Open Shop models

Algorithm Problem $O2|C_{\text{max}}$

1. $I = \text{set of jobs with } p_{1j} \leq p_{2j}; \quad J = \text{set of remaining jobs};$
2. IF $p_{1r} = \max \{ \max_{j \in I} p_{1j}, \max_{j \in J} p_{2j} \}$ then
   - order on $M_1$: $(I \setminus \{r\}, J, r);$ order on $M_2$: $(r, I \setminus \{r\}, J)$
   - $r$ first on $M_2$, than on $M_1$; all other jobs vice versa

\[
\begin{array}{|c|c|c|}
\hline
&M_1 & M_2 \\
I \setminus \{r\} & r & r \\
J & I \setminus \{r\} & J \\
r & & \\
\hline
\end{array}
\]
Open Shop models

Algorithm Problem $O2||C_{max}$

1. $I$ = set of jobs with $p_{1j} \leq p_{2j}$; $J$ = set of remaining jobs;
2. IF $p_{1r} = \max\{\max_{j \in I} p_{1j}, \max_{j \in J} p_{2j}\}$ then
   - order on $M_1$: $(I \setminus \{r\}, J, r)$; order on $M_2$: $(r, I \setminus \{r\}, J)$
   - $r$ first on $M_2$, than on $M_1$; all other jobs vice versa
3. ELSE IF $p_{2r} = \max\{\max_{j \in I} p_{1j}, \max_{j \in J} p_{2j}\}$ then
   - order on $M_1$: $(r, J \setminus \{r\}, I)$; order on $M_2$: $(J \setminus \{r\}, I, r)$
   - $r$ first on $M_1$, than on $M_2$; all other jobs vice versa
Open Shop models

Remarks Algorithm Problem $O2\|C_{max}$

- complexity: $O(n)$
- algorithm solves problem $O2\|C_{max}$ optimally
- Proof builds on fact that $C_{max}$ is either
  
  $- \sum_{j=1}^{n} p_{1j}$ or
  
  $- \sum_{j=1}^{n} p_{2j}$ or
  
  $- p_{1r} + p_{2r}$
Open Shop models

Remarks Algorithm Problem $O2||C_{max}$

- complexity: $O(n)$
- algorithm solves problem $O2||C_{max}$ optimally
- Proof builds on fact that $C_{max}$ is either
  \[- \sum_{j=1}^{n} p_{1j} \text{ or} \]
  \[- \sum_{j=1}^{n} p_{2j} \text{ or} \]
  \[- p_{1r} + p_{2r} \]

Problem $O3||C_{max}$

- Problem $O3||C_{max}$ is NP-hard
  Proof as Exercise (Reduction using PARTITION)
Open Shop models

Problem $O|pmtn|C_{\text{max}}$

- define $ML_i := \sum_{j=1}^{n} p_{ij}$ (load of machine $i$)
- define $JL_j := \sum_{i=1}^{m} p_{ij}$ (load of job $j$)
- $LB := \max\{\max_{i=1}^{m} ML_i, \max_{j=1}^{n} JL_j\}$ is a lower bound on $C_{\text{max}}$
Open Shop models

Problem $O|pmtn|C_{max}$

- define $ML_i := \sum_{j=1}^{n} p_{ij}$ (load of machine $i$)
- define $JL_j := \sum_{i=1}^{m} p_{ij}$ (load of job $j$)
- $LB := \max\{\max_{i=1}^{m} ML_i, \max_{j=1}^{n} JL_j\}$ is a lower bound on $C_{max}$
- **Theorem**: For problem $O|pmtn|C_{max}$ a schedule with $C_{max} = LB$ exists.
- Proof of the theorem is constructive and leads to a polynomial algorithm for problem $O|pmtn|C_{max}$
Open Shop models

Notations for Algorithm $O|pmtn|C_{max}$

- job $j$ (machine $i$) is called tight if $JL_j = LB$ ($ML_i = LB$)
- job $j$ (machine $i$) has slack if $JL_j < LB$ ($ML_i < LB$)
- a set $D$ of operations is called a decrementing set if it contain for each tight job and machine exactly one operation and for each job and machine with slack at most one operation
- **Theorem:** A decrementing set always exists and can be calculated in polynomial time
  (Proof based on maximal cardinality matchings; see e.g. P. Brucker: Scheduling Algorithms)
Open Shop models

Algorithm $O|pmtn|C_{\text{max}}$

REPEAT

1. Calculate a decrementing set $D$;
2. Calculate maximum value $\Delta$ with
   
   - $\Delta \leq \min_{(i,j) \in D} pij$
   - $\Delta \leq LB - ML_i$ if machine $i$ has slack and no operation in $D$
   - $\Delta \leq LB - JL_j$ if job $j$ has slack and no operation in $D$;
3. schedule the operations in $D$ for $\Delta$ time units in parallel;
4. Update values $p$, $LB$, $JL$, and $ML$

UNTIL all operations have been completely scheduled.
Open Shop models

Correctness Algorithm $O|\text{pmtn}|C_{max}$

- after an iteration we have: $LB_{new} = LB_{old} - \Delta$
- in each iteration a time slice of $\Delta$ time units is scheduled
- the algorithm terminates after at most $nm(n+m)$ iterations since in each iteration either
  - an operation gets completely scheduled or
  - one additional machine or job gets tight
### Open Shop models

**Example Algorithm** $O|pmtn|C_{\text{max}}$

<table>
<thead>
<tr>
<th>$p$</th>
<th>$ML$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 4 3 2</td>
<td>11</td>
</tr>
<tr>
<td>3 1 2 3</td>
<td>9</td>
</tr>
<tr>
<td>2 3 3 2</td>
<td>10</td>
</tr>
</tbody>
</table>

$LJ \begin{array}{|c|c|}
\hline
7 & 8 & 8 & 7 \\
LB = 11 \\
\hline
\end{array}$
## Open Shop models

**Example Algorithm** $O|pmtn|C_{max}$

<table>
<thead>
<tr>
<th>$\Delta = 3$</th>
<th>$p$</th>
<th>$ML$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 4 3 2</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>3 1 2 3</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>2 3 3 2</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td><strong>JL</strong> 7 8 8 7</td>
<td><strong>LB</strong> = 11</td>
<td></td>
</tr>
</tbody>
</table>
## Open Shop models

### Example Algorithm $O|pmtn|C_{\text{max}}$

<table>
<thead>
<tr>
<th>$\Delta = 3$</th>
<th>$p$</th>
<th>$ML$</th>
<th>$M_3$</th>
<th>$M_2$</th>
<th>$M_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 4 3 2</td>
<td>11</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$p$</td>
<td>3 1 2 3</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$JL$</td>
<td>7 8 8 7</td>
<td>$LB = 11$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lecture 7: Scheduling
Open Shop models

Example Algorithm $O|pmtn|C_{max}$

<table>
<thead>
<tr>
<th>$\Delta = 3$</th>
<th>$p$</th>
<th>$ML$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 4 3 2</td>
<td>11</td>
</tr>
<tr>
<td>$p$</td>
<td>3 1 2 3</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>2 3 3 2</td>
<td>10</td>
</tr>
<tr>
<td>$JL$</td>
<td>7 8 8 7</td>
<td>$LB = 11$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$p$</th>
<th>$ML$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 4 0 2</td>
<td>8</td>
</tr>
<tr>
<td>0 1 2 3</td>
<td>6</td>
</tr>
<tr>
<td>2 0 3 2</td>
<td>7</td>
</tr>
<tr>
<td>$JL$</td>
<td>4 5 5 7</td>
</tr>
</tbody>
</table>
## Open Shop models

### Example Algorithm $O|pmtn|C_{\text{max}}$

<table>
<thead>
<tr>
<th>$\Delta = 3$</th>
<th>$p$</th>
<th>$ML$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2 4 3 2$</td>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$3 1 2 3$</td>
<td>9</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$2 3 3 2$</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>$JL$</td>
<td>7 8 8 7</td>
<td>$LB = 11$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\Delta = 1$</th>
<th>$p$</th>
<th>$ML$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2 4 0 2$</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>$0 1 2 3$</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$2 0 3 2$</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>$JL$</td>
<td>4 5 5 7</td>
<td>$LB = 8$</td>
<td>3 4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Open Shop models

Example Algorithm $O|pmtn|C_{max}$

<table>
<thead>
<tr>
<th>$\Delta = 1$</th>
<th>$p$</th>
<th>$ML$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>2 4 0 2</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>0 1 2 3</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2 0 3 2</td>
<td>7</td>
</tr>
<tr>
<td>$JL$</td>
<td>4 5 5 7</td>
<td>$LB = 8$</td>
</tr>
</tbody>
</table>

$\Delta = 3$

| $p$         | 2 3 0 2 | 7    |
|             | 0 1 2 3 | 6    |
|             | 2 0 3 2 | 7    |
| $JL$        | 4 4 5 7 | $LB = 7$ |

$M_1$

$M_2$

$M_3$
### Open Shop models

**Example Algorithm** \( O|pmtn|C_{\text{max}} \)

<table>
<thead>
<tr>
<th>( \Delta = 3 )</th>
<th>( p )</th>
<th>( ML )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 3 0 2</td>
<td>7</td>
</tr>
<tr>
<td>( p )</td>
<td>0 1 2 3</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2 0 3 2</td>
<td>7</td>
</tr>
<tr>
<td>( JL )</td>
<td>4 4 5 7</td>
<td>( LB = 7 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \Delta = 2 )</th>
<th>( p )</th>
<th>( ML )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 0 0 2</td>
<td>4</td>
</tr>
<tr>
<td>( p )</td>
<td>0 1 2 0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2 0 0 2</td>
<td>4</td>
</tr>
<tr>
<td>( JL )</td>
<td>4 1 2 4</td>
<td>( LB = 4 )</td>
</tr>
</tbody>
</table>
### Open Shop models

#### Example Algorithm $O|pmtn|C_{\text{max}}$

<table>
<thead>
<tr>
<th>$\Delta = 2$</th>
<th>$p$</th>
<th>$ML$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>2 0 0 2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$JL$</td>
<td>4 1 2 4</td>
<td>$LB = 4$</td>
<td>3 4 7 9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\Delta = 1$</th>
<th>$p$</th>
<th>$ML$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>2 0 0 0</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>$JL$</td>
<td>2 1 0 2</td>
<td>$LB = 2$</td>
<td>3 4 7 9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $p$ represents the sequence of jobs.
- $ML$ represents the makespan.
- $JL$ and $LB$ are job lists and lower bounds, respectively.
Open Shop models

Example Algorithm $O|pmtn|C_{\text{max}}$

<table>
<thead>
<tr>
<th>$\Delta = 1$</th>
<th>$p$</th>
<th>$ML$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>2 0 0 0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0 1 0 0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 2</td>
<td>2</td>
</tr>
<tr>
<td>$JL$</td>
<td>2 1 0 2</td>
<td>$LB = 2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\Delta = 1$</th>
<th>$p$</th>
<th>$ML$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0 1 0 0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>$JL$</td>
<td>1 1 0 1</td>
<td>$LB = 1$</td>
</tr>
</tbody>
</table>

| $M_1$ | 3 2 2 4 1 |
| $M_2$ | 1 4 3 2   |
| $M_3$ | 2 3 1 4 4 |

| $JL$ | 3 4 7 9 10 |
| $ML$ | 3 4 7 9 10 |
| $LB$ | 3 4 7 9 10 |

18
Open Shop models

Final Schedule Example Algorithm $O|pmtn|C_{max}$

<table>
<thead>
<tr>
<th>$p$</th>
<th>$ML$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 4 3 2</td>
<td>11</td>
</tr>
<tr>
<td>3 1 2 3</td>
<td>9</td>
</tr>
<tr>
<td>2 3 3 2</td>
<td>10</td>
</tr>
</tbody>
</table>

$JL | 7 8 8 7 | LB = 11$

- 6 iterations
- $C_{max} = 11 = LB$
- sequence of time slices may be changed arbitrary
Job Shop models

Problem $J2||C_{max}$

- $I_1$: set of jobs only processed on $M_1$
- $I_2$: set of jobs only processed on $M_2$
- $I_{12}$: set of jobs processed first on $M_1$ and then on $M_2$
- $I_{21}$: set of jobs processed first on $M_2$ and then on $M_1$
- $\pi_{12}$: optimal flow shop sequence for jobs from $I_{12}$
- $\pi_{21}$: optimal flow shop sequence for jobs from $I_{21}$
Job Shop models

Algorithm Problem $J_2 || C_{max}$

1. on $M_1$ first schedule the jobs from $I_{12}$ in order $\pi_{12}$, than the jobs from $I_1$, and last the jobs from $I_{21}$ in order $\pi_{21}$

2. on $M_2$ first schedule the jobs from $I_{21}$ in order $\pi_{21}$, than the jobs from $I_2$, and last the jobs from $I_{12}$ in order $\pi_{12}$

<table>
<thead>
<tr>
<th>$M_2$</th>
<th>$I_{21}$</th>
<th>$I_2$</th>
<th>$I_{12}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>$I_{12}$</td>
<td>$I_1$</td>
<td>$I_{21}$</td>
</tr>
</tbody>
</table>
Job Shop models

Algorithm Problem $J2||C_{max}$

1. on $M_1$ first schedule the jobs from $I_{12}$ in order $\pi_{12}$, then the jobs from $I_1$, and last the jobs from $I_{21}$ in order $\pi_{21}$
2. on $M_2$ first schedule the jobs from $I_{21}$ in order $\pi_{21}$, than the jobs from $I_2$, and last the jobs from $I_{12}$ in order $\pi_{12}$

<table>
<thead>
<tr>
<th>$M_2$</th>
<th>$I_{12}$</th>
<th>$I_{21}$</th>
<th>$I_2$</th>
<th>$I_{12}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>$I_{12}$</td>
<td>$I_1$</td>
<td>$I_{21}$</td>
<td></td>
</tr>
</tbody>
</table>

Theorem: The above algorithm solves problem $J2||C_{max}$ optimally in $O(n \log(n))$ time.
Proof: almost straightforward!
Job Shop models

Problem $J||C_{max}$

- as a generalization of $F||C_{max}$, this problem is NP-hard
- it is one of the most treated scheduling problems in literature
- we presented
  - a branch and bound approach
  - a heuristic approach called the Shifing Bottleneck Heuristic
for problem $J||C_{max}$ which both depend on the disjunctive graph formulation
Job Shop models

Base of Branch and Bound

- The set of all active schedules contains an optimal schedule
- Solution method: Generate all active schedules and take the best
- Improvement: Use the generation scheme in a branch and bound setting
- Consequence: We need a generation scheme to produce all active schedules for a job shop
- → Approach: extend partial schedules
Job Shop models

Generation of all active schedules

- Notations: (assuming that already a partial schedule $S$ is given)
  - $\Omega$: set of all operations which predecessors have already been scheduled in $S$
  - $r_{ij}$: earliest possible starting time of operation $(i,j) \in \Omega$ w.r.t. $S$
  - $\Omega'$: subset of $\Omega$

- Remark: $r_{ij}$ can be calculated via longest path calculations in the disjunctive graph belonging to $S$
Job Shop models

Generation of all active schedules (cont.)

1. (Initial Conditions)
   \[ \Omega := \{ \text{first operations of each job} \}; \ r_{ij} := 0 \text{ for all } (i, j) \in \Omega; \]

2. (Machine selection)
   Compute for current partial schedule \[ t(\Omega) := \min_{(i, j) \in \Omega} \{ r_{ij} + p_{ij} \}; \]
   \[ i^* := \text{machine on which minimum is achieved}; \]

3. (Branching) \[ \Omega' := \{ (i^*, j) | r_{i^*, j} < t(\Omega) \} \]
   FOR ALL \((i^*, j) \in \Omega'\) DO
   (a) extend partial schedule by scheduling \((i^*, j)\) next on machine \(i^*\);
   (b) delete \((i^*, j)\) from \(\Omega\);
   (c) add job-successor of \((i^*, j)\) to \(\Omega\);
   (d) Return to Step 2
Job Shop models

Generation of all active schedules - example

Jobs: 1  (3, 1) → (2, 1) → (1, 1)  \( p_{31} = 4, p_{21} = 2, p_{11} = 1 \)

2  (1, 2) → (3, 2)  \( p_{12} = 3, p_{32} = 3 \)

3  (2, 3) → (1, 3) → (3, 3)  \( p_{23} = 2, p_{13} = 4, p_{33} = 1 \)

Partial Schedule:

\[ \begin{array}{c}
M_1 \\
M_2 \\
M_3 \\
4 \\
6
\end{array} \]
Job Shop models

Generation of all active schedules - example

Jobs: 1  \((3, 1) \rightarrow (2, 1) \rightarrow (1, 1)\)  \(p_{31} = 4, p_{21} = 2, p_{11} = 1\)

2  \((1, 2) \rightarrow (3, 2)\)  \(p_{12} = 3, p_{32} = 3\)

3  \((2, 3) \rightarrow (1, 3) \rightarrow (3, 3)\)  \(p_{23} = 2, p_{13} = 4, p_{33} = 1\)

Partial Schedule:

\[\Omega = \{(1, 1), (3, 2), (1, 3)\};\]
\[r_{11} = 6, r_{32} = 4, r_{13} = 3;\]
\[t(\Omega) = \min\{6 + 1, 4 + 3, 3 + 4\} = 7;\]
\[i^* = M1;\]
\[\Omega' = \{(1, 1), (1, 3)\}\]
Job Shop models

Generation of all active schedules - example (cont.)

Partial Schedule:

\[
\begin{array}{c|c}
 & \Omega = \{ (1, 1), (3, 2), (1, 3) \}; \\
M_1 & r_{11} = 6, r_{32} = 4, r_{13} = 3; \\
M_2 & t(\Omega) = \min\{6 + 1, 4 + 3, 3 + 4\} = 7; \\
M_3 & i^* = M_1; \\
\end{array}
\]

\[
\begin{array}{c|c}
 & \Omega' = \{ (1, 1), (1, 3) \} \\
M_1 & \\
M_2 & \\
M_3 & \\
\end{array}
\]

Extended partial schedules:
Job Shop models

Remarks on the generation:

• the given algorithm is the base of the branching

• nodes of the branching tree correspond to partial schedules

• Step 3 branches from the node corresponding to the current partial schedule

• the number of branches is given by the cardinality of $\Omega'$

• a branch corresponds to the choice of an operation $(i^*, j)$ to be schedules next on machine $i^*$
  → a branch fixes new disjunctions
Job Shop models

Disjunctions fixed by a branching

Node \( v \) with \( \Omega' = \{(i^*, j), (i^*, l)\} \)

- Node \( v' \) with selection \((i^*, j)\)
  - Add disjunctions \((i^*, j) \rightarrow (i^*, k)\)
    for all unscheduled operations \((i^*, k)\)

- Node \( v'' \) with selection \((i^*, l)\)
  - Add disjunctions \((i^*, l) \rightarrow (i^*, k)\)
    for all unscheduled operations \((i^*, k)\)

Consequence: Each node in the branch and bound tree is characterized by a set \( S' \) of fixed disjunctions
Job Shop models

Lower bounds for nodes of the branch and bound tree

- Consider node $V$ with fixed disjunctions $S'$:
- Simple lower bound:
  - calculate critical path in $G(S')$
  - $\rightarrow$ Lower bound $LB(V)$
Job Shop models

Lower bounds for nodes of the branch and bound tree

- Consider node $V$ with fixed disjunctions $S'$:
- Simple lower bound:
  - calculate critical path in $G(S')$
  - $\rightarrow$ Lower bound $LB(V)$
- Better lower bound:
  - consider machine $i$
  - allow parallel processing on all machines $\neq i$
  - solve problem on machine $i$
Job Shop models

1-machine problem resulting for better LB

1. calculate earliest starting times $r_{ij}$ of all operations $(i, j)$ on machine $i$ (longest paths from source in $G(S')$)

2. calculate minimum amount $q_{ij}$ of time between end of $(i, j)$ and end of schedule (longest path to sink in $G(S')$)

3. solve single machine problem on machine $i$:
   - respect release dates
   - no preemption
   - minimize maximum value of $C_{ij} + q_{ij}$

Result: head-body-tail problem (see Lecture 3)
Job Shop models

Better lower bound

- solve 1-machine problem for all machines
- this results in values $f_1, \ldots, f_m$
- $LB_{new}(V) = \max_{i=1}^{m} f_i$
Job Shop models

Better lower bound

- solve 1-machine problem for all machines
- this results in values $f_1, \ldots, f_m$
- $LB^{new}(V) = \max_{i=1}^{m} f_i$

Remarks:

- 1-machine problem is NP-hard
- computational experiments have shown that it pays off to solve these $m$ NP-hard problems per node of the search tree
- $20 \times 20$ job-shop instances are already hard to solve by branch and bound
Job Shop models

Better lower bound - example

Partial Schedule:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>$M_2$</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$M_3$</td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Corresponding graph $G(S')$:

Conjunctive arcs

Fixed disj.
Job Shop models

Better lower bound - example (cont.)

Graph $G(S')$ with processing times:

$LB(V)=l(U, (1, 2), (1, 3), (3, 3), V)=8$
Job Shop models

Better lower bound - example (cont.)

Graph $G(S')$ with processing times:

$$
\begin{align*}
3,1 & \quad 4 \\
2,1 & \quad 2 \\
1,1 & \quad 1 \\
3 & \\
1,2 & \\
3,2 & \\
2,3 & \\
1,3 & \\
3,3 & \\
2 & \\
4 & \\
1 & \\
\end{align*}
$$

$LB(V)=l(U, (1, 2), (1, 3), (3, 3), V)=8$

Data for jobs on Machine 1:

<table>
<thead>
<tr>
<th></th>
<th>green</th>
<th>blue</th>
<th>red</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{12}$</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_{12}$</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_{13}$</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_{13}$</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_{11}$</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_{11}$</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Opt. solution:

Opt = 8, $LB^{new}(V) = 8$
Job Shop models

Better lower bound - example (cont.)

Change $p_{11}$ from 1 to 2!

\[
\begin{align*}
&3,1 & 2,1 & 1,1 \\
&1,2 & 3,2 \\
&2,3 & 1,3 & 3,3
\end{align*}
\]

\[
\begin{align*}
U & \rightarrow 4 & 2 & 2 \\
& 3 \\
& 1,2 & 3,2 & 3 \\
& 2,3 & 1,3 & 3,3 \\
& 2 & 4 & 1 \\
V & \rightarrow
\end{align*}
\]

\[
LB(V) = l(U, (1,2), (1,3), (3,3), V) = 8
\]
Job Shop models

Better lower bound - example (cont.)

Change $p_{11}$ from 1 to 2!

Data for jobs on Machine 1:

<table>
<thead>
<tr>
<th>green</th>
<th>blue</th>
<th>red</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{12} = 0$</td>
<td>$r_{13} = 3$</td>
<td>$r_{11} = 6$</td>
</tr>
<tr>
<td>$q_{12} = 5$</td>
<td>$q_{13} = 1$</td>
<td>$q_{11} = 0$</td>
</tr>
</tbody>
</table>

Opt. solution:
$OPT = 9$, $LB^{new}(V) = 9$

$LB(V) = l(U, (1, 2), (1, 3), (3, 3), V)$
$= l(U, (3, 1), (2, 1), (1, 1), V) = 8$