Scheduling to Minimize Lateness

algorithms
minimize the maximum lateness

• need to schedule a series of $n$ jobs on a single processor
• the $i^{th}$ job requires $t_i$ units of processing time
• ... and has a deadline of $d_i$
• if job $i$ is scheduled to start at time $s$, it finishes at $f_i = s + t_i$
• the lateness of the $i^{th}$ job is $l_i = \max \{0, f_i - d_i\}$
• goal is to minimize $\max \{ l_i \mid 1 \leq i \leq n \}$
• input: $t_1, t_2, ..., t_n$ and $d_1, d_2, ..., d_n$
example

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_i$</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$d_i$</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

schedule by decreasing computation time:

$d_4=9$
$d_1=6$
$d_5=14$
$d_2=8$
$d_6=16$
$d_3=9$

lateness = 0
lateness = 15-9 = 6
optimal: schedule by increasing deadline

\[ d_1 = 6, \quad d_2 = 8, \quad d_3 = 9, \quad d_4 = 9, \quad d_5 = 14, \quad d_6 = 16 \]

lateness = 10 - 9 = 1
proof of optimality

• suppose there is some other “optimal” schedule that does not satisfy earliest deadline first
• it does not satisfy $d_1 \leq d_2 \leq \ldots \leq d_n$
• look for an inversion
• then there must be an $i$ such that $d_i > d_{i+1}$ (why does it exist?)
• specifically look for two neighbors that are out of order
• makes proof a bit simpler
• swap jobs $i$ and $i+1$ and show that lateness does not get worse
proof of optimality - continued

• we swap jobs \( i \) and \( i+1 \) where \( d_i > d_{i+1} \)
• before finish and lateness values: \( f_1, f_2, \ldots, f_n \) and \( l_1, l_2, \ldots, l_n \)
• after finish and lateness values: \( f'_1, f'_2, \ldots, f'_n \) and \( l'_1, l'_2, \ldots, l'_n \)
• note: by swapping adjacent jobs, other finish times don’t change
• ... so \( f_j = f'_j \) for all \( j \neq i \) or \( i+1 \)
• small note: we assume that in original schedule, one job starts as soon as another finishes
• GOAL: show that \( \max\{ l'_i, l'_{i+1} \} \leq \max\{ l_i, l_{i+1} \} \)
• this is enough since other lateness values don’t change
making the swap of jobs $i$ and $i+1$

- **before:** job $i$ starts at time (say) $s$
- **before:** $f_i = s + t_i$ and $f_{i+1} = s + t_i + t_{i+1}$
- **before:** and $l_{i+1} = s + t_i + t_{i+1} - d_{i+1}$

- **after:** $l'_i = f'_i - d_{i+1} = s + t_{i+1} - d_{i+1} \leq s + t_{i+1} + t_i - d_{i+1} = l_{i+1}$
- **after:** $l'_{i+1} = f'_{i+1} - d_i = f_{i+1} - d_i \leq f_{i+1} - d_{i+1} = l_{i+1}$
- ... (since $f_{i+1} = f'_{i+1}$ and $d_i > d_{i+1}$)

- Therefore: $\max\{ l'_i, l'_{i+1} \} \leq l_{i+1} \leq \max\{ l_i, l_{i+1} \}$  \hspace{1cm} \text{(done!)}