LPT Schedules

Largest Processing Time First
Analysis is for Independent Jobs

EXAMPLE:

\( m = 2 \)
\( T_1 = 3, T_2 = 3, T_3 = 2, T_4 = 2, T_5 = 2 \)
\( \hat{f} = 7, f^* = 6 \)

Therefore, \( \frac{\hat{f}}{f^*} = \frac{7}{6} \)

<table>
<thead>
<tr>
<th>T1</th>
<th>T3</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
<td>T4</td>
<td>T5</td>
</tr>
</tbody>
</table>

\( m = 3 \)
\( T_1 = 5, T_2 = 5, T_3 = 4, T_4 = 4, T_5 = 3, T_6 = 3, T_7 = 3 \)
\( \hat{f} = 11, f^* = 9 \)

Therefore, \( \frac{\hat{f}}{f^*} = \frac{11}{9} \)
General Case:

\[ T_i = 2m - \left\lfloor \frac{i+1}{2} \right\rfloor \quad 1 \leq i \leq 2m; \quad t_{2m+1} = m; \]
\[ \hat{f} = 4m - 1, \; \hat{f}^* = 3m \]
Therefore, \[ \frac{\hat{f}}{\hat{f}^*} = \frac{4}{3} - \frac{1}{3m} \]
Theorem: \( \frac{\hat{f}(I)}{f^*(I)} \leq \frac{4}{3} - \frac{1}{3m} \)

Proof:

- Obviously true when \( m = 1 \).
- So, assume \( m > 1 \).
- Contradiction. Suppose theorem is false.
- Let \( t_1 \geq t_2 \geq \ldots \geq t_n \) be an instance with \textit{smallest} number of tasks for which the theorem is false. Let \( \text{OPT} \) be any optimal scheduled for it.
- Let \( k \) be the index of a task with largest finish time in the LPT schedule.
- Claim: \( k = n \). o.w. smaller counter example.
Since there cannot be any idle time in the LPT schedule before time \((\hat{f} - t_n)\) and only \(T_1, T_2, \ldots, T_{n-1}\) can be scheduled in that time interval, we know that

\[
m \times (\hat{f} - t_n) \leq \sum_{1}^{n-1} t_i
\]

\[
\hat{f} - t_n \leq \frac{1}{m} \sum_{1}^{n-1} t_i
\]

\[
\hat{f} \leq \frac{1}{m} \sum t_i + \frac{m - 1}{m} t_n
\]

Clearly, \(f^* \geq \frac{1}{m} \sum t_i\)
Since our problem instance is a counter-example to the theorem, and we have established that 
\[ \hat{f} \leq \frac{1}{m} \sum t_i + \frac{m-1}{m} t_n, \] and 
\[ f^* \geq \frac{1}{m} \sum t_i, \] we know that

\[
\frac{4}{3} - \frac{1}{3m} < \frac{\hat{f}}{f^*} \leq 1 + \frac{(m - 1)t_n}{mf^*}
\]

\[
\left(\frac{1}{3} - \frac{1}{3m}\right)f^* < \frac{m - 1}{m} t_n
\]

\[
f^* < \frac{m}{m-1} t_n
\]

\[
f^* < \frac{3m}{m-1} t_n
\]

Therefore, OPT has at most 2 tasks per processor, i.e., \( n \leq 2m \).
Transforming schedule OPT

Apply the following three transformations on OPT to get OPT’. Stop when no further transformation is possible.

Type 1
\[ i \quad j \quad \rightarrow \quad j \quad i \]
\( ti < tj \)

Type 2
\[ i \quad i' \quad \rightarrow \quad i \quad j \quad i' \]
\( tj < ti \)

Type 3
\[ i \quad i' \quad \rightarrow \quad i \quad j' \quad \]
\( ti > tj \) and \( ti' > tj' \)
\[ j \quad j' \quad \rightarrow \quad j \quad i' \]
Clearly, finish time \((\text{OPT}')\) \(\leq\) finish time \((\text{OPT})\).

\(\text{OPT}':\) \(t_{k_1} \geq t_{k_2} \geq \cdots \geq t_{k_r} \geq t_{i_1} \geq t_{i_2} \geq \cdots \geq t_{i_s} \geq t_{j_s} \geq t_{j_{s-1}} \geq \cdots \geq t_{j_1}\)

Permutation of processors

Termination

- Type 1 transformation at most \(n^2\) times.
- \(\sum f_i^2\) decreases with Type 2 and 3 operations, where \(f_i\): finish time for processor \(i\).
Let $a < b$ be the finishing time of the two processors involved in the type 2 or type 3 interchange.

The $\sum f_i^2$ before the interchange minus the $\sum f_i^2$ after the interchange is

$$= a^2 + b^2 - (a + \delta)^2 - (b - \delta)^2$$

for some $0 < \delta < b - a$.

Therefore, $\sum f_i^2$ decreases with a type 2 or type 3 interchange.