Average Completion Time on Multiple Machines

\[ P||\sum C_j \]

<table>
<thead>
<tr>
<th>A</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
</tr>
<tr>
<td>E</td>
<td>6</td>
</tr>
<tr>
<td>F</td>
<td>9</td>
</tr>
<tr>
<td>G</td>
<td>12</td>
</tr>
<tr>
<td>H</td>
<td>20</td>
</tr>
<tr>
<td>I</td>
<td>50</td>
</tr>
<tr>
<td>J</td>
<td>60</td>
</tr>
</tbody>
</table>

What is the right algorithm?
Average Completion Time on Multiple Machines

- $P||\sum C_j$ – SPT is optimal.
- $P||\sum w_jC_j$ – Is WSPT optimal?

Example

<table>
<thead>
<tr>
<th>$j$</th>
<th>$w_j$</th>
<th>$p_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>99</td>
</tr>
</tbody>
</table>
Average Completion Time on Multiple Machines

• $P||\sum C_j$ – SPT is optimal.
• $P||\sum w_jC_j$ – Is WSPT optimal?

Example

\[
\begin{array}{ccc}
  j & w_j & p_j \\
  1 & 1 & 1 \\
  2 & 1 & 1 \\
  3 & 100 & 99 \\
\end{array}
\]

• $P||\sum w_jC_j$ is NP-complete.
• WSPT is a $(1 + \sqrt{2})/2$-approximation for $P||\sum w_jC_j$
\[ R | \sum C_j \]

- Can be solved as a matching problem.
- Left side node for each job \( j \)
- Right hand side node for the \( k \) th from last job on machine \( i \)

**Example**

<table>
<thead>
<tr>
<th></th>
<th>( J_1 )</th>
<th>( J_2 )</th>
<th>( J_3 )</th>
<th>( J_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_1 )</td>
<td>6</td>
<td>4</td>
<td>( \infty )</td>
<td>3</td>
</tr>
<tr>
<td>( M_2 )</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>( M_3 )</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>
LP for the matching problem

Variable \( x_{ijk} = 1 \) if \( j \) is the \( k \) th from last job on \( M_i \)

\[
\min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{n} k p_{ij} x_{ijk}
\]

s.t.

Each job runs
\[
\sum_{i=1}^{m} \sum_{k=1}^{n} x_{ijk} = 1 \quad j = 1 \ldots n
\]

Each machine/slot has at most 1 job
\[
\sum_{j=1}^{n} x_{ijk} \leq 1 \quad i = 1 \ldots m; k = 1 \ldots n
\]
\[
x_{ijk} \in \{0, 1\} \quad i = 1 \ldots m; j = 1 \ldots n; k = 1 \ldots n
\]

- Note that the may be unforced idleness e.g.

\[
\begin{array}{ccc}
J_1 & J_2 \\
M_1 & 1 & 1 \\
M_2 & 10 & 10 \\
\end{array}
\]
Algorithm is SRPT-FM. Shortest Remaining Processing Time on the Fastest Machines.

What about preemption in other models?
- P – doesn’t help
- R – NP-complete