5-6 A PERSONNEL ASSIGNMENT PROBLEM SOLVED BY THE BRANCH-AND-BOUND STRATEGY

We now show how a personnel assignment problem, which is NP-complete, can be solved efficiently by the branch-and-bound strategy. Let there be a linearly ordered set of persons $P = \{P_1, P_2, \ldots, P_n\}$, where $P_1 < P_2 < \cdots < P_n$. We may imagine that the ordering of persons is determined by some criterion, such as height, age, seniority and so on. Let there also be a set of jobs $J = \{J_1, J_2, \ldots, J_n\}$ and we assume that these jobs are partially ordered. Each person can be assigned to a job. Let P_i and P_j be assigned to jobs $f(P_i)$ and $f(P_j)$ respectively. We require that if $f(P_i) \leq f(P_j)$, then $P_i \leq P_j$. The function f can be interpreted as a feasible assignment which maps persons to their appropriate jobs. We also require that if $i \neq j$, then $f(P_i) \neq f(P_j)$.

Consider the following example. $P = \{P_1, P_2, P_3\}$, $J = \{J_1, J_2, J_3\}$ and the partial ordering of J is $J_1 \leq J_3$ and $J_2 \leq J_3$. In this case, $P_1 \rightarrow J_1$, $P_2 \rightarrow J_2$ and $P_3 \rightarrow J_3$ are feasible assignments while $P_1 \rightarrow J_1$, $P_2 \rightarrow J_3$ and $P_3 \rightarrow J_2$ are not.

We further assume that there is a cost C_{ij} incurred for a person P_i being assigned to job J_j. Let X_{ij} be 1 if P_i is assigned to J_j and 0 if otherwise. Then the total cost corresponding to a feasible assignment f is

$$\sum_{i,j} C_{ij} X_{ij}.$$

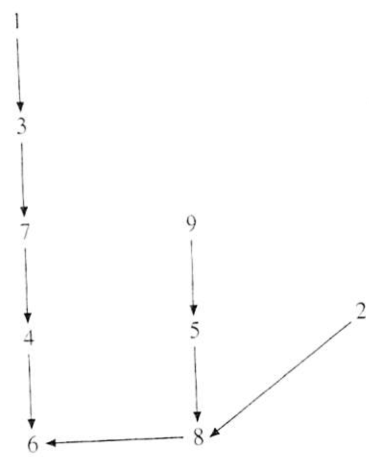
Our personnel assignment problem is precisely defined as follows: We are given a linearly ordered set of persons $P = \{P_1, P_2, \ldots, P_n\}$, where $P_1 < P_2 < \cdots < P_n$ and a partially ordered set of jobs $J = \{J_1, J_2, \ldots, J_n\}$. Cost C_{ij} is equal to the cost of assigning P_i to J_j. Each person is assigned to a job and no two persons are assigned to the same job. Our problem is to find an optimal feasible assignment which minimizes the following quantity

$$\sum_{i,j} C_{ij} X_{ij}.$$

Thus, our problem is an optimization problem and can be proved to be NP-hard. We shall not discuss the NP-hardness here.

To solve this problem, we shall use the notion "topological sorting". For a given partial ordering set S, a linear sequence $S_1, S_2, \ldots, S_n$ is topologically sorted with respect to S if $S_i \leq S_j$ in the partial ordering implies that S_i is located before S_j in the sequence. For instance, for the partial ordering shown in Figure 5–20, a corresponding topologically sorted sequence is 1, 3, 7, 4, 9, 2, 5, 8, 6.

FIGURE 5–20 A partial ordering.



Let $P_1 \rightarrow J_{k_1}, P_2 \rightarrow J_{k_2}, \ldots, P_n \rightarrow J_{k_n}$ be a feasible assignment. According to our problem definition, the jobs are partially ordered and persons are linearly ordered. Therefore, $J_{k_1}, J_{k_2}, \ldots, J_{k_n}$ must be a topologically sorted sequence with respect to the partial ordering of jobs. Let us illustrate our idea by an example. Consider $J = \{J_1, J_2, J_3, J_4\}$ and $P = \{P_1, P_2, P_3, P_4\}$. The partial ordering of J is illustrated in Figure 5–21.

FIGURE 5–21 A partial ordering of jobs.

The following are all the topologically sorted sequences:

J_1, J_2, J_3, J_4

J_1, J_2, J_4, J_3

J_1, J_3, J_2, J_4

J_2, J_1, J_3, J_4

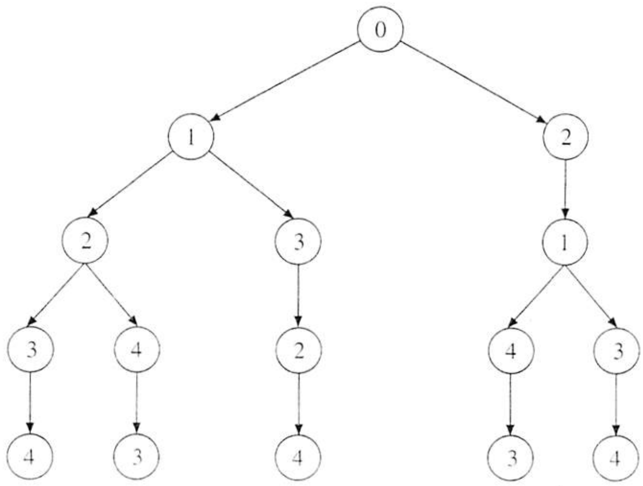
J_2, J_1, J_4, J_3.

Each sequence represents a feasible assignment. For instance, for the first sequence, it corresponds to the feasible assignment

$$P_1 \to J_1,\ P_2 \to J_2,\ P_3 \to J_3,\ P_4 \to J_4.$$

Tree searching techniques can be easily used to find all the topologically sorted sequences. For example, for the partial ordering shown in Figure 5–21, a tree showing all the topologically sorted sequences is shown in Figure 5–22.

FIGURE 5–22 A tree representation of all topologically sorted sequences corresponding to Figure 5–21.



The tree in Figure 5–22 is generated by using three basic steps.

(1) Take an element which is not preceded by any other element in the partial ordering.
(2) Select this element as an element in a topologically sorted sequence.
(3) Remove this element just selected from the partial ordering set. The resulting set is still partially ordered.

For instance, for the partial ordering shown in Figure 5–21, at the very beginning, J_1 and J_2 are the elements without predecessors. Thus, they are on the same level of the tree. Consider the node corresponding to 1. If we remove 1, the partially ordered set now contains 2, 3 and 4. Only 2 and 3 do not have predecessors in this new set. Therefore, 2 and 3 are generated.

Having described how the solution space of our problem can be described by a tree, we can now proceed to show how the branch-and-bound strategy can be used to find an optimal solution.

Given a cost matrix, we can compute a lower bound of our solutions immediately. This lower bound is obtained by reducing the cost matrix in such a way that it will not affect the solutions and there will be at least one zero in every row and every column and all remaining entries of the cost matrix are non-negative.

Note that if a constant is subtracted from any row or any column of the cost matrix, an optimal solution does not change. Consider Table 5–1 in which a cost matrix is shown. For this set of data, we can subtract 12, 26, 3 and 10 from rows 1, 2, 3 and 4 respectively. We can also subtract 3 from column 2 afterwards. The resulting matrix is a reduced cost matrix in which every row and every column contains at least one zero and the remaining entries of the matrix are all non-negative, as shown in Table 5–2. The total cost subtracted is $12 + 26 + 3 + 10 + 3 = 54$. This is a lower bound of our solutions.

TABLE 5–1 A cost matrix for a personnel assignment problem.

Persons \ Jobs	1	2	3	4
1	29	19	17	12
2	32	30	26	28
3	3	21	7	9
4	18	13	10	15

TABLE 5–2 A reduced cost matrix.

Persons \ Jobs	1	2	3	4	
1	17	4	5	0	
2	6	1	0	2	Total = 54
3	0	15	4	6	
4	8	0	0	5	

Figure 5–23 shows an enumeration tree associated with this reduced cost matrix. If the least lower bound is used, subsolutions which cannot lead to optimal solutions will be pruned in a much earlier stage, shown in Figure 5–24.

FIGURE 5–23 An enumeration tree associated with the reduced cost matrix in Table 5–2.

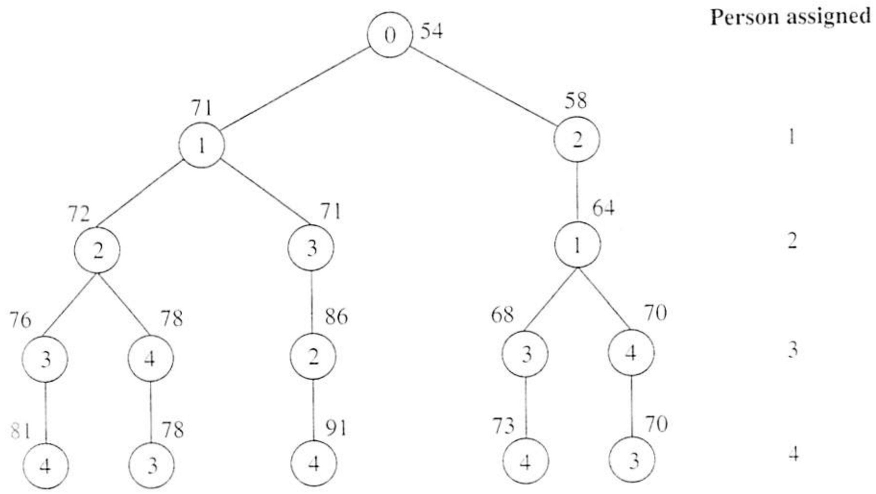
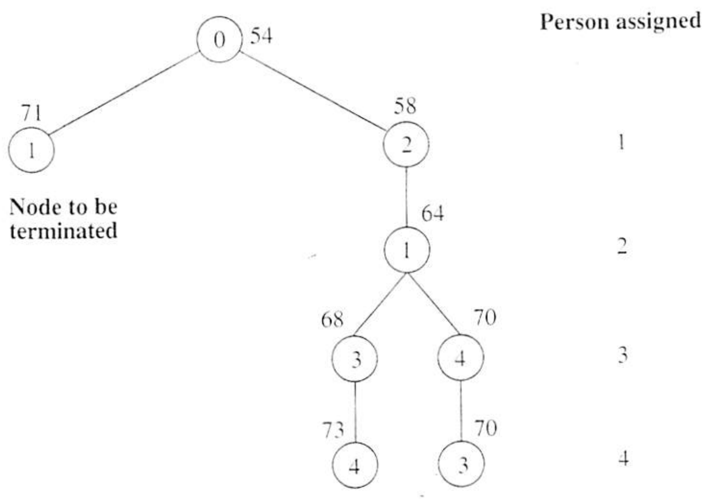


FIGURE 5–24 The bounding of subsolutions.



In Figure 5–24, we can see that after a solution with cost 70 is found, we can immediately bound all solutions starting with assigning P_1 to J_1 because its cost is 71 which is larger than 70.

Why did we subtract costs from the cost matrix? Suppose that we did not. Then, consider the node corresponding to assigning $P_1 \to J_1$. The cost associated with this node is only 29. Let us imagine that we have found a feasible solution with cost 70, namely assigning $P_1 \to J_2$, $P_2 \to J_1$, $P_3 \to J_4$ and $P_4 \to J_3$. Although we have found an upper bound, we cannot use it to bound the node corresponding to $P_1 \to J_1$ because its cost is only 29, lower than 70.

Look at Figure 5–24 again. We can now see that the cost associated with P_1 to J_1 is 71, instead of 29. Thus, a bound occurs. Why can we have such a high cost? This is because we have subtracted costs from the original cost matrix so that each row and each column contains a zero. Thus, after subtracting, we have a better lower bound for all solutions, namely 54. In other words, no solution can have a cost lower than 54. With this information, we know that the lower bound of assigning P_1 to J_1 is 54 + 17 = 71, instead of only 29. A higher lower bound will of course lead to an earlier termination.

5-7 THE TRAVELING SALESPERSON OPTIMIZATION PROBLEM SOLVED BY THE BRANCH-AND-BOUND STRATEGY

The traveling salesperson decision problem is an NP-complete problem. Thus, the traveling salesperson problem is hard to solve in worst cases. But, as will be shown in this section, the traveling salesperson problem can be solved by using the branch-and-bound strategy. That is, if we are lucky, an exhaustive search through the solution space may be avoided.

The basic principle of using the branch-and-bound strategy to solve the traveling salesperson optimization problem consists of two parts.

(1) There is a way to split the solution space.

(2) There is a way to predict a lower bound for a class of solutions. There is also a way to find an upper bound of an optimal solution. If the lower bound of a solution exceeds this upper bound, this solution cannot be optimal. Thus, we should terminate the branching associated with this solution.

The traveling salesperson problem can be defined on a graph, or planar points. If the traveling salesperson problem is defined on a set of planar points, many tricks can be used so that the algorithm can be even more efficient. In this