Branch & Bound

Job seq. with deadline.

Problem statement: to understand how to calculate $c^\wedge(x)$ & upper.

→ let there be n jobs with different processing times. with only one processor the jobs are executed on or before deadlines.

→ Each job i is given by a tuple (P_i, d_i, t_i) where t_i is the processing time required by job i.

→ If processing of job i is not completed by deadline d_i then penality P_i will occur.

→ The objective of this problem is to select a subset J such that penalty will be minimum among all possible subsets.

→ such a J should be optimal subset.

ex let $n = 4$

job index	P_i	d_i	t_i
1	5	1	1
2	10	3	2
3	6	2	1
4	3	1	1

find optimal subset J? & what will be the penalty?

sol^n state space tree can be drawn for proper select^n of job i for creating J.

Two ways → Fixed tuple size
variable tuple size.

1) Variable tuple size:-

$$\hat{c}(x) = \sum_{i<m,\ i \notin S_x} P_i \quad \text{where } m = \max\{i \mid i \in S_x\}$$

& S_x be ~~subject~~ subset of jobs selected for J at node x.

upper bound $u(x) = \sum_{i \notin S_x} P_i$.

Hence, $u(x)$ corresponds to the cost of the soln Sx for node x.

node 1 → root has 4 childrens – 1, 2, 3, 4 for $n=4$ ie. job 4. Hence

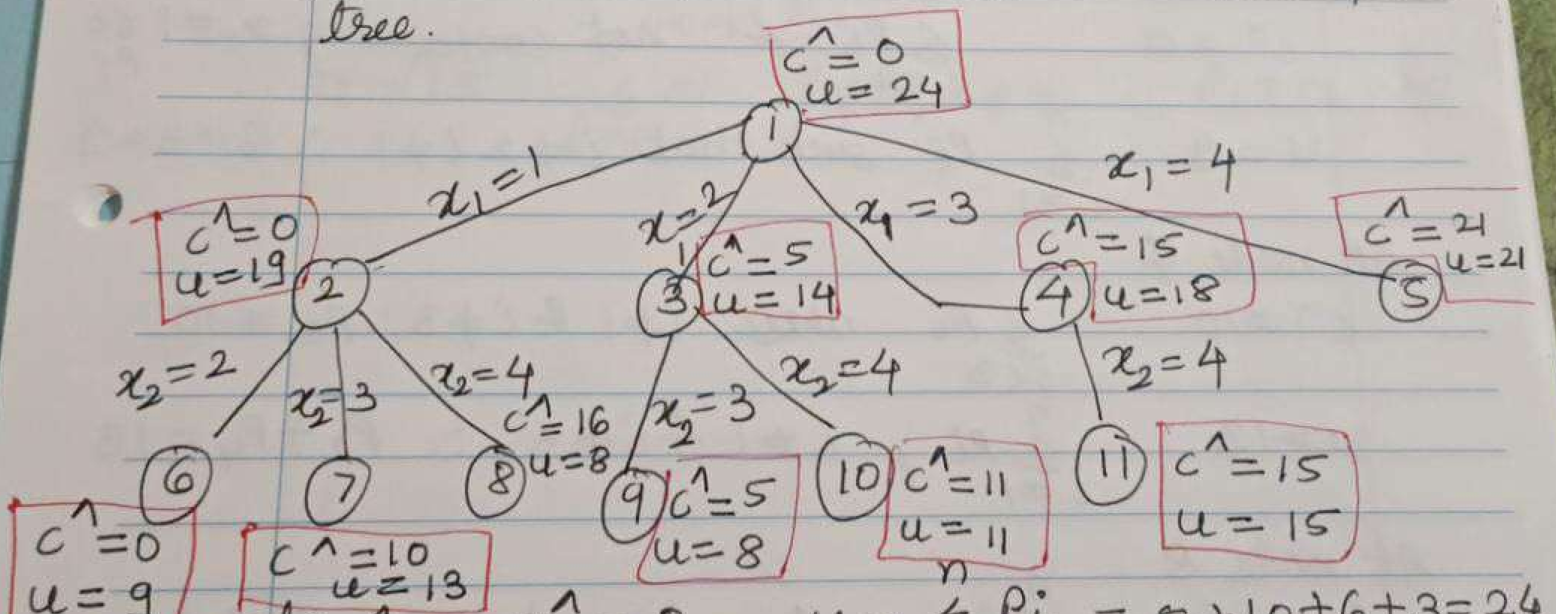
$x_1=1$, $x_1=2$, $x_1=3$ or $x_1=4$.

node 2 → with $x_1=1$ ∴ we can select either 2, 3 or 4.

Hence $x_2=2$, $x_2=3$ or $x_2=4$

⋮

In this fashion we draw the state space tree.



node 1 – $\hat{c} = 0$ $\quad u = \sum_{i=1}^{n} P_i = 5+10+6+3 = 24$

node 2 → $\hat{c} = 0$ $\quad \sum P_i$ where $i<0 = 0$

$u = 19$ $\quad$ sum of P_i excluding $x_1 =$

∴ $24-5 = 19$

MIT-WPU MIT WORLD PEACE UNIVERSITY

At node 3

$\hat{c} = 5$ $\quad \sum P_i$ where $i<2 = 5$ as $P_1=5$

$u = 14$ $\quad 24-10=14$ ie. ~~sum~~ $\sum P_i$ without $x_1=2$

At node 4

$\hat{c} = 15$ $\quad$ as $x_1=3$ $\sum P_i$, $i<3$.

$u = 18$ $\quad$ not consider P_3 $\sum P_i = 24-6=18$

At node 5

$\hat{c} = 21$ $\quad \sum P_i$, $i<4 = P_1+P_2+P_3$

$u = 21$ $\quad \sum P_i$ without P_4 ie. $24-3=21$

At node 6

$\hat{c} = 0$ $\quad \sum_{i<2} P_i$ & not containing $x_1=1$ ie P_1

$u = 9$ $\quad \sum_{i=1}^{n} P_i$ such that $i \neq 2$ $i \neq 1$ $\therefore 6+3=9$

At node 7

$\hat{c} = 10$ $\quad \sum_{i<3} P_i$ where $i \neq 1$ & $i \neq 3 \therefore P_2 = 10$

$u = 13$ $\quad \sum_{i=1}^{n} P_i$, $i \neq 1$ & $i \neq 3$ $\therefore P_2 + P_4 = 13$

At node 8

$\hat{c} = 16$ $\quad \sum_{i=1}^{n} P_i$, ~~$i<$~~, $i \neq 1$, $i \neq 4$ $\therefore P_2 + P_3 = 16$

$u = 16$ $\quad \sum_{i=1}^{n} P_i$, $i \neq 1$, $i \neq 4$ $P_2 + P_3 = 16$.

At node 9 :

$c^{\wedge} = 5 \quad \sum_{i<3} P_i \ , \ i \neq 2 \quad \therefore P_1 = 5$

$u = 8 \quad \sum_{i=1}^{n} P_i \ , \ i \neq 2, \ i \neq 3 \quad P_1 + P_4 = 8.$

At node 10 :

$c^{\wedge} = 1 \quad \sum_{i<4} P_i \ , \ i \neq 2 \quad \therefore P_1 + P_3 = 5 + 6 = 11$

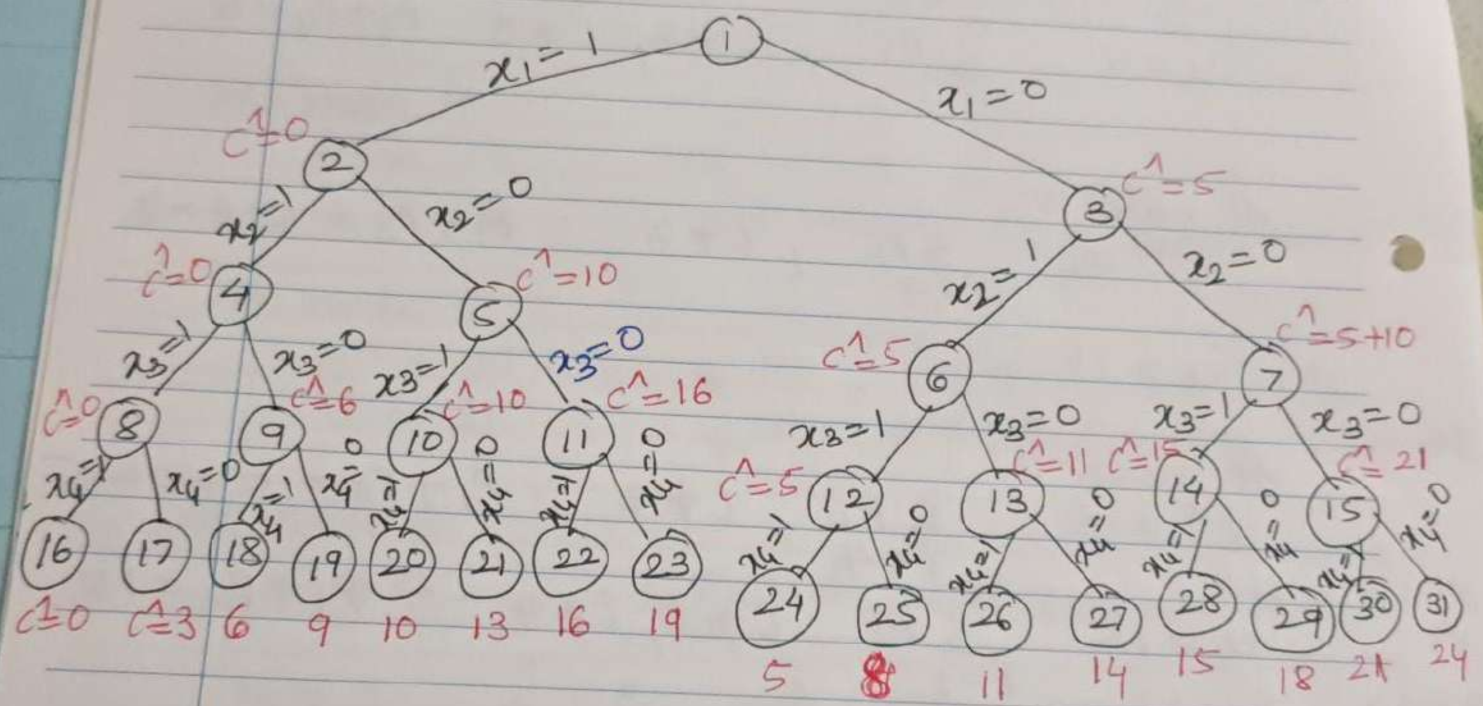
$u = 11$

At node 11 :

$c^{\wedge} = 15 \quad \sum_{i<4} P_i \ , \ i \neq 3 \quad \therefore P_1 + P_2 = 5 + 10 = 15$

$u = 15 \quad \sum_{i=1}^{n} P_i \quad i \neq 3, \ i \neq 4 \quad \therefore P_1 + P_2 = 15.$

Now we will draw fixed tuple size formulation :-

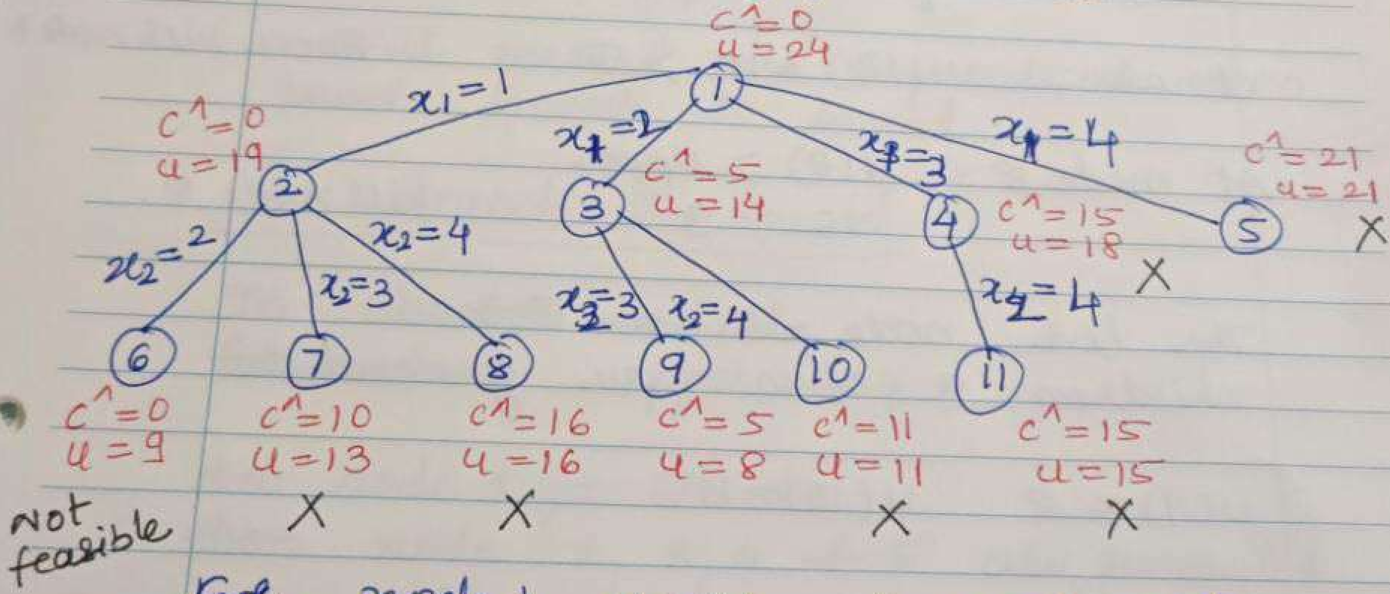


FIFO Branch & Bound

→ To understand FIFO branch & bound, we will consider the variable tuple size formulation.

→ The state space tree can be drawn as below –

→ For job sequencing problem as –



For node 1 $u = 24$ $\therefore$ upper $= 24$

at node 2, 3, 4 & 5 are generated.

$u(2) = 19$ $u(3) = 14$ $u(4) = 18$ $u(5) = 21$.

$\therefore$ minimum value $u(x) =$ upper $= 14$.

for node 4, $\hat{c}(4) = 15 > 14$ $\therefore$ kill node 4

for node 5, $\hat{c}(5) = 21 > 14$ $\therefore$ kill node 5

Node 2 & 3 become live nodes.

so, now we will consider node 2 & 3.

node 2 – E-node.

$\therefore u(6) = 9 \quad u(7) = 13 \quad u(8) = 16$.

$\therefore$ minimum of $u(x)$ = upper = 9.

At node 7 $\quad u(7) = 10 > 9 \quad \therefore$ Hence kill node 7

At node 8 $\quad u(8) >$ upper
$16 > 9 \quad \therefore$ kill node 8.

The live node = 3. = E-node.
children 9 & 10 are generated.

$\therefore u(9) = 8 \quad u(10) = 11$.

$\therefore$ min of $u(x)$ = upper = 8.

at node 10, $\hat{c}(10) = 11 > 8 \quad \therefore$ kill node 10.

Now, Node 6 becomes Enode
But as children of node 6 are infeasible,
we will not consider node 6.

only remaining live node = 9.

The only child of 9 is infeasible.

∴ there is no live node remaining, the minimum cost answer node is node 9.

& has a cost of 8.

∴ This method is referred as FIFO based branch & bound.

LC Branch & Bound.

In LC BB method, for node 1 upper = 24.

Now node 1 = E node. then node 2, 3, 4 & 5 are generated.

now, upper = min u(x) = 14.

As u^(4) > upper and u^(5) > upper. Hence node 4 & 5 are killed.

For node 2 & 3 → the cost c^(2) = 0.
∴ node 2 becomes as E-node.

Hence children 6, 7 & 8 are generated.

→ The upper = 9 ie $u(6) = 9$.

node 7 & 8 are killed.

→ The node 6 is selected as it has minimum cost.

But both the children of node 6 are infeasible. so kill node 6.

→ Now node 3 becomes E-node.
node 9 & 10 are generated.
upper = 8., $\hat{c}(10) >$ upper, kill node 10.

only node 9 is remain live.
∴ node 9 is E-node.
its only child is infeasible.

→ As there are no live node remaining, we will terminate search with node 9 as an answer node.

→ The principle idea is LC search is choosing of min. cost node each time.

→ This method of LC based BB with appropriate $\hat{c}(\cdot)$ & $u(\cdot)$ is called as LCBB (LIFOBB).

TSP BB

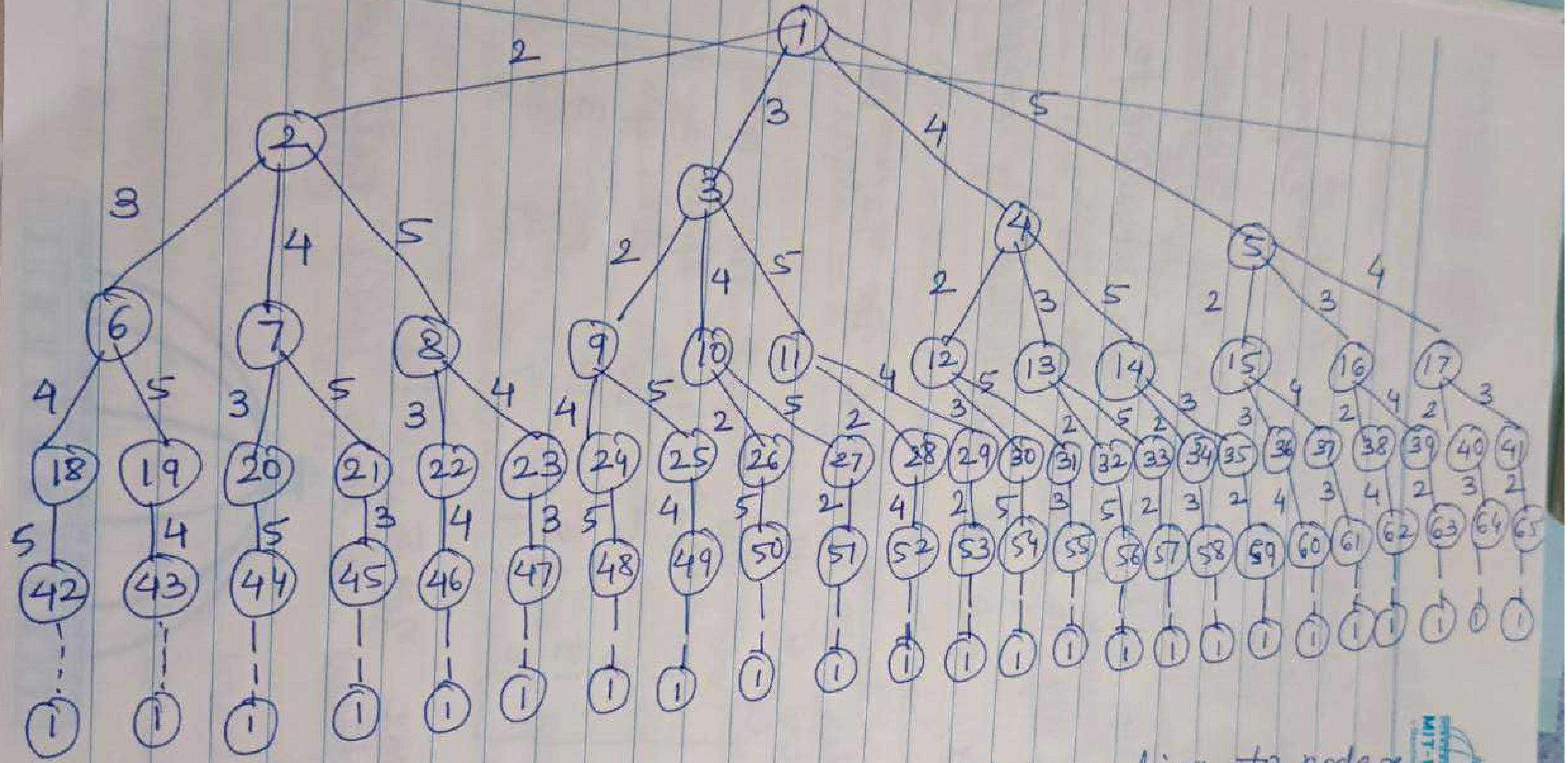
Problem statement :-
If there are n cities & cost of travelling from any city to any other city is given.
Then we have to obtain the cheapest round trip such that each city is visited exactly once & then returning to starting city to complete the tour.

Typically, TSP is represented by weighted graph.

ex Consider an instance for TSP is given by G as

$$G = \begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$$

sol^n There are $n = 5$ nodes.
we can draw a state space tree with 5 nodes.



Tour(x) is the path that begins at root & reaching to node x in a state space tree & returns to root.

In BB strategy, cost of each node x is computed.

The TSP problem is solved by choosing the node with optimum cost.

where x is a leaf node.

$c(x)$ = cost of tour(x)

$\hat{c}(x)$ = the approximation cost along the path from the root to x

Row minimization

To understand solving of TSP using BB approach, we will reduce the cost of matrix m by using the following formula

$$Red\_Row(m) = [m_{ij} - \min\{m_{ij} \mid 1 \le j \le n\}]$$

ex. $$m = \begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$$

find minimum of each row –

$r_1 = 10$

$r_2 = 2$

$r_3 = 2$

$r_4 = 3$

$r_5 = 4$

substract the row_min value from corresponding row.

$\therefore$ Hence $$Red\_Row(m) = \begin{bmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 12 & 0 & 3 & 12 & \infty \end{bmatrix}$$

Q. No.	1	2	3	4	5	6	7	8	9	10	Total
Marks/Grade											

column minimization:-

Now we will reduce the matrix by choosing minimum from each column. The formula for column reduction of matrix is:-

$$Red\_col(m) = m_{ji} - \min\{m_{ji} \mid 1 \leq j \leq n\}$$

where $m_{ji} < \infty$.

full reduction

let m be the cost matrix for TSP problem for n vertices the m is called reduced if each row & each column consists of either entirely ∞ entries or else contain at least one zero.

The full reduction can be achieved by applying both row reduction & column reduction ex :-

$$m = \begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$$

Row_Red (m) can be obtained as.
each row min val = 10
2
2
3
4
21

$$\therefore m = \begin{bmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 12 & 0 & 3 & 12 & \infty \end{bmatrix}$$

col Red (m) = 1 + 0 + 03 + 0 + 0 = 04

If row or column contains at least one zero ignore corresponding row or column.

Assignment/Test Page No.

$\therefore$ col- Red = $\begin{bmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix}$

The total Reduced cost will be

= cost (Red_Row(m) + cost (Red_Col(m))

= 21 + 4

= 25.

Then

$$\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 1 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$$

$\Downarrow$ Fully Reduced matrix

$$\begin{bmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix}$$

Dynamic Reduction

we obtained the total reduced cost = 25
That means all tours in the original graph have a length at least 25.

using dynamic reduction, we can make the choice of edge $i \rightarrow j$ with optimum cost.

Steps in dynamic reduction technique

1) Draw a state space tree with optimum cost at root node.

2) obtain the cost of matrix for path i to j by making i^{th} row & j^{th} col entries as ∞.
Also set $m[i][j] = \infty$

3) cost of corresponding node x with path $i \rightarrow j$ is optimum cost + reduced cost + $m[i][j]$

4) set node with minimum cost as E-node & generate its children.

Repeat step 1 to 4 for completing tour with optimum cost.

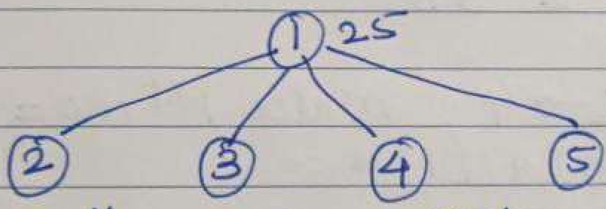
ex

$$m = \begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$$

Fully reduced matrix is

$$\begin{bmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix}$$

optimum cost $= 21 + 4 = 25$



→ consider path 1-2, make 1st row & 2nd col ∞ and set $m[2][1] = \infty$

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 11 & 2 & 0 \\ 0 & \infty & \infty & 0 & 2 \\ 15 & \infty & 12 & \infty & 0 \\ 11 & \infty & 0 & 12 & \infty \end{bmatrix}$$

ignore each row & col as it contains zero value.

$\therefore$ cost of node 2 = optimum cost + reduced cost + $m[1][2]$ old value

$= 25 + 0 + 10 = 35$

→ consider the path $1 \to 3$. Make 1^{st} row & 3^{rd} col is ∞ & $m[3][1] = \infty$

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & \infty & 2 & 0 \\ \infty & 3 & \infty & 0 & 2 \\ 15 & 3 & \infty & \infty & 0 \\ 11 & 0 & \infty & 12 & \infty \end{bmatrix}$$

↓

11

cost of node 3 = opt. cost + reduced cost + old val $m[1][3]$

$$= 25 + 11 + 17$$
$$= 53$$

→ consider $1 \to 4$, Make 1^{st} row = 4^{th} col = ∞ & $m[4][1] = \infty$

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & 0 \\ 0 & 3 & \infty & \infty & 2 \\ \infty & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & \infty & \infty \end{bmatrix}$$

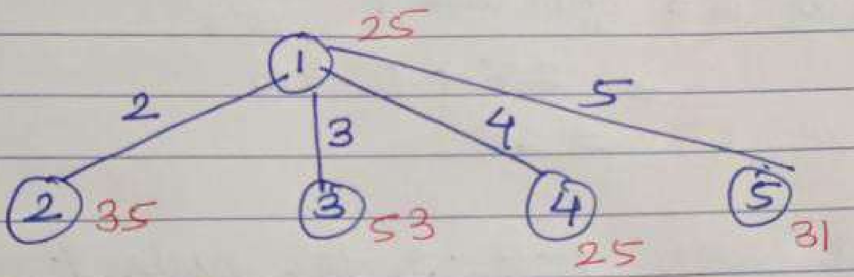
cost of node 4 = $25 + 0 + m[1][4]$ old val

$$= 25 + 0 + 0$$
$$= 25$$

→ consider $1 \to 5$, make 1^{st} row $= 5^{th}$ col $= \infty$
& $m[5][1] = \infty$

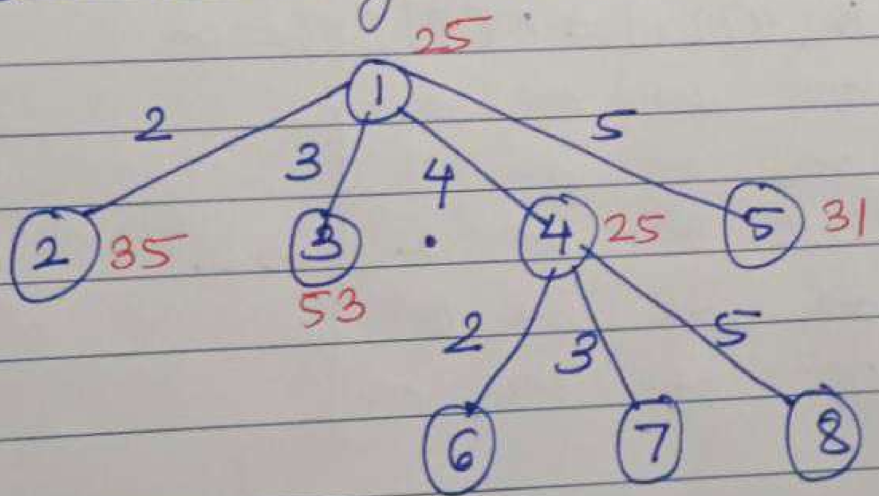
$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & 2 & \infty \\ 0 & 3 & \infty & 0 & \infty \\ 15 & 3 & 12 & \infty & \infty \\ \infty & 0 & 0 & 12 & \infty \end{bmatrix} \begin{array}{l} \\ \to 2 \\ \\ \to 3 \\ \\ \end{array}$$

cost of node 5 = opt. cost + reduced cost + $m[1][5]$ old value
= 25 + 5 + 1
= 31



cost of node 4 = 25 = minimum. set node 4 as E-node & generate its children 6, 7, 8.

→



Assignment/Test Page No

→ consider path 1-4-2 for node 6.
set 1^{st} row = 4^{th} row = ∞
set 2^{nd} col = 4^{th} col = ∞
$m[4][1] = m[2][1] = \infty$

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 11 & \infty & 0 \\ 0 & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 11 & \infty & 0 & \infty & \infty \end{bmatrix}$$

cost of node 6 = opt. cost + reduced cost + m[4][2] old value
= 25 + 0 + 3
= 28

→ consider path 1-4-3 for node 7.
set 1^{st} row = 4^{th} row = ∞
4^{th} col = 3^{rd} col = ∞
set $m[4][1] = m[3][1] = \infty$

$$\begin{array}{c} \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & \infty & \infty & 0 \\ \infty & 3 & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 11 & 0 & \infty & \infty & \infty \end{bmatrix} \begin{array}{l} \\ \\ \to 2 \\ \\ \\ \end{array} \\ \begin{array}{l} \downarrow \\ 11 \end{array} \end{array}$$

cost of node 7 = opt. cost + reduced cost + m[4][3] - old value
= 25 + 13 + 12
= 50

Maharashtra Academy of Engineering & Educational Research's

MAHARASHTRA INSTITUTE OF TECHNOLOGY, PUNE

ASSIGNMENT / TEST BOOKLET

Student's Name

Class Division

Subject Roll No Academic Year 20 -20

Assignment / Test No Date

Q. No.	1	2	3	4	5	6	7	8	9	10	Total
Marks/Grade											

Name & sign of the faculty Member

(Please start writing assignment/ test from here)

→ consider 1—4—5 for node 8.

set 1^{st} row = 4^{th} row = ∞

4^{th} col = 5^{th} col = ∞

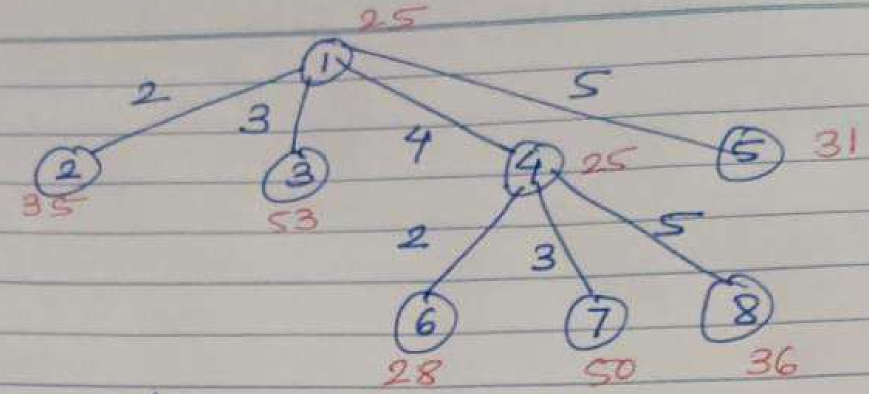
& $m[4][1] = m[5][1] = \infty$

$$\begin{array}{l}\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & \infty \\ 0 & 3 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & 0 & \infty & \infty \end{bmatrix} \begin{array}{l} \\ \rightarrow 11 \\ \\ \\ \\ \end{array}\end{array}$$

cost of node 8 = $25 + 11 + m[4][5]$

$= 36 + 0$

$= 36$



cost of node 6 = 28 = minimum
∴ node 6 becomes an E-node.
generate children 9 & 10.

→ consider path 1 − 4 − 2 − ~~3~~ for node 9.
set 1^{st} row = 4^{th} row = 2^{nd} row = ∞
4^{th} col = 2^{nd} col = 3^{rd} col = ∞
set $m[4,1] = m[2,1] = m[3,1] = \infty$

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 11 & \infty & \infty & \infty & \infty \end{bmatrix} \begin{array}{l} \\ \\ \rightarrow 2 \\ \\ \\ \end{array}$$

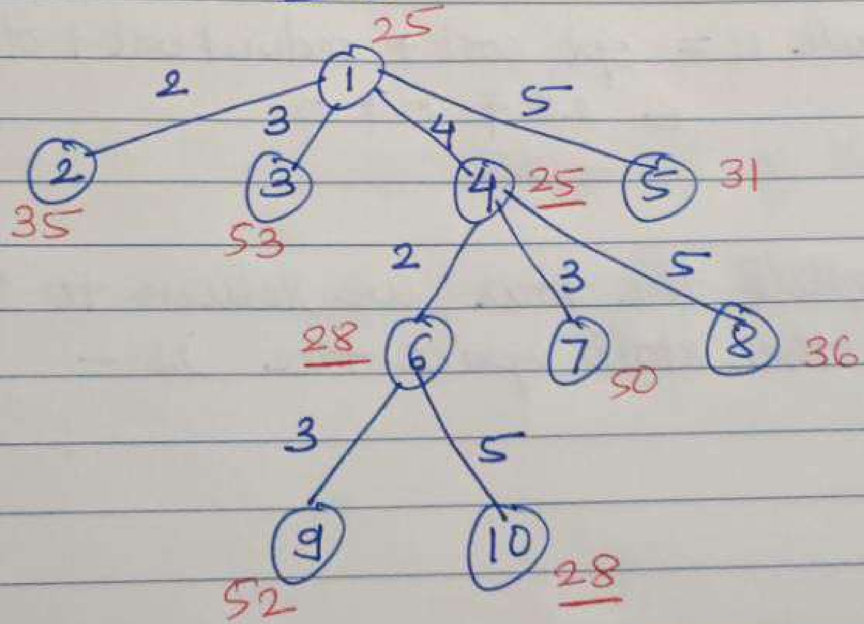
$\downarrow$
11

cost of node 9 = 28 + 13 + m[2,3] − old val
= 28 + 13 + 11
= 52

→ consider path 1-4-2-5 for node 10.
set 1^{st} row = 4^{th} row = 2^{nd} row = ∞
4^{th} col = 2^{nd} col = 5^{th} col = ∞
set $m[4,1] = m[2,1] = m[5,1] = \infty$

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty \end{bmatrix}$$

cost of node 10 = 28 + reduced cost + old val m[2][5]
= 28 + 0 + 0
= 28



cost of node 10 = 28 = minimum
∴ node 10 - becomes E-node now.

only child being generated is node 11.

consider path 1-4-2-5-3.

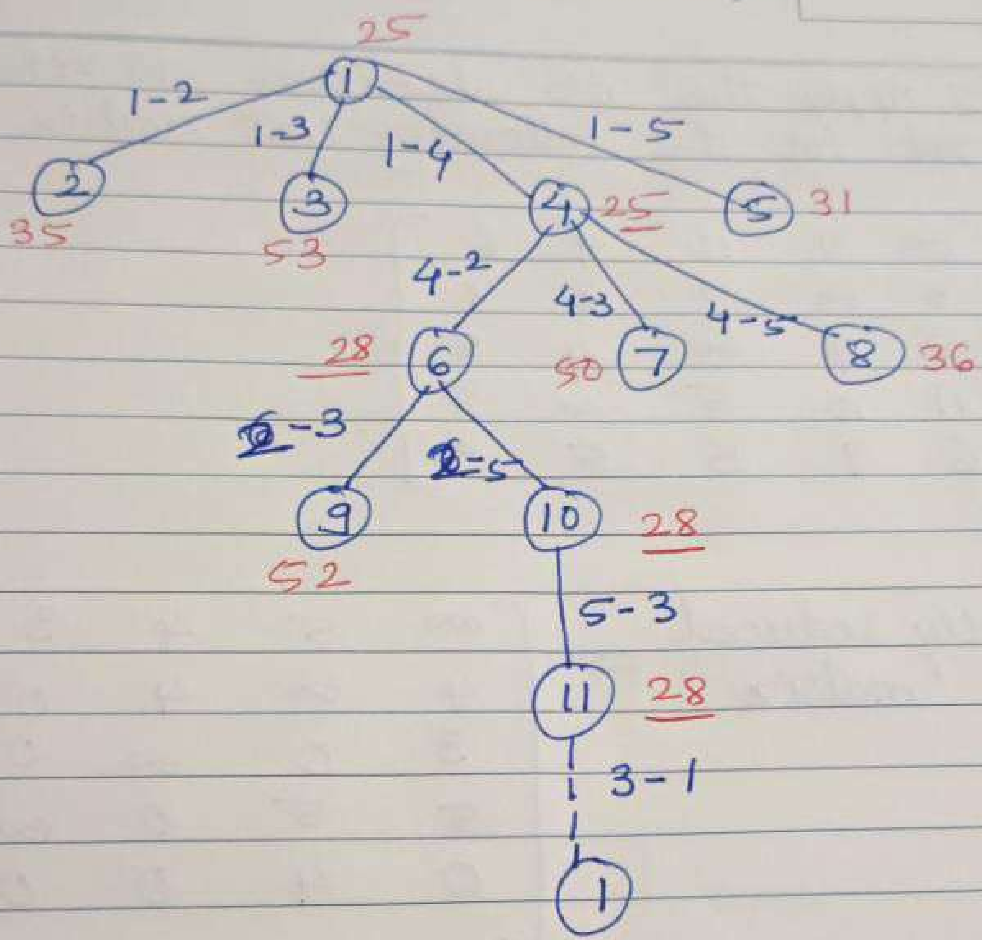
set 1^{st} row = 4^{th} row = 2^{nd} row = 5^{th} row = ∞
4^{th} col = 2^{nd} col = 5^{th} col = 3^{rd} col = ∞

set $m[4,1] = m[2,1] = m[5,1] = m[3,1] = \infty$

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

cost of node 11 = opt cost + reduced cost + old val $m[5,3]$
$= 28 + 0 + 0$
$= 28$

To complete the tour, we return to 1.
Hence, the state space tree is -



Hence, the optimum cost of the tour = 28

& path is 1 − 4 − 2 − 5 − 3 − 1

Ex[n] Apply the BB algorithm to solve the TSP for the following cost matrix –

$$\begin{bmatrix} \infty & 11 & 10 & 9 & 6 \\ 8 & \infty & 7 & 3 & 4 \\ 8 & 4 & \infty & 4 & 8 \\ 11 & 10 & 5 & \infty & 5 \\ 6 & 9 & 5 & 5 & \infty \end{bmatrix}$$

Soln

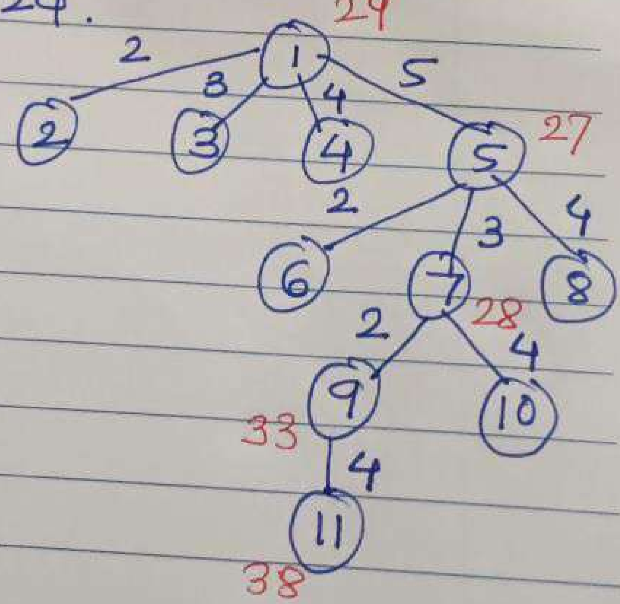
fully reduced matrix $= \begin{bmatrix} \infty & 5 & 4 & 3 & 0 \\ 4 & \infty & 4 & 0 & 1 \\ 3 & 0 & \infty & 0 & 4 \\ 5 & 5 & 0 & \infty & 0 \\ 0 & 4 & 0 & 0 & \infty \end{bmatrix}$

total reduced cost = total reduced raw cost + total reduced col cost

= 23 + 1

= 24.

The final state spae tree –



The tour with minimum cost = 38

path is 1–5–3–2–4–1