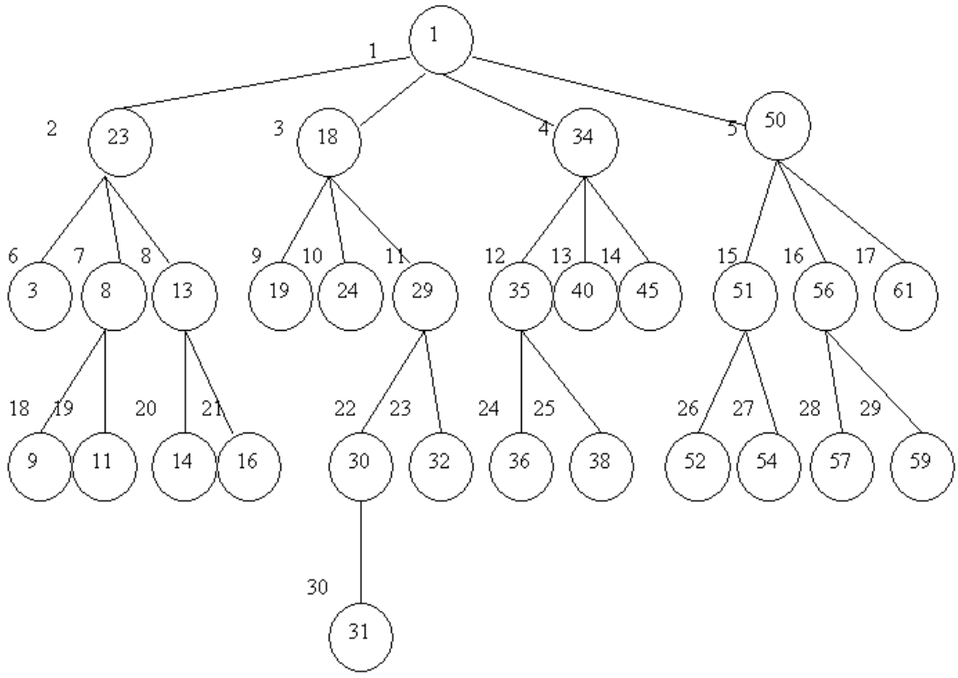
UNIT – VI

BRANCH AND BOUND

The term branch and bound refers to all state search methods in which all children of the E-node are generated before any other live node can become the Enode

• BFS:- like state space search will be called FIFO
• D Search:- like state space search will be called LIFO
Example:
A FIFO branch-and-bound algorithm searches the state space tree for eight queen problem as follows:



In the above LIFO and FIFO branch-and-bound the selection rule for the next Enode does not give any preference to a node that has a very good change of getting the search to an answer node quickly

Solution:
Least cost(LC) search:
The search for an answer node can often speeded by using an "intelligent" ranking function c(.) for lives nodes. The next E-node is selected on the basis of this ranking function

Definition:
A search strategy that uses a cost function c(x)=f(h(x))+g(x). To select the next E-node would always choose for its next E-node with least c(.). Hence, such a such strategy is called an LC-search(least cost search).

15-puzzle example:

1) To determine whether the goal state is reachable from the initial sate. Let position(i) be the position number in the initial sate of the tile numbered i. Then position (lb) will denote the position of the empty spot

	3	4	15
2		5	12
7	6	11	14
8	9	10	13

figure (a)

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

figure (b)

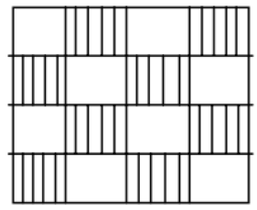


figure (c)

For any state let(i) be the number of tiles j such that j<I and position(j)>position(i).
e.g; Less(1)=0, less(4)=1& less(12)=6

Let x=1 if in the initial state the empty spot is at one of the shaded positions of figure(c). otherwise x=0.

Theorem:

The goal state of figure (b) is reachable from the initial state iff $\sum_i{}^{16}$ =less(i)+x is even.

2. Cost Estimation
One possible choice for g(x) is g(x)=number of nonblank tiles not in their goal position.

Example:
An LC search of the figure [part of the state space tree for the puzzle] will begin by using node 1 as the E-node.

a) All children of node 1 are generated and node 1 dies and leaves behind the line nodes 2,3,4 and 5.

b) The next node to become E-node is a live node with least c^(x). Now, c^(2)=1+4, c^(3)=1+4, c^(4)=1+2, c^(5) =1+4. Hence node 4 becomes Enode.

c) All children of node 1 are generated and node 1 dies and leaves behind the line nodes 2,3,4 and 5.

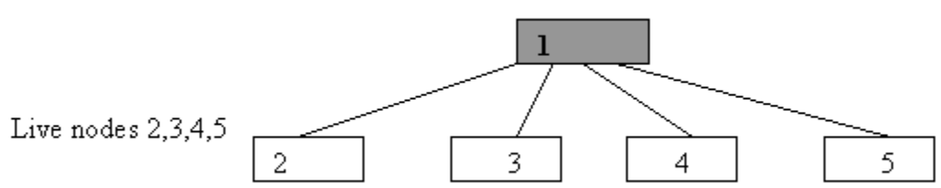
d) The next node to become E-node is a live node with least c^(x). Now, c^(2)=1+4, c^(3)=1+4, c^(4)=1+2, c^(5) =1+4. Hence node 4 becomes Enode.

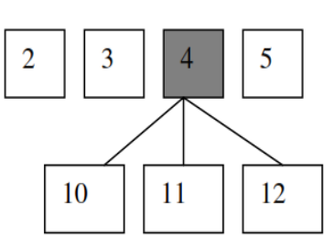
Fourth node children are generated. The live nodes at this time are 2,3,5,10,11, and 12.

C^(10)=2+1, c^(11)=2+3, c^(12)2+3 Hence node 10 becomes E-node [since the live node with least c^ is node 10]

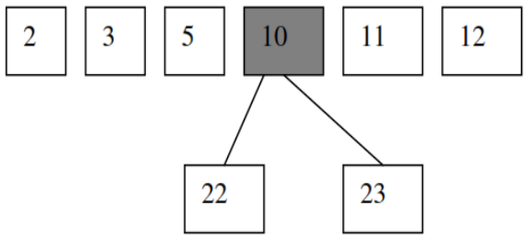
e) from node 10, nodes 22, and 23 are generated next. Hence node 23 is the goal node, the search terminates[c^=0]

figure(a)





Fig(b)



Node 4 dies
Live nodes 2,3,5,10,11,12
Least select node 10

fig(c)

FIFO BRANCH-AND-BOUND

[BFS+FIFO+Bounding Condition]

eg:

problem:

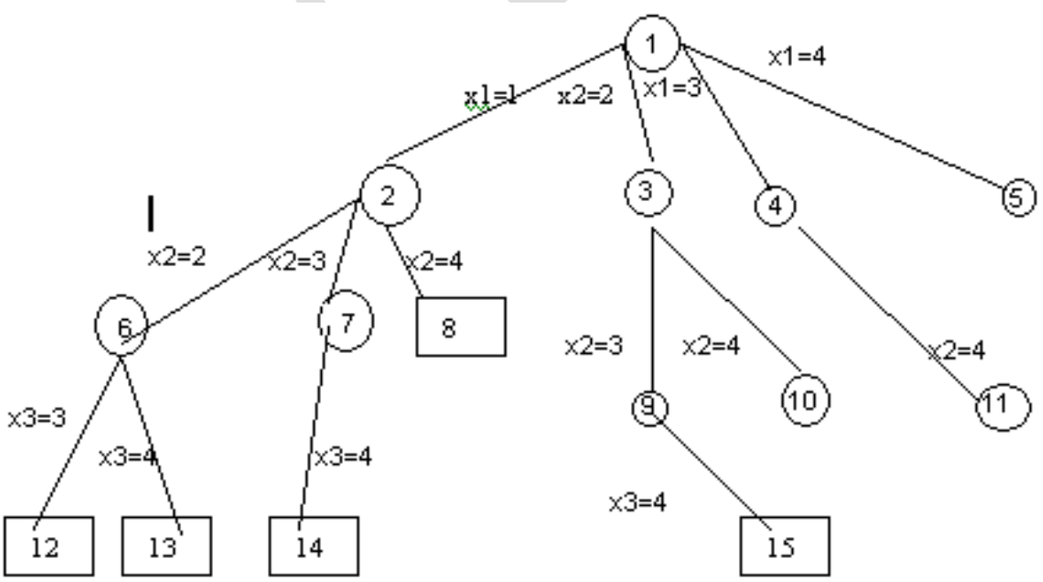
We are given
- n-jobs, one processor
- job i has associated with a three tuple(pi,di,ti)

Pi=penalty incurred when the processing not completed by the dead line di
Ti=required units of processing time

Objective is to select a subset J of n jobs such that all jobs in J can be completed by their deadlines & the penalty incurred is minimum among all possible subsets J.(a penalty can be incurred only on those jobs not in J).

Let n=4; (p1,d1,t1)=(5,1,1);(p2,d2,t2);=(10,3,2)
(p3,d3,t3)=(6,2,1);(p4,d4,t4);=(3,1,1)

Solution space tree for the above problem instance is:



◯ = Infeasible Subsets. Jobs index set {1,2,3,4}

□ = Answer nodes if x1=1;x2=4 in the tree J={1,4}

Bounding:
Cost Function: c(x)

For any circular node x, c(x) is the minimum penalty corresponding to any node in the sub tree with root x.
For example, c(x)= ∞ for sequence node
C(3)=8; c(2)=9;c(1)=8 etc.

A bound c^(x):
Let Sx be the subset of jobs selected for J at node x. if m=max{i | i∈Sx), then

$C^(x)=\sum_{i<m} pi$ is an estimate for c(x).

Example:
For mode 7, S7 ={1,3} and m=3 .
Therefore,

$\sum_{i<3} Pi=P2=10$

Upper bound u(x): (Cost of a minimum-cost answer node)
$U(x)=\sum_{i \text{ not belongs to } S2} Pi = \sum^{4}_{i=2} Pi = P2+P3+P4 =19$ S2={ 1 }

LC BRANCH AND BOUND
(LCBB for given Job Schedule Problem Instance)

Procedure:
Step 1: Set upper = ∞ or $\sum_{i=1 \text{ to } n} Pi$
Step 2: Node 1 is Enode. It is expanded. Children 2,3,4,5 are generated.
Step 3: C^(x) is calculated with each child node x of node1.
If u (x) is minimum than upper then upper will set to u(x).
Hence upper becomes u(3) i.e. 14.
C^(4), C^(5) > upper. So 4,5 nodes get deleted.
Step 4: Next E-Node is 2. Since C^(2) < C^(3) (Least cost BB).
Nodes 6,7,8 generated.
C^(7) > upper and 8 is infeasible, both are killed.
Step 5: Next E-Node is 6. Since out of all give nodes i.e. 6, 3.
C^(6) < C^(3) (i.e. Least cost BB)
All its children one generated i.e. 12, 13 but both are infeasible.
Step 6: Next E-Node is 3. Node 9 (child of 3) is generated. u(9) < upper and
C^(9) < upper. So upper becomes u(9) i.e. 8.
Node 10 is killed. Since C^(10) > upper i.e. 8.
Step 7: Next E-Node is node 9. It s child is infeasible. Here no live node remains. The Search terminates with node 9 representing minimum-cost as node.

Procedure with Example

X	1	2	3	4	5	6	7	8	9	10	11
C∧()	0	0	5	15	21	0	10	-	5	11	15
U()	24	19	14	18	21	9	10	16	8	1	15

- Set upper = ∞ (or upper =∑ $_{i=1 \text{ to } n}$ Pi) // upper bound on the cost of a minimum cost and node.
- Set node 1 as E-node. Generate child nodes.
- Upper will be set to 19 than 14(when node 3 is generated)
- If c^(x) for current generated child is > upper than kill the nodes. Hence nodes 4,5 get killed.
- Node 2 becomes next E-node. Generate children nodes 6,7,8. u(6)=9; hence upper=9. C^(x) for node 7 > upper. So 7 get killed. Node 8 is infeasible so it's killed.
- Node 3 becomes E-node I nod 9,10 generated. U(9)=8; hence upper=8. C^(10) > upper hence node 10 is killed.
- Node 6 becomes E-node; its children are infeasible.
- Node 9 becomes E-node; its child is infeasible.

Hence minimum cost answer node is 9, it has a cost of 8.

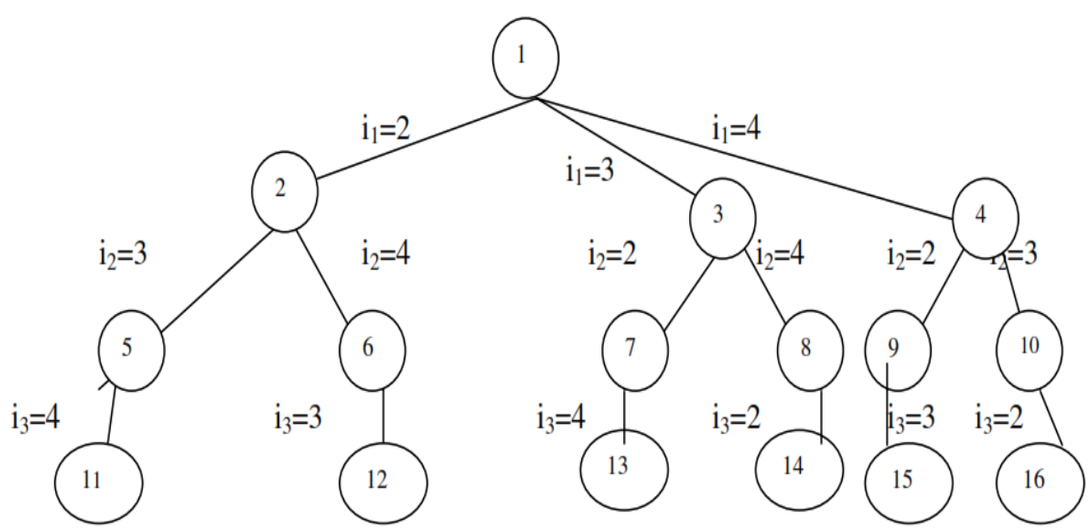
Travelling salesman problem

Let G=(V,E) be a directed graph
Let Cij be the cost of edge<i,j>,Cij=∞ if <i,j> does not belongs to E
Let | V | = n.
Every tour starts and ends at vertex 1.

Objective:
To find minimum cost tour.

Solution:
In order to use LC Branch & Bound to search the travelling salesperson tree, we need to define a cost function C()and other two functions C^(),u().

State space tree for the travelling salesperson problem with n=4 & i_0= i_1=1.



A cost estimation C^(.)such that C(A) ≥ C^(A) for all node A is obtained by defining C^(A) to be the length of the path defined at node A.

Procedure LCBB :

Step 1: A matrix is reduced by reducing rows and column of the matrix. A row(column) is said to reduced iff it contains atleast one zero and all remaining entries are non-negative.

Step 2: C (.) may be obtained by using the reduced cost matrix corresponding to G.
C(.) =[sum of minimum row value]+[sum of minimum column value]

Step 3: If an edge <i,j> in the tour, then change all entries in row i and column j of A to ∞.
Let A be a reduced cost matrix for node R. Let S be child of R.

Step 4: Set A(j,1) to ∞ .

Step 5: Reduce all rows and columns in the resulting matrix except for rows and columns Containing only ∞. Let the resulting matrix be B.

Step 6: Step 3 and Step 4 are valid as no tour in the sub tree S can contain edges of the type < i , k> or < k ,j >.

Step 7: If r is the total amount subtracted in Step 5 then C(S)=C(R)+A(i , j) + r.

Step 8: If leaf nodes C(.) = c( ) is easily computed as each leaf defines a unique tour.

Step 9: For the upper bound function u, we may use u(R) = ∞ for all nodes R.

Example: Cost Matrix

$$\begin{bmatrix} \infty & 7 & 3 & 12 & 8 \\ 3 & \infty & 6 & 14 & 9 \\ 5 & 8 & \infty & 6 & 18 \\ 9 & 3 & 5 & \infty & 11 \\ 18 & 14 & 9 & 8 & \infty \end{bmatrix}$$

Reduce by Row wise

$$\begin{bmatrix} \infty & 4 & 0 & 9 & 5 \\ 0 & \infty & 3 & 11 & 6 \\ 0 & 3 & \infty & 1 & 13 \\ 6 & 0 & 2 & \infty & 8 \\ 10 & 6 & 1 & 0 & \infty \end{bmatrix} \quad \begin{matrix} 3 \\ 3 \\ 5 \\ 3 \\ 8 \end{matrix}$$

Reduce by Column wise

$$\begin{bmatrix} \infty & 4 & 0 & 9 & 5 \\ 0 & \infty & 3 & 11 & 6 \\ 0 & 3 & \infty & 1 & 13 \\ 6 & 0 & 2 & \infty & 8 \\ 10 & 6 & 1 & 0 & \infty \end{bmatrix}$$

$$\begin{matrix} 0 & 0 & 0 & 0 & 5 \end{matrix}$$

$$\hat{c}(x)=[3+3+5+3+8]+[5\}=27$$

(1,2)- Make all the elements in 1St row and 2nd column to ∞ and A(2,1) to ∞ .

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 3 & 11 & 1 \\ 0 & \infty & \infty & 1 & 8 \\ 6 & \infty & 2 & \infty & 3 \\ 10 & \infty & 1 & 0 & \infty \end{bmatrix}$$

Reduce by Row wise:

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 2 & 10 & 0 \\ 0 & \infty & \infty & 1 & 8 \\ 4 & \infty & 0 & \infty & 1 \\ 10 & \infty & 1 & 0 & \infty \end{bmatrix} \begin{matrix} 1 \\ 0 \\ 2 \\ 0 \end{matrix}$$

Reduce by Column wise

(Same as previous matrix since every Column has an element 0)

C(S)=C(R)+A(i,j)+r.
=27 + 4 + 3 = 34

(1,3) 3

∞	∞	∞	∞	∞	
0	∞	∞	11	~~10~~	0
∞	~~2~~ 2	∞	~~10~~	~~8~~ 7 ~~6~~	1
6	0	∞	∞	~~8~~ 2	0
10	6	∞	0	∞	0
0	0		0	1	

27+0+2=29

(1,4) 4

∞	∞	∞	∞	∞	
0	∞	3	∞	~~10~~	0
0	3	∞	∞	~~8~~ 7	0
∞	0	2	∞	~~8~~ 2	0
~~10~~ 9	~~8~~ 5	~~10~~	∞	∞	1
0	0	0		1	

27+0+2=38

(1,5) 5

∞	∞	∞	∞	∞	
0	∞	~~3~~ 2	11	∞	0
∞	3	∞	1	∞	0
6	∞	~~2~~ 1	∞	∞	0
∞	6	~~1~~ 0	0	∞	0
0	0	1	0		

27+0+1=28

(1,5) is minimum

(1,5,2) 6

∞	∞	∞	∞	∞	
∞	∞	~~2~~ 0	~~11~~ 9 ~~8~~	∞	2
∞	∞	∞	~~1~~ 0	∞	0
~~6~~ ∞	∞	~~1~~ 0	∞	∞	1
∞	∞	∞	∞	∞	
0		0	1		

28+6+4=38

(1,5,13) 7

∞	∞	∞	∞	∞	
0	∞	∞	11	∞	0
∞	~~2~~ 2	∞	~~10~~	∞	1
6	0	∞	∞	∞	0
∞	∞	∞	∞	∞	
0	0		0		

(1, 5, 4) 8

∞	∞	∞	∞	∞	
0	∞	~~2~~ 1	∞	∞	0
0	3	∞	∞	∞	0
∞	0	~~1~~ 0	∞	∞	0
∞	∞	∞	∞	∞	

0 0 1

28 + 0 + 1 = 29

Since (1,5,3) & (1,5,4) = 29

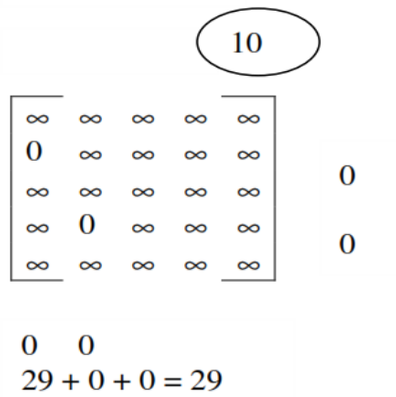
LCBB try for both case

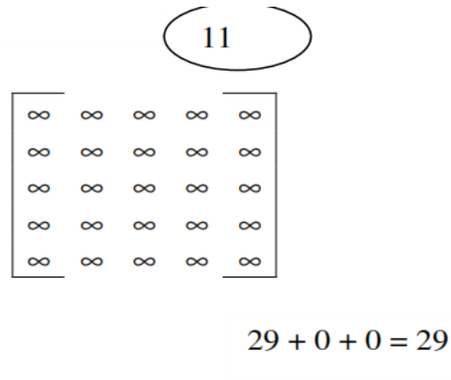
(1, 5, 3,2)

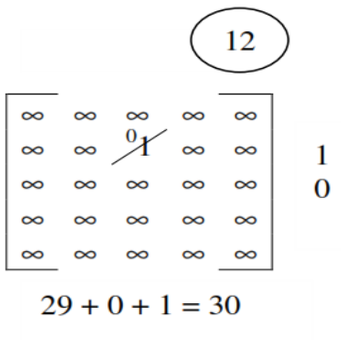
∞	∞	∞	∞	∞	
∞	∞	∞	~~0~~ 11	∞	11
∞	∞	∞	∞	∞	
6	∞	∞	0	∞	0
∞	∞	∞	∞	∞	

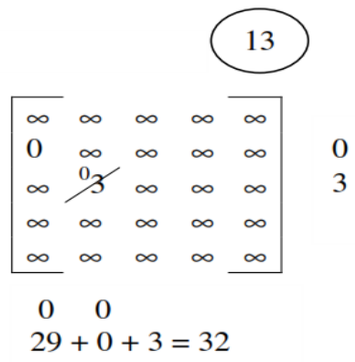
6 0

29 + 2 + 17 = 48



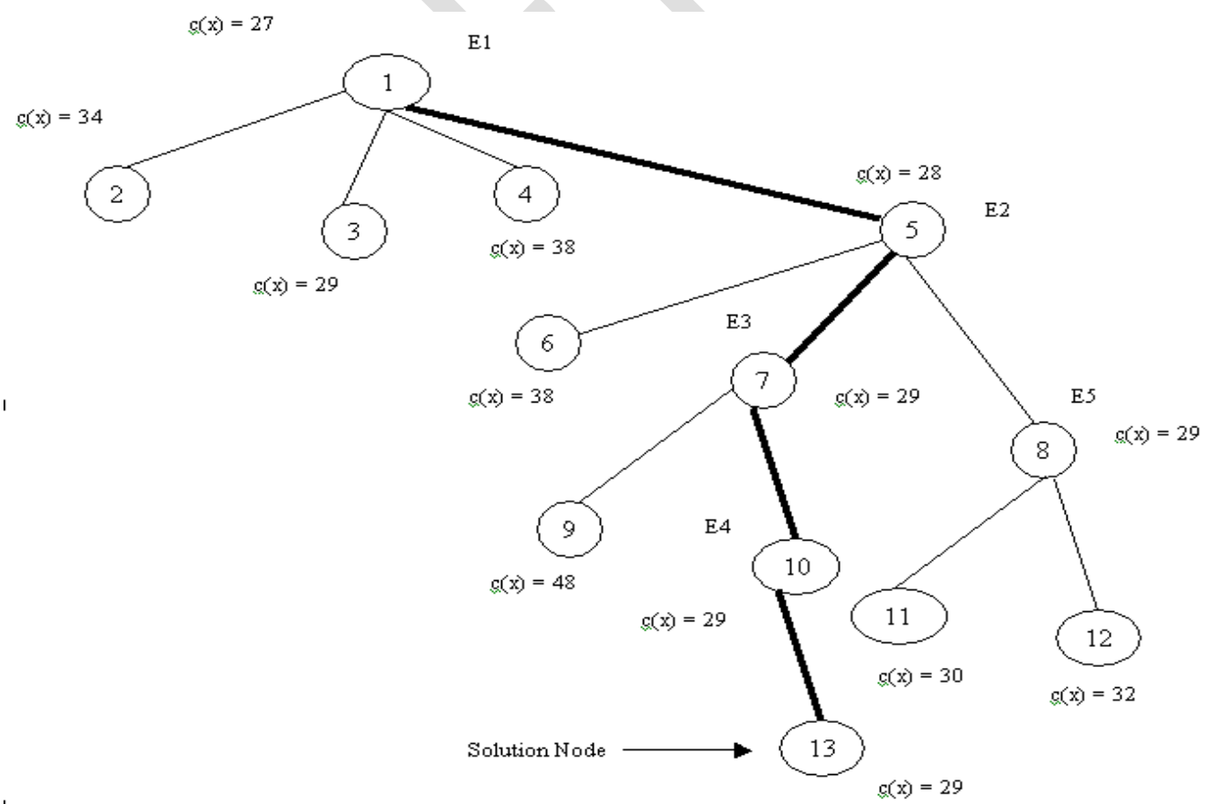






Hence the four has the path nodes 1 -> 5-> 2-> 4 -> 2->1
Cost that is 8+9+6+3+3 = 29

STATE SPACE TREE GRENERATED BY PROCEDURE LCBB



E1, E2, ……. E5 – are E $_{num}$ nodes E indicates order of selecting next E node.
(next E-nod e is selected based on Least Cost ie c ^(x)).