by Michael Soltys
January 20, 2005

Job Scheduling with Deadlines and Profits

We have n jobs, each of which takes unit time, and a processor on which we would like to schedule them in as profitable a manner as possible. Each job has a profit associated with it, as well as a deadline; if the job is not scheduled by its deadline, then we don't get its profit. Because each job takes the same amount of time, we will think of a schedule S as consisting of a sequence of job "slots" $1, 2, 3, \ldots$ where $S(t)$ is the job scheduled in slot t.

Formally, the input to the problem is a sequence of pairs $(d_1, g_1), (d_2, g_2), \ldots, (d_n, g_n)$ where g_i is a non-negative real number representing the profit obtainable from job i, and $d_i \in \mathbb{N}$ is the deadline for job i.

Definition 1 A *schedule* is an array $S(1), S(2), \ldots, S(d)$ where $d = \max d_i$ (i.e. the latest deadline, beyond which no jobs can be scheduled), 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).

Definition 3 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 $S(t) \longleftarrow i$
    [find the latest possible free slot 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).

Definition 4 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 $S(t) = S'(t)$.

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

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

PROOF: (of Lemma 1) The proof is by induction.

Basis Case: 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_{opt} be *some* optimal schedule that extends S. Let S' be the result of one more iteration through the loop where job i is considered. We must prove that S' continues to be promising, so the goal is to show that there is an optimal schedule S'_{opt} that extends S'. We consider two cases:

case (I) job i cannot be scheduled. Then $S' = S$, so we let $S'_{\text{opt}} = S_{\text{opt}}$, and we are done.

case (II) job i is scheduled at time t_0, so $S'(t_0) = i$ (whereas $S(t_0) = 0$) and t_0 is the latest possible time for job i in the schedule S.

subcase (a) job i is scheduled in S_{opt} at time t_1:

If $t_1 = t_0$ then, as in case (I), just let $S'_{\text{opt}} = S_{\text{opt}}$.

If $t_1 < t_0$ then let S'_{opt} be S_{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'_{opt} is feasible (why?), it extends S' (why?), and $P(S'_{\text{opt}}) = P(S_{\text{opt}})$ (why?).

The case $t_1 > t_0$ is not possible (why?).

subcase (b) job i is not scheduled in S_{opt}. Then we simply define S'_{opt} to be the same as S_{opt}, except $S'_{\text{opt}}(t_0) = i$. Since S_{opt} is feasible, so is S'_{opt}, and since S'_{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 $g_j \le g_i$.

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 $t_0 \le d_j$, so there was a slot for job j, and therefore it was scheduled). But S_{opt} extends S, and $S(t_2) = j \neq S_{\text{opt}}(t_2)$ — contradiction.

This finishes the proof of the Induction Step. $\square$

Exercise: Make sure you can answer all the "why's" in the above proof. Also, where in the proof of the claim you use the fact that $j \neq 0$?