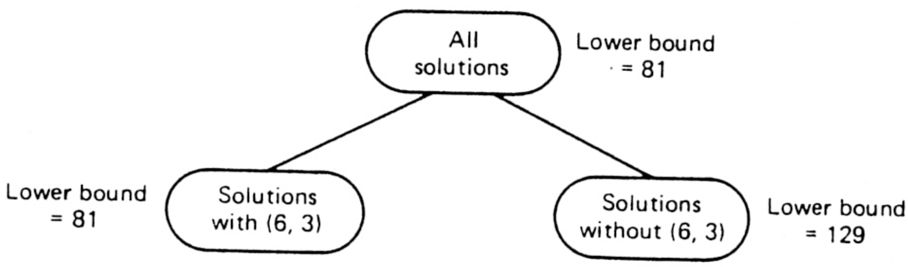
Reduction

Let us observe that a Hamiltonian cycle (cycle of length n) will contain exactly one element from each row of cost matrix (or weight matrix) W and exactly one element from each column of W. If a constant q is subtracted from any row or from any column of W, the cost of all Hamiltonian cycles (TS tours) is reduced by q. Therefore, the relative costs of different cycles remain the same. Thus the optimal tour remains optimal. If such a subtraction is done from rows and columns, such that each row and each column contains at least one zero while keeping remaining w_{ij}'s nonnegative, the total amount subtracted will be a lower bound on the cost of any solution. This process of subtracting constants from rows and columns is called *reduction*.

$$
\begin{array}{c}
\\ 1\\ 2\\ 3\\ 4\\ 5\\ 6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array}\\
\begin{bmatrix}
\infty & 3 & 93 & 13 & 33 & 9\\
4 & \infty & 77 & 42 & 21 & 16\\
45 & 17 & \infty & 36 & 16 & 28\\
39 & 90 & 80 & \infty & 56 & 7\\
28 & 46 & 88 & 33 & \infty & 25\\
3 & 88 & 18 & 46 & 92 & \infty
\end{bmatrix}
\end{array}
$$

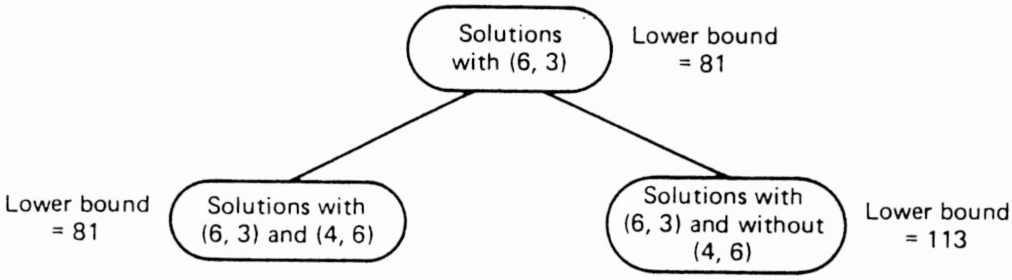
Let us now consider an example which is adapted from Reingold et al. [1977]. The cost matrix given above can be reduced by subtracting 3, 4, 16, 7, 25, and 3 from rows 1 through 6, respectively, and then subtracting 15 and 8 from columns 3 and 4, respectively, leaving the reduced matrix. Since a total of 81 was subtracted, 81 is a lower bound on the cost of all solutions for this problem.

$$
\begin{array}{c}
\\ 1\\ 2\\ 3\\ 4\\ 5\\ 6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array}\\
\begin{bmatrix}
\infty & 0 & 75 & 2 & 30 & 6\\
0 & \infty & 58 & 30 & 17 & 12\\
29 & 1 & \infty & 12 & 0 & 12\\
32 & 83 & 58 & \infty & 49 & 0\\
3 & 21 & 48 & 0 & \infty & 0\\
0 & 85 & 0 & 35 & 89 & \infty
\end{bmatrix}
\end{array}
$$



$$
\begin{array}{c}
\\ 1\\ 2\\ 3\\ 4\\ 5
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 4 & 5 & 6 \end{array}\\
\begin{bmatrix}
\infty & 0 & 2 & 30 & 6\\
0 & \infty & 30 & 17 & 12\\
29 & 1 & 12 & 0 & \infty\\
32 & 83 & \infty & 49 & 0\\
3 & 21 & 0 & \infty & 0
\end{bmatrix}
\end{array}
\qquad
\begin{array}{c}
\\ 1\\ 2\\ 3\\ 4\\ 5\\ 6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array}\\
\begin{bmatrix}
\infty & 0 & 27 & 2 & 30 & 6\\
0 & \infty & 10 & 30 & 17 & 12\\
29 & 1 & \infty & 12 & 0 & 12\\
32 & 83 & 10 & \infty & 49 & 0\\
3 & 21 & 0 & 0 & \infty & 0\\
0 & 85 & \infty & 35 & 89 & \infty
\end{bmatrix}
\end{array}
$$

Figure 3-40 *Splitting of solutions.*



$$\begin{array}{c|cccc} & 1 & 2 & 4 & 5 \\ \hline 1 & \infty & 0 & 2 & 30 \\ 2 & 0 & \infty & 30 & 17 \\ 3 & 29 & 1 & \infty & 0 \\ 5 & 3 & 21 & 0 & \infty \end{array} \qquad \begin{array}{c|ccccc} & 1 & 2 & 4 & 5 & 6 \\ \hline 1 & \infty & 0 & 2 & 30 & 6 \\ 2 & 0 & \infty & 30 & 17 & 12 \\ 3 & 29 & 1 & 12 & 0 & \infty \\ 4 & 0 & 51 & \infty & 17 & \infty \\ 5 & 3 & 21 & 0 & \infty & 0 \end{array}$$

Figure 3-41

no longer usable. This is enforced by setting entry (3, 4) to ∞. In general, then, if the edge added to the partial tour is from i_u to j_1 and the partial tour contains paths $(i_1, i_2, \ldots, i_u)$ and $(j_1, j_2, \ldots, j_k)$, the edge whose use is to be prevented is (j_k, i_1).

For splitting the leftmost node further, edge (2, 1) is the best; for when the zero in (2, 1) position is replaced with ∞, it allows 17 to be subtracted from the second row and 3 to be subtracted from the first column. This quantity is more than allowed by any other zero entry.

After splitting on edge (2, 1), the cost matrix on the left side is 3×3. Since we have included edge (2, 1) in the leftmost solutions, the edge (1, 2) is forbidden by setting (1, 2) entry to ∞. This matrix,

$$\begin{array}{c|ccc} & 2 & 4 & 5 \\ \hline 1 & \infty & 2 & 30 \\ 3 & 1 & \infty & 0 \\ 5 & 21 & 0 & \infty \end{array}$$

can be reduced by subtracting 1 from column two and 2 from row one. This produces the cost matrix

$$\begin{array}{c|ccc} & 2 & 4 & 5 \\ \hline 1 & \infty & 0 & 28 \\ 3 & 0 & \infty & 0 \\ 5 & 20 & 0 & \infty \end{array}$$

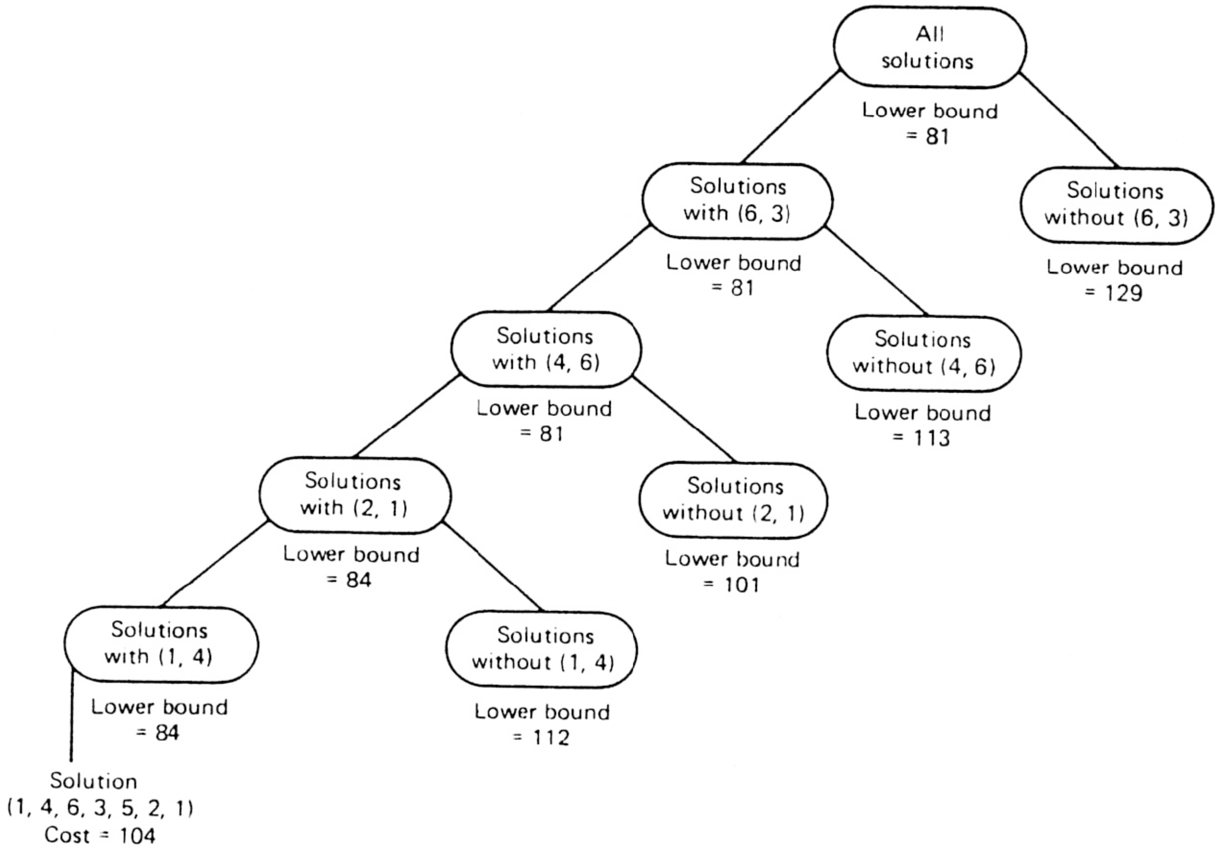


Figure 3-42 *Branch-and-bound binary tree.*

or form two separate paths. In either case, if the original network is directed (i.e., the distance matrix is asymmetric), we have no choice left in selection of the two remaining edges that complete the traveling salesman's tour. For instance, in the six-city illustrative problem that we have been solving, after edges (6, 3), (4, 6), (2, 1), and (1, 4) are selected, we have no choice but to add edges (3, 5) and (5, 2) to complete the traveling salesman's route.

Now we have obtained one TS route, namely 1, 4, 6, 3, 5, 2, 1, which costs 104. All nodes in the search tree (Fig. 3-42) with lower bound greater than 104 can be rejected, as they will not lead to a cheaper route. In Fig. 3-42 only one node has lower bound less than 104, and it must be expanded further. The node with lower bound of 101 includes edges (6, 3) and (4, 6) but excludes edge (2, 1). The cost matrix associated with this node is

$$
\begin{array}{c c}
 & \begin{array}{cccc} 1 & 2 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 5 \end{array} &
\begin{bmatrix}
\infty & 0 & 2 & 30 \\
\infty & \infty & 13 & 0 \\
26 & 1 & \infty & 0 \\
0 & 21 & 0 & \infty
\end{bmatrix}
\end{array}
$$

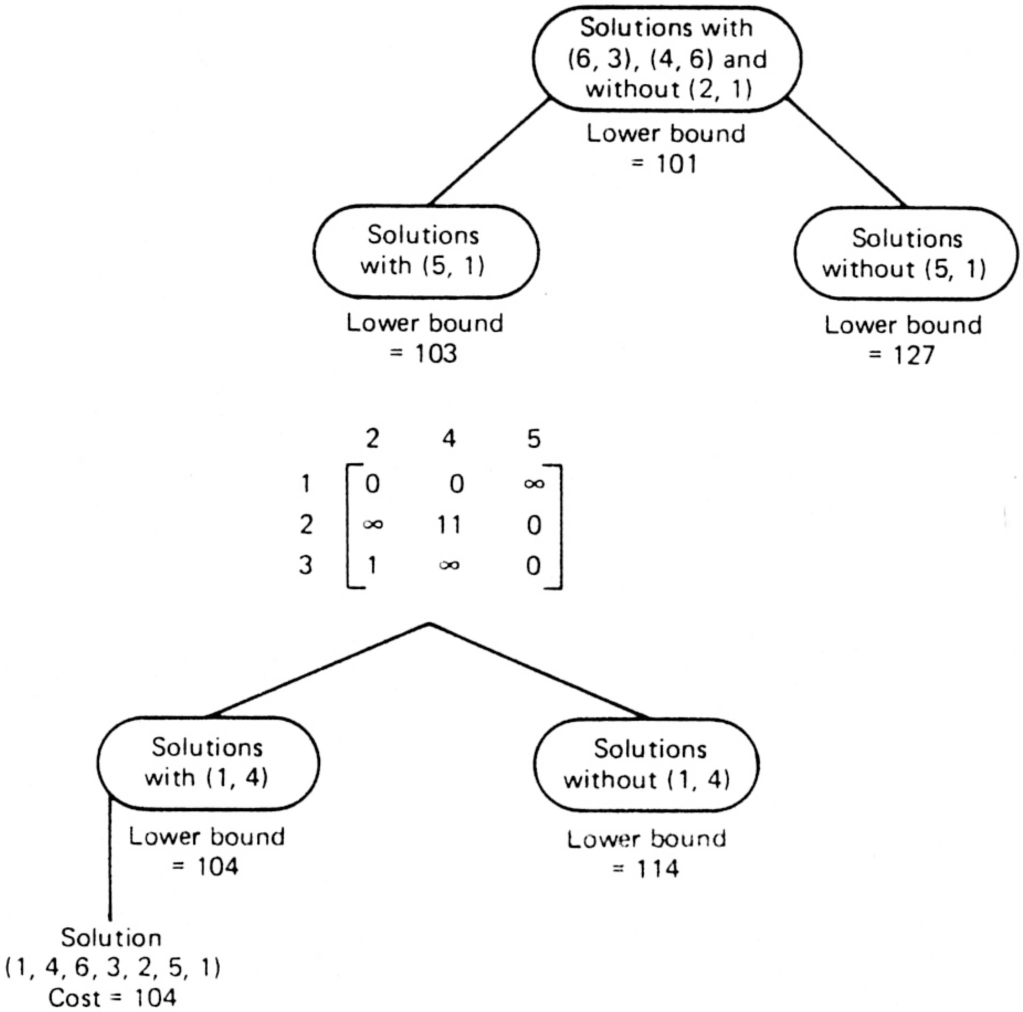

Solutions with (6, 3), (4, 6) and without (2, 1)
Lower bound = 101
Solutions with (5, 1)
Lower bound = 103
Solutions without (5, 1)
Lower bound = 127
$$\begin{array}{c|ccc} & 2 & 4 & 5 \\ \hline 1 & 0 & 0 & \infty \\ 2 & \infty & 11 & 0 \\ 3 & 1 & \infty & 0 \end{array}$$
Solutions with (1, 4)
Lower bound = 104
Solutions without (1, 4)
Lower bound = 114
Solution (1, 4, 6, 3, 2, 5, 1) Cost = 104

Figure 3-43

Algorithm 3-9(a): Reduction of Matrix A

```
function REDUCE(A);
begin
    rvalue ← 0; (* reduction value *)
    for i ← 1 to size do (* size of A *)
    begin
        rowred(i) ← smallest element in ith row;
        if rowred(i) > 0 then
        begin
            subtract rowred(i) from every finite element in ith row;
            rvalue ← rvalue + rowred(i)
        end
    end;
    for j ← 1 to size do
        begin
            colred(j) ← smallest element in jth column;
            if colred(j) > 0 then
            begin
                subtract colred(j) from every finite element in jth column;
                rvalue ← rvalue + colred(j)
            end
        end;
        REDUCE ← rvalue
     end
```

Algorithm 3-9(b): Selecting the Best Edge (r, c)

```
procedure BESTEDGE(A, size r, c, most);
begin
    most ← −∞;
    for i ← 1 to size do (* row *)
        for j ← 1 to size do (* column *)
            if a_ij = 0 then
            begin
                minr ← smallest entry in ith row, other than a_ij;
                minc ← smallest entry in jth column, other than a_ij;
                total ← minr + minc;
                if total > most then
                begin
                    most ← total;
                    r ← i; (* row index of best edge *)
                    c ← j ( *column index of best edge *)
                end
            end
end
```