second is obtained by adding the constraint $v \geq b + 1$. We need not do this for *all* the variables that have fractional values, just for one each time. We are guaranteed that every time we do this we will lose no integral solutions of the linear subprogram. Furthermore, if we join the integral solutions of the subprograms together, we get the integral solutions of the original linear program. Let us illustrate.

EXAMPLE 1 Suppose we want to

$$\begin{aligned} \text{Maximize} \quad & u = 3x + 4y, \\ \text{s.t.} \quad & 4x + 3y \leq 13, \\ & 3x + 2y \leq 7, \\ & x, y \geq 0 \quad \text{and integral.} \end{aligned}$$

If we solve the LP relaxation,

$$\begin{aligned} \text{Maximize} \quad & u = 3x + 4y, \\ \text{s.t.} \quad & 4x + 3y \leq 13, \\ & 3x + 2y \leq 7, \\ & x, y \geq 0. \end{aligned} \tag{27}$$

we obtain $x = 0$ and $y = 3.5$. The objective at this point is 14. In view of the fact that y must be integral, it follows that either $y \leq 3$ or $y \geq 4$. We form two subprograms: the first subprogram, LP1, consists of (27) together with the constraint $y \leq 3$; the second linear program, LP2, consists of the (27) together with the constraint $y \geq 4$. We indicate this as in Fig. 10.5. The circles are called nodes, and lines joining nodes to other nodes are called

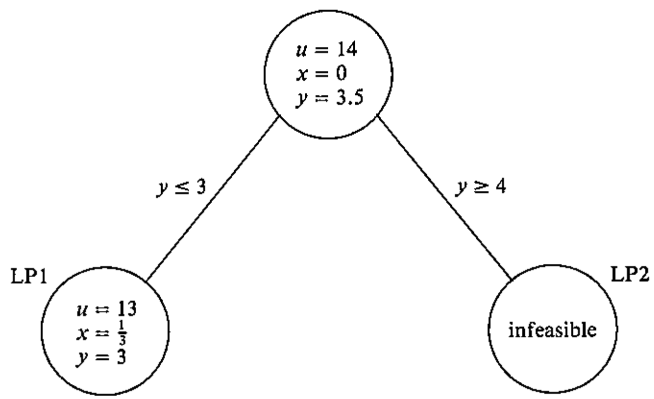


Figure 10.5.

branches. If we solve LP1, we find that the optimal solution is $x = 1/3$ and $y = 3$. Here, the objective function is 13. If we solve LP2, we find that the program is infeasible. Since we can get no further information from the node corresponding to LP2, we drop it from further consideration. Any node dropped in this way, or any node no longer in use, is called a *fathomed* node. Any other node is called a *dangling* node. The node corresponding to LP1 is dangling at this point.

Since the optimal solution of LP1 requires that $x = 1/3$ and we know that x must be integral, we branch on LP1 to form two programs, LP3 and LP4. LP3 consists of LP2 together with the constraint $x \leq 0$, and LP4 consists of LP2 together with the constraint $x \geq 1$. We note that since all variables are ≥ 0, the constraint $x \leq 0$ in LP3 forces x to be equal to zero. Our picture now is shown in Fig. 10.6.

Solving LP3, we find that $x = 0$, $y = 3$, and $u = 12$. We have found an integral solution that makes the objective function equal to 12. Thus, at this point our best integral solution is for LP3, and we know that the optimal value of the original linear program must at least 12. What about LP4? Is it possible that there is an integral solution to LP4 that is greater than 12? Theoretically, there is nothing to stop this from happening, and so we must solve LP4 also. There we find the solution $x = 1$ and $y = 2$. The objective value at this point is 11. The question facing us now is whether we branch

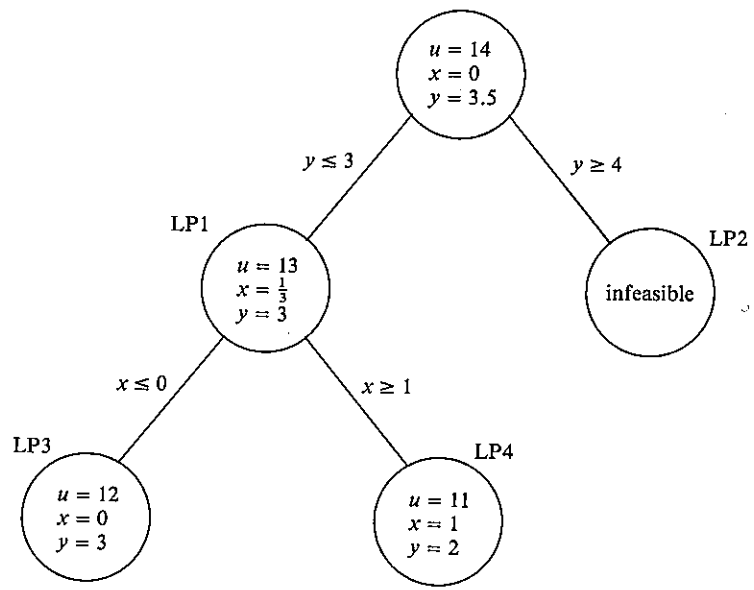


Figure 10.6.

again on these nodes to perhaps get better solutions. The answer is no, for the following reason: Whenever we add a constraint to a maximum program, the value of the objective function can only stay the same or decrease. Thus, to branch on node 4 makes no sense since our new program can only have an objective ≤ 11, and we already have a better value of the objective at LP3. Thus, node 4 is fathomed, as we can get no further useful information from it. Since adding a constraint can only decrease the objective in LP3, we have also fathomed that node. Thus, it no longer pays to branch further on any nodes, and the current best integral solution, $x = 0$, $y = 3$, is the optimal solution.

Let us give another example.

EXAMPLE 2 We wish to

$$\begin{aligned} \text{Maximize} \quad & u = 4x + 5y + 6z, \\ \text{s.t.} \quad & 3x + 2y + z \leq 9, \\ & 2x + y + 4z \leq 7, \\ & x, y, z \geq 0 \text{ and integral.} \end{aligned}$$

The solution process is summarized in Fig. 10.7. Let us go through the process. When we solve the LP relaxation, we obtain $u = 25$, $x = 0$, $y \approx 4.14$, $z \approx 0.71$. Since y is not integral, we may branch on y. The two branches are obtained by adding the constraint $y \leq 4$ to the LP relaxation to get node 1, and adding the constraint $y \geq 5$ to the LP relaxation to get node 2. Solving the program at node 1, we get $u = 24.6$, $x = 0.1$, $y = 4$, $z = 0.7$. The program corresponding to node 2 is infeasible. Now, node 1 is dangling, and we may branch on x to get nodes 3 and 4. Solving the program corresponding to node 3, we get $u = 24.5$, $x = 0$, $y = 4$, $z = 0.75$. Solving the program corresponding to node 4, we obtain $u = 21$, $x = 1$, $y \approx 2.7$, $z \approx 0.57$. So far we have no integral solutions. Nodes 3 and 4 are still dangling. We branch on node 3 to get nodes 5 and 6. The solution of the program at node 5 is integral. There $u = 20$, $x = z = 0$, and $y = 4$. The solution at node 6 is also integral: $u = 21$, $x = 0$, $y = 3$, $z = 1$. At this point, our best integral solution occurs at node 6. Nodes 5 and 6 are fathomed. Branching further on either of them will only serve to decrease the objective. Node 4 is still dangling; but there is no sense in branching on that node, since branching can only lead to an objective ≤ 21, and we have already obtained an integral solution where $u = 21$. Thus, we may consider node 4 fathomed. Since all nodes are fathomed, we have reached our optimal solution. It occurs at node 6, and it is $u = 21$ when $x = 0$, $y = 3$, and $z = 1$.

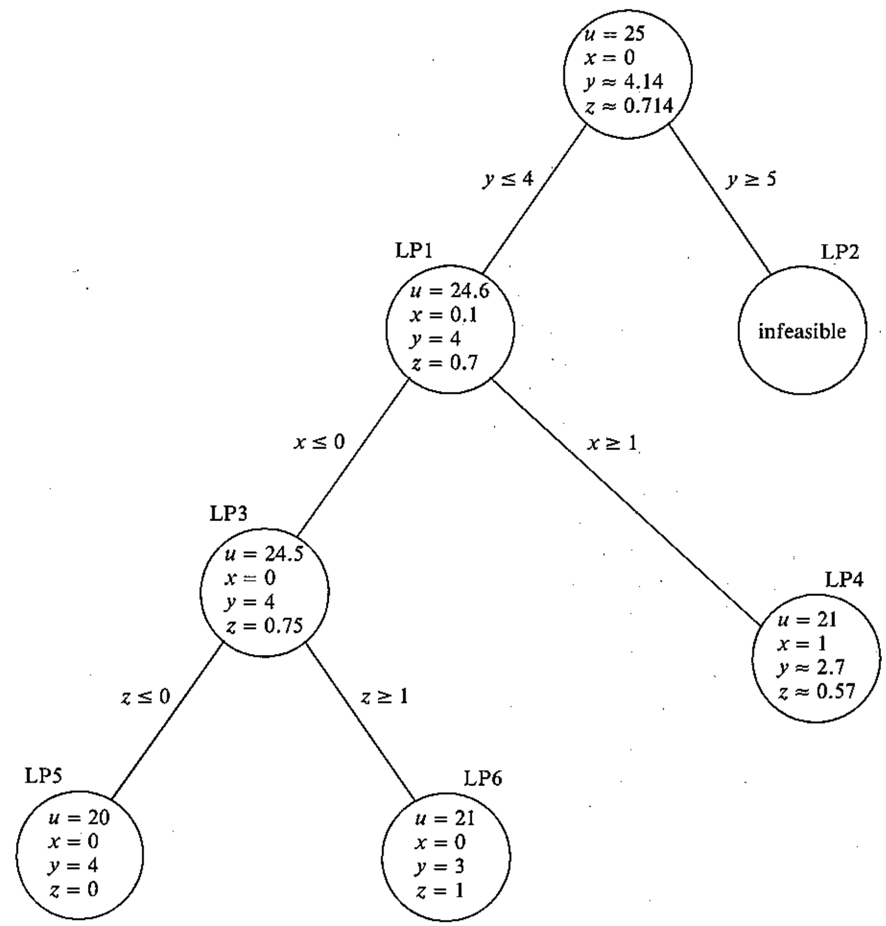


Figure 10.7.

One advantage of the branch and bound procedure is that as we proceed with it we often generate integral solutions along the way that are pretty good. If the solutions are acceptable to us, even though they are not optimal, we may stop at that point. This is especially useful when the branch and bound tree becomes very large. Also, sometimes we can obtain an integral solution of the program by inspection. This helps because then many nodes that we might have had to fathom otherwise will not need to be fathomed, because we know that they will not benefit us. For example, if in some problem we obtained $u = 45$ when $x = y = z = 3$, and we did this by inspection, then any node where u is less than or equal to 45 need not be studied. It is considered fathomed.

close the rounded solution is to the true optimal solution. Thus, if $d = 0.01$, our true solution is within 1% of the rounded solution, and so we are close to the optimal solution. For that reason, it probably pays to round when d is sufficiently small. In a similar manner, if u_I represents the current best integral value of u when using the branch and bound procedure, then when

$$\frac{u^* - u_I}{u_I}$$

is small, say β, we can be assured that the optimal integral solution is within $100\beta\%$ of u_I. That is, $u_I \le u_I^* \le (1 + \beta)u_I$.

EXERCISES 10.4

1. Use the branch and bound method to solve each of the following. Draw the branch and bound tree for each problem.

(a) Minimize $u = 5x + y$, s.t.
$3x + 2y \ge 4$,
$x \ge 2$,
$y \ge 0$,
x and y are integral.

(b) Minimize $u = 5x + y$, s.t.
$1.5 \le x \le 3.4$,
$2.1 \le y \le 2.7$,
x and y are integral.

(c) Maximize $u = 2x - y$, s.t.
$x + 2y \le 5$,
$3x - y \le 7$,
$x, y \ge 0$ and integral.

(d) Maximize $u = 3x + 4y$, s.t.
$2x + 3y \le 7$,
$x, y \ge 0$ and integral.

(e) Maximize $u = 2x + 3y$, s.t.
$x \ge y$,
$x + 2y \le 6$,
$2x + y \le 8$,
$x, y \ge 0$ and integral.

(f) Maximize $u = 5x - y - z$, s.t.
$2x + y + z \le 4$,
$x + 3y + 4z \le 1$,
$x, y, z \ge 0$ and integral.

(g) Maximize $u = 3x + 4y + 5z$, s.t.
$2x + 3y + z \le 6$,
$x + 3y + 4z \le 5$,
$x, y, z \ge 0$ and integral.

(h) Maximize $u = 3x - 2y + z$, s.t.
$4x + y + z \le 6$,
$3x + 2y + 3z \le 4$,
$x, y, z \ge 0$ and integral.

CHAPTER 11
NETWORK ANALYSIS

11.1 INTRODUCTION AND DEFINITIONS

An area of mathematics that has grown tremendously in the last 100 years is the subject of graph theory. The number of varied applications of this subject is enormous and continues to grow. In this chapter, we will study a special subdivision of graph theory that is closely connected to linear programming—network analysis. We will be brief, since our goal is only to show how linear programming may be used in other areas. More detailed discussions of the topics in this chapter may be found in operations research texts, management science texts, and, of course, graph theory and network analysis texts.

Loosely speaking, a *graph* is a collection of objects, called nodes or vertices (represented by dots or circles), together with a set of edges. What characterizes an edge is that it joins two vertices. (But not every two vertices need be joined by an edge.) Several examples are given in Fig. 11.1. In Fig. 11.1a, the graph has four vertices, labelled 1, 2, 3, and 4, and two edges. In Fig. 11.1b, the graph has four vertices and three edges; while in Fig. 11.1c, the graph has three vertices and one edge.

These and other questions will be discussed further once we have seen an example.

WOLSEY
"Integer Programming"

7.3 BRANCH AND BOUND: AN EXAMPLE

The most common way to solve integer programs is to use implicit enumeration, or *branch and bound*, in which linear programming relaxations provide the bounds. We first demonstrate the approach by an example:

$$z = \max 4x_1 - x_2 \tag{7.1}$$
$$7x_1 - 2x_2 \leq 14 \tag{7.2}$$
$$x_2 \leq 3 \tag{7.3}$$
$$2x_1 - 2x_2 \leq 3 \tag{7.4}$$
$$x \in Z^2_+. \tag{7.5}$$

Bounding. To obtain a first upper bound, we add slack variables x_3, x_4, x_5 and solve the linear programming relaxation in which the integrality constraints are dropped. The resulting optimal basis representation is:

$$\begin{array}{lccccccc}
\bar{z} = \max \frac{59}{7} & & & -\frac{4}{7}x_3 & -\frac{1}{7}x_4 & & & \\
 & x_1 & & +\frac{1}{7}x_3 & +\frac{2}{7}x_4 & & = & \frac{20}{7} \\
 & & x_2 & & +x_4 & & = & 3 \\
 & & & -\frac{2}{7}x_3 & +\frac{10}{7}x_4 & +x_5 & = & \frac{23}{7} \\
 & x_1, & x_2, & x_3, & x_4, & x_5 & \geq & 0.
\end{array}$$

Thus we obtain an upper bound $\bar{z} = \frac{59}{7}$, and a nonintegral solution $(\bar{x}_1, \bar{x}_2) = (\frac{20}{7}, 3)$. Is there any straightforward way to find a feasible solution? Apparently not. By convention, as no feasible solution is yet available, we take as lower bound $\underline{z} = -\infty$.

Branching. Now because $\underline{z} < \bar{z}$, we need to divide or branch. How should we split up the feasible region? One simple idea is to choose an integer variable that is basic and fractional in the linear programming solution, and split the problem into two about this fractional value. If $x_j = \bar{x}_j \notin Z^1$, one can take:

$$S_1 = S \cap \{x : x_j \leq \lfloor \bar{x}_j \rfloor\}$$
$$S_2 = S \cap \{x : x_j \geq \lceil \bar{x}_j \rceil\}.$$

It is clear that $S = S_1 \cup S_2$ and $S_1 \cap S_2 = \phi$. Another reason for this choice is that the solution $\bar{x}$ of $LP(S)$ is not feasible in either $LP(S_1)$ or $LP(S_2)$. This implies that if there is no degeneracy (i.e., multiple optimal LP solutions), then $\max\{\bar{z}_1, \bar{z}_2\} < \bar{z}$, so the upper bound will strictly decrease.

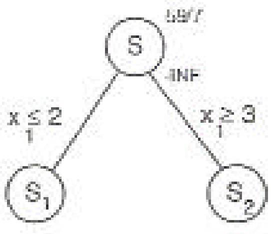


Fig. 7.6 Partial branch-and-bound tree 1

Following this idea, as $\bar{x}_1 = 20/7 \notin Z^1$, we take $S_1 = S \cap \{x : x_1 \leq 2\}$ and $S_2 = S \cap \{x : x_1 \geq 3\}$. We now have the tree shown in Figure 7.6. The subproblems (nodes) that must still be examined are called *active*.

Choosing a Node. The list of active problems (nodes) to be examined now contains S_1, S_2. We arbitrarily choose S_1.

Reoptimizing. How should we solve the new modified linear programs $LP(S_i)$ for $i = 1, 2$ without starting again from scratch?

As we have just added one single upper or lower bound constraint to the linear program, our previous optimal basis remains dual feasible, and it is therefore natural to reoptimize from this basis using the dual simplex algorithm. Typically, only a few pivots will be needed to find the new optimal linear programming solution.

Applying this to the linear program $LP(S_1)$, we can write the new constraint $x_1 \leq 2$ as $x_1 + s = 2, s \geq 0$, which can be rewritten in terms of the nonbasic variables as

$$-\tfrac{1}{7}x_3 - \tfrac{2}{7}x_4 + s = -\tfrac{6}{7}.$$

Thus we have the dual feasible representation:

$$\begin{array}{lccccccrl}
\overline{z}_1 = \max \frac{59}{7} & & & -\frac{4}{7}x_3 & -\frac{1}{7}x_4 & & & & \\
& x_1 & & +\frac{1}{7}x_3 & +\frac{2}{7}x_4 & & & = & \frac{20}{7} \\
& & x_2 & & +x_4 & & & = & 3 \\
& & & -\frac{2}{7}x_3 & +\frac{10}{7}x_4 & +x_5 & & = & \frac{23}{7} \\
& & & -\frac{1}{7}x_3 & -\frac{2}{7}x_4 & & +s & = & -\frac{6}{7} \\
& x_1, & x_2, & x_3, & x_4, & x_5, & s & \geq & 0.
\end{array}$$

After two simplex pivots, the linear program is reoptimized, giving:

$$\begin{array}{lccccccl}
\overline{z}_1 = \max \frac{15}{2} & & & & & -\frac{1}{2}x_5 & -3s & \\
& x_1 & & & & & +s & = 2 \\
& & x_2 & & & -\frac{1}{2}x_5 & +s & = \frac{1}{2} \\
& & & x_3 & & -x_5 & -5s & = 1 \\
& & & & x_4 & +\frac{1}{2}x_5 & +6s & = \frac{5}{2} \\
& x_1, & x_2, & x_3, & x_4, & x_5, & s & \geq 0
\end{array}$$

with $\overline{z}_1 = \frac{15}{2}$, and $(\overline{x}_1^1, \overline{x}_2^1) = (2, \frac{1}{2})$.

Branching. S_1 cannot be pruned, so using the same branching rule as before, we create two new nodes $S_{11} = S_1 \cap \{x : x_2 \leq 0\}$ and $S_{12} = S_1 \cap \{x : x_2 \geq 1\}$, and add them to the node list. The tree is now as shown in Figure 7.7.

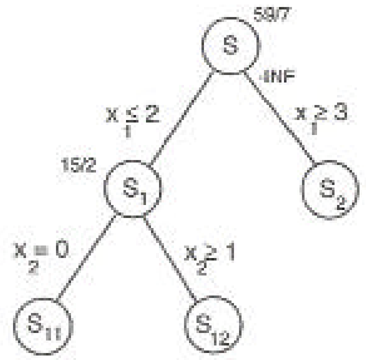


Fig. 7.7 Partial branch-and-bound tree 2

Choosing a Node. The active node list now contains S_2, S_{11}, S_{12}. Arbitrarily choosing S_2, we remove it from the node list and examine it in more detail.

Reoptimizing. To solve $LP(S_2)$, we use the dual simplex algorithm in the same way as above. The constraint $x_1 \geq 3$ is first written as $x_1 - t = 3, t \geq 0$, which expressed in terms of the nonbasic variables becomes:

$$\tfrac{1}{7}x_3 + \tfrac{2}{7}x_4 + t = -\tfrac{1}{7}.$$

From inspection of this constraint, we see that the resulting linear program

$$\begin{array}{lllllllll}
\overline{z}_2 = \max \frac{59}{7} & & & -\frac{4}{7}x_3 & -\frac{1}{7}x_4 & & & & \\
 & x_1 & & +\frac{1}{7}x_3 & +\frac{2}{7}x_4 & & & = & \frac{20}{7} \\
 & & x_2 & & +x_4 & & & = & 3 \\
 & & & -\frac{2}{7}x_3 & +\frac{10}{7}x_4 & +x_5 & & = & \frac{23}{7} \\
 & & & \frac{1}{7}x_3 & +\frac{2}{7}x_4 & & +t & = & -\frac{1}{7} \\
 & x_1, & x_2, & x_3, & x_4, & x_5, & t & \geq & 0
\end{array}$$

is infeasible, $\overline{z}_2 = -\infty$, and hence node S_2 is *pruned by infeasibility.*

Choosing a Node. The node list now contains S_{11}, S_{12}. Arbitrarily choosing S_{12}, we remove it from the list.

Reoptimizing. $S_{12} = S \cap \{x : x_1 \leq 2, x_2 \geq 1\}$. The resulting linear program has optimal solution $\overline{x}^{12} = (2, 1)$ with value 7. As $\overline{x}^{12}$ is integer, $z^{12} = 7$.

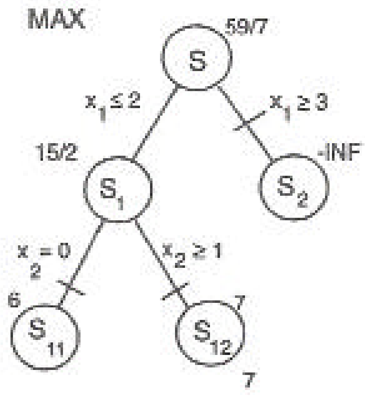


Fig. 7.8 Complete branch and bound tree

Updating the Incumbent. As the solution of $LP(S_{12})$ is integer, we update the value of the best feasible solution found $\underline{z} \leftarrow \max\{\underline{z}, 7\}$, and store the corresponding solution $(2,1)$. S_{12} is now *pruned by optimality.*

Choosing a Node. The node list now contains only S_{11}.

Reoptimizing. $S_{11} = S \cap \{x : x_1 \le 2, x_2 \le 0\}$. The resulting linear program has optimal solution $\overline{x}^{11} = (\frac{3}{2}, 0)$ with value 6. As $\underline{z} = 7 > \overline{z}_{11} = 6$, the node is *pruned by bound.*

Choosing a Node. As the node list is empty, the algorithm terminates. The incumbent solution $x = (2,1)$ with value $z = 7$ is optimal.

The complete branch-and-bound tree is shown in Figure 7.8. In Figure 7.9 we show graphically the feasible node sets S_i, the branching, the relaxations $LP(S_i)$, and the solutions encountered in the example.

7.4 LP-BASED BRANCH AND BOUND

In Figure 7.10 we present a flowchart of a simple branch and bound algorithm, and then discuss in more detail some of the practical aspects of developing and using such an algorithm.

Storing the Tree. In practice one does not store a tree, but just the list of *active* nodes or subproblems that have not been pruned and that still need to be explored further. Here the question arises of how much information one should keep. Should one keep a minimum of information and be prepared to repeat certain calculations, or should one keep all the information available? At a minimum, the best known dual bound and the variable lower and upper

Integer Programming by Branch-and-Bound

Example from [1988ECK], p 217 ff

Using Lindo semi-manually, the successive solutions are obtained, to observe the Branch-and-Bound methodology. The direct solution is:

PROBLEM

```
! Ecker & Kupferschmid, "A branch-and-bound example in detail", pp 218–225;
zMax = 15; X = (2, 3, 0)
max -3 x1 + 7 x2 + 12 x3
subject to
-3 x1 + 6 x2 + 8 x3 < 12
 6 x1 - 3 x2 + 7 x3 < 8
-6 x1 + 3 x2 + 3 x3 < 5
END
GIN 3
```

```
     OBJECTIVE FUNCTION VALUE
     1)      15.00000

 VARIABLE        VALUE          REDUCED COST
     X1        2.000000          3.000000
     X2        3.000000         -7.000000
     X3        0.000000        -12.000000
```

Branch-and-bound

PROBLEM 0

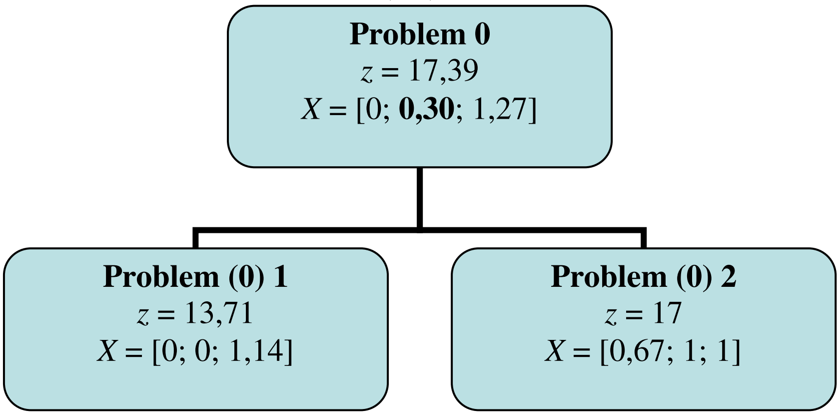
```
max -3 x1 + 7 x2 + 12 x3
subject to
-3 x1 + 6 x2 + 8 x3 < 12
 6 x1 - 3 x2 + 7 x3 < 8
-6 x1 + 3 x2 + 3 x3 < 5
END
```

```
LP OPTIMUM FOUND AT STEP      3
     OBJECTIVE FUNCTION VALUE

     1)      17.39394

 VARIABLE        VALUE          REDUCED COST
     X1        0.000000          0.590909
     X2        0.303030          0.000000
     X3        1.272727          0.000000
```

Branch around x_2: $x_2 \le 0$, $x_2 \ge 1$

PROBLEM (0) 1	**PROBLEM (0) 2**
max -3 x1 + 7 x2 + 12 x3 subject to -3 x1 + 6 x2 + 8 x3 < 12 6 x1 - 3 x2 + 7 x3 < 8 -6 x1 + 3 x2 + 3 x3 < 5 x2 < 0 END	max -3 x1 + 7 x2 + 12 x3 subject to -3 x1 + 6 x2 + 8 x3 < 12 6 x1 - 3 x2 + 7 x3 < 8 -6 x1 + 3 x2 + 3 x3 < 5 x2 > 1 END
LP OPTIMUM FOUND AT STEP 0 OBJECTIVE FUNCTION VALUE 1) 13.71429 VARIABLE VALUE REDUCED COST X1 0.000000 13.285714 X2 0.000000 0.000000 **X3** **1.142857** 0.000000	LP OPTIMUM FOUND AT STEP 1 OBJECTIVE FUNCTION VALUE 1) 17.00000 VARIABLE VALUE REDUCED COST **X1** **0.666667** 0.000000 X2 1.000000 0.000000 X3 1.000000 0.000000

$z_1 = 13{,}71$, $z_2 = 17$



P-2 is most promising and will give P-3 and P-4.

In the process of choosing the subproblem, it is impossible to predict the best choice. So, select the subproblem with the best, i.e., most promising, value of the objective function, z.

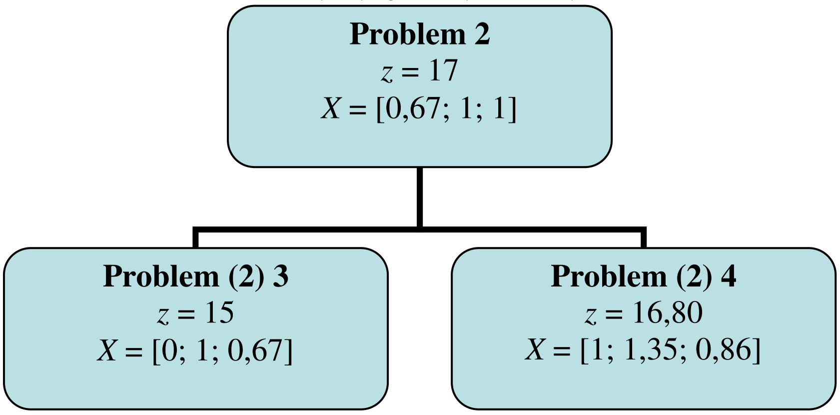
In the example, maybe we'll end up getting an integer solution with $z > z_1$ (not needing to explore its branches), so don't choose P-1.

Branch around x_1: $x_1 \le 0$, $x_1 \ge 1$.

PROBLEM (2) 3	**PROBLEM (2) 4**
max -3 x1 + 7 x2 + 12 x3 subject to -3 x1 + 6 x2 + 8 x3 < 12 6 x1 - 3 x2 + 7 x3 < 8 -6 x1 + 3 x2 + 3 x3 < 5 x2 > 1 x1 < 0 END	max -3 x1 + 7 x2 + 12 x3 subject to -3 x1 + 6 x2 + 8 x3 < 12 6 x1 - 3 x2 + 7 x3 < 8 -6 x1 + 3 x2 + 3 x3 < 5 x2 > 1 x1 > 1 END

LP OPTIMUM FOUND AT STEP 2 OBJECTIVE FUNCTION VALUE 1) 15.00000 VARIABLE VALUE REDUCED COST X1 0.000000 0.000000 X2 1.000000 0.000000 **X3 0.666667** 0.000000	LP OPTIMUM FOUND AT STEP 2 OBJECTIVE FUNCTION VALUE 1) 16.80303 VARIABLE VALUE REDUCED COST X1 1.000000 0.000000 **X2 1.348485** 0.000000 X3 0.863636 0.000000

$z_1 = 13{,}71$, $z_3 = 15$, $z_4 = 16{,}80$



P-4 is most promising and will give P-5 and P-6.

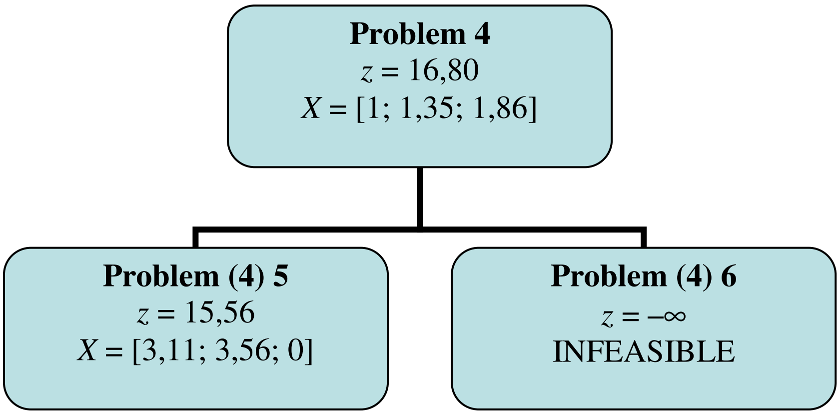
Branch around x_2: $x_2 \le 1$, $x_2 \ge 2$.

When there are more than one non-integer to branch from, choose the "least integer", i.e., the one nearest half unit (*e.g.*, 1,5 better than 1,4; 10,5 better than 11,5) — or arbitrarily.

Or (book) — Branch around x_3: $x_3 \le 0$, $x_3 \ge 1$.

PROBLEM (4) 5	PROBLEM (4) 6
max -3 x1 + 7 x2 + 12 x3 subject to -3 x1 + 6 x2 + 8 x3 < 12 6 x1 - 3 x2 + 7 x3 < 8 -6 x1 + 3 x2 + 3 x3 < 5 x2 > 1 x1 > 1 x3 < 0 END	max -3 x1 + 7 x2 + 12 x3 subject to -3 x1 + 6 x2 + 8 x3 < 12 6 x1 - 3 x2 + 7 x3 < 8 -6 x1 + 3 x2 + 3 x3 < 5 x2 > 1 x1 > 1 x3 > 1 END
LP OPTIMUM FOUND AT STEP 1 OBJECTIVE FUNCTION VALUE 1) 15.55556 VARIABLE VALUE REDUCED COST X1 3.111111 0.000000 **X2 3.555556** 0.000000 X3 0.000000 0.000000	INFEASIBLE

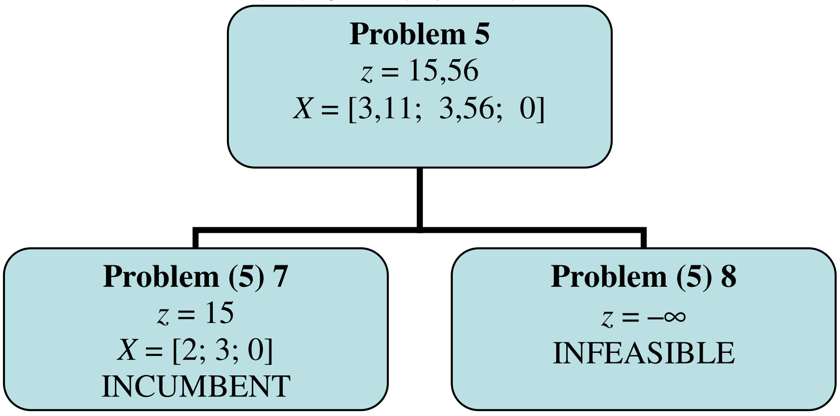
$z_1 = 13{,}71$, $z_3 = 15$, $z_5 = 15{,}56$.



P-5 is most promising and will give P-7 and P-8.
Branch around x_2: $x_2 \le 3$, $x_2 \ge 4$.

PROBLEM (5) 7	**PROBLEM (5) 8**
max -3 x1 + 7 x2 + 12 x3 subject to -3 x1 + 6 x2 + 8 x3 < 12 6 x1 - 3 x2 + 7 x3 < 8 -6 x1 + 3 x2 + 3 x3 < 5 x2 > 1 x1 > 1 x3 < 0 x2 < 3 END	max -3 x1 + 7 x2 + 12 x3 subject to -3 x1 + 6 x2 + 8 x3 < 12 6 x1 - 3 x2 + 7 x3 < 8 -6 x1 + 3 x2 + 3 x3 < 5 x2 > 1 x1 > 1 x3 < 0 x2 > 4 END
LP OPTIMUM FOUND AT STEP 4 OBJECTIVE FUNCTION VALUE 1) 15.00000 VARIABLE VALUE REDUCED COST X1 2.000000 0.000000 X2 3.000000 0.000000 X3 0.000000 0.000000 INCUMBENT	INFEASIBLE

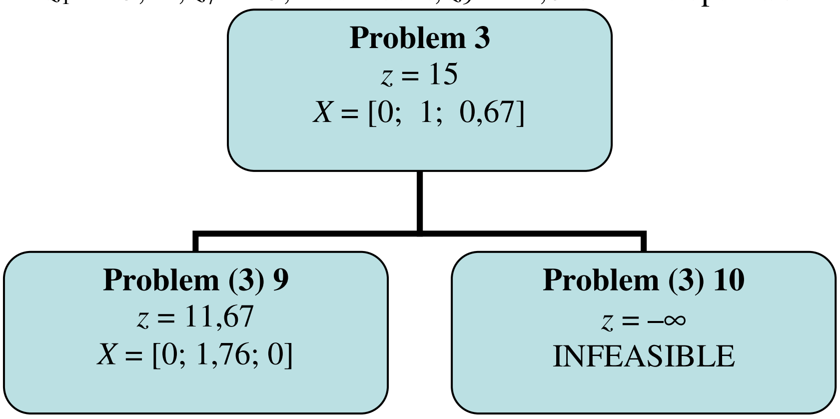
z_1 = 13,71, z_3 = 15, z_7 = 15, incumbent.



P-3 is most promising and will give P-9 and P-10.
(Nothing better than multiple solutions can be expected.)
Branch around x_3: $x_3 \le 0$, $x_3 \ge 1$.

PROBLEM (3) 9	**PROBLEM (3) 10**
max -3 x1 + 7 x2 + 12 x3 subject to -3 x1 + 6 x2 + 8 x3 < 12 6 x1 - 3 x2 + 7 x3 < 8 -6 x1 + 3 x2 + 3 x3 < 5 x2 > 1 x1 < 0 x3 < 0 END	max -3 x1 + 7 x2 + 12 x3 subject to -3 x1 + 6 x2 + 8 x3 < 12 6 x1 - 3 x2 + 7 x3 < 8 -6 x1 + 3 x2 + 3 x3 < 5 x2 > 1 x1 < 0 x3 > 1 END
LP OPTIMUM FOUND AT STEP 3 OBJECTIVE FUNCTION VALUE 1) 11.66667 VARIABLE VALUE REDUCED COST X1 0.000000 0.000000 **X2** **1.666667** 0.000000 X3 0.000000 0.000000	INFEASIBLE

z_1 = 13,71, z_7 = 15, incumbent, z_9 = 11,67: P-7 is optimal.



z_7 = 15, P-7 is optimal.

Compare with the solution in the book (next page). (Breadth-first, depth-first, mixed.)

In the 3.rd row of solutions, the rightmost (4.th) solution (“Infeasible”) is possibly wrong (without further influence).

Bibliography

–[1988ECK] Ecker, Joseph G., Michael Kupferschmid, 1988, “Introduction to Operations Research”, John Wiley, New York, NY (USA). ISBN 0-471-63362-3.

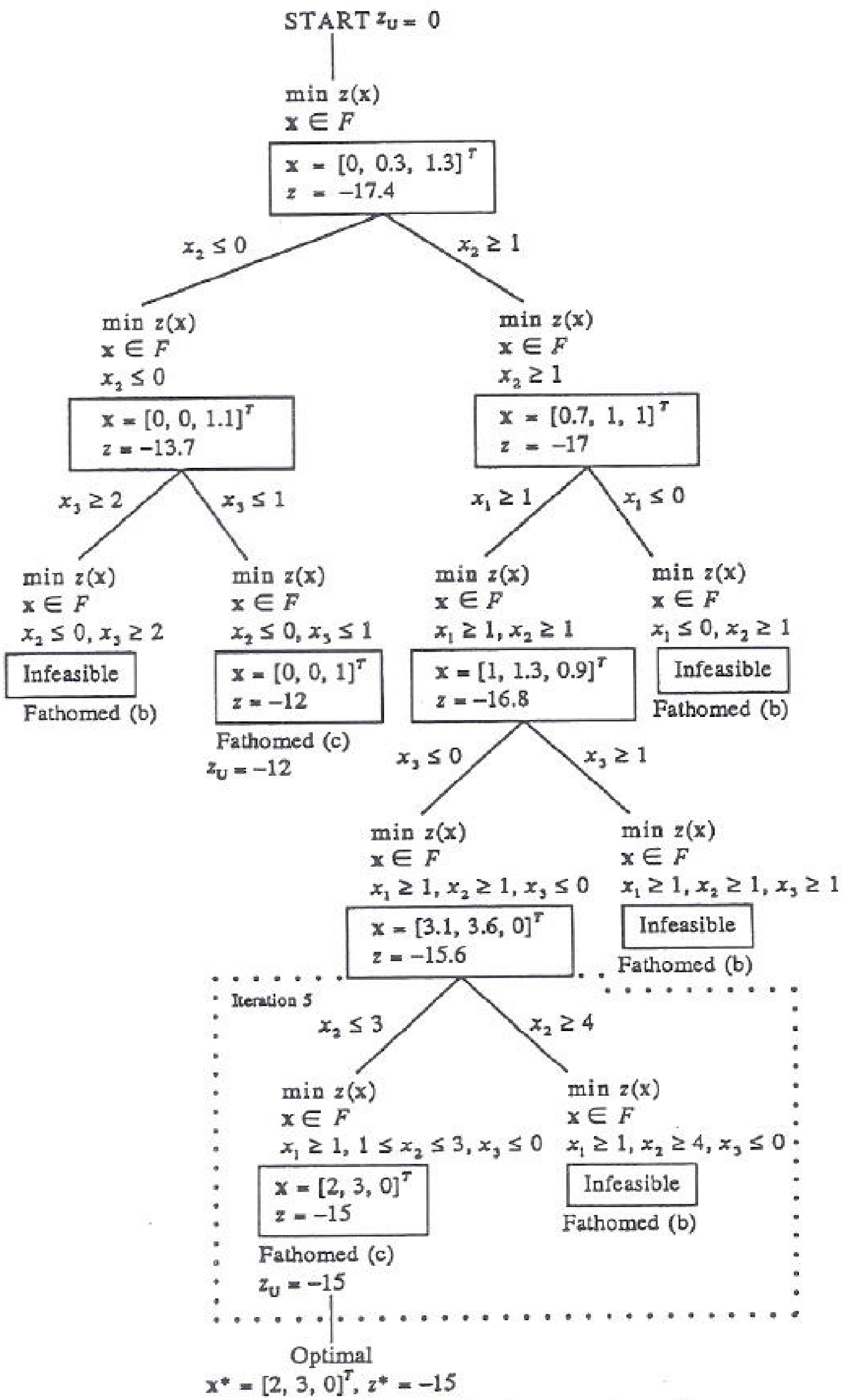


FIGURE 8.4e Complete branching diagram for the example problem.

❖

(Blank page)