Scheduling to Minimize Lateness

- You have a very busy month: $n$ assignments are due, with different deadlines

<table>
<thead>
<tr>
<th>Assignments</th>
<th>Deadlines</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>---</td>
</tr>
<tr>
<td>2:</td>
<td>---</td>
</tr>
<tr>
<td>3:</td>
<td>---</td>
</tr>
<tr>
<td>4:</td>
<td>---</td>
</tr>
</tbody>
</table>

- How should you schedule your time to “minimize lateness”?

Possible Greedy Approaches

- **Note**: it never hurts to schedule assignments consecutively with no “idle time” ⇒ schedule determined by order of assignments

- What order should we choose?
  - **Shortest Length**: ascending order of $t_j$.
  - **Earliest Deadline**: ascending order of $d_j$.
  - **Smallest Slack**: ascending order of $d_j - t_j$.

- Only earliest deadline first is optimal in all examples. Let’s prove it is always optimal.

Exchange Argument (False Start)

Assume jobs ordered by deadline $d_1 \leq d_2 \leq \ldots \leq d_n$, so the greedy ordering is simply

\[
A = 1, 2, \ldots, n
\]

**Claim**: $A$ is optimal

**Proof attempt**: Suppose for contradiction that $A$ is not optimal. Then, there is an optimal solution $O$ with $O \neq A$

- Since $O \neq A$, there must be two jobs $i$ and $j$ that are out of order in $O$ (e.g. $O = 1, 3, 2, 4$)
- Let’s swap $i$ and $j$ and show we get a better solution $O'$
  \[\implies O \text{ is not optimal}. \text{ Contradiction, so } A \text{ must be optimal.} \]

**Problem? $O'$ may still be optimal. Example.**
Exchange Argument (Correct)

Suppose $O$ optimal and $O \neq A$. Then we can modify $O$ to get a new solution $O'$ that is:

1. No worse than $O$
2. Closer to $A$ in some measurable way

$O$(optimal) → $O'$(optimal) → $O''$(optimal) → ... → $A$(optimal)

High-level idea: gradually transform $O$ into $A$ without hurting solution, thus preserving optimality.

Concretely: show 1 and 2 above.

Exchange Argument for Scheduling to Minimize Lateness

Recall $A = 1, 2, \ldots, n$. For $S \neq A$, say there is an inversion if $i$ comes before $i$ but $j < i$. Claim: if $S$ has an inversion, $S$ has a consecutive inversion—one where $i$ comes immediately before $j$.

Main result: let $O \neq A$ be an optimal schedule. Then $O$ has a consecutive inversion $i, j$. We can swap $i$ and $j$ to get a new schedule $O'$ such that:

1. Maximum lateness of $O'$ is no bigger than maximum lateness of $O$
2. $O'$ has one less inversion than $O$

Proof:

1. On board / next slide
2. Obvious

Wrap-Up

For any optimal $O \neq A$ we showed that we could transform $O$ to $O'$ such that:

1. $O'$ is still optimal
2. $O'$ has one less inversion than $A$

$O$(optimal) → $O'$(optimal) → $O''$(optimal) → ... → $A$(optimal)

Since there are at most $\binom{n}{2}$ inversions, by repeating the process a finite number of times we see that $A$ is optimal.

Proof of 1

Swapping a consecutive inversion ($i$ precedes $j$; $d_j \leq d_i$)

Consider the lateness $\ell'_k$ of each job $k$ in $O'$:

- If $k \notin \{i, j\}$, then lateness is unchanged: $\ell'_k = \ell_k$
- Job $j$ finishes earlier in $O'$ than $O$: $\ell'_j \leq \ell_j$
- Finish time of $i$ in $O'$ = finish time of $j$ in $O$. Therefore
  \[ \ell'_i = f'_i - d_i = f_j - d_i \leq f_j - d_j = \ell_j \]

Conclusion: $\max_k \ell'_k \leq \max_k \ell_k$. Therefore $O'$ is still optimal.