) problem given by (3.7).

one-to-one correspondence with the vertices of the polyhedron. In fact, applying the simplex method to this problem, one discovers that the sequence of vertices visited by the algorithm is

$$(0,0,0) \quad \longrightarrow \quad (0,0,1) \quad \longrightarrow \quad (1,0,1) \quad \longrightarrow \quad (3,2,0).$$

The example we've been considering has the nice property that every vertex is formed by the intersection of exactly three of the facets. But consider now the following problem:

$$\begin{array}{lrcl} \text{maximize} & x_1 + 2x_2 + 3x_3 & & \\ \text{subject to} & x_1 \qquad + 2x_3 & \leq & 2 \\ & x_2 + 2x_3 & \leq & 2 \\ & x_1,\ x_2,\ x_3 & \geq & 0\ . \end{array} \tag{3.7}$$

Algebraically, the only difference between this problem and the previous one is that the right-hand side of the first inequality is now a 2 instead of a 3. But look at the polyhedron of feasible solutions shown in Figure 3.2. The vertex $(0,0,1)$ is at the intersection of four of the facets, not three as one would "normally" expect. This vertex does not correspond to one basic feasible solution; rather, there are four degenerate basic feasible solutions, each representing it. We've seen two of them before. Indeed, dictionaries (3.1) and (3.2) correspond to two of these degenerate

dictionaries (in fact, dictionary (3.1) is the dictionary one obtains after one pivot of the simplex method applied to problem (3.7)).

We end by considering the geometric effect of the perturbation method for resolving degeneracy. By perturbing the right-hand sides, one moves the planes that determine the facets. If the moves are random or chosen with vastly different magnitudes (all small), then one would expect that each vertex in the perturbed problem would be determined by exactly three planes. That is, degenerate vertices from the original problem get split into multiple nearby vertices in the perturbed problem. For example, problem (3.6) can be thought of as a perturbation of degenerate problem (3.7) (the perturbation isn't small, but it also isn't so large as to obscure the effect). Note how the degenerate vertex in Figure 3.2 appears as two vertices in Figure 3.1.

Exercises

3.1 Solve the following linear program using the perturbation method to resolve degeneracy:

$$\begin{array}{lrcl} \text{maximize} & 10x_1 - 57x_2 - 9x_3 - 24x_4 & & \\ \text{subject to} & 0.5x_1 - 5.5x_2 - 2.5x_3 + 9x_4 & \leq & 0 \\ & 0.5x_1 - 1.5x_2 - 0.5x_3 + x_4 & \leq & 0 \\ & x_1 & \leq & 1 \\ & x_1,\ x_2,\ x_3,\ x_4 & \geq & 0\,. \end{array}$$

Note: The simple pivot tool with the *Lexicographic* labels can be used to check your arithmetic:

www.princeton.edu/∼rvdb/JAVA/pivot/simple.html

3.2 Solve the following linear program using Bland's rule to resolve degeneracy:

$$\begin{array}{lrcl} \text{maximize} & 10x_1 - 57x_2 - 9x_3 - 24x_4 & & \\ \text{subject to} & 0.5x_1 - 5.5x_2 - 2.5x_3 + 9x_4 & \leq & 0 \\ & 0.5x_1 - 1.5x_2 - 0.5x_3 + x_4 & \leq & 0 \\ & x_1 & \leq & 1 \\ & x_1,\ x_2,\ x_3,\ x_4 & \geq & 0\,. \end{array}$$

3.3 Using today's date (MMYY) for the seed value, solve 10 possibly degenerate problems using the online pivot tool:

www.princeton.edu/∼rvdb/JAVA/pivot/lexico.html

3.4 Consider the linear programming problems whose right-hand sides are identically zero:

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} a_{ij} x_j \leq 0 & i = 1, 2, \ldots, m \\ & x_j \geq 0 & j = 1, 2, \ldots, n. \end{array}$$

Show that either $x_j = 0$ for all j is optimal or else the problem is unbounded.

3.5 Consider the following linear program:

$$\begin{array}{lrcl} \text{maximize} & x_1 + 3x_2 & & \\ \text{subject to} & -2x_1 & \leq & -5 \\ & x_1 & \geq & 0. \end{array}$$

Show that this problem has feasible solutions but no vertex solutions. How does this reconcile with the fundamental theorem of linear programming (Theorem 3.4)?

3.6 Suppose that a linear programming problem has the following property: its initial dictionary is not degenerate and, when solved by the simplex method, there is never a tie for the choice of leaving variable.

(a) Can such a problem have degenerate dictionaries? Explain.
(b) Can such a problem cycle? Explain.

3.7 Consider the following dictionary:

$$\begin{array}{rcl} \zeta = 5 + 2x_2 - 2x_3 + 3x_5 & & \\ \hline x_6 = 4 - 2x_2 - x_3 + x_5 & & \\ x_4 = 2 - x_2 + x_3 - x_5 & & \\ x_1 = 6 - 2x_2 - 2x_3 - 3x_5 & . & \end{array}$$

(a) List all pairs (x_r, x_s) such that x_r could be the entering variable and x_s could be the leaving variable.
(b) List all such pairs if the largest-coefficient rule for choosing the entering variable is used.
(c) List all such pairs if Bland's rule for choosing the entering and leaving variables is used.

Notes

The first example of cycling was given by Hoffman (1953). The fact that any linear programming problem that cycles must have at least six variables and three constraints was proved by Marshall and Suurballe (1969).

Early proofs of the fundamental theorem of linear programming (Theorem 3.4) were constructive, relying, as in our development, on the existence of a variant of the simplex method that works even in the presense of degeneracy. Hence, finding such variants occupied the attention of early researchers in linear programming. The *perturbation method* was first suggested by A. Orden and developed independently by Charnes (1952). The essentially equivalent *lexicographic method* first appeared in Dantzig et al. (1955). Theorem 3.3 was proved by Bland (1977).

For an extensive treatment of degeneracy issues see Gal (1994).