# Greedy Algorithms II

## A greedy algorithm for job scheduling with profits

Reference: Course Readings, pages 20–30.

One processor, $n$ jobs, $\{1, 2, \ldots, n\}$. Each job takes unit time to execute. Job $i$ has a deadline $d_i \in \mathbb{N}$, and profit $g_i > 0$. We want to schedule the jobs in such a way that we maximize the total profit (the sum of all the profits).

**Definition 1** A *schedule* is an array $S(1), S(2), \ldots, S(d)$ where $d = \max d_i$ (i.e. the latest deadline), such that if $S(t) = i$, then job $i$ is scheduled at time $t$, $1 \leq t \leq d$. If $S(t) = 0$, then no job is scheduled at time $t$.

**Definition 2** A schedule $S$ is *feasible* if (a) If $S(t) = i > 0$, then $t \leq d_i$ (i.e. every scheduled job meets its deadline), and (b) If $t_1 \neq t_2$ and $S(t_1) \neq 0$, then $S(t_1) \neq S(t_2)$ (i.e. each job is scheduled at most once).

Let the total profit of schedule $S$ be $P(S) = g_{S(1)} + g_{S(2)} + \ldots + g_{S(d)}$, where $g_0 = 0$.

We want to find a feasible schedule $S$ whose profit $P(S)$ is as large as possible. This can be accomplished with the following greedy algorithm:

Sort the jobs in non-increasing order of profits: $g_1 \geq g_2 \geq \ldots \geq g_n$
$d \longleftarrow \max_i d_i$
for $t : 1..d$
    $S(t) \longleftarrow 0$
end for
for $i : 1..n$
    Find the largest $t$ such that ($S(t) = 0$ and $t \leq d_i$), and let $S(t) \longleftarrow i$
    (that is, find the latest possible time meeting the deadline)
end for

**Theorem 1** The greedy solution above is optimal (i.e. the profit $P(S)$ of the schedule $S$ computed by this greedy algorithm is as large as possible).

[1]These notes were taken originally by Eugenia Ternovskaia in Steven Cook's csc364 lecture, Spring 99, and they have been edited for this term.

**Definition 3** A schedule is *promising* if it can be extended to an optimal schedule. Schedule $S'$ extends schedule $S$ iff for all $1 \leq t \leq d$, if $S(t) \neq 0$, then $S(t) = S'(t)$.

**Example 1** If $S = (2, 0, 0, 0, 3)$, and $S' = (2, 0, 1, 0, 3)$, then $S'$ extends $S$.

**Lemma 1** The following is a loop invariant of the above greedy algorithm: *$S$ is promising.*

Exercise: Why does the above Lemma imply Theorem 1? (Hint: simple observation).

*Proof of Lemma* 1. Initially $S = (0, 0, \ldots, 0)$ is promising (take any optimal schedule; it is an extension).

Induction Step: Suppose that $S$ is promising, and let $S'$ be the result of one more iteration through the loop. We have to show that $S'$ continues to be promising, that is, we have to show that there is an optimal schedule that extends $S'$.

Since $S$ is promising it can be extended to an optimal solution $S_{\text{opt}}$.

**case (a)** job $i$ cannot be scheduled. Then $S' = S$, so $S'$ is promising as $S_{\text{opt}}$ extends it.

**case (b)** job $i$ is scheduled at time $t_0$, so $S'(t_0) = i$ and $t_0$ is the latest possible time for job $i$ given the schedule $S$.

**subcase (1)** job $i$ is scheduled in $S_{\text{opt}}$ at time $t_1$. If $t_1 = t_0$ then we are done (why?). If $t_1 < t_0$ then let $S'_{\text{opt}}$ be $S_{\text{opt}}$ except that we interchange $t_0$ and $t_1$, that is: $S'_{\text{opt}}(t_0) = S_{\text{opt}}(t_1) = i$ and $S'_{\text{opt}}(t_1) = S_{\text{opt}}(t_0)$. Then $S'_{\text{opt}}$ is feasible (why?), and $P(S'_{\text{opt}}) = P(S_{\text{opt}})$, so $S'_{\text{opt}}$ is still optimal, and $S'_{\text{opt}}$ extends $S'$. The case where $t_1 > t_0$ is not possible (why?).

**subcase (2)** job $i$ is not scheduled in $S_{\text{opt}}$. Then we simply define $S'_{\text{opt}}$ to be the same as $S_{\text{opt}}$, except $S'_{\text{opt}}(t_0) = i$. Since $S_{\text{opt}}$ is feasible, so is $S'_{\text{opt}}$, and since $S'_{\text{opt}}$ extends $S'$, we only have to show that $P(S'_{\text{opt}}) = P(S_{\text{opt}})$. This follows from the following claim:

**Claim:** Let $S_{\text{opt}}(t_0) = j$. Then $g_j \leq g_i$.

We prove the claim by contradiction: assume that $g_j > g_i$ (note that in this case $j \neq 0$). Then job $j$ was considered before job $i$. Since job $i$ was scheduled at time $t_0$, job $j$ must have been scheduled at time $t_2 \neq t_0$ (we know that job $j$ was scheduled in $S$ since $S(t_0) = 0$, and $t_0 \leq d_j$, so there was a slot for job $j$, and therefore it was scheduled). But $S_{\text{opt}}$ extends $S$, and $S(t_2) = j \neq S_{\text{opt}}(t_2)$ — contradiction.