# IOE543 - Job Shop Problem Set

## March 15, 2016

**TABLE OF CONTENTS**

**PAGES**

**Example 1: Decomposition with Johnson's Rule……2**

**Example 2: Total enumeration…….3-8**

**Example 3: Branch-and-Bound with Simple Lower Bounds on $C_{max}$…….9-13**

**Example 4: Branch-and-Bound with Improve Bound on $C_{max}$ using $1|r_j|L_{max}$…….14-19**

***Example:*** Optimal schedule for $J_2 \,||\, C_{max}$

| *Job* | *Sequence* | *Processing Time* |
|---|---|---|
| 1 | 1,2 | $p_{11}= 2$ , $p_{21}= 5$ |
| 2 | 1,2 | $p_{12}= 6$ , $p_{22}= 3$ |
| 3 | 2,1 | $p_{23}= 8$ , $p_{13}= 6$ |

Decompose this instance of $J_2 \,||\, C_{max}$ into two instances of $F_2 \,||\, C_{max}$. Solve using the method in Lecture 14.

$J_{1,2} = \{1,2\}$ , $J_{2,1} = \{3\}$

$J_{1,2}$ schedule : Set I = {1} , Set II = {2}

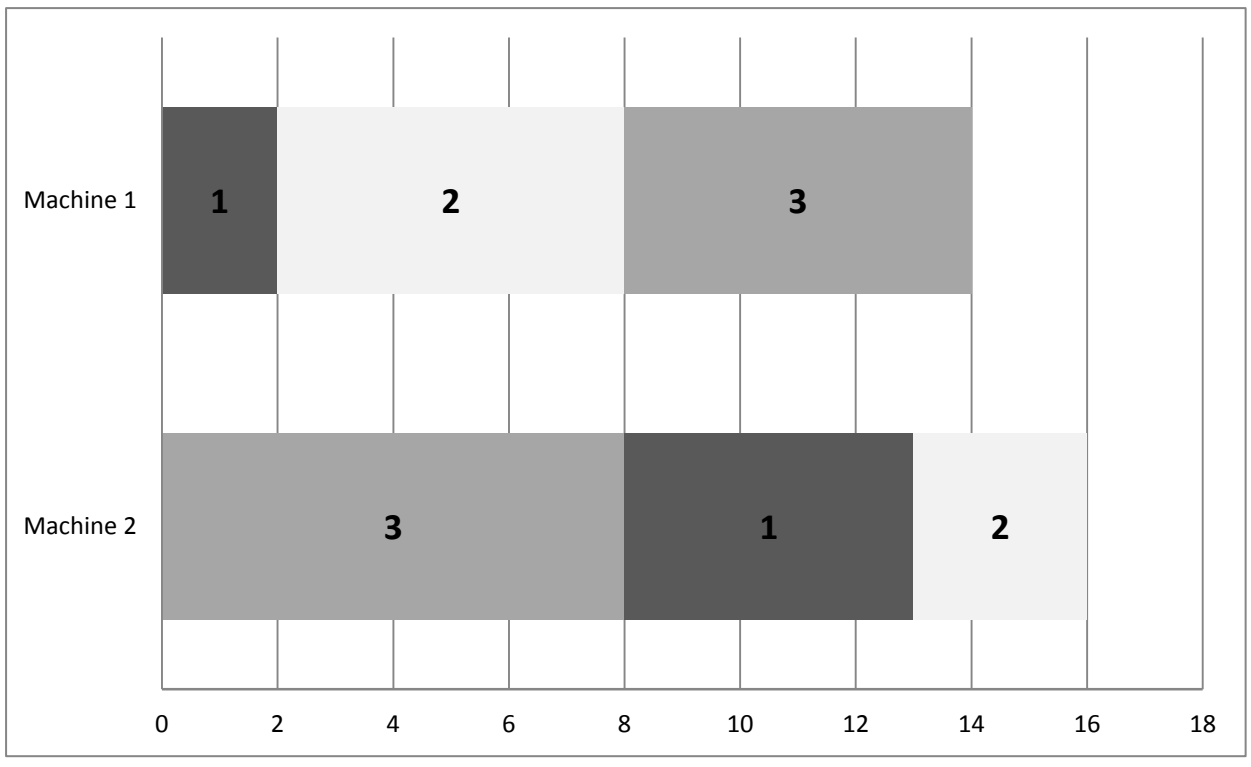
$\Longrightarrow$ sequence is $1 \rightarrow 2$ on machine 1

$J_{2,1}$ schedule : Set I = {3} , Set II = {∅}

$\Longrightarrow$ optimal schedule is :

| *Machine* | *Sequence* |
|---|---|
| 1 | 1 , 2 , 3 |
| 2 | 3 , 1 , 2 |

***Gantt Chart***



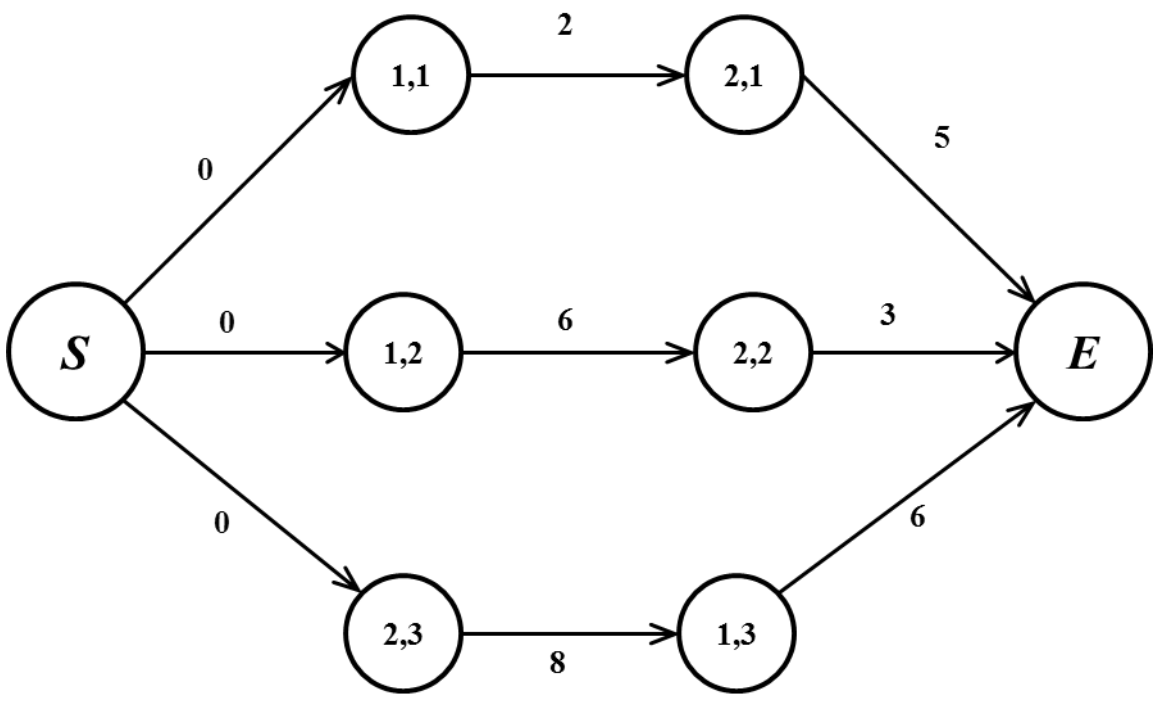

$\boldsymbol{C_{max} = 16}$

***Example:*** Total enumeration for $J_2 \,||\, C_{max}$

| *Job* | *Sequence* | *Processing Time* |
|---|---|---|
| 1 | 1,2 | $p_{11}= 2$ , $p_{21}= 5$ |
| 2 | 1,2 | $p_{12}= 6$ , $p_{22}= 3$ |
| 3 | 2,1 | $p_{23}= 8$ , $p_{13}= 6$ |

See the complete branch-and-bound tree on page 9 for complete details about the finished tree.

Start with the initial graph with conjunctive arcs to branch on node 0 (the root node):



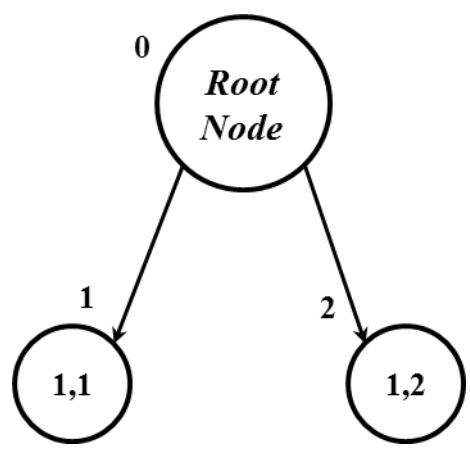

*Node 0:*

**Ω** = { (1,1) , (1,2) , (2,3) }

**t(Ω)** = min( 0+2 , 0+6 , 0+8 ) = 2

***i**** = 1

⟹ **Ω'** = { (1,1) , (1,2) }



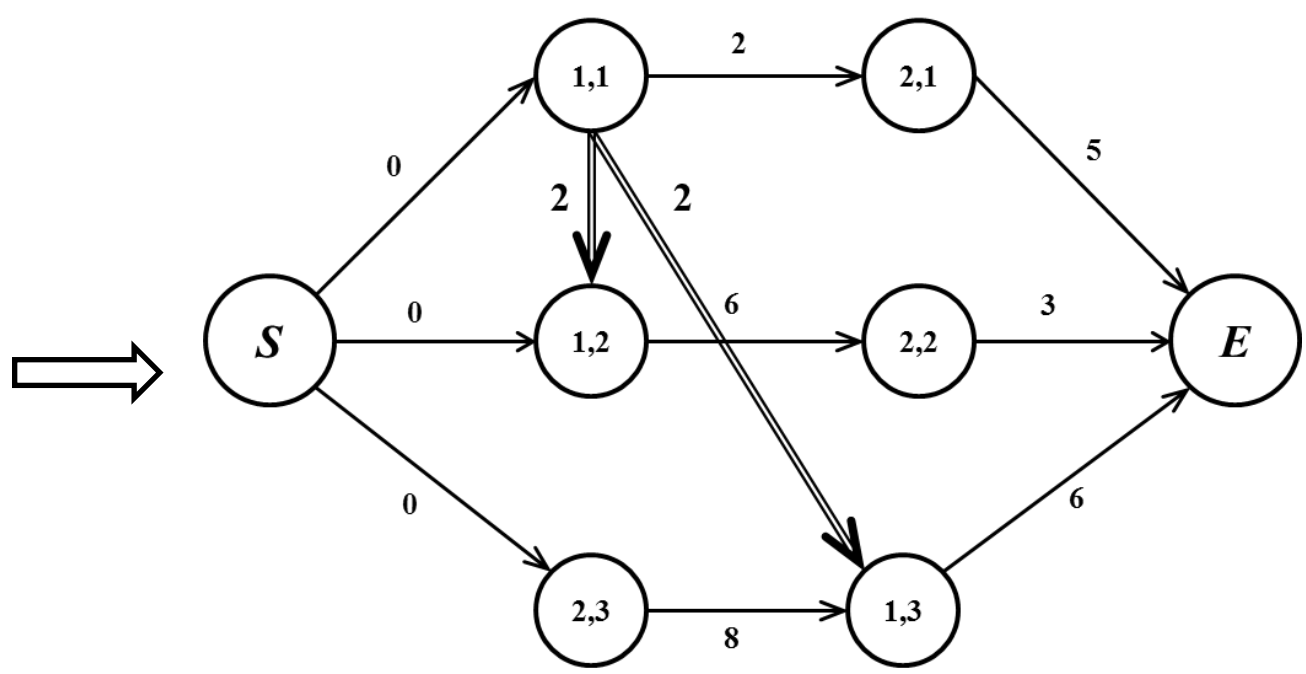

*Node 1:*

**Ω** = { (2,1) , (1,2) , (2,3) }

**t(Ω)** = min( 2+5 , 2+6 , 0+8 ) = 7

***i**** = 2

⟹ **Ω'** = { (2,1) , (2,3) }



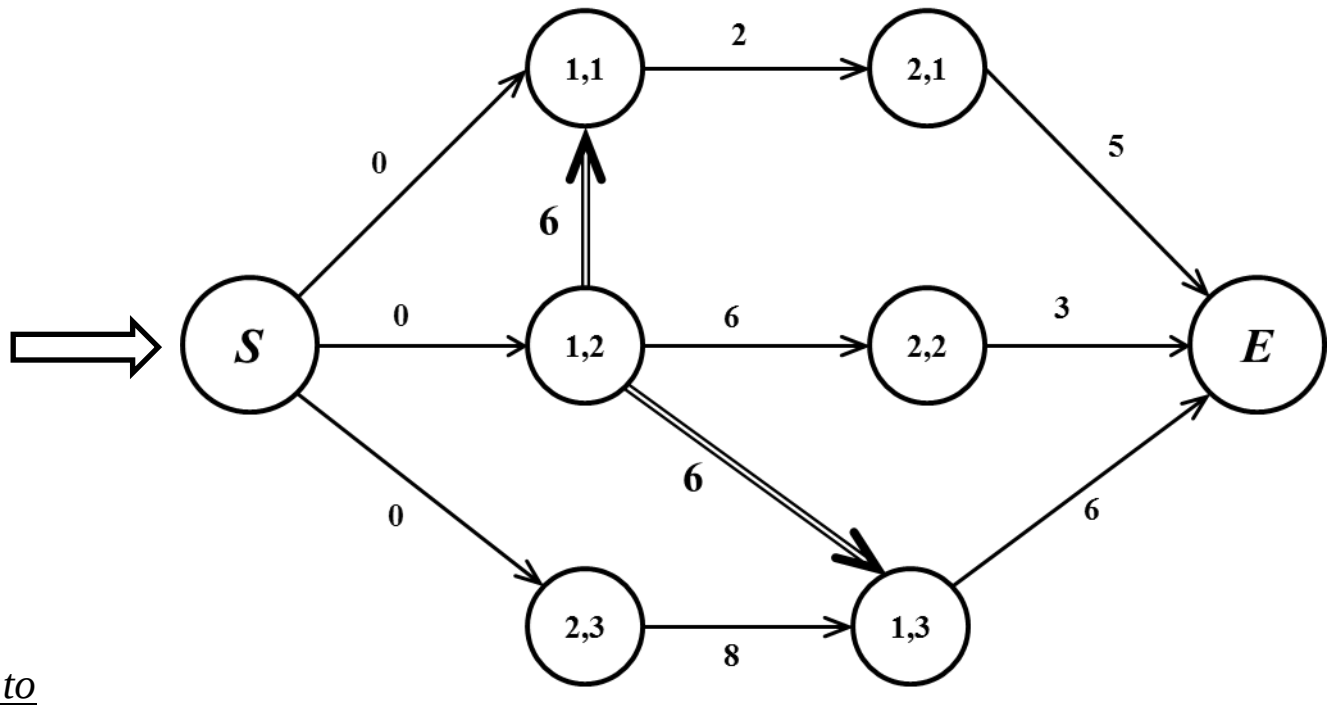
*Node 2:*

**Ω** = { (1,1) , (2,2) , (2,3) }

**t(Ω)** = min( 6+2 , 6+3 , 0+8 ) = 8

***i**** = 1

⟹ **Ω'** = { (1,1) }




*Note: we arbitrarily picked machine 1 to branch on since there was a tie.*

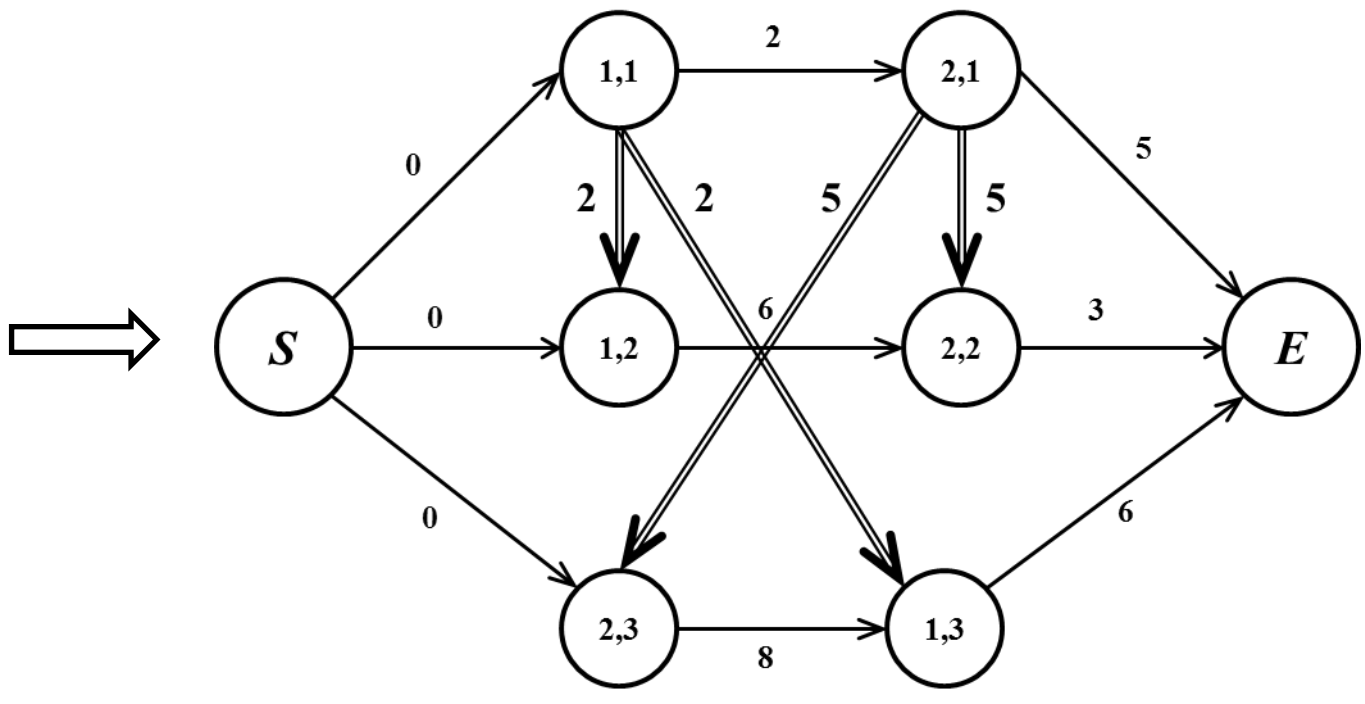
*Node 3:*

**Ω** = { (1,2) , (2,3) }

**t(Ω)** = min( 2+6 , 7+8 ) = 8

***i**** = 1

⟹ **Ω'** = { (1,2) }



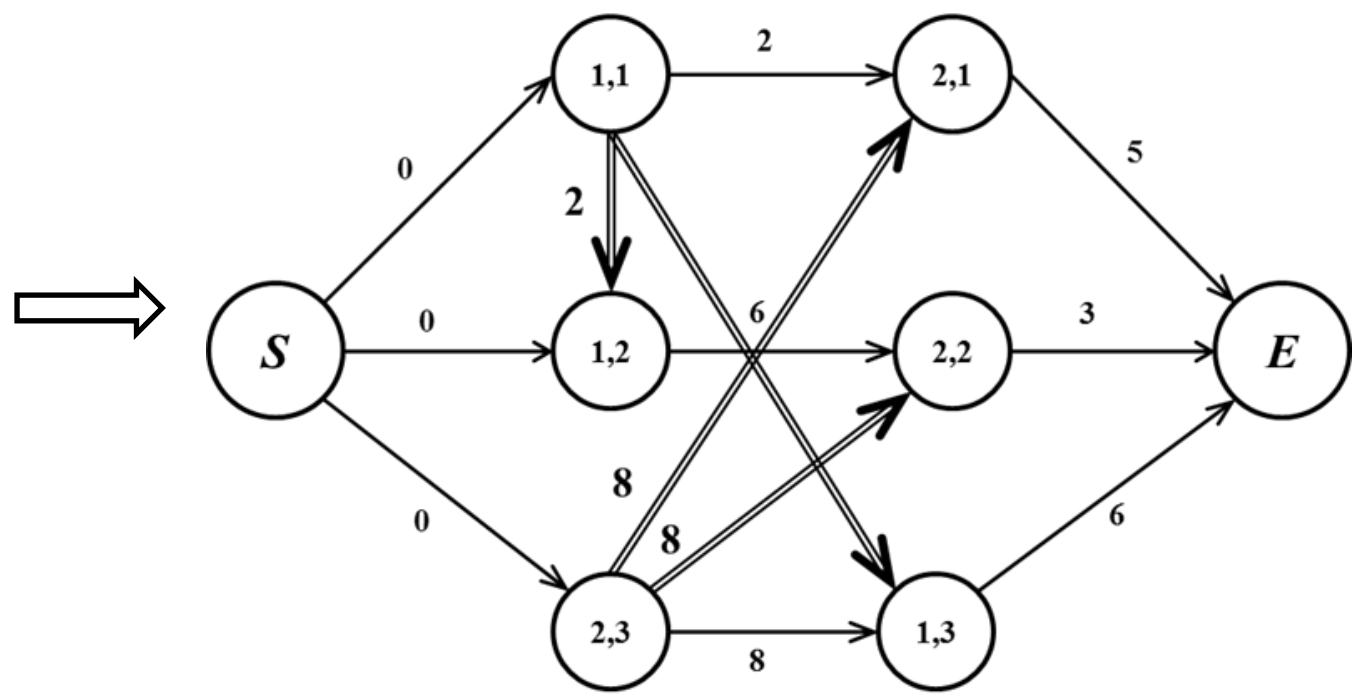

*Node 4:*

**Ω** = { (2,1) , (1,2) , (1,3) }

**t(Ω)** = min( 8+5 , 2+6, 8+6 ) = 8

***i**** =1

⟹ **Ω'** = { (1,2) }



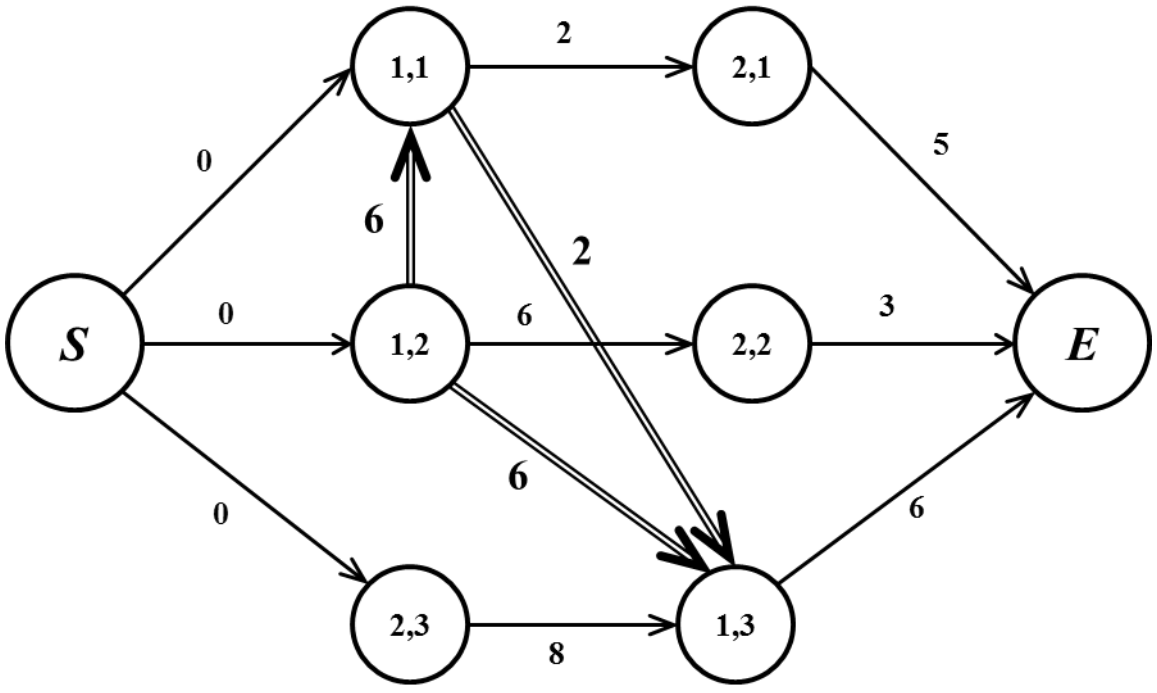
*Node 5:*

**Ω** = { (2,1) , (2,2) , (2,3) }

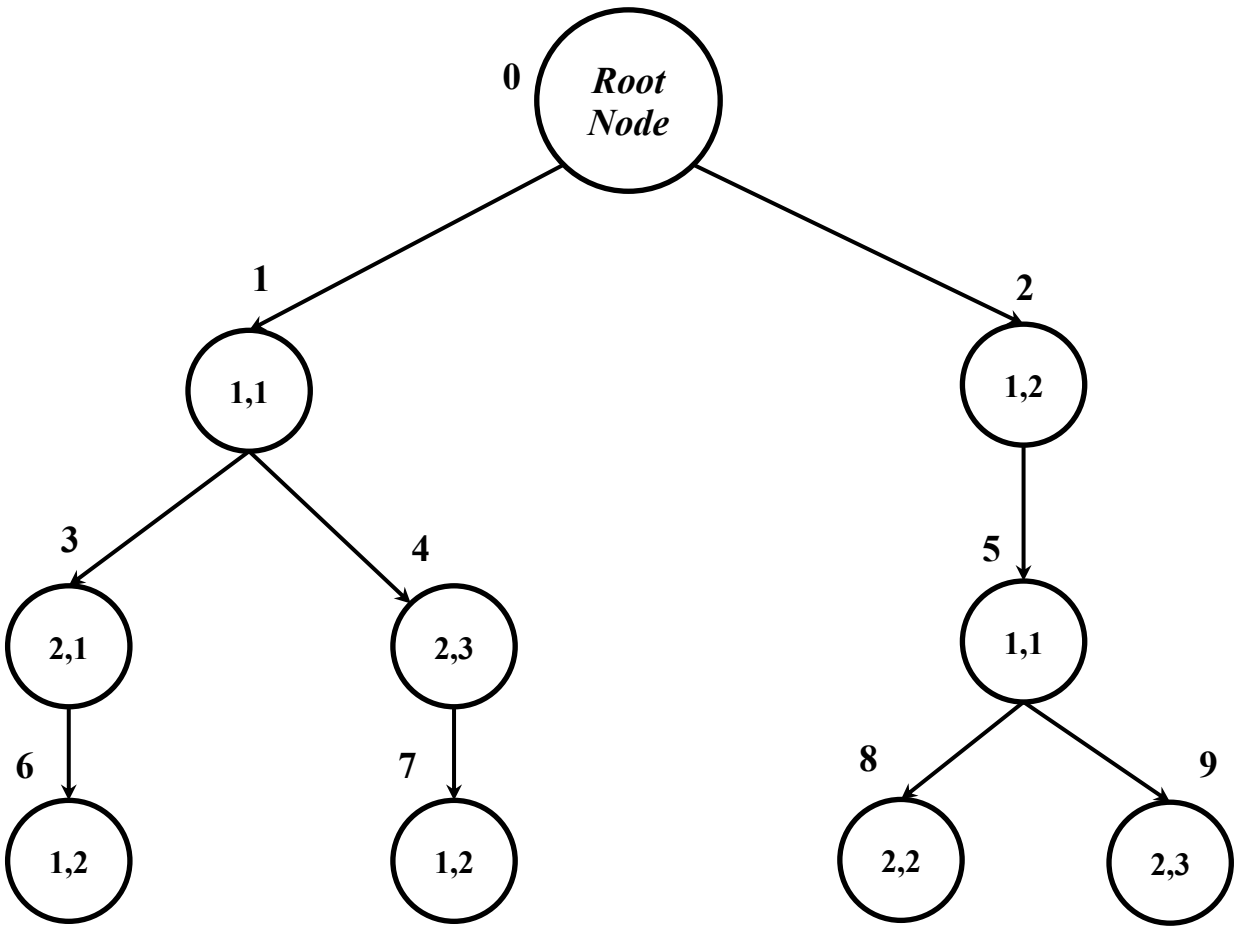
**t(Ω)** = min( 8+5 , 6+3 , 0+8 ) = 8

***i**** = 2

$\Longrightarrow$ **Ω'** = { (2,2) , (2,3) }




Following is the current branch-and-bound tree up to this point. Nodes 6,7,8,9,10 are pending nodes awaiting processing.




Process each of the pending nodes one at a time to generate the remainder of the tree.

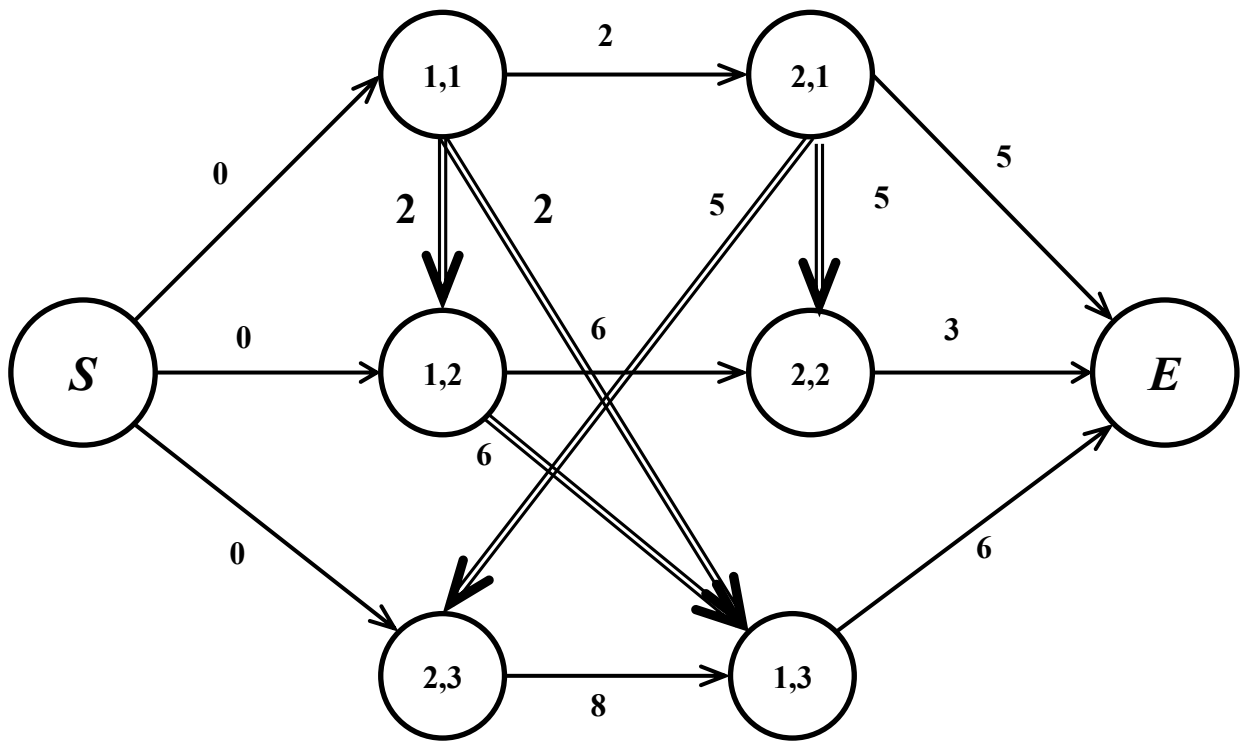
*Node 6:*

**Ω** = { (2,2) , (2,3) }

**t(Ω)** = min( 8+3 , 7+8 ) = 11

***i**** = 2

$\Longrightarrow$ **Ω'** = { (2,2) , (2,3) }

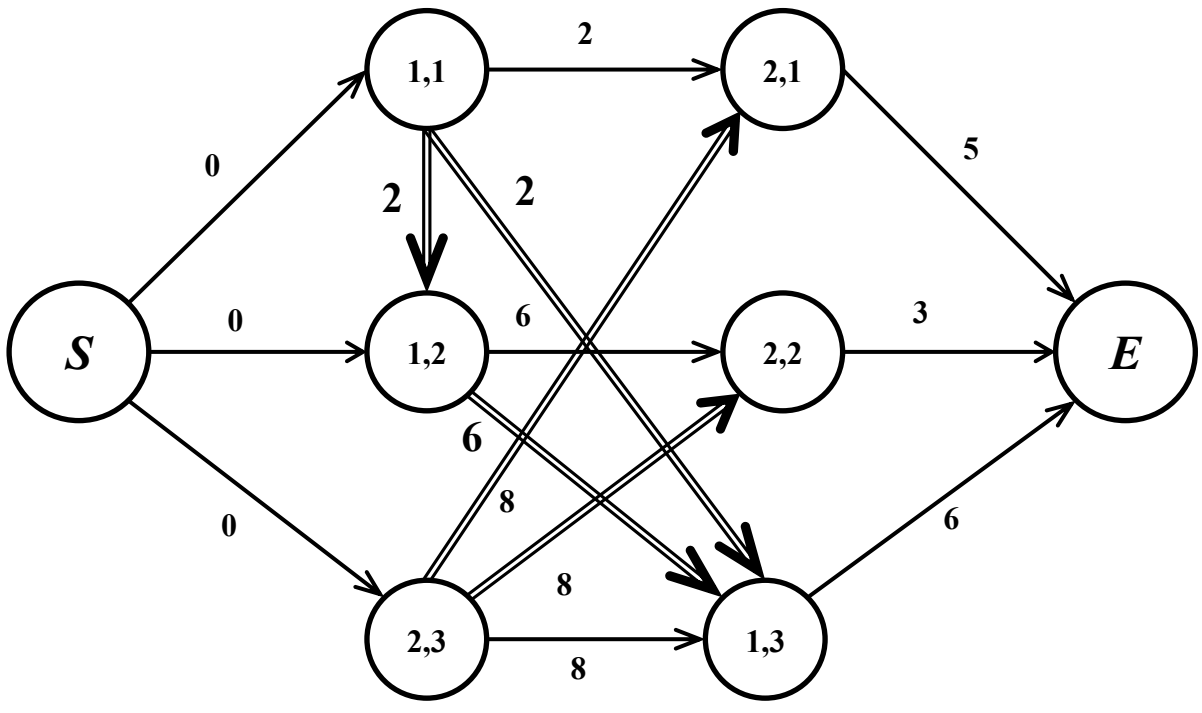



*Node 7:* **Ω** = { (2,1), (2,2), (1,3) }

**t(Ω)** = min( 8+5 , 8+3, 8+6 ) = 11

***i**** = 2

$\Longrightarrow$ **Ω'** = { (2,1) , (2,2) }



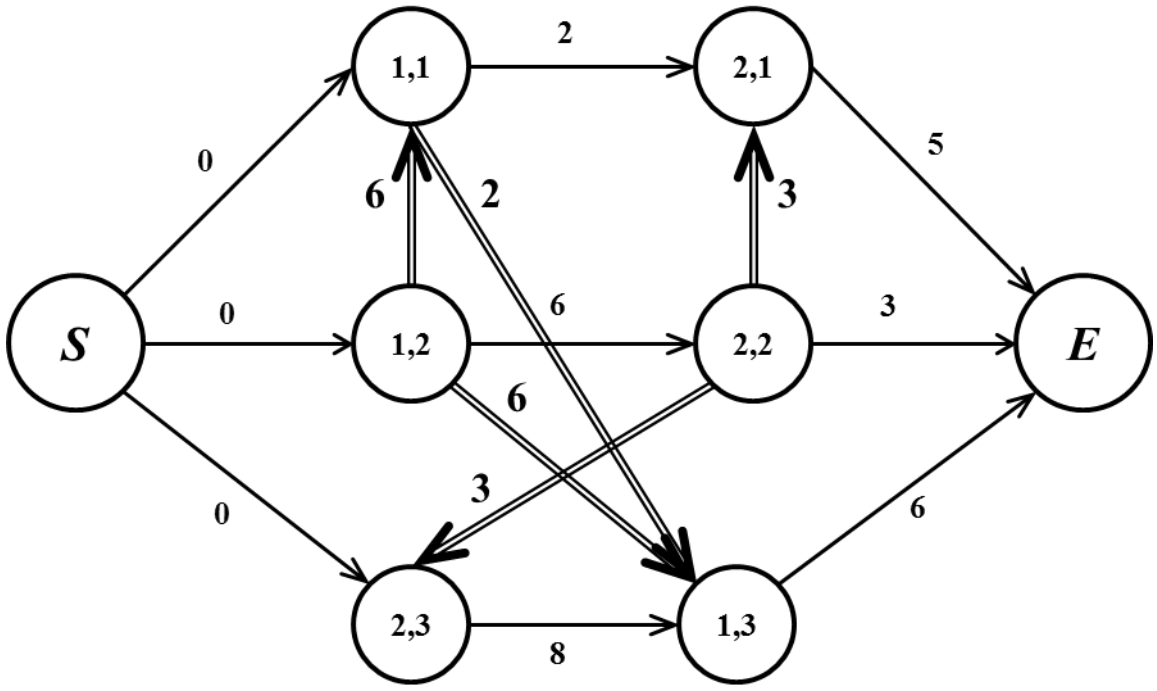

*Node 8:*

**Ω** = { (2,1) , (2,3) }

**t(Ω)** = min( 9+5 , 9+8 ) =14

***i**** = 2

$\Longrightarrow$ **Ω'** = { (2,1) , (2,3) }



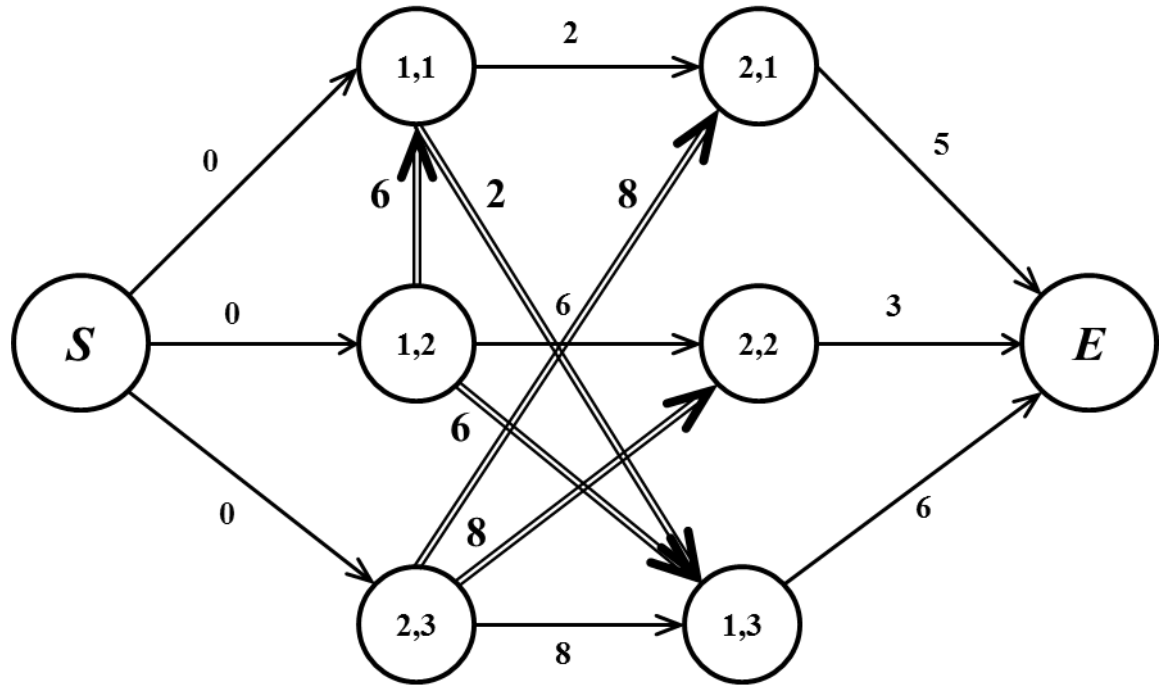
*Node 9:*

**Ω** = { (2,1) , (2,2) , (1,3) }

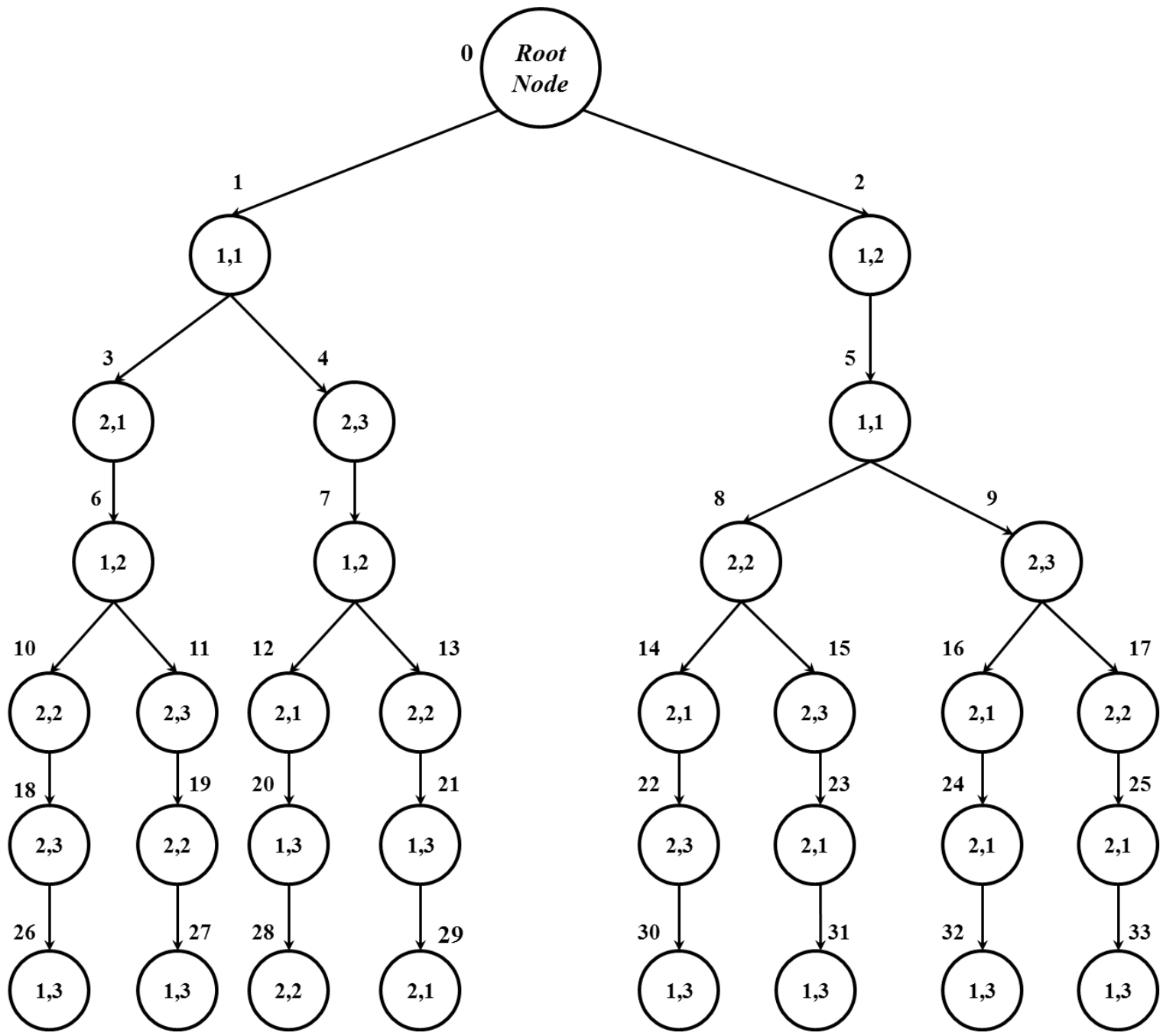
**t(Ω)** = min( 8+5 , 8+3 , 8+6 ) = 11

***i**** = 2

⟹ **Ω'** = { (2,1) , (2,2) }




Continuing on branching on nodes6,7,8,9, and so on results in the following complete branch and bound tree since the remaining levels are trivially determined.




Note that Node 28 corresponds to the optimal schedule found previously by decomposing the problem into two flow shops and applying Johnson's rule. However, there are multiple optimal solutions.

***Example:*** Branch-and-Bound for $J_2 \,||\, C_{max}$

| *Job* | *Sequence* | *Processing Time* |
|---|---|---|
| 1 | 1,2 | $p_{11}= 2$ , $p_{21}= 5$ |
| 2 | 1,2 | $p_{12}= 6$ , $p_{22}= 3$ |
| 3 | 2,1 | $p_{23}= 8$ , $p_{13}= 6$ |

In this example we will compute lower bounds for each node and fathom nodes whenever possible to avoid generating the entire branch-and-bound tree.

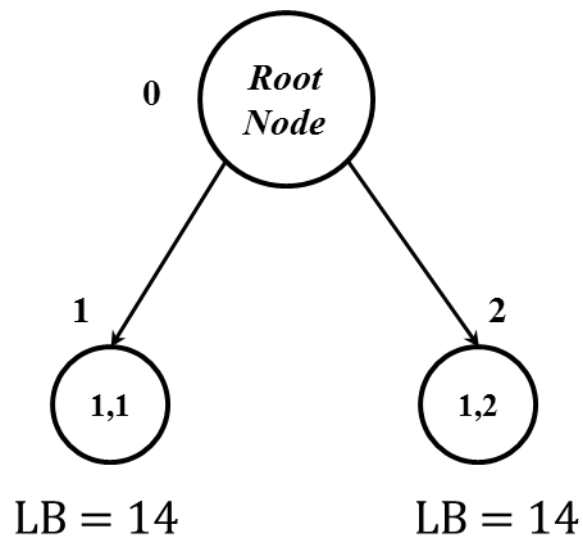
*Node 0:*

**Ω** = { (1,1) , (1,2) , (2,3) }

**t(Ω)** = min( 0+2 , 0+6 , 0+8 ) = 2

***i**** = 1

$\Longrightarrow$ **Ω'** = { (1,1) , (1,2) }




Randomly pick *Node 2* from the two pending nodes with Lower Bound (LB) = 14 and branch to (1,1).

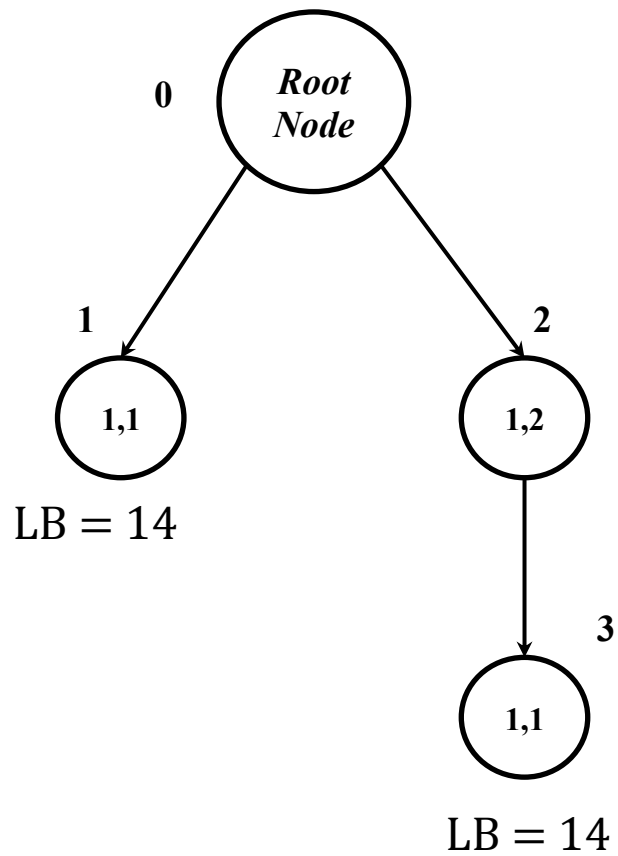
*Node 2:*

**Ω** = { (1,1) , (2,2) , (2,3) }

**t(Ω)** = min( 6+2 , 6+3 , 0+8 ) = 8

***i**** = 1

$\Longrightarrow$ **Ω'** = { (1,1) }




*Note: we arbitrarily picked machine 1 to branch on since there was a tie.*

Evaluate node 3 to get lower bound of 14. Randomly select node 3 (from nodes 1 and 3, both with LB=14) for branching.

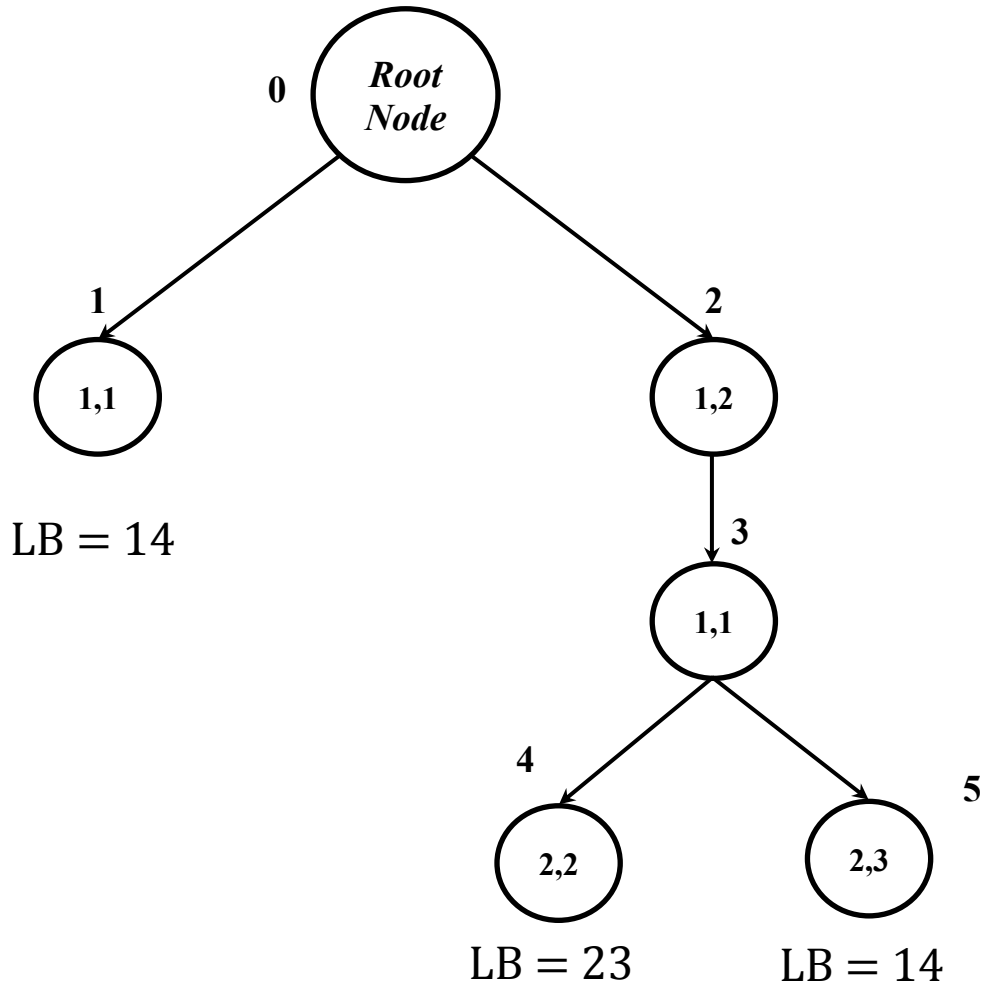
*Node 3:*

**Ω** = { (2,1) , (2,2) , (2,3) }

**t(Ω)** = min( 8+5 , 6+3 , 0+8 ) = 8

***i**** = 2

$\Longrightarrow$ **Ω'** = { (2,2), (2,3) }




Branch to two new nodes (2,2) and (2,3) with resulting lower bonds of LB=23 and LB=14 respectively.

Randomly select *Node 5* of the two pending nodes with Lower Bound (LB) = 14.

*Node 5:*

**Ω** = { (2,1) , (2,2) , (1,3) }

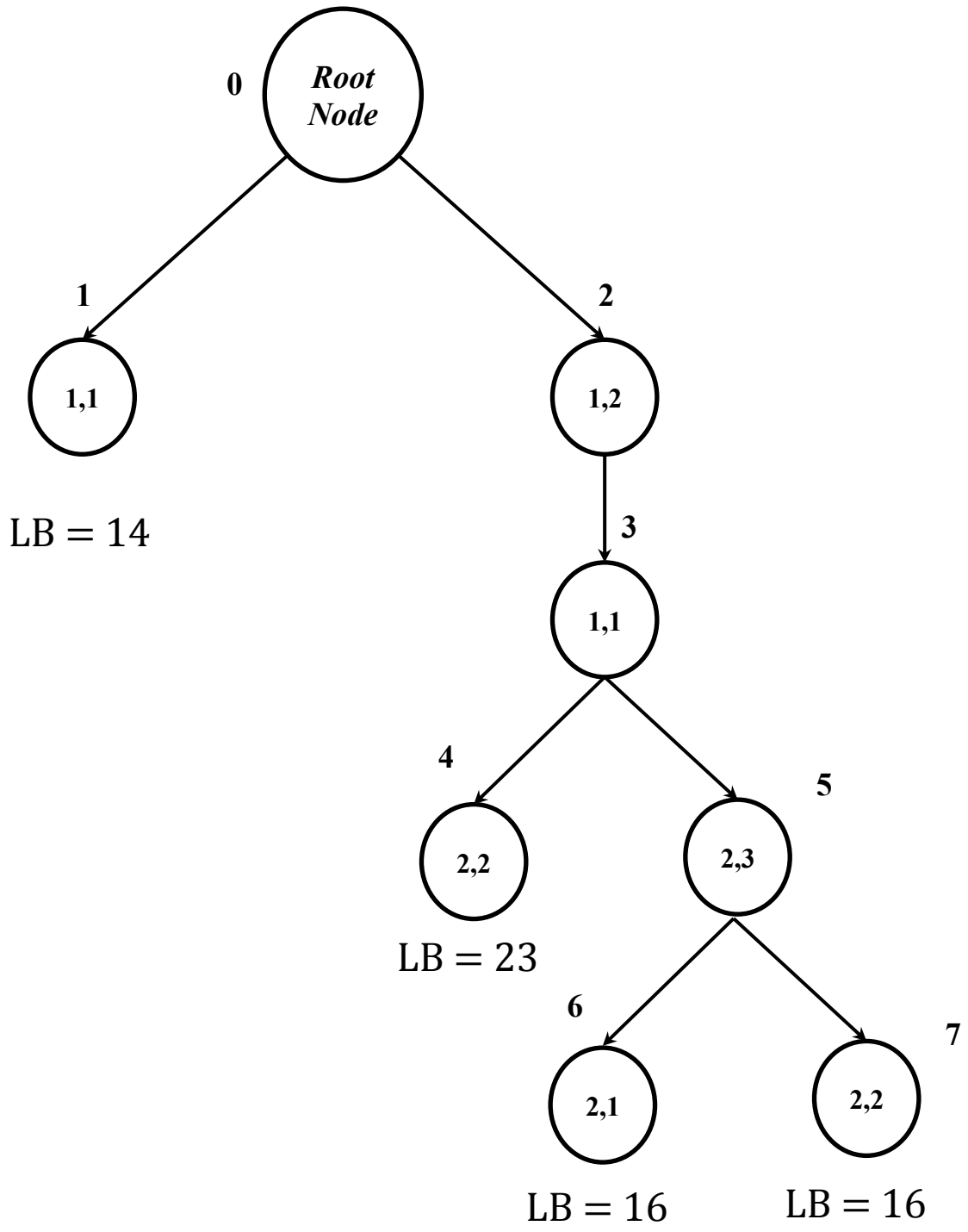
**t(Ω)** = min( 8+5 , 8+3 , 8+6 ) =11

***i**** = 2

$\Longrightarrow$ **Ω'** = { (2,1) , (2,2) }

Branch to two new nodes (2,1) and (2,2) which both have LB=16.

The new branch-and-bound tree is now:




Select *Node 1* for branching since it has the best bound with Lower Bound (LB) = 14.

<u>*Node 1:*</u>

**Ω** = { (2,1) , (1,2) , (2,3) }

**t(Ω)** = min( 2+5 , 2+6 , 0+8 ) = 7

***i**** = 2

$\Longrightarrow$ **Ω'** = { ( 2,1 ) , (2,3) }

Branch to two new nodes for (2,1) and (2,3) (nodes 8 and 9), which have lower bounds of LB=21 and LB=14, respectively. Select node 9 since it has the best bound of 14.

*Node 9*

**Ω** = { (2,1) , (1,2) , (1,3) }

**t(Ω)** = min( 8+5 , 2+6, 8+6 ) = 8

***i**** =12

$\Longrightarrow$ **Ω'** = { (1,2) }

Branch to a new node for (1,2) (node 10) with LB=14. Select node 10 since it has the best bound of 14.

*Node 10:*

**Ω** = { (2,1), (2,2), (1,3) }

**t(Ω)** = min( 8+5 , 8+3, 8+6 ) = 11

***i**** = 2

$\Longrightarrow$ **Ω'** = { (2,1) , (2,2) }

Branch to two new nodes for (2,1), (2,2) (nodes 11 and 12) each with LB=16. Randomly select node 7 of the 3 pending nodes with LB=16.

*Node 7:*

**Ω** = { (2,1), (1,3) }

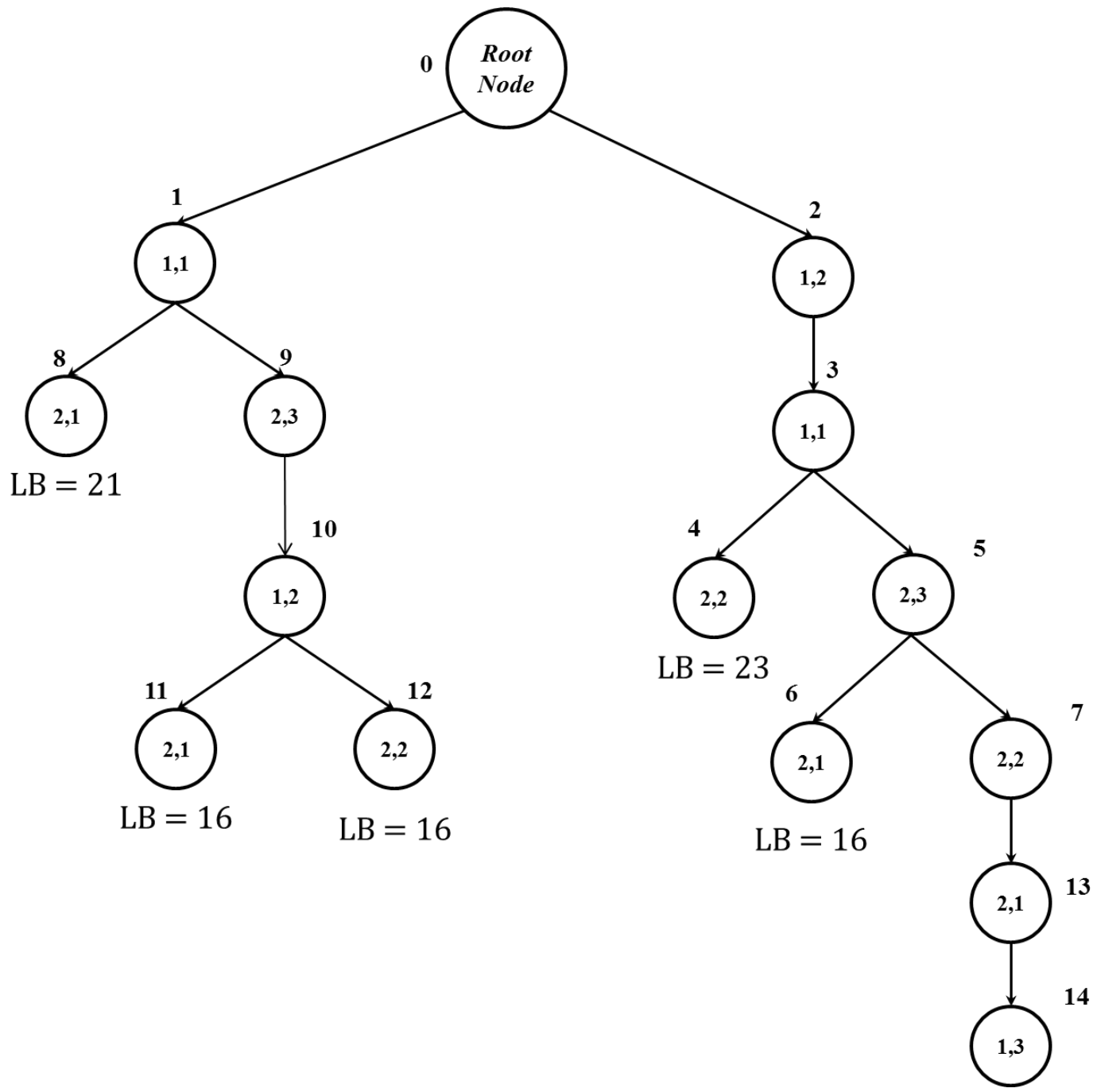
**t(Ω)** = min( 11+5 , 2+6 , 0+8 ) = 7

***i**** = 1

$\Longrightarrow$ **Ω'** = { (2,1) }

This creates a new node for (2,1) (node 13) with LB=16. Randomly selecting node 13 of the three pending nodes with LB = 16. This leads to a feasible solution at Node 14 with $C_{max}$ = 16.

The complete branch-and-bound tree is now:




Feasible candidate solution

At Node 14 with $C_{max}$ = 16

Once *Node 14* is reached all remaining pending nodes can be **<u>fathomed</u>** since **LB ≥ 16** for all pending nodes.

*Optimal Solution*

| *Machine* | *Sequence* |
|---|---|
| 1 | 2 , 1 , 3 |
| 2 | 3 , 2 , 1 |

$\boldsymbol{C_{max}}$ **= 16**

***Example:*** *Branch-and-Bound* using improved bound based on $1|r_j|L_{max}$

| *Job* | *Sequence* | *Processing Time* |
|---|---|---|
| 1 | 1,2 | $p_{11}=2$ , $p_{21}=5$ |
| 2 | 1,2 | $p_{12}=6$ , $p_{22}=3$ |
| 3 | 2,1 | $P_{23}=8$ , $p_{13}=6$ |

For this example it will be helpful to refer back to the Job Shop example from Lecture 16 to use the directed networks for each of the nodes to be evaluated (note that the node numbering differs).

Start by evaluating the root node of the tree which results in branching to two new nodes.

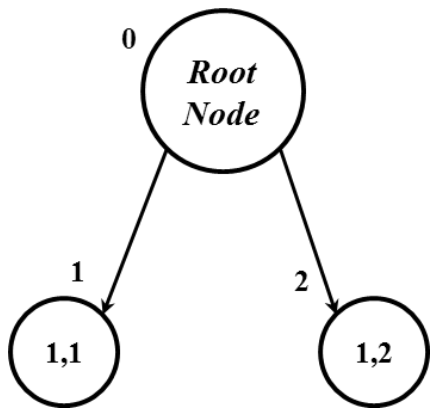
*Node 0:*

**Ω** = { (1,1) , (1,2) , (2,3) }

**t(Ω)** = min( 0+2 , 0+6 , 0+8 ) = 2

***i**** = 1

$\Longrightarrow$ **Ω'** = { (1,1) , (1,2) }




Now compute lower bounds for *Nodes 1 and 2*. For each node solve the corresponding instances of $1|r_j|L_{max}$ for each of two machines. Thus, we will solve 4 instances of $1|r_j|L_{max}$ in total.

*Node 1:* The initial lower bound is **14**.

For *Machine 1*:

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 2 | 6 | 6 |
| $r_j$ | 0 | 2 | 8 |
| $d_j$ | 5 | 11 | 14 |

$\Longrightarrow$ 1→2→3 achieves $L_{max}=0$

For *Machine 2*:

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 5 | 3 | 8 |
| $r_j$ | 2 | 8 | 0 |
| $d_j$ | 14 | 14 | 8 |

$\Longrightarrow$ 3→2→1 achieves $L_{max}=2$

Therefore the lower bound is **LB = 14+2 = 16**.

*Node 2:* The initial lower bound is **14**.

For *Machine 1*:

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 2 | 6 | 6 |
| $r_j$ | 6 | 0 | 8 |
| $d_j$ | 9 | 7 | 14 |

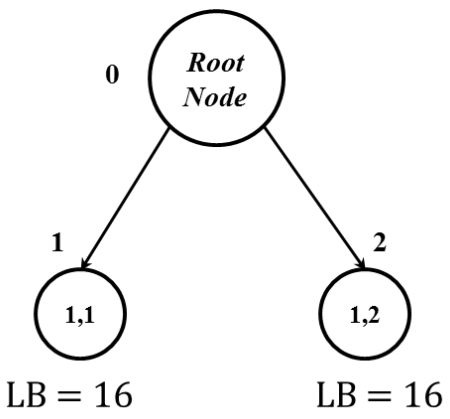
$\Longrightarrow$ 2→1→3 achieves $L_{max}=0$

For *Machine 2:*

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 5 | 3 | 8 |
| $r_j$ | 8 | 6 | 0 |
| $d_j$ | 14 | 14 | 8 |

$\Longrightarrow$ 3→2→1 achieves $L_{max}=2$

Therefore the lower bound is **LB = 14+2 = 16** and the branch-and-bound tree is:




Randomly select *Node 2* since both pending nodes have **LB = 16**

*Node 2:*

**Ω** = { (1,1) , (2,2) , (2,3) }

**t(Ω)** = min( 6+2 , 6+3 , 0+8 ) = 8

***i**** = 1 (select (1,1) randomly to branch since there is a tie)

$\Longrightarrow$ **Ω'** = { (1,1) }

*Node 3:* The initial lower bound is **14**.

For *Machine 1:*

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 2 | 6 | 6 |
| $r_j$ | 6 | 0 | 8 |
| $d_j$ | 8 | 6 | 14 |

$\Longrightarrow$ 2→1→3 achieves $L_{max}=0$

For *Machine 2:*

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 5 | 3 | 8 |
| $r_j$ | 8 | 6 | 0 |
| $d_j$ | 14 | 14 | 8 |

$\Longrightarrow$ 3→2→1 achieves $L_{max}$=2

Therefore the lower bound is **LB = 14+2 = 16**.

Randomly select *Node 3* since both pending nodes have **LB = 16**

*Node 3:*

**Ω** = { (2,1) , (2,2) , (2,3) }

**t(Ω)** = min( 8+5 , 6+3 , 0+8 ) = 8

***i**** = 2

$\Longrightarrow$ **Ω'** = { (2,2) , (2,3) }

*Node 4:* The initial lower bound is **23**.

For *Machine 1:*

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 2 | 6 | 6 |
| $r_j$ | 6 | 0 | 17 |
| $d_j$ | 17 | 6 | 23 |

$\Longrightarrow$ 2→1→3 achieves $L_{max}$=0

For *Machine 2:*

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 5 | 3 | 8 |
| $r_j$ | 9 | 6 | 9 |
| $d_j$ | 23 | 9 | 17 |

$\Longrightarrow$ 2→3→1 achieves $L_{max}$=0

Therefore the lower bound is **LB = 23 = 23**.

*Node 5:* The initial lower bound is **14**.

For *Machine 1:*

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 2 | 6 | 6 |
| $r_j$ | 6 | 0 | 8 |
| $d_j$ | 8 | 6 | 14 |

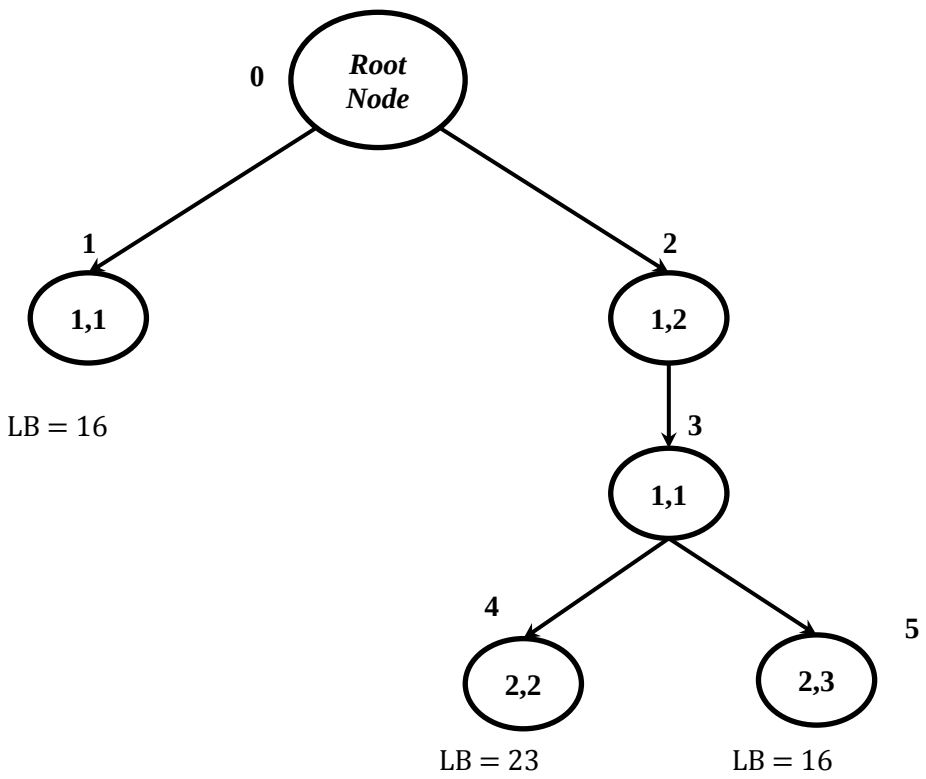
$\Longrightarrow$ 2→1→3 achieves $L_{max}$=0

For *Machine 2:*

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 5 | 3 | 8 |
| $r_j$ | 8 | 8 | 0 |
| $d_j$ | 14 | 14 | 8 |

$\Longrightarrow$ 3→1→2 achieves $L_{max}$=2

Therefore the lower bound is **LB = 14+2 = 16**.




Randomly select *Node 5* of the two pending nodes with **LB = 16.**

*Node 5:*

**Ω** = { (2,1) , (2,2) , (1,3) }

**t(Ω)** = min( 8+5 , 8+3 , 8+6 ) = 11

***i**** = 2

$\Longrightarrow$ **Ω'** = { (2,1) , (2,2) }

Now compute the lower bounds for the two new nodes branching from Node 5.

*Node 6:* The initial lower bound is **16**.

For *Machine 1*:

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 2 | 6 | 6 |
| $r_j$ | 6 | 0 | 8 |
| $d_j$ | 8 | 6 | 16 |

$\Longrightarrow$ 2→1→3 achieves $L_{max}=0$

For *Machine 2*:

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 5 | 3 | 8 |
| $r_j$ | 8 | 13 | 0 |
| $d_j$ | 13 | 16 | 8 |

$\Longrightarrow$ 3→1→2 achieves $L_{max}=0$

Therefore the lower bound is **LB = 16+0 = 16**.

*Node 7:* The initial lower bound is **16**.

For *Machine 1*:

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 2 | 6 | 6 |
| $r_j$ | 6 | 0 | 8 |
| $d_j$ | 10 | 8 | 16 |

$\Longrightarrow$ 2→1→3 achieves $L_{max}=0$

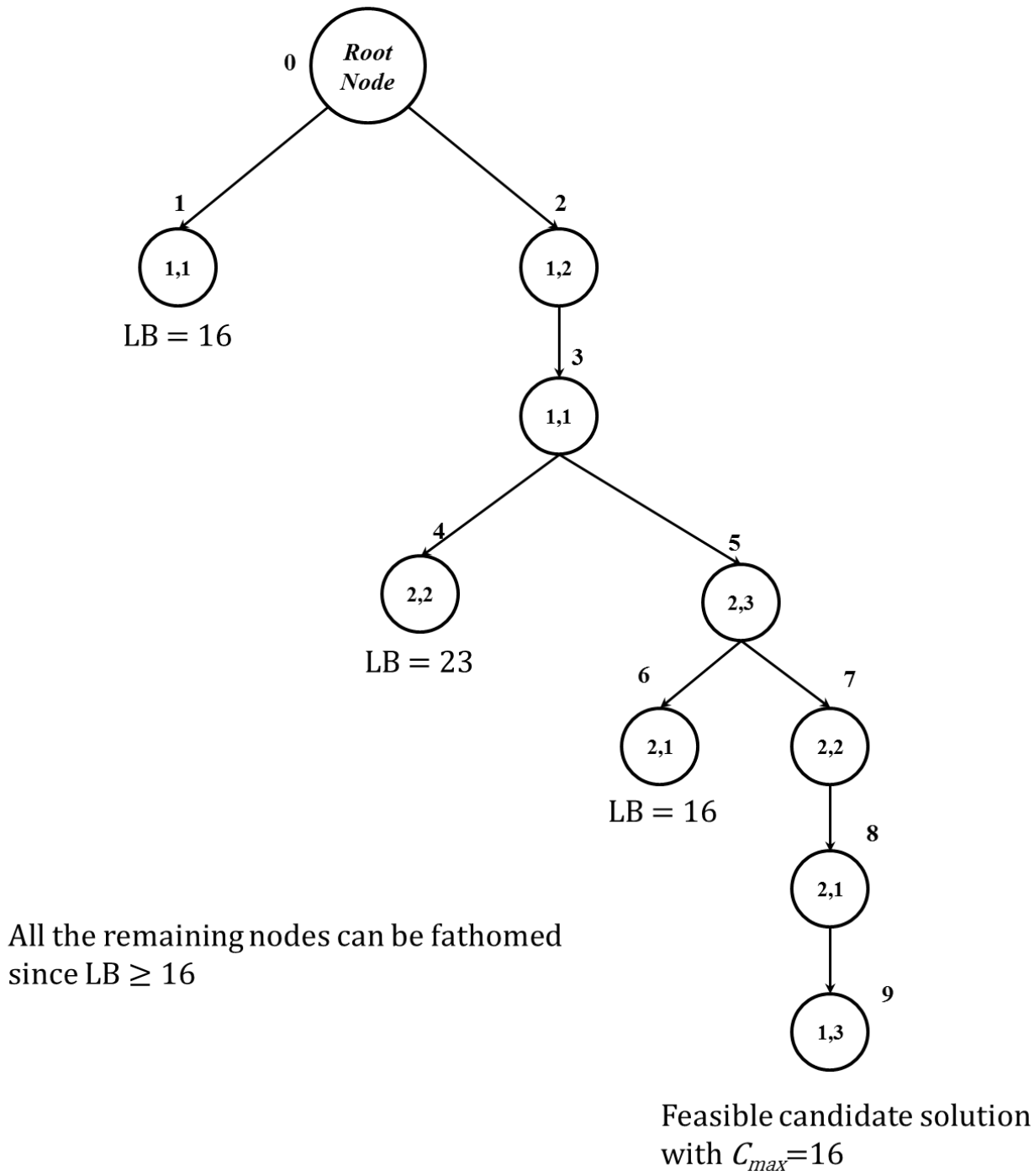
For *Machine 2*:

| Jobs | 1 | 2 | 3 |
|---|---|---|---|
| $p_j$ | 5 | 3 | 8 |
| $r_j$ | 11 | 8 | 0 |
| $d_j$ | 16 | 11 | 8 |

$\Longrightarrow$ 3→2→1 achieves $L_{max}=0$

Therefore the lower bound is **LB = 16+0 = 16**.

Randomly selecting *Node 7* of the pending nodes with **LB = 16** results in the following final *Branch-and-Bound* tree.




Note that we avoided branching on Node 1 as a result of the tighter bounds generated using $1||r_j|L_{max}$ in this example.

*Optimal Solution :*

| ***Machine*** | ***Sequence*** |
|---|---|
| 1 | 2,1,3 |
| 2 | 3,2,1 |

$C_{max} = 16$