Branch-and-Bound Approach for Machine-Scheduling Problem

Example 10 illustrates how a branch-and-bound approach may be used to schedule jobs on a single machine. See Baker (1974) and Hax and Candea (1984) for a discussion of other branch-and-bound approaches to machine-scheduling problems.

EXAMPLE 10 Branch-and-Bound Machine Scheduling

Four jobs must be processed on a single machine. The time required to process each job and the date the job is due are shown in Table 63. The delay of a job is the number of days after the due date that a job is completed (if a job is completed on time or early, the job's delay is zero). In what order should the jobs be processed to minimize the total delay of the four jobs?

Solution Suppose the jobs are processed in the following order: job 1, job 2, job 3, and job 4. Then the delays shown in Table 64 would occur. For this sequence, total delay $= 0 + 6 + 3 + 7 = 16$ days. We now describe a branch-and-bound approach for solving this type of machine-scheduling problem.

Because a possible solution to the problem must specify the order in which the jobs are processed, we define

$$x_{ij} = \begin{cases} 1 & \text{if job } i \text{ is the } j\text{th job to be processed} \\ 0 & \text{otherwise} \end{cases}$$

The branch-and-bound approach begins by partitioning all solutions according to the job that is *last* processed. Any sequence of jobs must process some job last, so each sequence of jobs must have $x_{14} = 1$, $x_{24} = 1$, $x_{34} = 1$, or $x_{44} = 1$. This yields four branches with nodes 1–4 in Figure 23. After we create a node by branching, we obtain a lower bound on the total delay (D) associated with the node. For example, if $x_{44} = 1$, we know that job 4 is the last job to be processed. In this case, job 4 will be completed at the end of day $6 + 4 + 5 + 8 = 23$ and will be $23 - 16 = 7$ days late. Thus, any schedule having

TABLE 63
Durations and Due Date of Jobs

Job	Days Required to Complete Job	Due Date
1	6	End of day 8
2	4	End of day 4
3	5	End of day 12
4	8	End of day 16

TABLE 64
Delays Incurred If Jobs Are Processed in the Order 1–2–3–4

Job	Completion Time of Job	Delay of Job
1	6	0
2	$6 + 4 = 10$	$10 - 4 = 6$
3	$6 + 4 + 5 = 15$	$15 - 12 = 3$
4	$6 + 4 + 5 + 8 = 23$	$23 - 16 = 7$

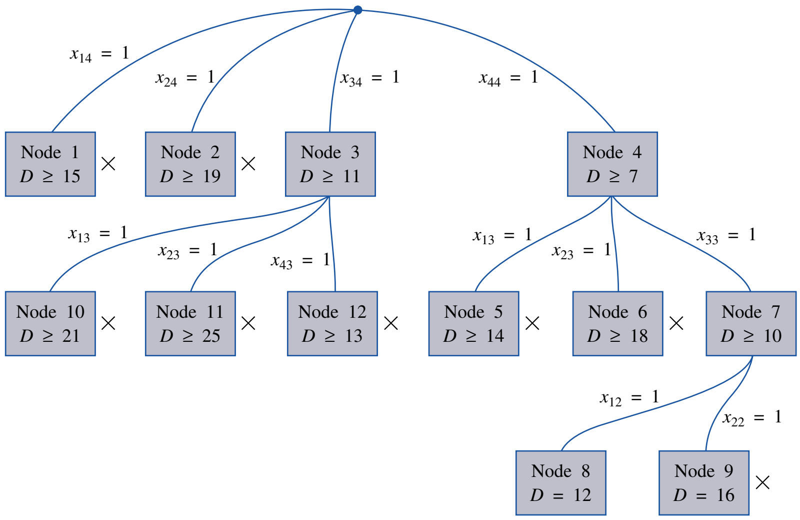

FIGURE 23
Branch-and-Bound Tree for Machine-Scheduling Problem

$x_{44} = 1$ must have $D \geq 7$. Thus, we write $D \geq 7$ inside node 4 of Figure 23. Similar reasoning shows that any sequence of jobs having $x_{34} = 1$ will have $D \geq 11$, $x_{24} = 1$ will have $D \geq 19$, and $x_{14} = 1$ will have $D \geq 15$. We have no reason to exclude any of nodes 1–4 from consideration as part of the optimal job sequence, so we choose to branch on a node. We use the jumptracking approach and branch on the node that has the smallest bound on D: node 4. Any job sequence associated with node 4 must have $x_{13} = 1$, $x_{23} = 1$, or $x_{33} = 1$. Branching on node 4 yields nodes 5–7 in Figure 23. For each new node, we need a lower bound for the total delay. For example, at node 7, we know from our analysis of node 1 that job 4 will be processed last and will be delayed by 7 days. For node 7, we know that job 3 will be the third job processed. Thus, job 3 will be completed after $6 + 4 + 5 = 15$ days and will be $15 - 12 = 3$ days late. Any sequence associated with node 7 must have $D \geq 7 + 3 = 10$ days. Similar reasoning shows that node 5 must have $D \geq 14$, and node 6 must have $D \geq 18$. We still do not have any reason to eliminate any of nodes 1–7 from consideration, so we again branch on a node. The jumptracking approach directs us to branch on node 7. Any job sequence associated with node 7 must have either job 1 or job 2 as the second job processed. Thus, any job sequence associated with node 7 must have $x_{12} = 1$ or $x_{22} = 1$. Branching on node 7 yields nodes 8 and 9 in Figure 23.

Node 9 corresponds to processing the jobs in the order 1–2–3–4. This sequence yields a total delay of 7 (for job 4) + 3 (for job 3) + $(6 + 4 - 4)$ (for job 2) + 0 (for job 1) = 16 days. Node 9 is a feasible sequence and may be considered a candidate solution having $D = 16$. We now know that any node that cannot have a total delay of less than 16 days can be eliminated.

Node 8 corresponds to the sequence 2–1–3–4. This sequence has a total delay of 7 (for job 4) + 3 (for job 3) + $(4 + 6 - 8)$ (for job 1) + 0 (for job 2) = 12 days. Node 8 is a feasible sequence and may be viewed as a candidate solution with $D = 12$. Because node 8 is better than node 9, node 9 may be eliminated from consideration.

Similarly, node 5 (having $D \geq 14$), node 6 (having $D \geq 18$), node 1 (having $D \geq 15$), and node 2 (having $D \geq 19$) can be eliminated. Node 3 cannot yet be eliminated, because it is still possible for node 3 to yield a sequence having $D = 11$. Thus, we now branch on node 3. Any job sequence associated with node 3 must have $x_{13} = 1$, $x_{23} = 1$, or $x_{43} = 1$, so we obtain nodes 10–12.

For node 10, $D \geq$ (delay from processing job 3 last) + (delay from processing job 1 third) $= 11 + (6 + 4 + 8 - 8) = 21$. Because any sequence associated with node 10 must have $D \geq 21$ and we have a candidate with $D = 12$, node 10 may be eliminated.

For node 11, $D \geq$ (delay from processing job 3 last) + (delay from processing job 2 third) = 11 + (6 + 4 + 8 − 4) = 25. Any sequence associated with node 11 must have $D \geq 25$, and node 11 may be eliminated.

Finally, for node 12, $D \geq$ (delay from processing job 3 last) + (delay from processing job 4 third) = 11 + (6 + 4 + 8 − 16) = 13. Any sequence associated with node 12 must have $D \geq 13$, and node 12 may be eliminated.

With the exception of node 8, every node in Figure 23 has been eliminated from consideration. Node 8 yields the delay-minimizing sequence $x_{44} = x_{33} = x_{12} = x_{21} = 1$. Thus, the jobs should be processed in the order 2–1–3–4, with a total delay of 12 days resulting.

Branch-and-Bound Approach for Traveling Salesperson Problem

EXAMPLE 11 Traveling Salesperson Problem

Joe State lives in Gary, Indiana. He owns insurance agencies in Gary, Fort Wayne, Evansville, Terre Haute, and South Bend. Each December, he visits each of his insurance agencies. The distance between each agency (in miles) is shown in Table 65. What order of visiting his agencies will minimize the total distance traveled?

Solution Joe must determine the order of visiting the five cities that minimizes the total distance traveled. For example, Joe could choose to visit the cities in the order 1–3–4–5–2–1. Then he would travel a total of 217 + 113 + 196 + 79 + 132 = 737 miles.

To tackle the traveling salesperson problem, define

$$x_{ij} = \begin{cases} 1 & \text{if Joe leaves city } i \text{ and travels next to city } j \\ 0 & \text{otherwise} \end{cases}$$

Also, for $i \neq j$,

$$c_{ij} = \text{distance between cities } i \text{ and } j$$
$$c_{ii} = M, \text{ where } M \text{ is a large positive number}$$

It seems reasonable that we might be able to find the answer to Joe's problem by solving an assignment problem having a cost matrix whose ijth element is c_{ij}. For instance, suppose we solved this assignment problem and obtained the solution $x_{12} = x_{24} = x_{45} = x_{53} = x_{31} = 1$. Then Joe should go from Gary to Fort Wayne, from Fort Wayne to Terre Haute, from Terre Haute to South Bend, from South Bend to Evansville, and from Evansville to Gary. This solution can be written as 1–2–4–5–3–1. An itinerary that begins and ends at the same city and visits each city once is called a **tour.**

TABLE 65
Distance between Cities in Traveling Salesperson Problem

Day	Gary	Fort Wayne	Evansville	Terre Haute	South Bend
City 1 Gary	0	132	217	164	58
City 2 Fort Wayne	132	0	290	201	79
City 3 Evansville	217	290	0	113	303
City 4 Terre Haute	164	201	113	0	196
City 5 South Bend	58	79	303	196	0