CS 7800: Advanced Algorithms

Greedy Algs I:

- Interval Scheduling
- Minimum Lateness Scheduling

Jonathan Ullman

09-13-2022

Optimization

feasible solutions

objective function

$$\text{Given} \quad \underbrace{F, g}_{\text{input}} \qquad \text{find } y^* \in \operatorname*{argmax}_{y \in F} g(y)$$

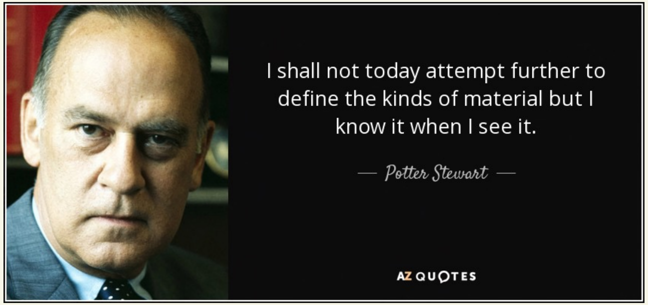
Discrete:

- shortest (s,t) path
- minimum spanning tree
- optimal bin packing
- max-weight matching

Continuous:

- maximum flow
- optimal parameters for an SVM

Greedy Algorithms



Typically: make one "pass" over the input and make irrevocable choices

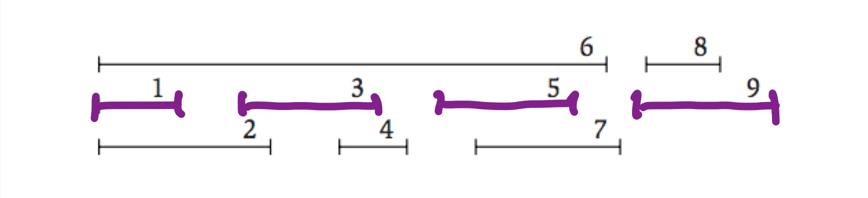
Why?

- speed
- simplicity
- easy to understand

Today: two examples, two proof styles

- Interval scheduling ⇔ induction
- Minimum lateness ⇔ exchange argument

Interval Scheduling



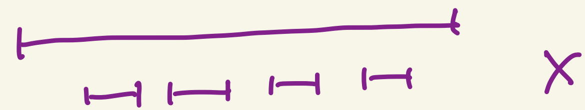
<u>Input</u>: n intervals $[s_i, f_i]$ (assume all s_i, f_i are distinct)

<u>Output</u>: A non-overlapping subset of intervals of maximum size

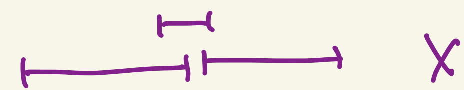
Interval Scheduling

What to sort on?

① Sort by s_i



② Sort by length $f_i - s_i$



③ Sort by f_i

the one that works

Interval Scheduling

<u>Algorithm:</u>

Sort by finish time so $f_1 \le f_2 \le \dots \le f_n$

Let $S = \emptyset$

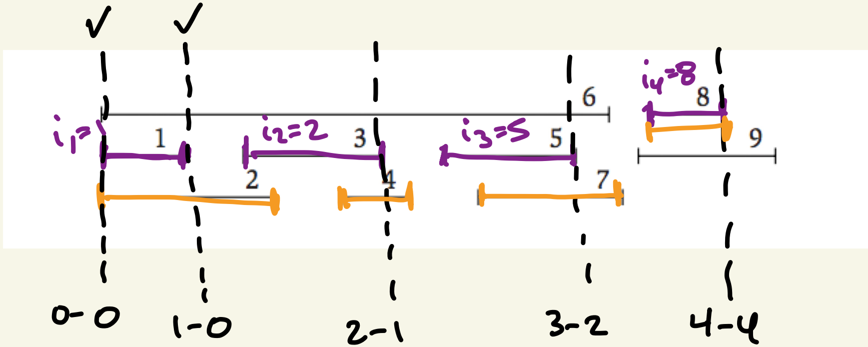
For $i = 1 \dots n$:

- If $[s_i, f_i]$ conflicts with S: continue
- Else: add $[s_i, f_i]$ to S

Return S

Implement in $O(n \log n)$ time.

Proof of Correctness



- Let S be the output of greedy $i_1, i_2, \ldots i_\ell$

- Let T be the optimal schedule be $j_1, j_2, \ldots, j_k$

Clm: For every $m = 1, 2, \ldots, \ell$, after my interval m completes, T has completed $\le m$ intervals

Proof Cont'd

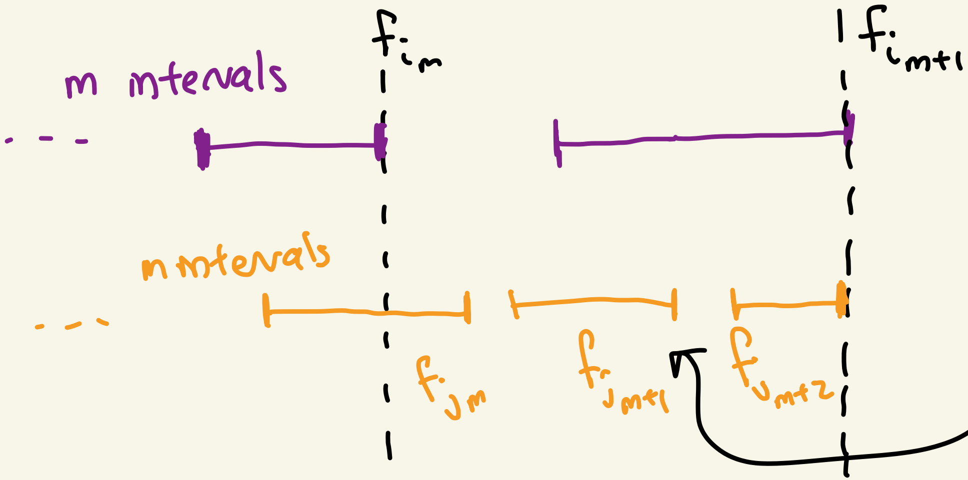
- Let S be the output of greedy $i_1, i_2, \dots i_\ell$
- Let T be the optimal schedule be $j_1, j_2, \dots, j_k$

Clm: For every $m = 1, 2, \dots, \ell$, after my interval m completes, T has completed $\leq m$ intervals

Proof by Induction:

Base Case: True for $m=1$

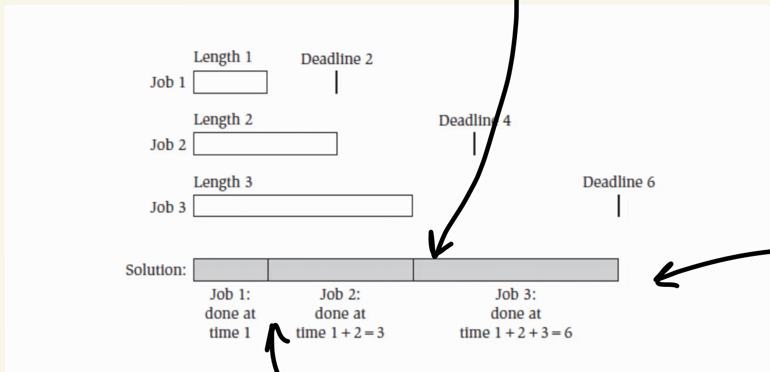
Inductive Step: Assume true for m, show it's true for $m+1$



If true for m, but not true for $m+1$ then we conclude that this interval wasn't considered by greedy

Minimum Lateness Schedule

job 2 finishes at $t=3$



job 3 finishes at $t=6$

job 1 finishes at $t=1$

<u>Input:</u> n jobs of length l_i deadline d_i

<u>Output:</u> An ordering of the jobs $i_1, i_2, \ldots, i_n$ (job i finishes at time f_i)

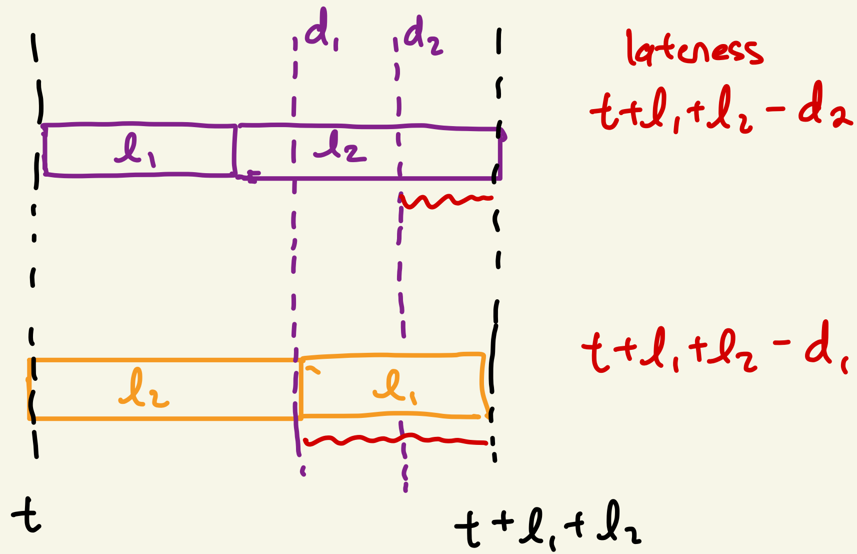
that minimizes $\max_i \max\{f_i - d_i, 0\}$

Proof of Correctness ($n=2$)

Algorithm:

Sort by deadline

$d_1 \le d_2 \le \dots \le d_n$

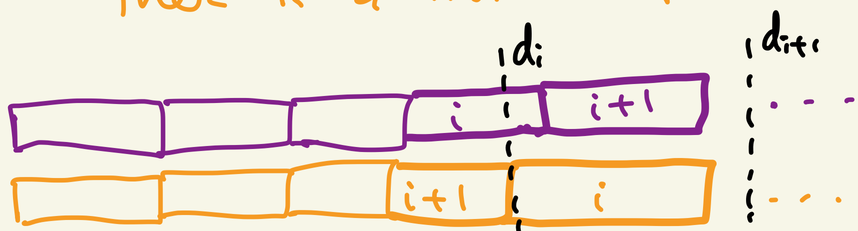


$$\max\{\max\{0, t+l_1-d_1\}, \max\{0, t+l_1+l_2-d_2\}\}$$

$$\le \max\{\max\{0, t+l_2-d_2\}, \max\{0, t+l_1+l_2-d_1\}\}$$

Proof of Correctness (General Case)

- Let S be greedy (sorted by deadline)
- Let T be optimal (not sorted by deadline)
  - $\Rightarrow$ some pair of jobs where $d_i < d_j$ but j comes before i
  - $\Rightarrow$ some pair $i, i+1$ where $d_i < d_{i+1}$ but $i+1$ comes first
  - $\Rightarrow$ there is a first such pair



If we only scheduled $i, i+1$ then greedy has smaller lateness

Flipping $i, i+1$ only improves the lateness of T