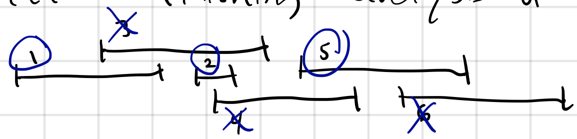
2 Feb 2018 — Scheduling to Minimize Lateness (§4.2)

Theme: Analyzing Greedy Algs via "Exchange Arguments"

But first ... finishing analysis of Earliest Finish Time.



Recall: $f_1 \le f_2 \le \dots \le f_n$ are jobs' finish time.

LAST LECTURE'S INDUCTION HYPOTHESIS:

The number of intervals with finish time $\le f_j$ chosen by our algorithm is at least as great as the # of int. with f.t. $\le f_j$ in any other ~~set~~ non-conflicting set.

Base case $j = 1$.

EFT chooses 1 interval that finishes $\le f_1$. There is only one such interval, so no other non-conflicting set contains more than one.

Induction step. $j > 1$. Assume induction hypothesis true for $1 \le i \le j-1$. ("strong induction")
Consider any non-conflicting interval set, S.
Must show S contains no more intervals finishing $\le f_j$ than our algorithm's set does.

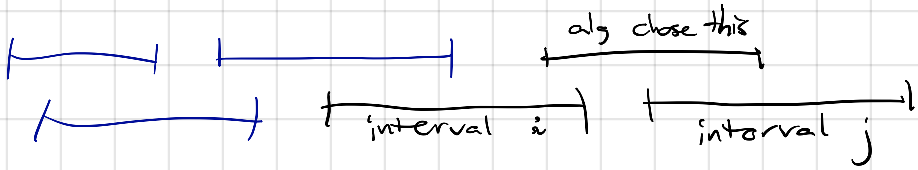
Case 1. Interval j doesn't belong to S.
Induction hypothesis for $j-1$ says

$$\begin{pmatrix}\text{\# of intervals alg chooses}\\ \text{ending} \le f_{j-1}\end{pmatrix} \ge \begin{pmatrix}\text{\# intervals in } S\\ \text{ending} \le f_{j-1}\end{pmatrix}$$

$$\begin{array}{ccc} \wedge\!\!\!\shortparallel & & \shortparallel \\ \begin{pmatrix}\text{\# interval alg chooses}\\ \text{ending} \le f_j\end{pmatrix} & & \begin{pmatrix}\text{\# intervals in } S\\ \text{ending} \le f_j\end{pmatrix}\end{array}$$

Case 2. Interval j does belong to S.

Observation: our algorithm chooses some interval, I, which is either equal to interval j or I finishes some time between start & finish times of interval j.



Let f_i be the latest finish time that occurs before the start of interval j.

Apply induction hypothesis for i.

$$\begin{pmatrix}\#\text{ intervals alg chooses}\\ \text{ending} \le f_i\end{pmatrix} \geq \begin{pmatrix}\#\text{ intervals in } S\\ \text{ending} \le f_i\end{pmatrix}$$

$$\underbrace{1 + \big(\text{— same thing —}\big)}_{\begin{pmatrix}\#\text{ intervals alg chooses}\\ \text{ending} \le f_j\end{pmatrix}} \geq \underbrace{1 + \big(\text{— same thing —}\big)}_{\begin{pmatrix}\#\text{ intervals in } S\\ \text{ending} \le f_j\end{pmatrix}}$$

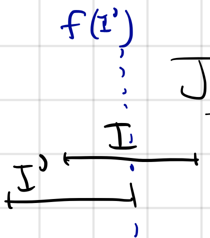
$\boxtimes$

Alternate proof of correctness. Method is "augment the algorithm to produce a certificate of its own correctness."
A piece of evidence that can be easily produced as alg. runs, and allows a third party to easily check the solution is correct.

Let $F = \{\text{finishing times of the intervals that EFT chooses}\}$

Observation. Every one of the intervals in the problem input contains a time step that is an element of F.

Justification. Consider interval I. Either EFT chose I, then $f(I) \in F$. Or EFT deleted I when choosing an interval I' that finishes earlier and conflicts with I.



Then $f(I') \in F$ and is in the interior of I.

If S is any set of non-conflicting intervals, each of them contains a <u>distinct</u> element of F.

$$\therefore \quad \underset{\text{any other valid solution}}{|S|} \leq |F| = \underset{\text{our solution}}{|\text{EFT set}|}$$

QED!

Scheduling to minimize lateness.

<u>Input</u>: Set of n jobs.

Each is specified by giving

t_i = amount of time required to process job i

d_i = deadline of job i.

Global start time, s, when a resource becomes available

<u>Goal</u>. Assign to each job a contiguous interval of length t_i.

The assigned intervals shouldn't overlap except at endpoints.

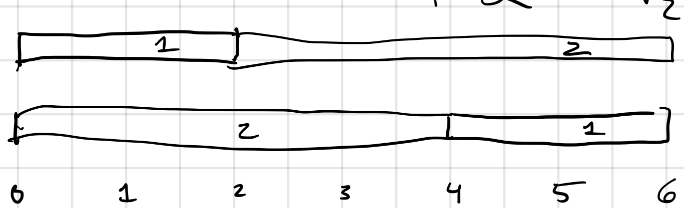
No assigned interval starts before s.

Minimize maximum lateness:

$$\underset{\text{scheduler}}{\text{minimize}} \ \max_i \ (\underset{\substack{\uparrow \\ \text{finish time}}}{f_i} - \underset{\substack{\uparrow \\ \text{deadline}}}{d_i})^+ \qquad x^+ := \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases}$$

Example. $s = 0$ $\quad d_1 = 1 \quad d_2 = 4$
$\quad t_1 = 2 \quad t_2 = 4$

This schedule is optimal.



$(f_1 - d_1)^+ = 1 \quad (f_2 - d_2) = 2$

$(f_1 - d_1)^+ = 5 \quad (f_2 - d_2) = 0$

Earliest Deadline First (EDF).

Sequence jobs in order of increasing deadline.

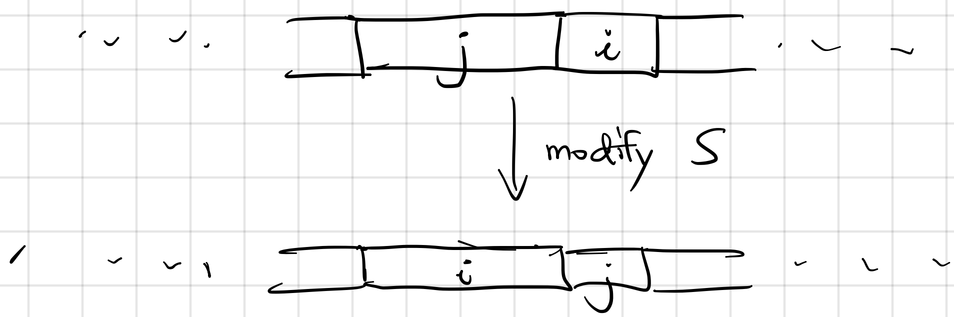
Proof of correctness is via "exchange argument".
Show that every other schedule is suboptimal. ⟶ or equally as good as our solution
By process of elimination, ours is optimal. ⟶ by proposing a way to improve it. ... or make it more similar to ours.

(Actually this is an oversimplification that ignores tie-breaking.)

Suppose WLOG that jobs are numbered s.t. $d_1 \leq \cdots \leq d_n$.
Observation 1. If S is any schedule with "idle time" then it can be improved by pushing jobs together.

The only decision we really must make is a sequencing decision.

If S is an ordering of the jobs other than $1, 2, \ldots, n$ then there must be $i < j$ such that S contains j, i in consecutive order.



This modification leaves finish time of every job besides i, j unchanged. f_i gets earlier, improves lateness of i. f_j gets later, lateness of j may get worse.

(new lateness of j) $\leq$ (old lateness of i)

$\therefore$ exchanging i with j can only improve (or remain the same) the maximum lateness.

After a sequence of such exchanges, we obtain the EDF schedule, and the lateness (weakly) improved along the way, so S is not better than EDF.