When an LP solution contains several fractional integer variables, the decision of which integer variable to branch on next is needed. The following rules are commonly used for choosing a branching variable:

1. Variable with fractional value closest to 0.5
2. Variable with highest impact on objective function
3. Variable with the least index

A decision is also needed as to which unpruned node to explore first. The most commonly used search strategies include

1. Depth-first (last-in first-out; solve the most recently generated subproblem first)
2. Best-bound-first (best upper bound; branch on the active node with greatest z-value)

The goal of the depth-first strategy is to quickly obtain a primal feasible integer solution whose objective function value z^k is a lower bound on the given IP problem and can be used to prune nodes by optimality (rule 3). The best-bound-first strategy chooses the active node with the best upper bound (for maximization problem). The goal is to minimize the total number of nodes evaluated in the B&B tree. Performance of these branching rules depends on the problem structure. In practice, a compromise between the two is adopted. That is, apply the depth-first strategy first to get one feasible integer solution, followed by a mixture of both strategies.

Example 11.2 Solve the following mixed integer problem using branch-and-bound approach. At each step, apply the rule of best-bound-first, and at each node, select the variable with least index to branch first.

$$\begin{aligned}
\text{Maximize} \quad & z = -y_1 + 2y_2 + y_3 + 2x_1 \\
\text{subject to} \quad & y_1 + y_2 - y_3 + 3x_1 \le 7 \\
& y_2 + 3y_3 - x_1 \le 5 \\
& 3y_1 + x_1 \ge 2 \\
& y_1, y_2, y_3 \ge 0 \text{ and integer} \\
& x_1 \ge 0
\end{aligned}$$

After solving the LP relaxation, we obtain an LP optimum $y_1 = 6/11$, $y_2 = 59/11$, $y_3 = 0$, $x_1 = 4/11$, and $z = 120/11$. This solution violates the integer requirements of y_1 and y_2. We use this solution as the root node the branch-and-bound tree in Figure 11.5. The number of each node indicates the sequence of subproblems evaluated. Note

nal integer variables, the decision of
d. The following rules are commonly

).5
e function

ned node to explore first. The most

t recently generated subproblem first)
ch on the active node with greatest

ckly obtain a primal feasible integer
is a lower bound on the given IP
mality (rule 3). The best-bound-first
st upper bound (for maximization
nber of nodes evaluated in the B&B
pends on the problem structure. In
opted. That is, apply the depth-first
ion, followed by a mixture of both

ger problem using branch-and-bound
und-first, and at each node, select the

$2y_2 + y_3 + 2x_1$

$+ 3x_1 \leq 7$

≤ 5

2

0 and integer

n LP optimum $y_1 = 6/11$, $y_2 = 59/11$,
violates the integer requirements of y_1
branch-and-bound tree in Figure 11.5.
nce of subproblems evaluated. Note

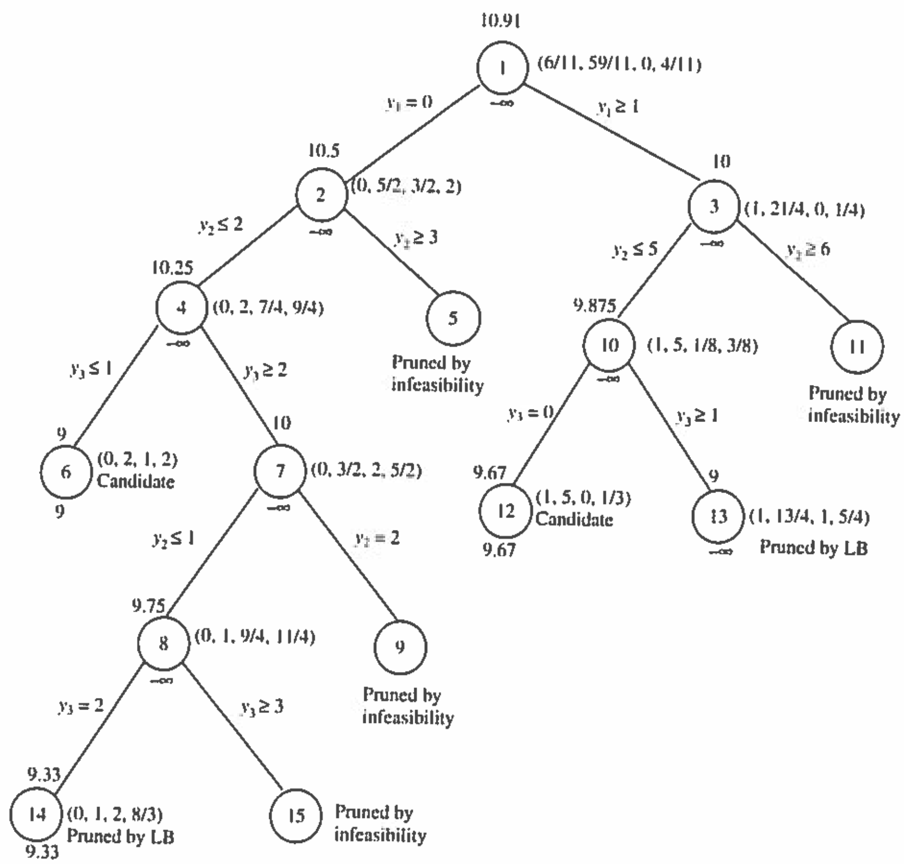


FIGURE 11.5 Branch-and-bound tree for Example 11.2 using best-bound-first.

that at node 1, the constraint $y_1 \leq 0$ was indicated on the left branch, but since $y_1 \geq 0$, y_1 has to be fixed at 0. At node 7, the constraint $y_2 \geq 2$ was intended to be added, but if we trace back along node 7, we would see that the constraint of $y_2 \leq 2$ was already added at node 2. Combining these two constraints, we have $y_2 = 2$. So is the constraint of $y_3 = 2$ at node 8. The problem is finally optimized at node 12, where (1, 5, 0, 1/3) is the optimal solution, with objective value 9.67.

Figure 11.6 depicts the branch-and-bound tree for the same problem, where the "depth-first" rule is applied, and at each node, the variable (violating an integer constraint) with the largest absolute value cost coefficient is chosen to branch first. Ties are broken arbitrarily.

Depth-first is sometimes called last-in first-out (LIFO) because it solves the most recently generated subproblem first. It tends to pursue paths to the depths of the tree, then backtrack to where that path started, and finally plunge down into another

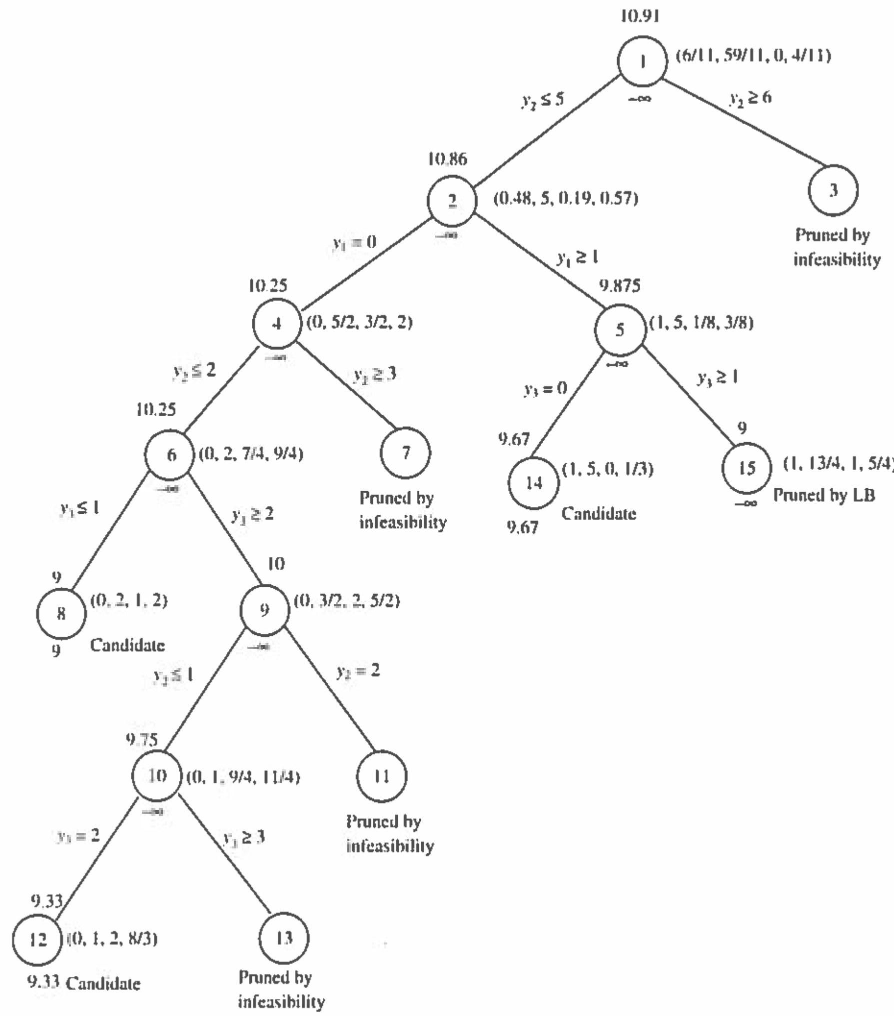


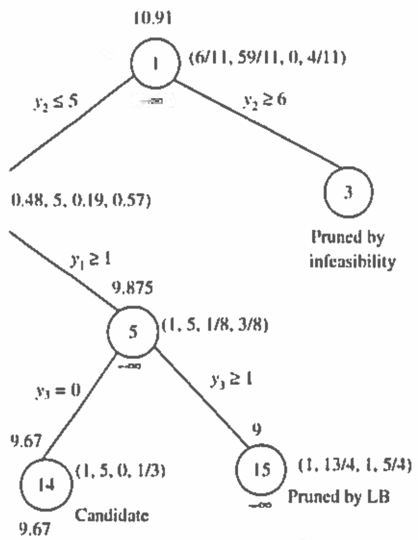
FIGURE 11.6 Branch-and-bound tree for Example 11.2 using depth-first.

depth search. Yet another name for depth-first is "backtracking." Best-bound-first is sometimes called "jumptracking" because it leads to searches that jump back and forth across the tree.

11.1.2 Branch-and-Bound Algorithm

Now we describe the general branch-and-bound algorithm using the following notation.

S = the given IP problem
S_{LP} = the LP relaxation of S



or Example 11.2 using depth-first.

t is "backtracking." Best-bound-first is leads to searches that jump back and

ound algorithm using the following

y_{LP} = the solution to the LP relaxation of the given IP
$\bar{z}$ = lowest (best) upper bound on z^* of the given IP problem
$\underline{z}$ = highest (best) lower bound on z^* of the given IP problem

These are *global* bounds that are periodically updated as the branching proceeds down the various paths in the tree, but are *not* shown on the tree. In Example 11.2, $\bar{z} = 10.91$ throughout and $\underline{z}$ was $-\infty$, then 9, 9.33, and 9.67 at nodes 1, 8, 12, and 14, respectively. Next comes more notation. Let

S^k = subproblem k of problem S
S^k_{LP} = the LP relaxation of subproblem k
z^k = the optimum objective value of S^k
$\bar{z}^k$ = best (lowest) upper bound of subproblem S^k (shown above node k)
$\underline{z}^k$ = best (highest) lower bound of subproblem S^k (shown below node k)
y^k_{LP} = the optimum solution of the LP subproblem S^k_{LP}
$\bar{y}_j$ = noninteger value of integer variable y_j (current numerical value of y_j)
$\lfloor a \rfloor$ = the largest integer $\le a$ (or rounding down a)
$\lceil a \rceil$ = the smallest integer $\ge a$ (or rounding up a)

We now formally describe the B&B algorithm

Step 0 (Initialization). Solve the LP relaxation (S_{LP}) of the given IP problem (S). If it is infeasible, so is the IP problem—terminate. If the LP optimum solution satisfies the integer requirement, the IP problem is solved—terminate. Otherwise, initialize the best upper bound ($\bar{z}$) by the optimal objective value of problem S_{LP} and the best lower bound by $\underline{z} = -\infty$. Place S^k_{LP} on the active list of nodes (subproblems). Initially, there is no incumbent solution.

Step 1 (Choosing a Node). If the active list is empty, terminate. The incumbent solution y^* is optimal. Otherwise, choose a node (subproblem) S^k with S^k_{LP} by one of the rules (e.g., depth-first, best-bound-first, etc.)

Step 2 (Updating Upper Bound). Solve and set $\bar{z}^k$ equal to the LP optimum objective value. Keep the optimum LP solution y^k_{LP}.

Step 3 (Prune by Infeasibility). If S^k_{LP} has no feasible solution, prune the current node and go to step 1. Otherwise, go to step 4.

Step 4 (Prune by Bound). If $\bar{z}^k \le \underline{z}$, prune the current node and go to step 1. Otherwise, go to step 5.

Step 5 (Updating Lower Bound and Prune by Optimality).

(a) If the LP optimum y^k_{LP} is integer, a feasible solution to S is found, an incumbent solution to the given problem. Set $\underline{z}^k = y^k_{LP}$ and compare $\underline{z}^k$ with $\underline{z}$. If $\underline{z}_k > \underline{z}$, set $\underline{z} = \underline{z}_k$, otherwise $\underline{z}$ does not change. The current node is pruned because no better solution can be branched down from this node. Go to step 1.

(b) If the LP optimum y^k_{LP} is noninteger, go to step 6.

Step 6 (Branching). From the current node S^k choose a variable y_j with fractional value to generate two subproblems S_1^k and S_2^k defined by

$$S_1^k = S^k \cap \{y : y_j \leq \lfloor \bar{y}_j \rfloor\}$$
$$S_2^k = S^k \cap \{y : y_j \geq \lceil \bar{y}_j \rceil\}$$

Place both of these two nodes in the active list and go to step 1.

11.2 CUTTING PLANE APPROACH

In geometry, an equation in two variables is called a *plane* and an equation in n variables a *hyperplane*, strictly speaking. For simplicity, however, both in practice are often referred to as a plane, regardless of the number of variables. Strictly speaking, an inequality constraint in n variables is called a *half-space*, not a hyperplane. But an inequality constraint can always be converted to an equation by adding or subtracting a nonnegative slack variable. The term *cutting plane* is often used for an equality or inequality constraint that can cut off a fractional part of an LP feasible region, without excluding any *integer* feasible solution. In the cutting plane approach, one or more such cutting planes are added to the current LP simplex tableau, which in turn are resolved for a new LP optimum. This process is repeated until the prescribed integer requirements are satisfied. In this text, the collection of all such cutting plane methods will be called a *cutting plane approach* (more specifically, a *dual cutting plane approach*, due to the use of the dual simplex method for LP reoptimization).

11.2.1 Dual Cutting Plane Approach

A large variety of cutting plane methods were developed during the 1950s and 1960s. Among them, the most prominent ones belong to the class of *dual* cutting plane approach such as the fractional and mixed cutting plane methods developed by Gomory. This class shares a common solution algorithm when they are utilized as a *stand-alone* solver.

Step 1. Solve the integer program as if it were a linear program. If it is infeasible, so is the integer program and then stop. Else if an LP optimal solution satisfying the integer requirements is found, then the IP is solved. Otherwise, go to step 2.

Step 2. Select a row to be a *generating row* (or *source row*) from the LP optimum simplex tableau.

Step 3. Derive a cut constraint from the generating row and augment it to the current tableau, resulting in a primal infeasible solution.

Step 4. Apply the dual simplex method to reoptimize the augmented linear program. If the new LP optimum satisfies the integer requirements, the original MIP program is solved. Otherwise, go to step 2.

^{k} choose a variable y_j with fractional
S_2^k defined by

$y_j \le \lfloor \bar{y}_j \rfloor\}$

$\ge \lceil \bar{y}_j \rceil\}$

st and go to step 1.

s called a *plane* and an equation in
simplicity, however, both in practice
of the number of variables. Strictly
es is called a *half-space*, not a hyper-
oe converted to an equation by adding
term *cutting plane* is often used for an
ff a fractional part of an LP feasible
olution. In the cutting plane approach,
current LP simplex tableau, which in
rocess is repeated until the prescribed
ne collection of all such cutting plane
ch (more specifically, a *dual cutting*
plex method for LP reoptimization).

veloped during the 1950s and 1960s.
g to the class of *dual* cutting plane
utting plane methods developed by
algorithm when they are utilized as

a linear program. If it is infeasible,
. Else if an LP optimal solution
l, then the IP is solved. Otherwise,

source row) from the LP optimum

ng row and augment it to the current
ion.

nize the augmented linear program.
er requirements, the original MIP

The main difference among various methods of cutting plane is *how* a cut constraint is generated. The main requirement is that a generated cut constraint must be *valid*, meaning that its addition will result in cutting off the current LP optimal point but will not eliminate any *integer feasible* solution. In other words, every valid cut has two properties:

- The current optimal solution to the LP relaxation problem will violate the cut constraints.
- Any feasible point to the corresponding IP or MIP problems will satisfy the cut constraint.

The class of dual cutting plane methods begins with an optimal LP solution and requires application of a series of dual simplex steps to reoptimize a series of new LP problems, each adding one or more constraints to the current simplex tableau (although some cuts may be dropped from later considerations).

There is another class of cutting plane methods known as the *primal cutting plane approach*. This approach commonly begins with a primal simplex tableau and creates a series of primal simplex tableaux, from which cuts are generated. As a result, all the subsequent simplex tableaux will remain primal feasibility and dual infeasibility. The primal simplex method is applied throughout the process until both primal and dual solutions are feasible, in which case an optimum solution if found. We shall not describe them in detail. The interested reader may refer the Section 11.5.

11.2.2 Fractional Cutting Plane Method

The fractional cutting plane method is capable of solving *pure* integer programs. This method requires that the starting IP problem must contain *all-integer coefficients* so that all slack variables, including those that are added after introduction of cuts, are ensured to be nonnegative integers. Note that the integer assumption of the starting IP problem does not limit the problem application because any coefficients consisting of *rational numbers* can always be made integers by multiplying an appropriate number.

The fractional cutting plane method begins with an *optimal* simplex tableau of the LP relaxation given below (recall that we use y_j to denote integer variables):

$$\begin{aligned}
\text{Maximize} \quad & z + \sum_k \bar{d}_k y_k = \bar{d}_0 & & \\
\text{subject to} \quad & y_{B_i} + \sum_k \bar{g}_{ik} y_k = \bar{b}_i & & i = 1, 2, \ldots, m \\
& y_{B_i} \ge 0 & & i = 1, 2, \ldots, m \\
& y_k = 0 & & k = 1, 2, \ldots, p
\end{aligned} \tag{11.1}$$

where y_{B_i} and y_k denote basic variables and nonbasic variables, respectively. Note that the current LP optimum solution is $y_{B_i} = \bar{b}_i$ and $y_k = 0$, in which some $\bar{b}_i$ are assumed to be noninteger. Optimality conditions require that $\bar{d}_k \ge 0$ for all k.

To find an integer solution, we arbitrarily select a row with $\bar{b}_i$ noninteger. The selected row, say r, is called a *source row* or *generating row*, from which a fractional cut will be generated. Consider the source row

$$y_{B_r} + \sum_k \bar{g}_{rk} y_k = \bar{b}_r \quad (k = 1, \ldots, p)$$

which can be rewritten by separating fractional and integral parts of all data:

$$y_{B_r} + \sum_k \{(\bar{g}_{rk} - \lfloor \bar{g}_{rk} \rfloor) + \lfloor \bar{g}_{rk} \rfloor\} y_k = (\bar{b}_r - \lfloor \bar{b}_r \rfloor) + \lfloor \bar{b}_r \rfloor$$

where $\lfloor a \rfloor$ denotes the largest integer $\leq a$. For example, $\lfloor 5.4 \rfloor = 5$, $\lfloor -1.8 \rfloor = -2$, and $\lfloor 3 \rfloor = 3$. The fractional part is always ≥ 0. For simplicity, we let

$$f_{rk} = \bar{g}_{rk} - \lfloor \bar{g}_{rk} \rfloor$$
$$f_{r0} = b_r - \lfloor b_r \rfloor$$

be the fractional parts of tableau coefficients and the RHS of row r. Rearranging the terms, we have

$$y_{B_r} + \sum_k \lfloor \bar{g}_{rk} \rfloor y_k - \lfloor \bar{b}_r \rfloor = f_{r0} - \sum_k f_{rk} y_k \tag{11.2}$$

Now in order for y_{B_r} and y_k ($k = 1, \ldots, p$) to be integer, both the left-hand and right-hand sides of (11.2) must be integer. By the definition of congruence, we have

$$f_{r0} - \sum_k f_{rk} y_k = 0 \ (\mathrm{mod}\ 1) \tag{11.3}$$

But since $f_{r0} - \sum_k f_{rk} y_k \leq f_{r0} < 1$, the necessary condition for integrality becomes

$$f_{r0} - \sum_k f_{rk} y_k \leq 0$$

$$\text{or} \quad \sum_k f_{rk} y_k = f_{r0} \quad \text{(Gomory fractional cutt)} \tag{11.4}$$

$$\text{or} \quad \sum_k -f_{rk} y_k + s = -f_{r0} \quad \text{(Gomory fractional cut)} \tag{11.5}$$

where $s \geq 0$ is Gomory's slack variable associated with the fractional cut. Note that all f_{rk} and f_{r0} must be nonnegative fractions, that is, $0 \leq f_{rk}$ ($k = 1, \ldots, p$) and $f_{r0} \leq 1$.

Example 11.3 Solve the pure IP problem in (11.1) using the cutting plane method.

elect a row with $\bar{b}_i$ noninteger. The *erating row*, from which a fractional

$= 1, \ldots, p)$

and integral parts of all data:

$= (\bar{b}_r - \lfloor \bar{b}_r \rfloor) + \lfloor \bar{b}_r \rfloor$

mple, $\lfloor 5.4 \rfloor = 5$, $\lfloor -1.8 \rfloor = -2$, and simplicity, we let

$\rfloor$

$]$

l the RHS of row r. Rearranging the

$f_{r0} - \sum_k f_{rk} y_k$ (11.2)

) be integer, both the left-hand and e definition of congruence, we have

mod 1) (11.3)

y condition for integrality becomes

0

y fractional cutt) (11.4)

mory fractional cut) (11.5)

with the fractional cut. Note that all , $0 \le f_{rk}$ ($k = 1, \ldots, p$) and $f_{r0} \le 1$.

.1) using the cutting plane method.

TABLE 11.1 Tableau 1 for Example 11.3

Basic Variable	z	y_1	y_2	y_3	y_4	y_5	RHS
z	1	0	0	0	17/7	16/7	179/7
y_1	0	1	0	0	3/7	2/7	39/7
y_2	0	0	1	0	−1/7	−3/7	8/7
y_3	0	0	0	1	5/7	8/7	58/7

We first add a nonnegative slack variable to each inequality constraint:

$$\begin{aligned} \text{Maximize} \quad & z = 5y_1 - 2y_2 \\ \text{subject to} \quad & -y_1 + 2y_2 + y_3 = 5 \\ & 3y_1 + 2y_2 + y_4 = 19 \\ & -y_1 - 3y_2 + y_5 = -9 \\ & y_1, y_2, y_3, y_4, y_5 \ge 0 \text{ and integer} \end{aligned}$$

We then solve the LP relaxation, yielding the optimal simplex tableau shown in Table 11.1.

The optimum solution is noninteger: $y_1 = 39/7$, $y_2 = 8/7$, $y_3 = 58/7$, $y_4 = y_5 = 0$, and $z = 179/7$. We arbitrarily select y_1 row as the source row and generate the following fractional cut.

$$\frac{3}{7}y_4 + \frac{2}{7}y_5 \ge \frac{4}{7}$$

or

$$-\frac{3}{7}y_4 - \frac{2}{7}y_5 + s_1 = -\frac{4}{7}$$

where s_1 is called *Gomory's slack variable* to differentiate from the ordinary slack variable. Appending the equation to tableau 1, we obtain tableau 2 (Table 11.2).

Applying the dual simplex iteration, s_1 is replaced by y_4 yielding tableau 3 (Table 11.3).

Repeating generation of fractional cuts and application of dual simplex iterations, the reader may verify the subsequent simplex tableaux 4 through 6 (Tables 11.4–11.6).

TABLE 11.2 Tableau 2 for Example 11.3

Basic Variable	z	y_1	y_2	y_3	y_4	y_5	s_1	RHS
z	1	0	0	0	17/7	16/7	0	179/7
y_1	0	1	0	0	3/7	2/7	0	39/7
y_2	0	0	1	0	−1/7	−3/7	0	8/7
y_3	0	0	0	1	5/7	8/7	0	58/7
s_1	0	0	0	0	−3/7	−2/7	1	−4/7

TABLE 11.3 Tableau 3 for Example 11.3

Basic Variable	z	y_1	y_2	y_3	y_4	y_5	s_1	RHS
z	1	0	0	0	0	2/3	17/3	67/3
y_1	0	1	0	0	0	0	1	5
y_2	0	0	1	0	0	−1/3	−1/3	4/3
y_3	0	0	0	1	0	2/3	5/3	22/3
y_4	0	0	0	0	1	2/3	−7/3	4/3

TABLE 11.4 Tableau 4 for Example 11.3

Basic Variable	z	y_1	y_2	y_3	y_4	y_5	s_1	s_2	RHS
z	1	0	0	0	0	2/3	17/3	0	67/3
y_1	0	1	0	0	0	0	1	0	5
y_2	0	0	1	0	0	−1/3	−1/3	0	4/3
y_3	0	0	0	1	0	2/3	5/3	0	22/3
y_4	0	0	0	0	1	2/3	−7/3	0	4/3
s_2	0	0	0	0	0	−2/3	−2/3	1	−1/3

TABLE 11.5 Tableau 5 for Example 11.3

Basic Variable	z	y_1	y_2	y_3	y_4	y_5	s_1	s_2	RHS
z	1	0	0	0	0	0	5	1	22
y_1	0	1	0	0	0	0	1	0	5
y_2	0	0	1	0	0	0	0	−1/2	3/2
y_3	0	0	0	1	0	0	1	1	3
y_4	0	0	0	0	1	0	−3	1	1
y_5	0	0	0	0	0	1	1	−3/2	1/2
s_3	0	0	0	0	0	0	0	−1/2	−1/2

TABLE 11.6 Tableau 6 for Example 11.3

Basic Variable	z	y_1	y_2	y_3	y_4	y_5	s_1	s_2	s_3	RHS
z	1	0	0	0	0	0	5	0	2	21
y_1	0	1	0	0	0	0	1	0	0	5
y_2	0	0	1	0	0	0	0	0	1	2
y_3	0	0	0	1	0	0	1	0	2	2
y_4	0	0	0	0	1	0	−3	0	2	0
y_5	0	0	0	0	0	1	1	0	−3	2
s_2	0	0	0	0	0	0	0	1	−2	1

y_4	y_5	s_1	RHS
0	2/3	17/3	67/3
0	0	1	5
0	−1/3	−1/3	4/3
0	2/3	5/3	22/3
1	2/3	−7/3	4/3

y_5	s_1	s_2	RHS
2/3	17/3	0	67/3
0	1	0	5
−1/3	−1/3	0	4/3
2/3	5/3	0	22/3
2/3	−7/3	0	4/3
−2/3	−2/3	1	−1/3

y_4	y_5	s_1	s_2	RHS
0	0	5	1	22
0	0	1	0	5
0	0	0	−1/2	3/2
0	0	1	1	3
1	0	−3	1	1
0	1	1	−3/2	1/2
0	0	0	−1/2	−1/2

y_5	s_1	s_2	s_3	RHS
0	5	0	2	21
0	1	0	0	5
0	0	0	1	2
0	1	0	2	2
0	−3	0	2	0
1	1	0	−3	2
0	0	1	−2	1

Because all basic variables are integer, we have an integer optimum $y_1 = 5$, $y_2 = 2$, and $z = 21$. The solution is the same as that obtained by branch-and-bound approach.

The cutting plane approach often takes a large number of cuts to reach an integer solution even for a small or moderate sized IP problem, although it can be shown that the fractional cutting plane method is ensured to converge to an IP optimum after a finite number of cuts. Here, we arbitrarily select a source row, although alternative rules may be applied to select other source rows. For example, we may select a source row r with f_{r0} closest to 0.5, or select a row with the largest f_{r0}. However, no evidence shows that a certain selection rule is better than the others in all cases.

11.2.3 Mixed Integer Cutting Plane Method

The mixed integer cutting plane method, also due to Gomory, can be used to solve the following MIP problem:

$$\begin{aligned}
\text{Maximize} \quad & z = \sum_j c_j x_j + \sum_k d_k y_k \\
\text{subject to} \quad & \sum_j a_{ij} x_j + \sum_k g_{ik} y_k \le b_i \quad (i = 1, 2, \ldots, m) \\
& x_j \ge 0 \quad (j = 1, 2, \ldots, n) \\
& y_k \ge 0 \text{ and integer} \quad (k = 1, 2, \ldots, p)
\end{aligned}$$

Essentially, the solution procedure is similar to that of the fractional cutting plane method. It generates a valid cut from the optimal simplex tableau of the LP relaxation of the MIP problem. Any row r associated with y_k, which is basic but has fractional right-hand side, may be chosen to generate the cut. Just like the fractional cuts, each of the generated mixed cuts will violate primal feasibility and will be restored to primal feasibility after applying the dual simplex method.

Let the nonzero coefficients ($\bar{a}_{rj}$) of the continuous variables x_j ($j \in J$) be partitioned into two sets: positive coefficients ($\bar{a}_{rj} > 0$) and negative coefficients ($\bar{a}_{rj} < 0$). Also, let $f_{rk} = \bar{g}_{rk} - \lfloor \bar{g}_{rk} \rfloor$ and $f_{r0} = \bar{b}_r - \lfloor \bar{b}_r \rfloor$ as before. It can be shown that a mixed cut due to Gomory can be derived (see Section 11.5) as

$$\sum_{j:\bar{a}_{rj}>0} \bar{a}_{rj} x_j + \sum_{j:\bar{a}_{rj}<0} \left(\frac{f_{r0}}{f_{r0}-1}\right) \bar{a}_{rj} x_j + \sum_{k:f_{rk} \le f_{r0}} f_{rk} y_k + \sum_{k:f_{rk} > f_{r0}} \frac{f_{r0}(1-f_{rk})}{1-f_{r0}} y_k \ge f_{r0}$$

We use the following numerical problem to show how to generate a mixed integer cut. The remaining procedure will be similar to that in Example 11.3, except that no rows corresponding to continuous variables are used for source rows to generate cuts.

Example 11.4 Solve the given MIP problem using a cutting plane approach.

$$\begin{aligned} \text{Maximize} \quad & z = 5x_1 + 3x_2 + 7y_1 + 2y_2 \\ \text{subject to} \quad & 7x_1 + 8x_2 + 9y_1 + 3y_2 \le 43 \\ & 11x_1 + 4x_2 + 4y_1 + 5y_2 \le 51 \\ & \mathbf{x} \ge \mathbf{0} \\ & \mathbf{y} \ge \mathbf{0} \text{ and integer} \end{aligned}$$

Solving the LP relaxation, we obtain an LP optimum $y_1 = 43/9$, with the following source row:

$$\frac{7}{9}x_1 + \frac{8}{9}x_2 + y_1 + \frac{1}{3}y_2 + \frac{1}{9}s_1 = \frac{43}{9}$$

Here, we have all positive coefficients and no negative coefficients for the continuous variables. Compute

$$\begin{aligned} f_{r0} &= \frac{7}{9} \\ f_{r1} &= 0 \\ f_{r2} &= \frac{1}{3} \end{aligned}$$

and we obtain a mixed integer cut

$$\frac{7}{9}x_1 + \frac{8}{9}x_2 + \frac{1}{3}y_2 + \frac{1}{9}s_1 \ge \frac{7}{9}$$

In Exercise 11.10, the reader is asked to continue on this example.

11.3 GROUP THEORETIC APPROACH

Gomory showed that the coefficient *row* vectors of the derived inequalities form a finite set that is closed under the operation of addition when the arithmetic operations are taken modulo 1 (i.e., integer parts are dropped). Such a set forms what is called a *group*. Furthermore, this group can have at most D elements, where D is the absolute value of the determinant of the current LP basis. If the starting basis is an identity matrix, then this group contains exactly D elements.

Gomory also showed that by relaxing nonnegative (but not integer) requirements of the current basic variables, an integer program can be transformed into one in which the *columns* of constraint coefficients and the right-hand side are elements of an abelian group. If this group problem (in terms of nonbasic variables only) is solved and a solution containing nonnegative values for all variables is obtained, then the