Some comments on proving the correctness of greedy algorithms

Vassos Hadzilacos

In class we proved the correctness of the greedy algorithm for interval scheduling by employing a “greedy-stays-ahead” argument. Specifically, we proved that if the set of (compatible) jobs constructed by the greedy algorithm listed left-to-right are

\[ j_1, j_2, \ldots, j_k \]

and the jobs in an optimal set listed left-to-right are

\[ j^*_1, j^*_2, \ldots, j^*_m, \]

then \( f(j_t) \leq f(j^*_t) \) for every \( t = 1, 2, \ldots, k \). Using this, we then argued that the set of jobs produced by the greedy algorithm is, in fact, optimal.

There is an alternative proof of a similar nature that uses what might be called the “promising set” argument. Specifically, we prove that:

**Lemma 1** For every iteration \( t \) of the loop, the set of jobs that have been added to the set \( A \) at the end of that iteration is a compatible set of jobs that is a subset of some optimal set.

**Proof.** By induction on \( t \). The basis \( t = 0 \) is obvious, since at the end of the “0-th iteration” of the loop, i.e., just before we enter the loop for the first time, \( A \) is empty.

For the induction step, let \( A_t \) be the set of jobs in \( A \) at the end of iteration \( t \), for some \( t \geq 0 \). Assume (by induction hypothesis) that \( A_t \) is compatible and is a subset of some optimal set \( A^* \) of jobs. Let \( A_{t+1} \) be the set of jobs in \( A \) at the end of iteration \( t + 1 \). We will prove that (a) \( A_{t+1} \) is compatible, and (b) \( A_{t+1} \) is a subset of some optimal set \( \hat{A} \) of jobs. (Note that \( \hat{A} \) may be a different optimal set than \( A^* \).) Part (a) is easy to see since the job added to \( A \) in iteration \( t + 1 \), if any, starts after all jobs in \( A_t \) have finished. Part (b) is likewise easy to see if no job is added to \( A \) in iteration \( t + 1 \), or if the job \( j \) added to \( A \) in iteration \( t + 1 \) happens to be in \( A^* \). It remains to show (b) if some job \( j \) is added to \( A \) in iteration \( t + 1 \), and \( j \notin A^* \).

In that case, first we note that \( A^* \) contains some job \( j^* \) that conflicts with \( j \): for, otherwise, we could add \( j \) to \( A^* \) and obtain a set of compatible jobs with more jobs than \( A^* \), contradicting that \( A^* \) is optimal. Second, we note that \( A^* \) does not contain two jobs that conflict with \( j \): for, otherwise, both of them would start after all the jobs in \( A_t \) finished (because \( A_t \) is, by induction hypothesis, a subset of \( A^* \), which is optimal and hence compatible), and one of them, call it \( j' \), would finish before \( j \); so the greedy algorithm would have added \( j' \), rather than \( j \), to \( A \) in iteration \( t + 1 \). So, there is exactly one job \( j^* \) in \( A^* - A_t \) that conflicts with \( j \). Define \( \hat{A} = (A^* - \{j^*\}) \cup \{j\} \); in other words \( \hat{A} \) is obtained from \( A^* \) by replacing \( j^* \) with \( j \). Since \( j^* \) is the only job in \( A^* \) that is conflicts with \( j \), \( \hat{A} \) is also a compatible set of jobs: \( j \) is compatible with every job in \( A_t \) (otherwise it would not have been added to \( A \) in iteration \( t + 1 \) by the greedy algorithm), and is compatible with every job in \( A^* - A_t \) except for \( j^* \). Furthermore, \( \hat{A} \) has the same number of jobs as the optimal set \( A^* \). So, \( \hat{A} \) is optimal. Finally, by construction, \( \hat{A} \) contains all the jobs in \( A_t \) as well as \( j \). Thus, \( A_t \cup \{j\} = A_{t+1} \) is a subset of the optimal set \( \hat{A} \), as wanted. \[ \square \]
We can now prove that the set $A$ returned by the greedy algorithm is optimal: By the preceding lemma, for some optimal set $A^*$, $A \subseteq A^*$. We claim that, in fact, $A = A^*$. For, otherwise, there would exist some job $j \in A^* - A$. Since $A^*$ is optimal, $j$ is compatible with every job in $A^*$ other than itself. Since $A \subseteq A^*$ and $j \notin A$, $j$ is compatible with every job in $A$. So, when $j$’s turn comes to be considered for inclusion in $A$, it would be added to it, contradicting that $j \notin A$. So, $A = A^*$, which means that the greedy algorithm returns an optimal set.

The “greedy-stays-ahead” and “promising set” arguments are very similar to each other in spirit. Both capture the intuition that the greedy algorithm builds an optimal solution incrementally so that the partial solution constructed at each stage is “optimal so far”.

The “switching” argument we saw in the proof of the greedy algorithm for minimum-lateness scheduling has a different flavour. There we prove that the greedy algorithm produces an optimal solution by showing how to transform an arbitrary optimal solution to the one constructed by the greedy algorithm through a series of “switching steps”, each of which preserves optimality. This style of proof also uses induction (or its first cousin, the well-ordering principle), but in a different way. We don’t do induction on the number of steps through which the algorithm builds the (as it turns out, optimal) solution. Rather, we do induction on the number of optimality-preserving steps required to gradually “massage” the arbitrarily chosen optimal solution into the solution produced by the greedy algorithm.