Unit-4

Branch and Bound Method

Branch and Bound method searches a state space tree using any search mechanism in which all the children of the current node (also called live node) are generated before another node is expanded. It avoids the generation of subtrees that do not contain a answer node. There are three approaches: FIFO (Queue) (BFS) or LIFO (STACK) (DFS) and Least cost (LC) method which explores the nodes with minimum cost answer.

At each stage of tree exploration, bound for a particular node is calculated and checked if this bound will be able to generate solution. The bound indicates how far we are from the solution. The feasible solution is one, which gives the maximum or minimum value for the given objective function.

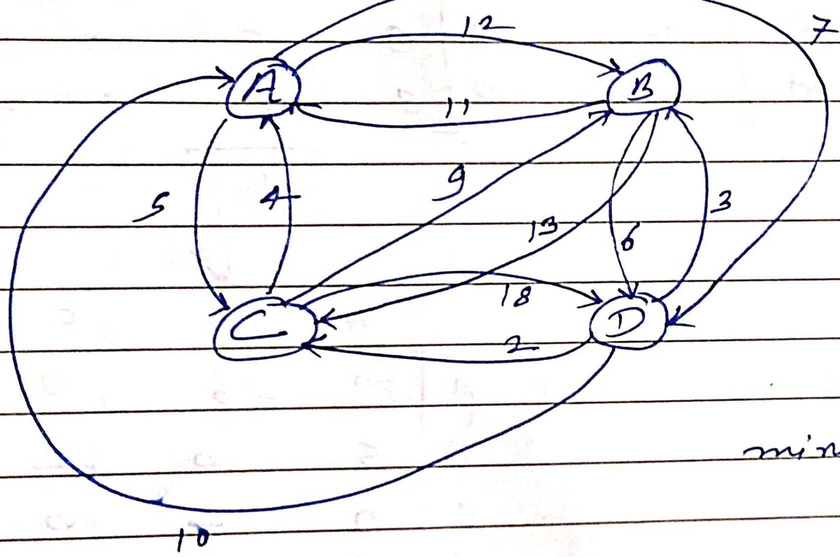
For example, the objective of Knapsack is maximum profits, where TSP is to find minimum path costs.

Travelling Salesman Problem
(Branch and Bound Method)

Problem states that a salesman wants to visit a certain number of cities. He knows the distance or cost or time of journey between every pair of cities. The problem is to select a path that starts from his city, passes through each city once and return to home city in the shortest possible distance or with least time

The problem is represented in Graph $G(V, E)$, which defines an instances of travelling of salesperson. The cost or the distance is shown as the weight of the edge in the graph G, denoted by C_{ij} for edge (i, j).
If $i = j$, $C_{ij} = \infty$, which means the salesman cann't go from city i to city i (i.e there is no meaning of going from a source city to itself.

We will ~~start~~ understand the problem from example:



The problem is shown in the form of distance b/w cities, where salesperson to travel. with minimum cost.

<u>step-1</u> :- Construct adjacency matrix as

To city →

$$
\text{From city} \quad
\begin{array}{c}
 \\ A \\ B \\ C \\ D
\end{array}
\begin{array}{c}
\begin{array}{cccc} A & B & C & D \end{array} \\
\begin{bmatrix}
\infty & 12 & 5 & 7 \\
11 & \infty & 13 & 6 \\
4 & 9 & \infty & 18 \\
10 & 3 & 2 & \infty
\end{bmatrix}
\end{array}
$$

Row specify from city
Colum specify To city

The edge (i, j) has weight ∞, so,
edge (A,A), (B,B), (C,C) and (D,D) assign to ∞.
Weights for other edges are provided in the cost matrix.

Step-2: Find reduced matrix R, ~~by selecting the minimum~~ of & substracting the smallest element from each row and each column, so that in each row and each column one zero to get. It is constructed to get better minimum cost value.

$$
\begin{array}{c c c}
\begin{array}{c|cccc|c}
 & A & B & C & D & \\
\hline
A & \infty & 12 & 5 & 7 & -5 \\
B & 11 & \infty & 13 & 6 & -6 \\
C & 4 & 9 & \infty & 18 & -4 \\
D & 10 & 3 & 2 & \infty & -2
\end{array}
& \Rightarrow &
\begin{array}{c|cccc|}
 & A & B & C & D \\
\hline
A & \infty & 7 & 0 & 2 \\
B & 5 & \infty & 7 & 0 \\
C & 0 & 5 & \infty & 14 \\
D & 8 & 1 & 0 & \infty \\
\hline
 & & -1 & &
\end{array}
\end{array}
$$

$$\Downarrow$$

$$
\begin{array}{c|cccc|}
 & A & B & C & D \\
\hline
A & \infty & 6 & 0 & 2 \\
B & 5 & \infty & 7 & 0 \\
C & 0 & 4 & \infty & 14 \\
D & 8 & 0 & 0 & \infty
\end{array}
$$

Now all column & rows are having at least 1 zero. Here make a node for total reduced cost in preparation of this reduced matrix

$$= 5 + 6 + 4 + 2 + 1 = 18$$

This reduced cost will become the root of the search tree corresponding to given graph.

Step-3 Assume source vertex A be the root (but It can be any vertex).
For every child node of A, ~~there~~ check if the child node is the leaf node?

If it is not a leaf node then find reduced cost matrix of this child node as follows:

1. If A is the root node or parent node and B is the child node there is an edge between two nodes as (A, B). Change all entries in row A and column B of the reduced matrix to ∞. This prevents the use of any more edges leaving vertex A or entering vertex B.
2. Set (row B, column 1) to ∞
3. Reduce all rows and columns in the resulting matrix except the rows and columns containing only ∞.

$$\begin{array}{c|c} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 7 & 0 \\ 0 & \infty & \infty & 14 \\ 8 & \infty & 0 & \infty \end{bmatrix} \end{array}$$

Just ensure here, each row & each column contains at one zero entry otherwise apply step-2 again. {Except ∞ entry}

Here total amount subtracted will be denoted by T, here no rows or columns reduced by weight, so T = 0.
Cost of the child node is then calculated by

cost of child node = cost of parent node + R(A, B) + T

Where R(A, B) is the value of Reduced cost matrix in row A, column B, in step-2 of reduced matrix.

So, cost of child node, B = 18 + R(A, B) + T
= 18 + 6 + 0 = 24

For edge $A \to C$, find reduced cost matrix for C.

Make row A, column C elements to ∞, in reduced matrix cost of step-2:

$$
\begin{array}{c|cccc|c}
 & A & B & C & D & \\
\hline
A & \infty & \infty & \infty & \infty & \\
B & 5 & \infty & \infty & 0 & \\
C & \infty & 4 & \infty & 14 & -4 \\
D & 8 & 0 & \infty & \infty & \\
\hline
 & -5 & & & &
\end{array}
\Rightarrow
\begin{array}{c|cccc|}
 & A & B & C & D \\
\hline
A & \infty & \infty & \infty & \infty \\
B & 0 & \infty & \infty & 0 \\
C & \infty & 0 & \infty & 10 \\
D & 3 & 0 & \infty & \infty
\end{array}
$$

Cost of child node, $C = 18 + R(A,C) + T$

$= 18 + R(A,C) + (4+5)$

$= 18 + 0 + 9 = 27$

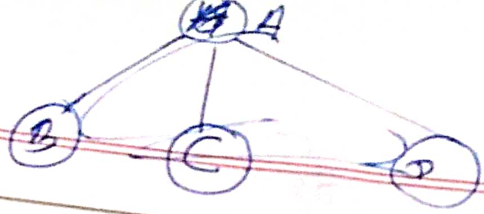
For edge $A \to D$, find reduced cost matrix for D. Make row A, column D all ∞, and element (D, 1) also ∞, in reduced cost matrix of step 2. Thus, matrix is

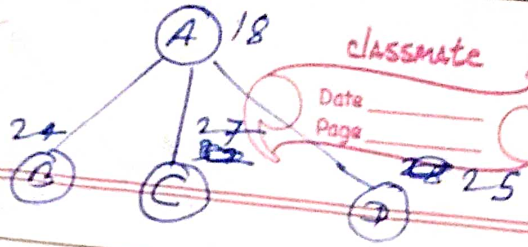
$$
\begin{array}{c|cccc|c}
 & A & B & C & D & \\
\hline
A & \infty & \infty & \infty & \infty & \\
B & 5 & \infty & 7 & \infty & -5 \\
C & 0 & 4 & \infty & \infty & \\
D & \infty & 0 & 0 & \infty &
\end{array}
\xrightarrow[\text{matrix}]{\text{Reduced}}
\begin{array}{c|cccc|}
 & A & B & C & D \\
\hline
A & \infty & \infty & \infty & \infty \\
B & 0 & \infty & 2 & \infty \\
C & 0 & 4 & \infty & \infty \\
D & \infty & 0 & 0 & \infty
\end{array}
$$

Cost of child node, $D = 18 + R(A,D) + T$

$= 18 + 2 + 5 = 25$

After processing of vertices from A to B, A to C, A to D, the search tree constructed is as follows





Choose node B have a minimum cost value as a promising solution. In other words, path (A, B) is the most promising solution.

From B, there is a path to vertices C and D. Now, the reduced cost matrices for these paths have to calculated.

For edge $A \to B \to C$, change entries in row A, row B and column C to ∞; in reduced cost matrix of step 2. Also change R(B,1) and R(C,1) to ∞. Thus matrix is:

$$\begin{array}{c|cccc|c} & A & B & C & D & \\ \hline A & \infty & \infty & \infty & \infty & \\ B & \infty & \infty & \infty & \infty & \\ C & \infty & 4 & \infty & 14 & -4 \\ D & 8 & 0 & \infty & \infty & \\ \hline & -8 & & & & \end{array} \xRightarrow{\text{Reduced matrix}} \begin{array}{c|cccc} & A & B & C & D \\ \hline A & \infty & \infty & \infty & \infty \\ B & \infty & \infty & \infty & \infty \\ C & \infty & 0 & \infty & 10 \\ D & 0 & 0 & \infty & \infty \end{array}$$

Cost of child node, C with parent B

$= \text{Cost of } B + R(B, C) + T$

$= 24 + 7 + 12 = 43$

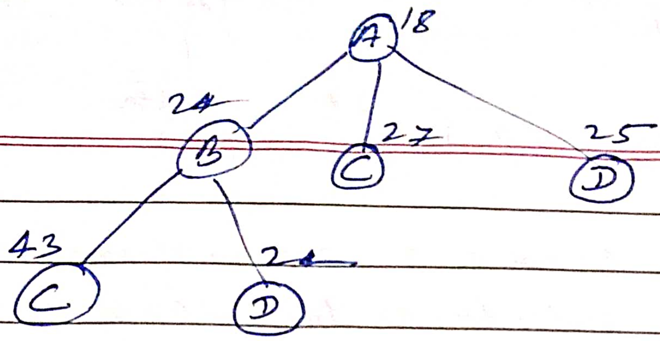
For edge $A \to B \to D$, change entries in row A, row B & column D to ∞, in reduced cost of step 2. Also change R(B,1) and R(D,1) to ∞. Thus matrix is

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & \infty & \infty & \infty & \infty \\ B & \infty & \infty & \infty & \infty \\ C & 0 & 4 & \infty & \infty \\ D & \infty & 0 & 0 & \infty \end{array}$$

Cost of child node, D with parent B

$= \text{cost of } B + R(B, D) + T$

$= 24 + 0 + 0 = 24$



Cost of A → B → D is most promising, choose them as solution.

From vertex D, the only path for vertex C is remaining. All other paths are either discarded or have been already traversed.

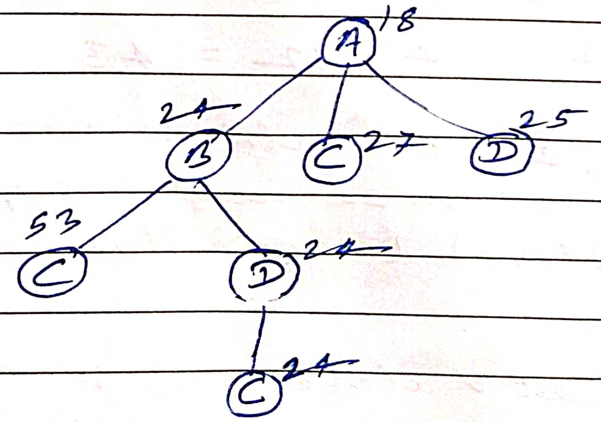
Vertex C is thus the leaf node, each leaf node defines a unique tour. Cost of reaching from vertex D to C according to reduced matrix R is

$$R(D, C) = 0$$

So, cost of node C = cost of parent node D + R(D, C)

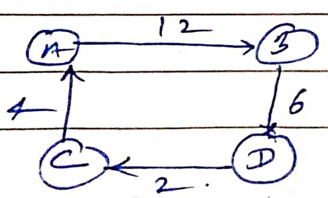
$$= 24 + 0 = 24$$

The search tree that gives minimum cost path for given graph is thus.



Leaf node with minimum cost is C with 24. Thus feasible path is

A → B → D → C → A



$\boxed{\text{Optimal Cost} = 24}$