# Greedy Algorithms (continued)

Recall: a greedy alg. solves an optimization problem by making a series of choices. Each choice is made without "looking ahead" to see the consequences of the choice — the algorithm just chooses the option that seems best right now.

Before the strike, we started talking about the following scheduling problem.

We have $n$ jobs.
Each takes 1 unit of time to complete
Job $i$ has a deadline $d_i$ $(1 \le i \le n)$
and a profit $p_i$. $(1 \le i \le n)$

We want to choose subset of the jobs and an ordering to do them in that maximizes the total profit of all jobs that are completed on time.

(A job that is not completed before its deadline does not provide any profit).

We came up with the algorithm:

Sort jobs so that $p_1 \ge p_2 \ge \cdots \ge p_n$.
For $i = 1..n$
  if $\exists$ a free time slot before $d_i$
    then schedule job $i$ in last such slot.
end for.

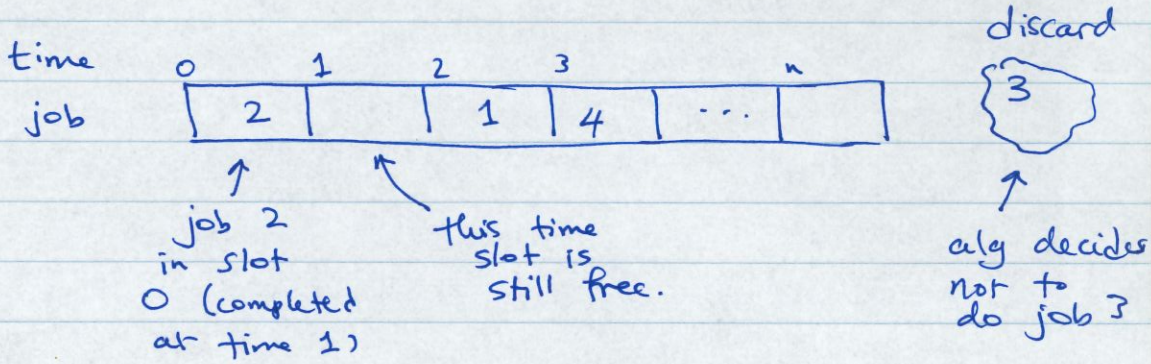
Clearly, this produces a feasible schedule (we only schedule a job in a time slot if that slot is empty)

We must just prove solution produced is optimal

Recall: a good way to prove greedy alg is correct is to show that it never makes a bad choice: at any time, ∃ optimal solution that extends the partial solution constructed so far by the algorithm.

In this case, how can we describe a partial solution at end of the $i^{th}$ iteration of the loop?

- it is an assignment of jobs 1..i either to timeslots or to "discard job".
- we'll draw the partial schedule like this:




(Formally, the partial schedule ~~at time~~ after i iterations is a function

$$S_i : \{1, 2, \ldots, i\} \to \{0, 1, 2, \ldots, n-1, \text{discard}\}$$

In the example above

$S_i(2) = 0$ $\quad$ $S_i(4) = 3$.

$S_i(3) = \text{discard}$ $\quad$ $S_i(1) = 2$. )

An ~~optimal~~ complete schedule is a function $S: \{1, \ldots, n\} \to \{0, \ldots, n-1, \text{discard}\}$

Assume no schedule ever puts a job after its deadline - instead of doing this it just discards job.

What does it mean to say $S^*$ is an extension of the partial schedule $S_i$?

- intuitively $S^*$ schedules jobs $1 \ldots i$ in exactly the same way $S_i$ does

(Formally, $\forall j \leq i, \; S_i(j) = S^*(j)$.)

Note this is basically a loop invariant.

**Claim** $\forall i \; (0 \leq i \leq n)$ $\exists$ optimal schedule $S^*$ that is an extension of $S_i$.

**Proof** by induction on $i$.

**Base** $i = 0$ The claim $\forall j \leq i \; S_i(j) = S^*(j)$ is vacuously true since there are no jobs $j \leq 0$.

**Ind. Step.** Let $i \geq 1$.
Assume $\exists$ optimal $S^*$ that extends $S_{i-1}$.
We prove $\exists$ optimal $\hat{S}$ that extends $S_i$.

As usual, the proof reflects the structure of the code: we consider 2 cases depending on whether the "if" condition is true.

**Case I** $\exists$ empty slot before $d_i$ in $S_{i-1}$.
Then the alg. schedules job $i$ in the last such slot, $t$.

**Case I(a)** If $S^*(i) = t$ also, then $\hat{S} = S^*$ is an optimal extension of $S_i$.

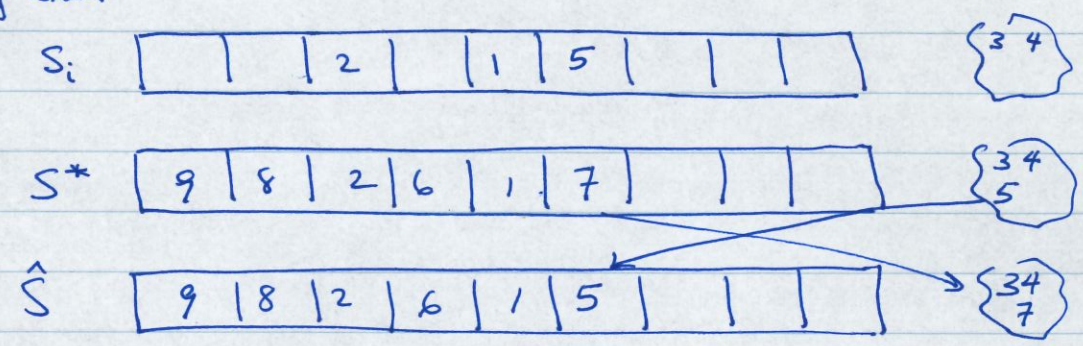
Otherwise, we will have to change $S^*$ slightly to make it an extension of $S_i$:

Case I(b) If $S^*(i) =$ discard, then let $\hat{S}$ be the schedule obtained from $S^*$ by replacing the job in slot $t$ (if any) by $i$.

Note that the job that is replaced (if any) must be bigger than $i$ (since $S_{i-1}$ and $S^*$ agree on what to do with jobs $1 \ldots i-1$ and $S_{i-1}$ has no job in slot $t$) and therefore have profit $\leq p_i$ (since jobs are sorted).

Thus, total profit of $\hat{S}$ is at least as big as profit of $S^*$, and $\hat{S}$ is an extension of $S_i$.

Picture of example if $i = 5$



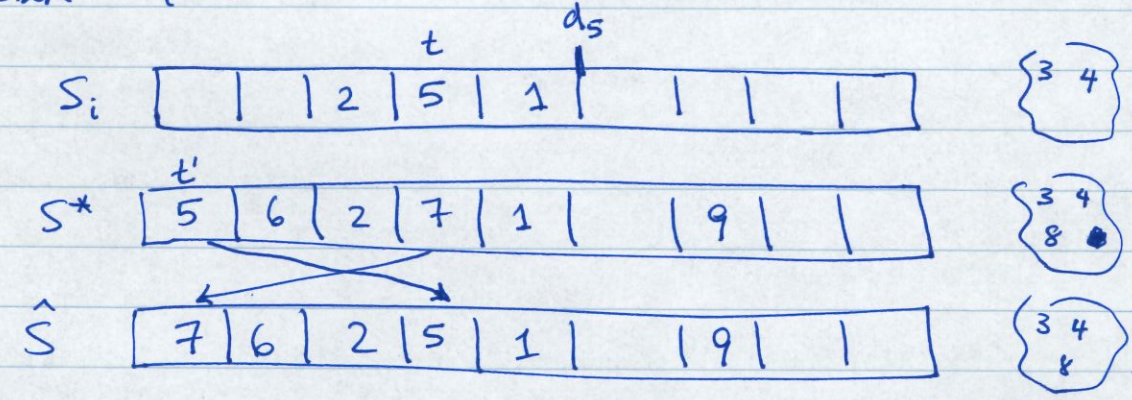

Case I(c) If $S^*(i) = t' \neq t$.

First, notice that $t' < t$ since $t$ is the last free slot before $d_i$ in $S_{i-1}$ (and therefore in $S^*$).

Let $\hat{S}$ be the schedule obtained from $S^*$ by swapping the contents of $t'$ and $t$.

Note: profit of $\hat{S}$ = profit of $S^*$ since job $i$ is done on time in both and the job done in slot $t$ of $S^*$ (if any) is done even earlier in $\hat{S}$, so it will still be on time.
And $\hat{S}$ is an extension of $S_i$

Picture $i=5$




Case II There is no empty slot before $d_i$ in $S_{i-1}$.

Then $S_i$ discards job $i$.
But $S^*$ must too, because it agrees with $S_{i-1}$ on how to schedule jobs $1 \ldots i-1$ and therefore has no slots before $d_i$ in which it could do job $i$.
So $S^*$ is an extension of $S_i$.

This completes the proof of the claim by induction.

When the loop terminates, we know there is an optimal solution $S^*$ such that $S_n(j) = S^*(j) \quad \forall j \leq n$.
I.e. $S_n$ is the optimal solution $S^*$.