IO543: Scheduling

Lecture 16

Brian Denton
Associate Professor, IOE

IOE Bld 2893
btdenton@umich.edu

Assignments

- Assignment 5 due Thursday
- Assignment 6 is up on Ctools; Due March 31

Job Shop Methods Summary

Methods for solving $Jm \mid \mid C_{max}$

- For $J2 \mid \mid C_{max}$ decompose into two instances of $\mathrm{F2} \mid \mid C_{max}$ and use Johnson's rule
- For $Jm \mid \mid C_{max}$ use branch-and-bound
  - Total enumeration ✓
  - Simple bound on C_{max} ✓
  - <u>Improved bound</u> on C_{max} using $1 \mid r_j \mid L_{max}$
- **Shifting Bottleneck Heuristic** for $Jm \mid \mid C_{max}$

Improved Branch-and-Bound

A **stronger lower bound** can be obtained for each node using $1 \mid r_j \mid L_{max}$

- For each machine, for a given partial schedule, find the optimal schedule to the instance of $1 \mid r_j \mid L_{max}$ with:
  - **Release times** determined by: the longest path from node *S* to node (i, j)
  - **Due dates** determined by: $(\mathrm{C_{max}}$ lower bound) – (longest path from node (i, j) to node *E)* + (operation (i, j) processing time)
- New lower bound = $\mathrm{C_{max}} + \max\{\mathrm{L_{max}}(\mathrm{i})\}$

Example

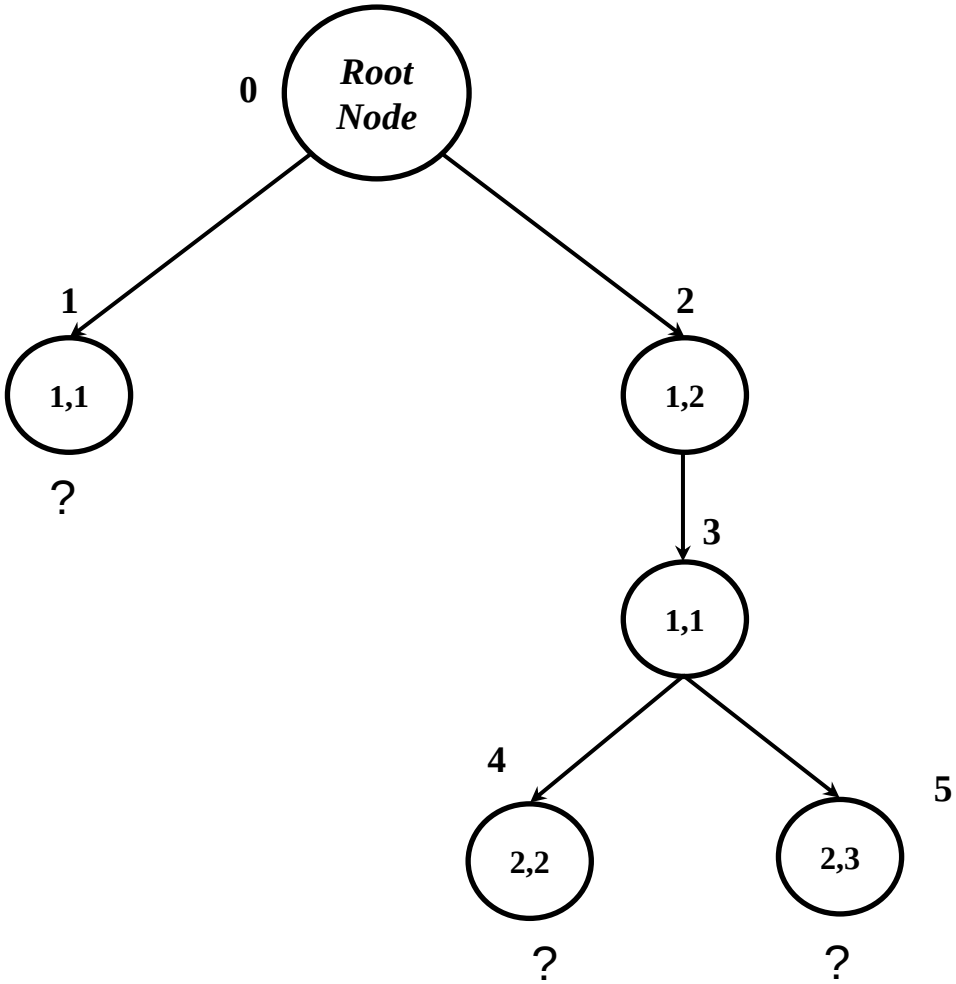
Example: Imrpoved Branch-and-Bound for $J_2 \| C_{max}$

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	$p_{23} = 8$, $p_{13} = 6$

Improved Branch-and-Bound

Example:

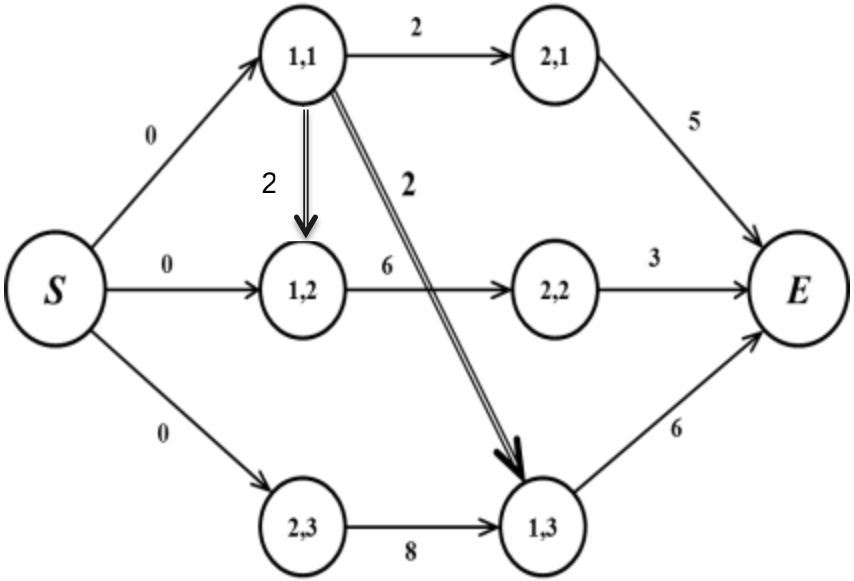
Assume we are given the following branch-and-bound tree. Compute lower bounds for nodes 1, 4, and 5, using $1 \mid r_j \mid L_{max}$



Improved Branch-and-Bound

Example:

For node 1 initial lower bound is 14. First solve the following instance of $1\,|r_j\,|L_{max}$ for machine 1



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

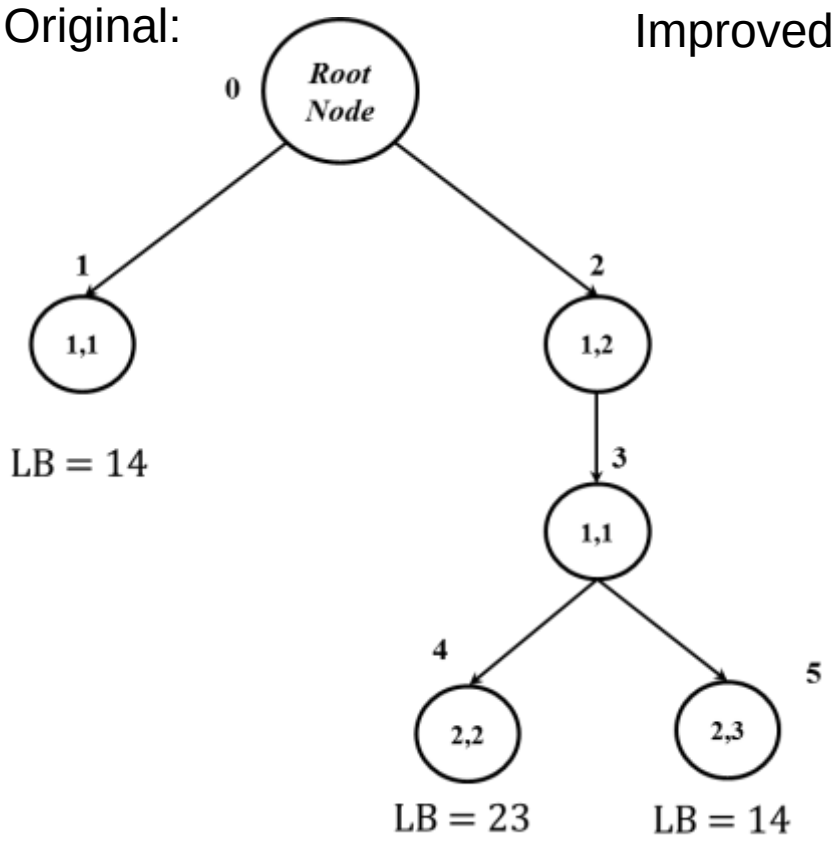
Sequence 1→ 2→ 3 achieves Lmax = 0

Next, solve the instance of $1\,|r_j\,|L_{max}$ for machine 2 ($L_{max} = 2$). The new lower bound is LB = 14 + 2 = 16

Improved Branch-and-Bound

Continuing to evaluate the lower bounds for nodes 4 and 5 results in a new branch-and-bound tree with tighter bounds:

Original:

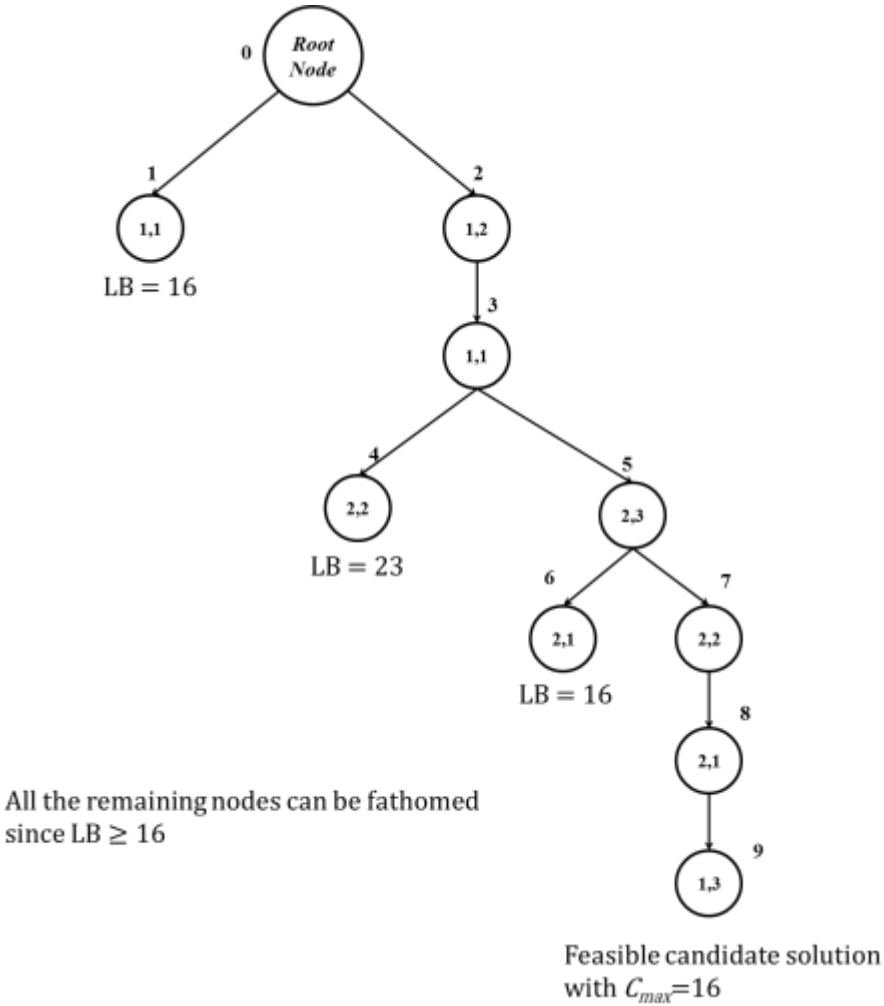


Improved:

?

Improved Branch-and-Bound

Final branch-and bound tree using the improved bound:



Optimal Solution:

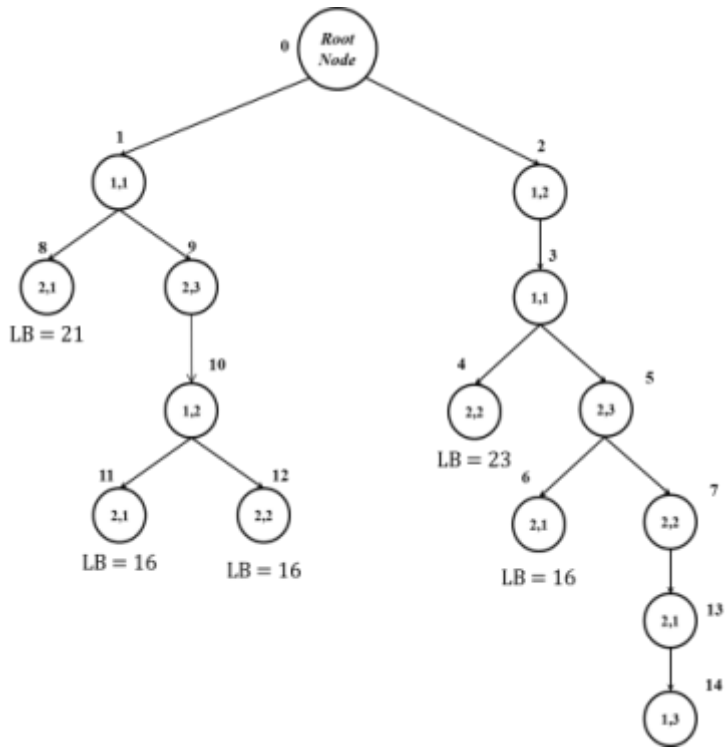
Machine	*Sequence*
1	2 , 1 , 3
2	3 , 2 , 1

$C_{max} = 16$

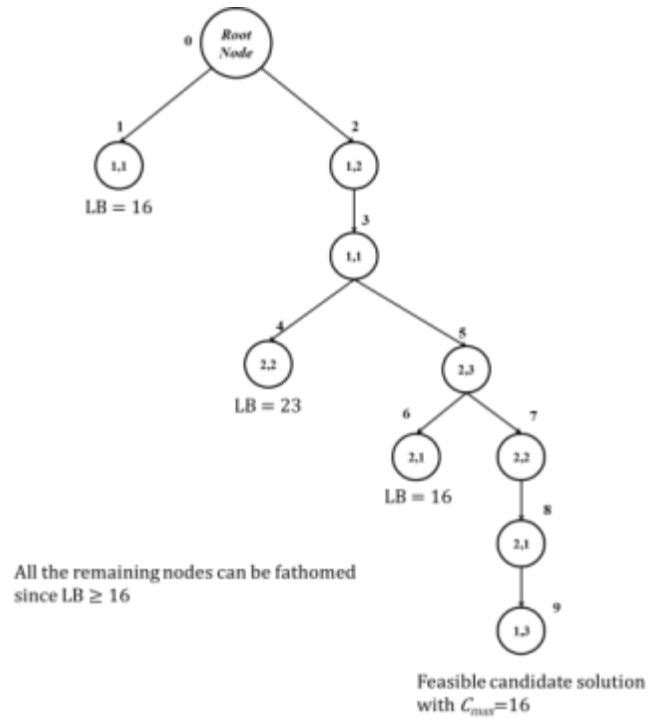
Original Branch-and-Bound

Compare to the original branch-and-bound tree using the weaker bound that does not use $1 \mid r_j \mid L_{max}$

Original:



New:



Shifting Bottleneck Heuristic

- Branch-and-bound is effective for small instances of $Jm||C_{max}$
- Heuristics are required for large instances of $Jm||C_{max}$
- The **Shifting Bottleneck Heuristic** is the most well known heuristic for $Jm||C_{max}$

Shifting Bottleneck Heuristic

Basic idea:

- Select machines one at a time to sequence jobs
- At each iteration select a machine that is **likely to be the bottleneck**
- Sequence jobs on the selected machine by solving $1 \mid r_j \mid L_{max}$

Shifting Bottleneck Heuristic

Shifting Bottleneck Heuristic:

Step 1: Let M be the set of all machines and M_0 be the set of <u>selected</u> machines. Set $\mathrm{M}_0 = \{\emptyset\}$, and set $\mathrm{C}_{\max}(\mathrm{M}_0)$ equal to longest path in the network with no disjunctive arcs.

Step 2: For each machine in $\mathrm{M} - \mathrm{M}_0$ generate an instance of $1 \left| r_j \right| L_{max}$. Let $\mathrm{L}_{\max}(\mathrm{i})$ be L_{max} for machine *i.*

Step 3: Let

$$\mathrm{L}_{\max}(\mathrm{k}) = \max_{\{\mathrm{i} \in M - M_0\}} \{\mathrm{L}_{\max}(\mathrm{i})\}$$

Sequence machine *k* according to the sequence obtained in Step 2. Add machine k to M_0. Add disjunctive arcs for machine k based on Step 2.

Step 4: For each machine $i \in \{M_0 - k\}$ solve the new $1 \left| r_j \right| L_{max}$

Step 5: If $\mathrm{M}_0 = M$ then STOP; otherwise go to Step 2.

Example

Example: Shifting Bottleneck Heuristic 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	$p_{23} = 8$, $p_{13} = 6$

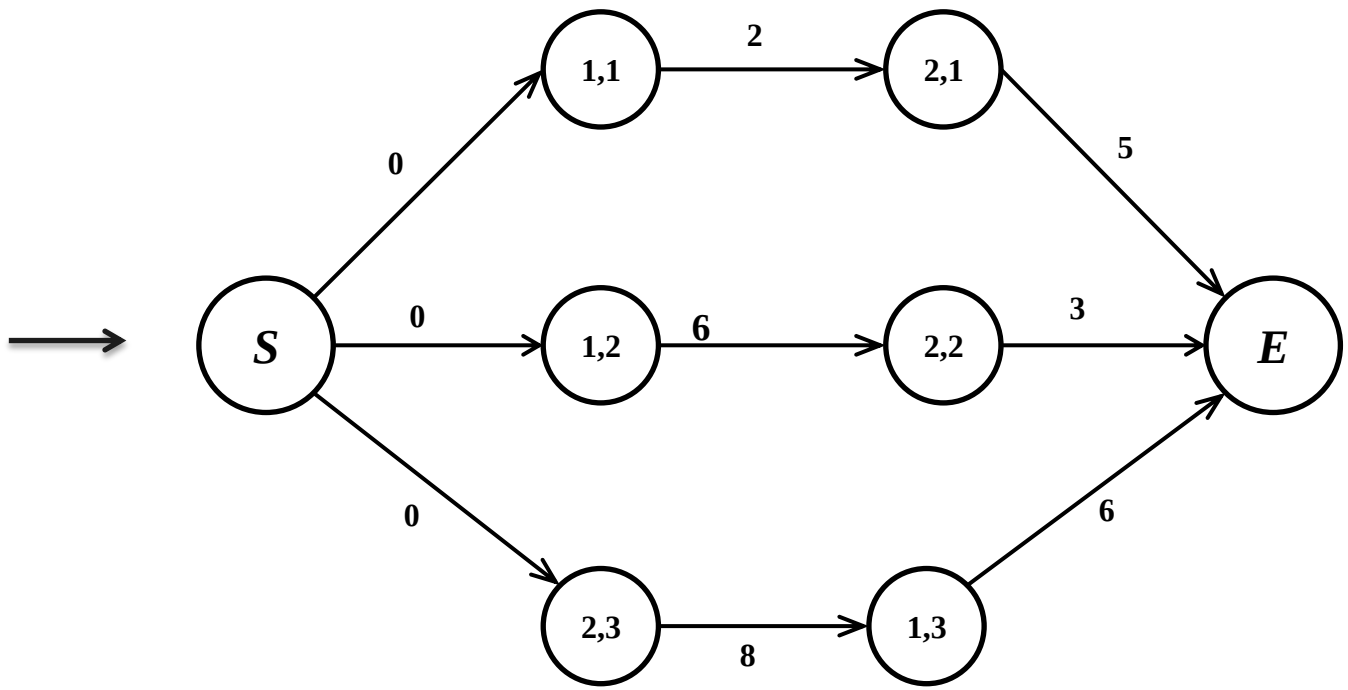
There are two machines. Therefore there will be **two iterations** of the heuristic.

Example

Iteration1 :

Step 1: Step 1: $M_0 = \{\emptyset\}, \;\; C_{max}(M_0) = 14$

Setup initial instances of $1 \,|r_j\,| L_{max}$ for step 2 using network with conjunctive arcs



Example

Iteration 1:

Step 1: $M_0 = \{\emptyset\}, \quad C_{max}(M_0) = 14$

Step 2: Solve an instance of $1 \,|r_j\,| L_{max}$ for machines 1 and 2

Machine 1:

Jobs	1	2	3
p_j	2	6	6
r_j	0	0	8
d_j	9	11	14

$\longrightarrow$ 1→ 2→ 3, $L_{max} = 0$

Machine 2:

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

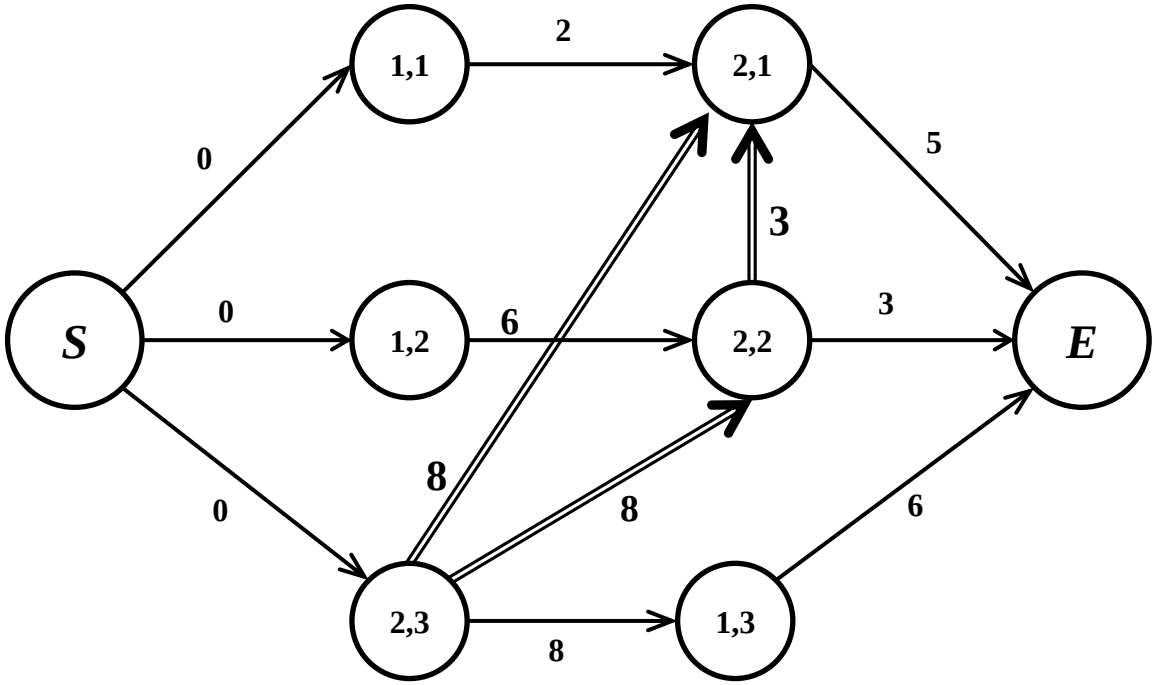
$\longrightarrow$ 3→2→ 1, $L_{max} = 2$

Example

Iteration 1 (continued):

Step 3: Select machine 2 since $L_{max} = 2$ is higher of the two. Sequence jobs in the order 3→ 2→ 1. Set k = 2 and $M_0 = \{2\}$.

This leads to the following directed network: →



Example

Iteration 1 (continued):

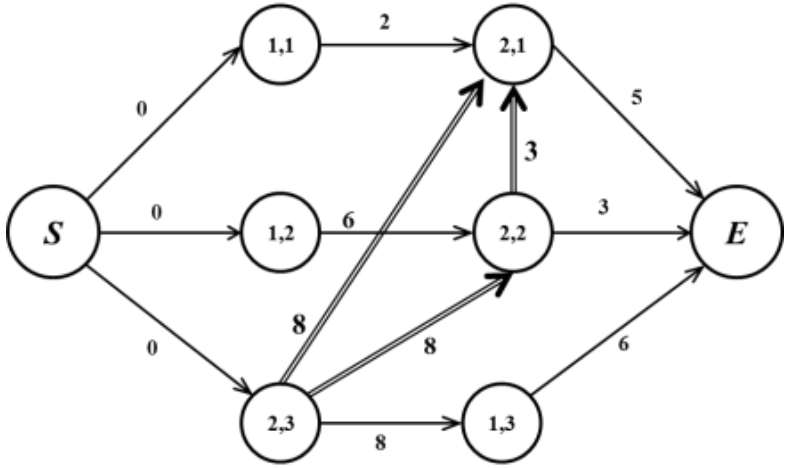
Step 4: The set of selected machines $\{M_0 - k\}$ is empty, so no resequencing is required.

Step 5: Since $\mathrm{M}_0 \neq M$ therefore go to Step 2.

Iteration 1 complete

Example

Iteration 2:



Step 2: Solve the following instance of $1\,|r_j\,|L_{max}$ for machine 1 based on the above network:

Jobs	1	2	3
p_j	2	6	6
r_j	0	0	8
d_j	11	8	16

$\longrightarrow$ 2→ 1→ 3, $L_{max} = 0$

Example

Iteration 2 (continued):

Step 3: Sequence jobs on machine 1 in the order 2→ 1→ 3. Set k = 1 and $M_0 = \{1,2\}$.

Step 4: The set of selected machines $\{M_0 - k\} = \{2\}$, so **<u>resequence</u>** machine 2. Sequence 3→ 2→ 1 is still optimal.

Step 5: Since $\mathrm{M}_0 = M$ therefore STOP.

Example

Solution:

Machine	*Sequence*
1	2 , 1 , 3
2	3 , 2 , 1

$$C_{max} = 16$$

Note this is the same solution as obtained using (a) decomposition and Johnson' rule and (b) branch-and-bound.

Since this is a heuristic it is not guaranteed to find the optimal solution, but often it does.

Delayed Precedence Constraints

Sometimes the shifting bottleneck heuristic can result in <u>infeasible solutions</u>. Additional (delayed precedence) constraints may be necessary to avoid subtours at each iteration.

Consider the following example:

Job	*Sequence*	*Processing Time*
1	1,2	$p_{11} = 1$, $p_{21} = 1$
2	2,1	$p_{22} = 1$, $p_{12} = 1$
3	3	$p_{33} = 4$
4	3	$p_{34} = 4$