Integer Programming: Branch-and-Bound Algorithm

FIRST APPROXIMATION

An integer program is a linear program with the added requirement that all variables be integers (see Chapter 1). Therefore, a *first approximation* to the solution of any integer program may be obtained by ignoring the integer requirement and solving the resulting linear program by one of the techniques already presented. If the optimal solution to the linear program happens to be integral, then this solution is also the optimal solution to the original integer program (see Problem 6.3). Otherwise—and this is the usual situation—one may round the components of the first approximation to the nearest feasible integers and obtain a *second approximation*. This procedure is often carried out, especially when the first approximation involves large numbers, but it can be inaccurate when the numbers are small (see Problem 6.5).

BRANCHING

If the first approximation contains a variable that is not integral, say x_j^*, then $i_1 < x_j^* < i_2$, where i_1 and i_2 are consecutive, nonnegative integers. Two new integer programs are then created by augmenting the original integer program with either the constraint $x_j \le i_1$ or the constraint $x_j \ge i_2$. This process, called *branching*, has the effect of shrinking the feasible region in a way that eliminates from further consideration the current nonintegral solution for x_j but still preserves all possible integral solutions to the original problem. (See Problem 6.8.)

Example 6.1 As a first approximation to the integer program

$$\begin{aligned} \text{maximize:}\quad & z = 10x_1 + x_2 \\ \text{subject to:}\quad & 2x_1 + 5x_2 \le 11 \\ \text{with:}\quad & x_1 \text{ and } x_2 \text{ nonnegative and integral} \end{aligned} \tag{6.1}$$

we consider the associated linear program obtained by deleting the integral requirement. By graphing, the solution is readily found to be $x_1^* = 5.5$, $x_2^* = 0$, with $z^* = 55$. Since $5 < x_1^* < 6$, branching creates the two new integer programs

$$\begin{aligned} \text{maximize:}\quad & z = 10x_1 + x_2 \\ \text{subject to:}\quad & 2x_1 + 5x_2 \le 11 \\ & x_1 \qquad\quad \le 5 \\ \text{with:}\quad & x_1 \text{ and } x_2 \text{ nonnegative and integral} \end{aligned} \tag{6.2}$$

$$\begin{aligned} \text{maximize:}\quad & z = 10x_1 + x_2 \\ \text{subject to:}\quad & 2x_1 + 5x_2 \le 11 \\ & x_1 \qquad\quad \ge 6 \\ \text{with:}\quad & x_1 \text{ and } x_2 \text{ nonnegative and integral} \end{aligned} \tag{6.3}$$

For the two integer programs created by the branching process, first approximations are obtained by again ignoring the integer requirements and solving the resulting linear programs. If either first approximation is still nonintegral, then the integer program which gave rise to that first approximation becomes a candidate for further branching.

Example 6.2 Using graphical methods, we find that program (6.2) has the first approximation $x_1^* = 5$, $x_2^* = 0.2$, with $z^* = 50.2$, while program (6.3) has no feasible solution. Thus, program (6.2) is a candidate for further branching. Since $0 < x_2^* < 1$, we augment (6.2) with either $x_2 \le 0$ or $x_2 \ge 1$, and obtain the two new programs

$$\begin{aligned} \text{maximize:} \quad & z = 10x_1 + x_2 \\ \text{subject to:} \quad & 2x_1 + 5x_2 \le 11 \\ & x_1 \qquad\quad \le 5 \\ & \qquad\quad x_2 \le 0 \\ \text{with:} \quad & x_1 \text{ and } x_2 \text{ nonnegative and integral} \end{aligned} \tag{6.4}$$

(in which $x_2 = 0$ is forced) and

$$\begin{aligned} \text{maximize:} \quad & z = 10x_1 + x_2 \\ \text{subject to:} \quad & 2x_1 + 5x_2 \le 11 \\ & x_1 \qquad\quad \le 5 \\ & \qquad\quad x_2 \ge 1 \\ \text{with:} \quad & x_1 \text{ and } x_2 \text{ nonnegative and integral} \end{aligned} \tag{6.5}$$

With the integer requirements ignored, the solution to program (6.4) is $x_1^* = 5$, $x_2^* = 0$, with $z^* = 50$, while the solution to program (6.5) is $x_1^* = 3$, $x_2^* = 1$, with $z^* = 31$. Since both these first approximations are integral, no further branching is required.

BOUNDING

Assume that the objective function is to be *maximized*. Branching continues until an integral first approximation (which is thus an integral solution) is obtained. The value of the objective for this first integral solution becomes a lower bound for the problem, and all programs whose first approximations, integral or not, yield values of the objective function smaller than the lower bound are discarded.

Example 6.3 Program (6.4) possesses an integral solution with $z^* = 50$; hence 50 becomes a lower bound for the problem. Program (6.5) has a solution with $z^* = 31$. Since 31 is less than the lower bound 50, program (6.5) is eliminated from further consideration, *and would have been so eliminated even if its first approximation had been nonintegral.*

Branching continues from those programs having nonintegral first approximations that give values of the objective function greater than the lower bound. If, in the process, a new integral solution is uncovered having a value of the objective function greater than the current lower bound, then this value of the objective function becomes the new lower bound. The program that yielded the old lower bound is eliminated, as are all programs whose first approximations give values of the objective function smaller than the new lower bound. The branching process continues until there are no programs with nonintegral first approximations remaining under consideration. At this point, the current lower-bound solution is the optimal solution to the original integer program.

If the objective function is to be *minimized*, the procedure remains the same, except that upper bounds are used. Thus, the value of the first integral solution becomes an upper bound for the problem, and programs are eliminated when their first approximate z-values are greater than the current upper bound.

One always branches from that program which appears most nearly optimal. When there are a number of candidates for further branching, one chooses that having the largest z-value, if the objective function is to be maximized, or that having the smallest z-value, if the objective function is to be minimized.

Additional constraints are added one at a time. If a first approximation involves more than one nonintegral variable, the new constraints are imposed on that variable which is furthest from being an integer; i.e., that variable whose fractional part is closest to 0.5. In case of a tie, the solver arbitrarily chooses one of the variables.

Finally, it is possible for an integer program or an associated linear program to have more than one optimal solution. In both cases, we adhere to the convention adopted in Chapter 1, arbitrarily designating one of the solutions as the optimal one and disregarding the rest.

Solved Problems

6.1 Draw a schematic diagram (tree) depicting the results of Examples 6.1 through 6.3.

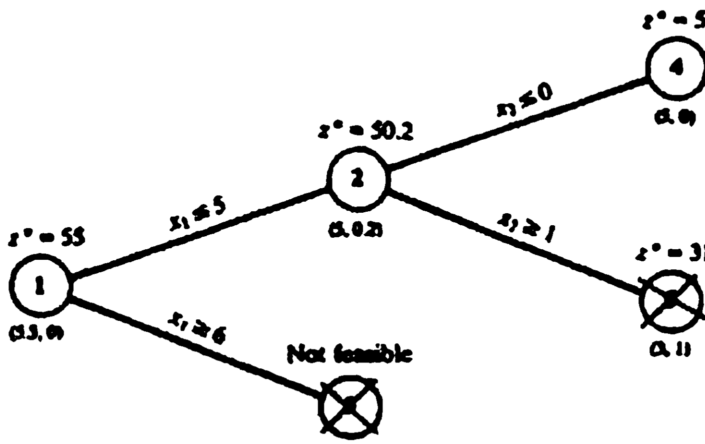

Fig. 6-1

See Fig. 6-1. The original integer program, here (*6.1*), is designated by a circled 1, and all other programs formed through branching are designated in the order of their creation by circled successive integers. Thus, programs (*6.2*) through (*6.5*) are designated by circled 2 through 5, respectively. The first approximate solution to each program is written by the circle designating the program. Each circle (program) is then connected by a line to that circle (program) which generated it via the branching process. The new constraint that defined the branch is written above the line. Finally, a large cross is drawn through a circle if the corresponding program has been eliminated from further consideration. Hence, branch 3 was eliminated because it was not feasible; branch 5 was eliminated by bounding in Example 6.3. Since there are no nonintegral branches left to consider, the schematic diagram indicates that program 1 is solved with $x_1^* = 5$, $x_2^* = 0$, and $z^* = 50$.

6.2

$$\begin{aligned}\text{maximize:}\quad & z = 3x_1 + 4x_2\\ \text{subject to:}\quad & 2x_1 + x_2 \le 6\\ & 2x_1 + 3x_2 \le 9\\ \text{with:}\quad & x_1 \text{ and } x_2 \text{ nonnegative and integral}\end{aligned}$$

Neglecting the integer requirement, we obtain $x_1^\circ = 2.25$, $x_2^\circ = 1.5$, with $z^\circ = 12.75$, as the solution to the associated linear program. Since x_2° is further from an integral value than x_1°, we use it to generate the branches $x_2 \leq 1$ and $x_2 \geq 2$.

Program 2

maximize: $z = 3x_1 + 4x_2$

subject to: $2x_1 + x_2 \leq 6$

$2x_1 + 3x_2 \leq 9$

$x_2 \leq 1$

with: x_1, x_2 nonnegative and integral

Program 3

maximize: $z = 3x_1 + 4x_2$

subject to: $2x_1 + x_2 \leq 6$

$2x_1 + 3x_2 \leq 9$

$x_2 \geq 2$

with: x_1, x_2 nonnegative and integral

The first approximation to Program 2 is $x_1^\circ = 2.5$, $x_2^\circ = 1$, with $z^\circ = 11.5$; the first approximation to Program 3 is $x_1^\circ = 1.5$, $x_2^\circ = 2$, with $z^\circ = 12.5$. These results are shown in Fig. 6-2. Since Programs 2 and 3 both have nonintegral first approximations, we could branch from either one; we choose Program 3 because it has the larger (more nearly optimal) value of the objective function. Here $1 < x_1^\circ < 2$, so the new programs are

Program 4

maximize: $z = 3x_1 + 4x_2$

subject to: $2x_1 + x_2 \leq 6$

$2x_1 + 3x_2 \leq 9$

$x_2 \geq 2$

$x_1 \leq 1$

with: x_1, x_2 nonnegative and integral

Program 5

maximize: $z = 3x_1 + 4x_2$

subject to: $2x_1 + x_2 \leq 6$

$2x_1 + 3x_2 \leq 9$

$x_2 \geq 2$

$x_1 \geq 2$

with: x_1, x_2 nonnegative and integral

There is no solution to Program 5 (it is infeasible), while the solution to Program 4 with the integer constraints ignored is $x_1^\circ = 1$, $x_2^\circ = 7/3$, with $z^\circ = 12.33$. See Fig. 6-2. The branching can continue from either Program 2 or Program 4; we choose Program 4 since it has the greater z-value. Here $2 < x_2^\circ < 3$, so

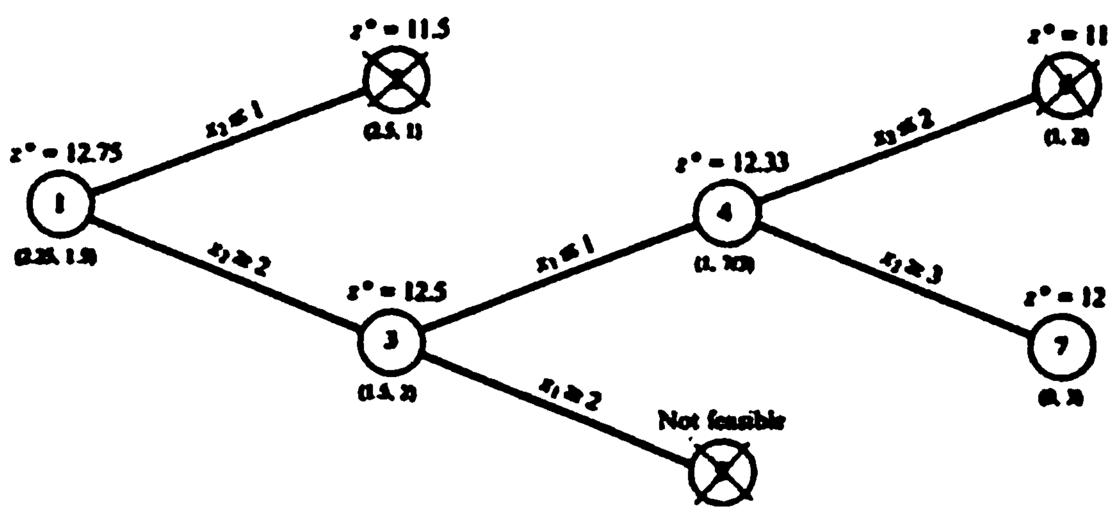

Fig. 6-2

the new programs are

Program 6

$$\begin{aligned} \text{maximize:}\quad & z = 3x_1 + 4x_2 \\ \text{subject to:}\quad & 2x_1 + x_2 \le 6 \\ & 2x_1 + 3x_2 \le 9 \\ & x_2 \ge 2 \\ & x_1 \le 1 \\ & x_2 \le 2 \end{aligned}$$

with: x_1, x_2 nonnegative and integral

Program 7

$$\begin{aligned} \text{maximize:}\quad & z = 3x_1 + 4x_2 \\ \text{subject to:}\quad & 2x_1 + x_2 \le 6 \\ & 2x_1 + 3x_2 \le 9 \\ & x_2 \ge 2 \\ & x_1 \le 1 \\ & x_2 \ge 3 \end{aligned}$$

with: x_1, x_2 nonnegative and integral

The solution to Program 6 with the integer constraints ignored is $x_1^* = 1$, $x_2^* = 2$, with $z^* = 11$. Since this is an integral solution, $z = 11$ becomes a lower bound for the problem: any program yielding a z-value smaller than 11 will henceforth be eliminated. The first approximation to Problem 7 is $x_1^* = 0$, $x_2^* = 3$, with $z^* = 12$. Since this is an integral solution with a z-value greater than the current lower bound, $z = 12$ becomes the new lower bound, and the program that generated the old lower bound, Program 6, is eliminated from further consideration, as is Program 2. Figure 6-2 now shows no branches left to consider other than the one corresponding to the current lower bound. Consequently, this branch gives the optimal solution to Program 1: $x_1^* = 0$, $x_2^* = 3$, with $z^* = 12$.

6.3 Solve Problem 1.9.

Dropping the integer requirements from program (*1*) of Problem 1.9, we solve the associated linear program first, to find (see Problem 5.4): $x_1^* = 2$, $x_2^* = 18$, $x_3^* = 0$, $x_4^* = 20$, $x_5^* = 0$, $x_6^* = 5$, with $z^* = 45$. This is the first approximation. Since it is integral, however, it is also the optimal solution to the original integer program.

6.4 Solve Problem 1.6.

Ignoring the integer requirements in program (*4*) of Problem 1.6, we obtain $x_1^* = x_2^* = 0$, $x_3^* = 1666.67$, $x_4^* = 5000$, with $z^* = 55000$, as the first approximation. Since x_3^* is not integral, we branch to two new programs, and solve each with the integer constraints ignored. The results are indicated in Fig. 6-3. Program 3 possesses an integral solution with a z-value greater than the z-value of Program 2. Consequently, we eliminate Program 2 and accept the solution to Program 3 as the optimal one: $x_1^* = 1$, $x_2^* = 0$, $x_3^* = 1667$, $x_4^* = 4999$, with $z^* = \$55000$.

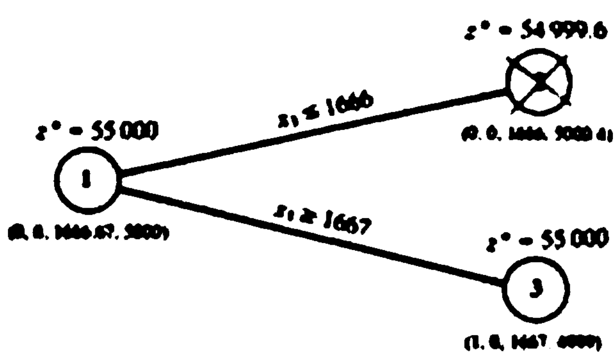

Fig. 6-3

6.5 Discuss the errors involved in rounding the first approximations to the original programs in Problems 6.2 and 6.4 to integers and then taking these answers as the optimal ones.

The first approximatiom in Problem 6.2 was $x_1^* = 2.25$, $x_2^* = 1.5$. We wish to round to the closest integer point *in the feasible region*. Now, of the four integer points surrounding the first approximation, only one, (2, 1), is found to lie in the feasible region. Thus we take $x_1^* = 2$, $x_2^* = 1$, with a corresponding $z^* = 10$, as the proposed optimal solution. The true optimal solution was found as $z^* = 12$; thus the rounded solution deviates from the true solution by more than 16 percent.

The first approximation in Problem 6.4 was $x_1^* = x_2^* = 0$, $x_3^* = 1666.67$, $x_4^* = 5000$. Rounding x_3^* down, to remain feasible, we obtain $x_1^* = x_2^* = 0$, $x_3^* = 1666$, $x_4^* = 5000$ as the estimated coordinates of the optimal solution. The corresponding z-value, \$54 996, deviates from the true solution, $z^* = \$55\,000$, by less than 0.008 percent.

6.6

$$\begin{aligned} \text{minimize:} \quad & z = x_1 + x_2 \\ \text{subject to:} \quad & 2x_1 + 2x_2 \geq 5 \\ & 12x_1 + 5x_2 \leq 30 \\ \text{with:} \quad & x_1 \text{ and } x_2 \text{ nonnegative and integral} \end{aligned}$$

A first approximation to this program is $x_1^* = 2.5$, $x_2^* = 0$, with $z^* = 2.5$. Rounding x_1^* up, thereby remaining feasible, we have $x_1^* = 3$, $x_2^* = 0$, with $z^* = 3$, as an estimate of the optimal solution to the original program. Observe, however, that for integral values of the variables, the objective function must itself be integral. The z-value for the first approximation, $z^* = 2.5$, provides a lower bound for the optimal objective; consequently, the optimal objective cannot be smaller than 3. Since we have an estimate which attains the value 3, the estimate must be optimal; i.e., $x_1^* = 3$, $x_2^* = 0$, with $z^* = 3$.

6.7 Solve the *knapsack problem* formulated in Problem 1.8.

The simplex method could be used to find the first approximation for program (*3*) of Problem 1.8. A more efficient procedure is the following:

The critical factor in determining whether an item is taken is not its weight or value per se but the ratio of the two, its value per pound. We denote this factor as *desirability*, adjoin it to the data, and construct Table 6-1, where the items are listed in order of decreasing desirability. To obtain the optimal solution to the knapsack problem with the integer constraints ignored, we simply take as much of each item as possible (without exceeding the 60-lb weight limit), beginning with the most desirable. It follows from Table 6-1 that the first approximation consists in all of item 2 (the most desirable one), all of item 5 (the next most desirable item), and 30 lb of item 3: $x_1^* = 0$, $x_2^* = 1$, $x_3^* = 30/35$, $x_4^* = 0$, $x_5^* = 1$, with $z^* = 135$.

Table 6-1

Item	Weight, lb	Value	Desirability, value/lb
2	23	60	2.61
5	7	15	2.14
3	35	70	2.00
1	52	100	1.92
4	15	15	1.00

Since this first approximation is nonintegral, we branch by augmenting the original constraints with either $x_3 \leq 0$ or $x_3 \geq 1$. Before doing so, however, we note that since x_3 is required to be nonnegative, the constraint $x_3 \leq 0$ can be tightened to $x_3 = 0$; and since at most one of an item will be taken, the constraint $x_3 \geq 1$ can be tightened to $x_3 = 1$. This is indicated in the tree diagram, Fig. 6-4.

Dropping the integer requirements, we determine the optimal solutions to both Programs 2 and 3 in Fig. 6-4, using Table 6-1 to find the best mix consistent with the constraints. For Program 2, we obtain $x_1^* = 30/52$, $x_2^* = 1$, $x_3^* = 0$, $x_4^* = 0$, $x_5^* = 1$, with $z^* = 132.69$; and for Program 3, $x_1^* = 0$, $x_2^* = 1$, $x_3^* = 1$, $x_4^* = 0$, $x_5^* = 2/7$, with $z^* = 134.28$.

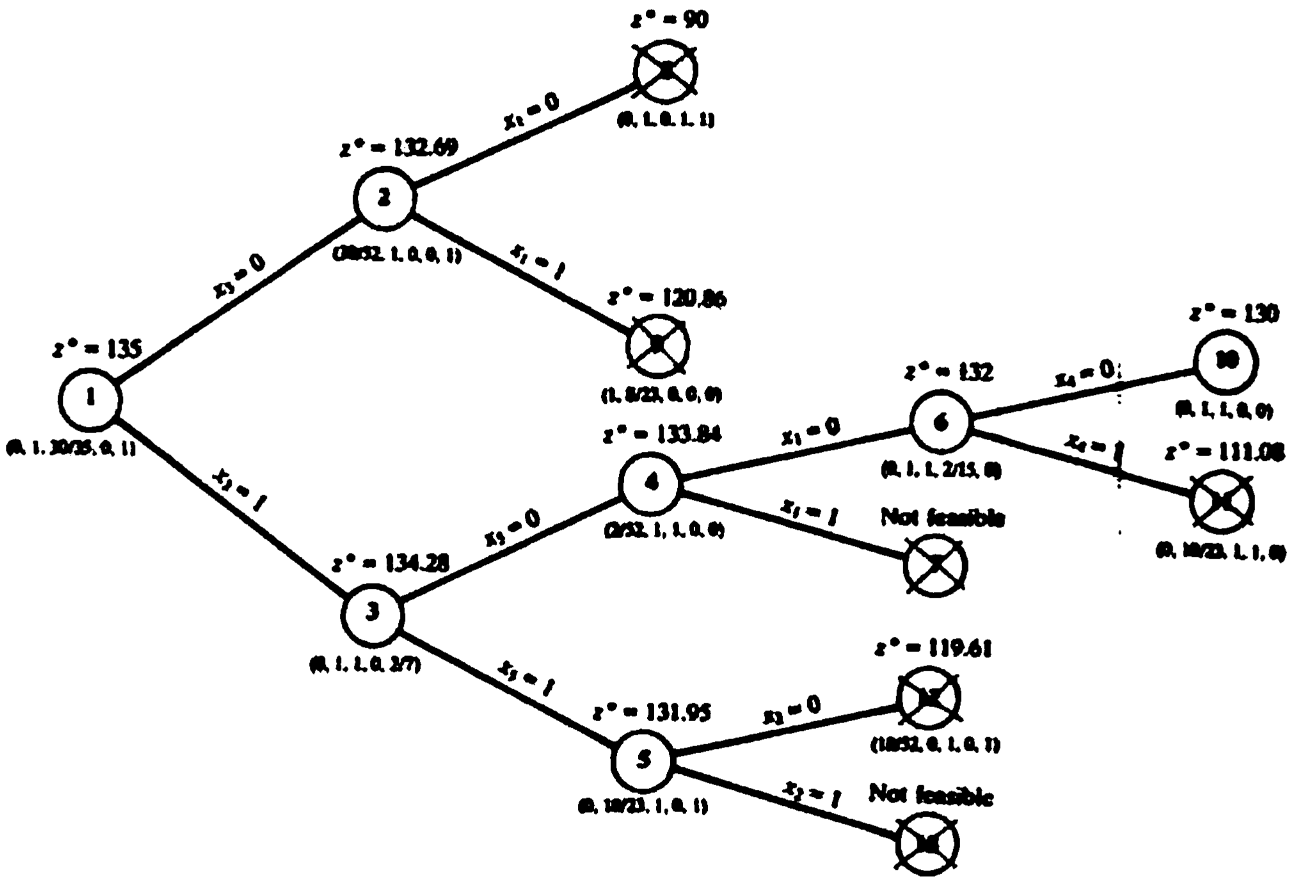

Fig. 6-4

Continuing the branch-and-bound process, we complete Fig. 6-4. The first integral solution is obtained in Program 8, with $z^* = 90$. A second integral solution is obtained in Program 10, with $z^* = 130$. Since this second z-value is larger than the first, we eliminate Program 8, as well as Programs 9 and 11. Program 5, however, possesses a z-value greater than the current lower bound, so that we must still branch from it. The resulting Program 12 has a z-value smaller than 130, while Program 13 is infeasible; hence they too are eliminated. We remain with only Program 10; therefore, its solution—take only items 2 and 3, for a total value of 130—is the optimal solution.

Much of the branch-and-bound process might have been avoided. We know in advance that either $x_3 = 0$ or $x_3 = 1$ in the optimal solution. If $x_3 = 0$, then Program 2 coincides with the original program, and the z-value 132.69 obtained when the integer requirements are dropped (thereby expanding the feasible region) must be greater than, or at least equal to, the true optimum. Similarly, if $x_3 = 1$, we see from Program 3 that the true optimum cannot exceed 134.28. Whichever the case, we are assured that the true optimum is less than 135. But, for integral values of the variables, z is integral; in fact, it is a multiple of 5, since the values of the items are multiples of 5. Therefore, the true optimum is at most 130. Now, rounding the first approximate solution to Program 3 gives $x_1^* = 0, x_2^* = 1, x_3^* = 1, x_4^* = 0, x_5^* = 0$, with $z^* = 130$. Consequently, this solution is optimal.

6.8 Discuss the geometrical significance of making the first branch in Problem 6.2.

The feasible region for Problem 6.2 with the integer requirements ignored is the shaded region in Fig. 6-5(*a*); the feasible region for Problem 6.2 as given is the set of all integer points (marked with crosses) belonging to the shaded region. The first approximation is the circled extreme point.

As a result of branching, the feasible region for Program 2, with the integer constraints ignored, is Region I in Fig. 6-5(*b*), whereas Region II in the same figure represents the feasible region for Program 3 with the integer requirements neglected. Observe that Regions I and II together contain all the feasible integer points of Fig. 6-5(*a*), and only those integer points. Hence, if the original integer program has an

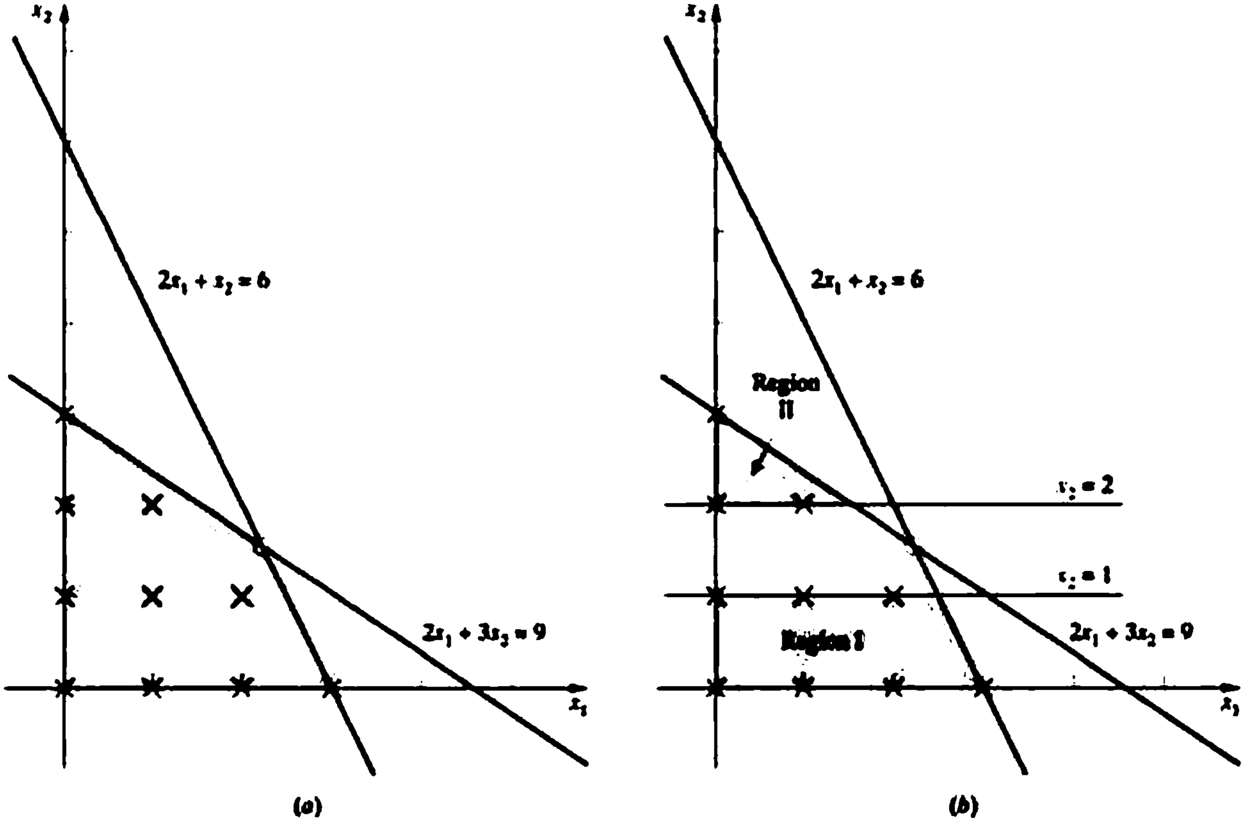

Fig. 6-5

optimal solution (as it does, in this case), that solution will be optimal for one of the two new integer programs. Conversely, *if the two new integer programs have optimal solutions, one of these solutions (the one with the larger z-value, in the case of a maximization problem) will be optimal for the original integer program.* The validity of the bounding technique follows from the parenthetical remark just made.

Supplementary Problems

Solve the following problems by use of the branch-and-bound algorithm.

6.9

$$\begin{aligned} \text{maximize:} \quad & z = x_1 + 2x_2 + x_3 \\ \text{subject to:} \quad & 2x_1 + 3x_2 + 3x_3 \le 11 \\ \text{with:} \quad & \text{all variables nonnegative and integral} \end{aligned}$$

6.10

$$\begin{aligned} \text{maximize:} \quad & z = x_1 + 2x_2 + 3x_3 + x_4 \\ \text{subject to:} \quad & 3x_1 + 2x_2 + x_3 + 4x_4 \le 10 \\ & 5x_1 + 3x_2 + 2x_3 + 5x_4 \le 5 \\ \text{with:} \quad & \text{all variables nonnegative and integral} \end{aligned}$$

6.11

$$\begin{aligned} \text{maximize:} \quad & z = 2x_1 + 10x_2 + x_3 \\ \text{subject to:} \quad & 5x_1 + 2x_2 + x_3 \le 15 \\ & 2x_1 + x_2 + 7x_3 \le 20 \\ & x_1 + 3x_2 + 2x_3 \le 25 \\ \text{with:} \quad & \text{all variables nonnegative and integral} \end{aligned}$$

6.12

$$\begin{aligned} \text{minimize: } & z = 10x_1 + 2x_2 + 11x_3 \\ \text{subject to: } & 2x_1 + 7x_2 + x_3 = 4 \\ & 5x_1 + 8x_2 - 2x_3 = 17 \\ \text{with: } & \text{all variables nonnegative and integral} \end{aligned}$$

6.13 Problem 1.20.

6.14 Solve Problem 6.7 by applying the branch-and-bound algorithm directly to program (*3*) of Problem 1.8 and compare this procedure with the approach taken in Problem 6.7.

Integer Programming: Cut Algorithms

At each stage of branching in the branch-and-bound algorithm the current feasible region (for the current program with integer restrictions ignored) is *cut* into two smaller regions (one of them may be empty) by the imposition of two new constraints derived from the first approximation to the current program. This splitting is such that the optimal solution to the current program must show up as the optimal solution to one of the two new programs (Problem 6.8). The cut algorithms of the present chapter operate essentially in like fashion, the only difference being that a single new constraint is added at each stage, whereby the feasible region is diminished without being split.

THE GOMORY ALGORITHM

The new constraints are determined by the following three-step procedure. (See Problem 7.5.)

STEP 1: In the current final simplex tableau, select one (any one) of the nonintegral variables and, *without assigning zero values to the nonbasic variables*, consider the constraint equation represented by the row of the selected variable.

Example 7.1 The simplex tableau

	x_1	x_2	x_3	x_4	x_5	
x_3	$-1/2$	0	1	$-7/3$	1/2	11/2
x_2	1/2	1	0	-1	1/4	1
	4	0	0	1	3/4	25/2

gives the optimal solution (i.e., the current first approximation) as $x_3^* = 11/2$, $x_2^* = 1$, with each of the nonbasic variables x_1^*, x_4^*, and x_5^* set equal to zero. The noninteger assignment for x_3^* came from the first row of the tableau, which represents the constraint

$$-\tfrac{1}{2}x_1 + x_3 - \tfrac{7}{3}x_4 + \tfrac{1}{2}x_5 = \tfrac{11}{2} \tag{7.1}$$

STEP 2: Rewrite each fractional coefficient and constant in the constraint equation obtained from Step 1 as the sum of an integer and a *positive* fraction between 0 and 1. Then rewrite the equation so that the left-hand side contains only terms with fractional coefficients (and a fractional constant), while the right-hand side contains only terms with integral coefficients (and an integral constant).

Example 7.2 Equation (*7.1*) becomes

$$(-1 + \tfrac{1}{2})x_1 + x_3 + (-3 + \tfrac{2}{3})x_4 + (0 + \tfrac{1}{2})x_5 = 5 + \tfrac{1}{2}$$

or

$$\tfrac{1}{2}x_1 + \tfrac{2}{3}x_4 + \tfrac{1}{2}x_5 - \tfrac{1}{2} = 5 + x_1 - x_3 + 3x_4 \tag{7.2}$$

STEP 3: Require the left-hand side of the rewritten equation to be nonnegative. The resulting inequality is the new constraint.

Example 7.3 From (7.2),

$$\tfrac{1}{3}x_1 + \tfrac{2}{3}x_4 + \tfrac{1}{3}x_5 - \tfrac{1}{3} \ge 0 \qquad \text{or} \qquad \tfrac{1}{3}x_1 + \tfrac{2}{3}x_4 + \tfrac{1}{3}x_5 \ge \tfrac{1}{3}$$

is the new constraint.

COMPUTATIONAL CONSIDERATIONS

Computing time is saved by appending the new constraint inequality obtained from Step 3 to the constraint equations described in the current final simplex tableau rather than to the algebraically equivalent constraints given in the original program. (See Problem 7.1.)

The Gomory cut algorithm may not converge; that is, an integral solution may not be obtained regardless of the number of iterations. Generally, however, if the algorithm does converge, it converges reasonably quickly. For this reason, an upper limit on the number of iterations to be attempted is often established before the computation is initiated. If the integral solution is not obtained within this bound, the algorithm is abandoned.

There are no theoretical reasons for choosing between the Gomory and branch-and-bound algorithms. The branch-and-bound algorithm is the newer of the two procedures, and appears to be favored slightly among practitioners.

Solved Problems

7.1

$$\begin{aligned} \text{maximize:} \quad & z = 2x_1 + x_2 \\ \text{subject to:} \quad & 2x_1 + 5x_2 \le 17 \\ & 3x_1 + 2x_2 \le 10 \\ \text{with:} \quad & x_1 \text{ and } x_2 \text{ nonnegative and integral} \end{aligned} \tag{1}$$

Ignoring the integer requirements and applying the simplex method to the resulting linear program, we obtain Tableau 1 as the optimal tableau after one iteration.

	x_1	x_2	x_3	x_4	
x_3	0	11/3	1	−2/3	31/3
x_1	1	2/3	0	1/3	10/3
	0	1/3	0	2/3	20/3

Tableau 1

	x_1	x_2	x_3	x_4	x_5	
x_3	0	0	1	−5/2	11/6	17/2
x_1	1	0	0	0	1/3	3
x_2	0	1	0	1/2	−1/2	1/2
	0	0	0	1/2	1/6	13/2

Tableau 2

The first approximation to program (1), therefore, is $x_1^* = 10/3$, $x_3^* = 31/3$, $x_2^* = x_4^* = 0$. Both x_1^* and x_3^* are nonintegral. Arbitrarily selecting x_1^*, we consider the constraint represented by the second row of Tableau 1, the row defining x_1^*; namely,

$$x_1 + \tfrac{2}{3}x_2 + \tfrac{1}{3}x_4 = \tfrac{10}{3}$$

Writing each fraction as the sum of an integer and a fraction between 0 and 1, we have

$$x_1 + (0 + \tfrac{2}{3})x_2 + (0 + \tfrac{1}{3})x_4 = 3 + \tfrac{1}{3} \qquad \text{or} \qquad \tfrac{2}{3}x_2 + \tfrac{1}{3}x_4 - \tfrac{1}{3} = 3 - x_1$$

Requiring the left-hand side of this equation to be nonnegative, we obtain

$$\tfrac{2}{3}x_2 + \tfrac{1}{3}x_4 - \tfrac{1}{3} \ge 0 \qquad \text{or} \qquad 2x_2 + x_4 \ge 1$$

as the new constraint. Rewriting the constraints of the original program (*1*) in the forms suggested by Tableau 1 and adding the new constraint, we generate the new program

$$\begin{aligned}
\text{maximize:}\quad & z = 2x_1 + x_2 + 0x_3 + 0x_4 \\
\text{subject to:}\quad & \tfrac{11}{3}x_2 + x_3 - \tfrac{2}{3}x_4 = \tfrac{31}{3} \\
& x_1 + \tfrac{2}{3}x_2 \qquad + \tfrac{1}{3}x_4 = \tfrac{10}{3} \qquad\qquad (2) \\
& \qquad 2x_2 \qquad\quad + x_4 \geq 1 \\
\text{with:}\quad & \text{all variables nonnegative and integral}
\end{aligned}$$

A surplus variable, x_5, and an artificial variable, x_6, are introduced into the inequality constraint of (*2*), and then the two-phase method is applied, with x_1, x_3, and x_6 as the initial set of basic variables. The optimal Tableau 2 is obtained after only one iteration. The first approximation to program (*2*) is thus $x_1^* = 3$, $x_2^* = 1/2$, $x_3^* = 17/2$, $x_4^* = x_5^* = 0$. Choosing x_2^* to generate the new constraint, we obtain from the third row of Tableau 2

$$\tfrac{1}{2}x_4 + \tfrac{1}{2}x_5 - \tfrac{1}{2} \geq 0 \qquad \text{or} \qquad x_4 + x_5 \geq 1$$

This, combined with the constraints of program (*2*) in the forms suggested by Tableau 2, gives the new integer program

$$\begin{aligned}
\text{maximize:}\quad & z = 2x_1 + x_2 + 0x_3 + 0x_4 + 0x_5 \\
\text{subject to:}\quad & \qquad\qquad x_3 - \tfrac{5}{2}x_4 + \tfrac{11}{6}x_5 = \tfrac{17}{2} \\
& x_1 \qquad\qquad\qquad\quad + \tfrac{1}{3}x_5 = 3 \qquad\qquad (3) \\
& \quad x_2 \qquad + \tfrac{1}{2}x_4 - \tfrac{1}{2}x_5 = \tfrac{1}{2} \\
& \qquad\qquad\qquad x_4 + x_5 \geq 1 \\
\text{with:}\quad & \text{all variables nonnegative and integral}
\end{aligned}$$

Ignoring the integer constraint and applying the two-phase method to program (*3*), with x_1, x_2, x_3, and x_7 (artificial) as the initial basic set, we obtain the optimal Tableau 3.

	x_1	x_2	x_3	x_4	x_5	x_6	
x_3	0	0	1	−13/3	0	11/6	20/3
x_1	1	0	0	−1/3	0	1/3	8/3
x_2	0	1	0	1	0	−1/2	1
x_5	0	0	0	1	1	−1	1
	0	0	0	1/3	0	1/6	19/3

Tableau 3

A new iteration of the process is started from $x_1^* = 8/3$ in Tableau 3. This results in a program whose solution is integral, with $x_1^* = 3$, $x_2^* = 0$, and $z^* = 6$. This solution is then the optimal solution to integer program (*1*).

7.2 Discuss the geometrical significance of the first added constraint in Problem 7.1.

Initially, the feasible region consists of all points in the first quadrant having integral coordinates that satisfy

$$2x_1 + 5x_2 \leq 17 \qquad \text{and} \qquad 3x_1 + 2x_2 \leq 10$$

These are the points marked by crosses in Fig. 7-1(*a*).

The constraint added to the original program (*1*) was $2x_2 + x_4 \geq 1$; it led to program (*2*). Solving the

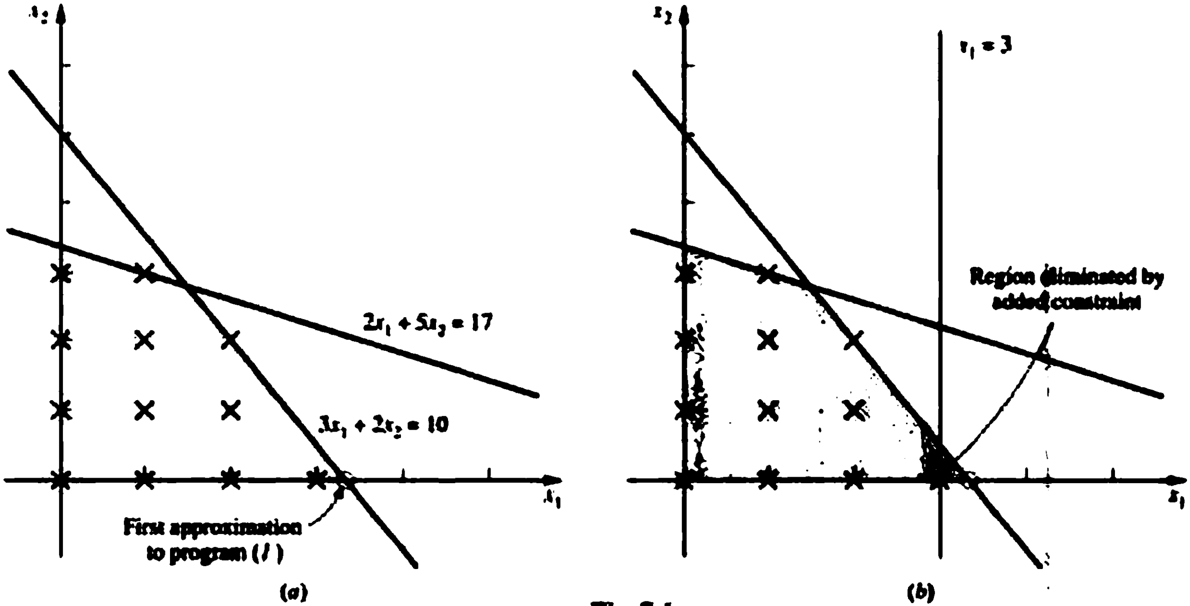

Fig. 7-1

second constraint equation of program (2) for x_4 and substituting the result into the new constraint, we have

$$2x_2 + (10 - 3x_1 - 2x_2) \geq 1 \qquad \text{or} \qquad x_1 \leq 3$$

The effect of imposing $x_1 \leq 3$ is indicated in Fig. 7-1(*b*): a small piece containing the current first approximation is sliced off the feasible region. No integer point, however, is lost.

7.3 Solve Problem 1.12.

The first approximation to this integer program (see Problem 3.20 with the variables relabeled) is $x_{12}^* = 700$, $x_{13}^* = 500$, $x_{21}^* = 1000$, $x_{11}^* = x_{22}^* = x_{23}^* = 0$, with $z^* = 27\,600$¢. Since this first approximation is integral, it is also the optimal solution to the integer program. Under this optimal schedule, 700 boxes will be shipped from factory 1 to retailer 2, 500 boxes from factory 1 to retailer 3, and 1000 boxes from factory 2 to retailer 1. The total shipping cost is \$276.

7.4 Solve Problem 1.5.

Program (4) of Problem 1.5, brought into standard form, is

$$\begin{aligned}
\text{minimize:}\quad & z = 20x_1 + 22x_2 + 18x_3 + 0x_4 + 0x_5 + 0x_6 + 0x_7 + 0x_8 + Mx_9 + Mx_{10} \\
\text{subject to:}\quad & 4x_1 + 6x_2 + x_3 - x_4 + x_9 = 54 \\
& 4x_1 + 4x_2 + 6x_3 - x_5 + x_{10} = 65 \\
& x_1 + x_6 = 7 \\
& x_2 + x_7 = 7 \\
& x_3 + x_8 = 7 \\
\text{with:}\quad & \text{all variables nonnegative and integral}
\end{aligned} \qquad (1)$$

Ignoring the integer restrictions and solving this program by the two-phase method, we obtain Tableau 1 after three iterations. The first approximation to program (1) is thus $x_1^* = 1.75$, $x_2^* = 7$, $x_3^* = 5$, with $z^* = 279$.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
x_1	1	0	0	−0.3	0.005	0	−1.6	0	1.75
x_3	0	0	1	0.2	−0.2	0	0.4	0	5
x_6	0	0	0	0.3	−0.05	1	1.6	0	5.25
x_2	0	1	0	0	0	0	1	0	7
x_8	0	0	0	−0.2	0.2	0	−0.4	1	2
	0	0	0	2.4	2.6	0	2.8	0	−279

Tableau 1

Now, this first approximation may be rounded to the feasible integral solution $x_1 = 2$, $x_2 = 7$, $x_3 = 5$, with $z = 284$. It follows that the desired minimum cannot exceed 284. On the other hand, referring to the original program (4) of Problem 1.5, we see that for integral values of the variables z is an *even* integer; hence, in view of the lower bound $z^* = 279$ provided by the first approximation, the minimal z cannot be less than 280. Therefore, the minimal z can only be 280, 282, or 284, and we are guaranteed that the error committed in taking $(2, 7, 5)^T$ as the optimal solution is at worst

$$\frac{284 - 280}{280} = 1.43\%$$

(Starting from Tableau 1, one finds after six iterations of the Gomory algorithm that $(2, 7, 5)^T$ is in fact the optimal solution.)

7.5 Develop the Gomory cut algorithm.

Consider the optimal tableau that results from applying the simplex method to an integer program with the integer requirements ignored, and assume that one of the basic variables, x_b, is nonintegral. The constraint equation corresponding to the tableau row that determined x_b must have the form

$$x_b + \sum y_j x_j = y_0 \tag{1}$$

where the sum is over all nonbasic variables. The y-terms are the coefficients and the constant term appearing in the tableau row determining x_b. Since x_b is obtained from (1) by setting the nonbasic variables equal to zero, it follows that y_0 is also nonintegral.

Write each y-term in (1) as the sum of an integer and a nonnegative fraction less than 1:

$$y_j = i_j + f_j \qquad \text{and} \qquad y_0 = i_0 + f_0$$

Some of the f_j may be zero, but f_0 is guaranteed to be positive. Equation (1) becomes

$$x_b + \sum (i_j + f_j)x_j = i_0 + f_0$$

or

$$x_b + \sum i_j x_j - i_0 = f_0 - \sum f_j x_j \tag{2}$$

If each x-variable is required to be integral, then the left-hand side of (2) is integral, which forces the right-hand side also to be integral. But, since each f_j and x_j is nonnegative, so too is $\sum f_j x_j$. The right-hand side of (2) then is an integer which is smaller than a positive fraction less than 1; that is, a nonpositive integer.

$$f_0 - \sum f_j x_j \le 0 \qquad \text{or} \qquad \sum f_j x_j - f_0 \ge 0$$

This is the new constraint in the Gomory algorithm.

7.6 Develop another cut algorithm.

Consider (1) of Problem 7.5. If each nonbasic variable x_j is zero, then $x_b = y_0$ is nonintegral. If x_b is to become integer-valued, then at least one of the nonbasic x_j must be made different from zero. Since all variables are required to be nonnegative and integral, it follows that at least one nonbasic variable must be made greater than or equal to 1. This in turn implies that the sum of all the nonbasic variables must be made greater than or equal to 1. If this condition is used as the new constraint to be adjoined to the original integer program, we have the cut algorithm first suggested by Danzig.

7.7 **Use the cut algorithm developed in Problem 7.6 to solve**

$$\begin{aligned} \text{maximize:} \quad & z = 3x_1 + 4x_2 \\ \text{subject to:} \quad & 2x_1 + x_2 \le 6 \\ & 2x_1 + 3x_2 \le 9 \\ \text{with:} \quad & x_1 \text{ and } x_2 \text{ nonnegative and integral} \end{aligned}$$

Introducing slack variables x_3 and x_4 and then solving the resulting program, with the integer requirements ignored, by the simplex method, we obtain Tableau 1.

	x_1	x_2	x_3	x_4	
x_1	1	0	0.75	−0.25	2.25
x_2	0	1	−0.5	0.5	1.5
	0	0	0.25	1.25	12.75

Tableau 1

The first approximation is, therefore, $x_1^* = 2.25$, $x_2^* = 1.5$, which is not integral. The nonbasic variables are x_3 and x_4, so the new constraint is $x_3 + x_4 \ge 1$. Appending this constraint to Tableau 1, after the introduction of surplus variable x_5 and artificial variable x_6, and solving the resulting program by the two-phase method, we generate Tableau 2.

	x_1	x_2	x_3	x_4	x_5	
x_1	1	0	0	−1	0.75	1.5
x_2	0	1	0	1	−0.5	2
x_3	0	0	1	1	−1	1
	0	0	0	1	0.25	12.25

Tableau 2

It follows from Tableau 2 that $x_1^* = 1.5$, $x_2^* = 2$, $x_3^* = 1$, with x_4 and x_5 nonbasic. Since this solution is nonintegral, we take $x_4 + x_5 \ge 1$ as the new constraint. Adjoining this constraint to Tableau 2, after the introduction of surplus variable x_6 and artificial variable x_7, and solving the resulting program by the two-phase method, we generate Tableau 3.

	x_1	x_2	x_3	x_4	x_5	x_6	
x_1	1	0	0	−1.75	0	0.75	0.75
x_2	0	1	0	1.5	0	−0.5	2.5
x_3	0	0	1	2	0	−1	2
x_5	0	0	0	1	1	−1	1
	0	0	0	0.75	0	0.25	12.25

Tableau 3

From Tableau 3, the current optimal solution is nonintegral, with nonbasic variables x_4 and x_6. The new constraint is thus $x_4 + x_6 \geq 1$. Adjoining it to Tableau 3 and solving the resulting program by the two-phase method, we obtain $x_1^* = 0$, $x_2^* = 3$, with $z^* = 12$. Since this solution is integral, it is the optimal solution to the original integer program.

Supplementary Problems

7.8 Use the Gomory algorithm to

$$\begin{aligned} \text{maximize:} \quad & z = x_1 + 9x_2 + x_3 \\ \text{subject to:} \quad & x_1 + 2x_2 + 3x_3 \leq 9 \\ & 3x_1 + 2x_2 + 2x_3 \leq 15 \\ \text{with:} \quad & \text{all variables nonnegative and integral} \end{aligned}$$

7.9 Solve Problem 1.3 by the Gomory algorithm.

7.10 Solve Problem 6.9 by the Gomory algorithm.

7.11 Solve Problem 6.10 by the Gomory algorithm.

7.12 Solve Problem 6.11 by the Gomory algorithm.

7.13 Solve Problem 6.9 by the cut algorithm of Problem 7.6.