Comments on proving the correctness of (some) greedy algorithms

Vassos Hadzilacos

The greedy algorithm for interval scheduling is shown below.

```plaintext
1    sort the given intervals by increasing finish time
2    for each interval i in sorted order do
3        if i overlaps no interval in A then A := A ∪ {i}
4    return A
```

Following are two ways of proving that this algorithm is correct, i.e., that it returns a maximum cardinality subset of the given set of intervals in which no intervals overlap.

The “greedy-stays-ahead” proof

Consider the set of intervals $A$ constructed by the algorithm. By the test in line 3, this set is feasible: no two intervals in it overlap. Let $j_1, j_2, \ldots, j_k$ be the list of these intervals in left-to-right order (this is well defined since the intervals do not overlap). Note that this is the order in which the algorithm adds these intervals to $A$. Let $A^*$ be any optimal set of intervals and $j^*_1, j^*_2, \ldots, j^*_m$ be the list of intervals in $A^*$ listed left-to-right (this is also well defined, since $A^*$ is certainly feasible and so its intervals do not overlap).

Recall that the start and finish times of interval $j$ are denoted $s(j)$ and $f(j)$, respectively.

**Lemma 1 ("Greedy-stays-ahead" lemma)** For every $t$, $1 \leq t \leq k$, $f(j_t) \leq f(j^*_t)$.

**Proof.** By induction on $t$. The basis $t = 1$ is obvious by the algorithm (the first interval chosen by the algorithm is an interval with minimum finish time).

For the induction step, suppose that $f(j_t) \leq f(j^*_t)$. We will prove that $f(j_{t+1}) \leq f(j^*_{t+1})$. Suppose, for contradiction, that $f(j^*_{t+1}) < f(j_{t+1})$. This means that $j^*_{t+1}$ was considered by the greedy algorithm before $j_{t+1}$. Since $f(j_t) \leq f(j^*_t) \leq s(j^*_{t+1}) < f(j^*_{t+1})$, the interval $j^*_{t+1}$ was considered by the greedy algorithm after $j_t$. So, at the time $j^*_{t+1}$ was considered by the greedy algorithm, it was not added to $A$ (otherwise, it and not $j_{t+1}$ would have been the $(t+1)$-th interval added to $A$). Therefore, $j^*_{t+1}$ starts before $j_t$ finishes. So, $s(j^*_{t+1}) < f(j_t) \leq f(j^*_t) \leq f(j^*_{t+1})$. This means that $j^*_{t+1}$ overlaps $j^*_t$, contradicting that $A^*$ is optimal (and therefore feasible). Therefore $f(j_{t+1}) \leq f(j^*_{t+1})$, which completes the induction step. □

Using this lemma, we can prove that the greedy algorithm is correct.

**Theorem 2** The set of intervals $A$ produced by the greedy algorithm is optimal.

**Proof.** If not, $k < \ell$, and $A^*$ contains an interval $j^*_{k+1}$. By Lemma 1, $f(j_k) \leq f(j^*_k)$. Since $A^*$ is optimal and therefore feasible, $f(j^*_k) \leq s(j^*_{k+1}) < f(j^*_{k+1})$. So (a) the greedy algorithm considers interval $j^*_{k+1}$ after interval $j_k$, and (b) $j^*_{k+1}$ does not overlap any of the intervals of $A$. Thus, the greedy algorithm should add it to $A$, contradicting that $A$ has only $k$ intervals. So $k \geq \ell$, and since $A$ is feasible, it is also optimal. □

---

1 The first inequality is by the induction hypothesis, the second because $A^*$ is optimal and therefore feasible, and the third because every interval starts before it finishes.
The “promising set” proof

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

**Lemma 3 (“Promising set” lemma)** For every iteration \( i \) of the loop there is an optimal set of intervals \( A_i^* \) that contains the set of intervals \( A_i \) in variable \( A \) at the end of that iteration.

**Proof.** By induction on \( i \). The basis \( i = 0 \) 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, \( A_0 \) is empty.

For the induction step, assume (by induction hypothesis) that \( A_i \) is a subset of some optimal set \( A_i^* \) of intervals. We will prove that \( A_{i+1} \) is a subset of some optimal set \( A_{i+1}^* \) of intervals. (Note that \( A_{i+1}^* \) may be a different optimal set than \( A_i^* \).) This is obvious if no interval is added to \( A \) in iteration \( i + 1 \), or if the interval \( j \) added to \( A \) in iteration \( i + 1 \) happens to be in \( A_i^* \): In both of these cases, we take \( A_{i+1}^* = A_i^* \) and we are done. So, suppose that some interval \( j \) is added to \( A \) in iteration \( i + 1 \) and \( j \not\in A_i^* \).

In this case, first we note that \( A_i^* \) contains some interval \( j^* \) that overlaps with \( j \): for, otherwise, we could add \( j \) to \( A_i^* \) and obtain a feasible set with more intervals than \( A_i^* \), contradicting that \( A_i^* \) is optimal. Second, we note that \( A_i^* \) does not contain two intervals that overlap with \( j \): for, otherwise, both of them would start after all the intervals in \( A_i \) finish (because \( A_i \) is, by induction hypothesis, a subset of \( A_i^* \), which is optimal and hence feasible), and one of them, call it \( j' \), would finish before \( j \) finishes; so the greedy algorithm would have added \( j' \), rather than \( j \), to \( A \) in iteration \( i + 1 \). So, there is exactly one interval \( j^* \) in \( A_i^* - A_i \) that overlaps with \( j \). Let \( A_{i+1}^* \) be the set obtained from \( A_i^* \) by replacing \( j^* \) with \( j \): i.e., \( A_{i+1}^* = (A_i^* - \{j^*\}) \cup \{j\} \). The set \( A_{i+1}^* \), (a) is feasible (since \( j^* \) is the only interval in \( A_i^* \) that overlaps with \( j \)); (b) has the same number of intervals as the optimal set \( A_i^* \); and (c) contains \( A_i \cup \{j\} = A_{i+1} \) (by construction). So, \( A_{i+1}^* \) is an optimal set that contains all the intervals in \( A_{i+1} \), as wanted. \( \square \)

We now prove Theorem 2 using Lemma 3 instead of Lemma 1.

**Proof.** By Lemma 3, there is an optimal set \( A^* \supseteq A \). We claim that, in fact, \( A = A^* \). Suppose, for contradiction, that there is some interval \( j \in A^* - A \). Since \( A^* \) is optimal (and hence feasible), \( j \) does not overlap with any interval in \( A^* \) other than itself. Since \( A \subseteq A^* \) and \( j \not\in A \), \( j \) does not overlap with any interval in \( A \). So, when \( j \) was considered for inclusion in \( A \) by the algorithm, it would have been added to it, contradicting that \( j \not\in A \). Thus, \( A = A^* \), and so the greedy algorithm returns an optimal set. \( \square \)

**Remarks on “greedy-stays-ahead”, “promising-set”, and “exchange” proofs**

The “greedy-stays-ahead” and “promising set” arguments both capture the intuition that a greedy algorithm builds an optimal solution incrementally so that the partial solution constructed at each stage is “optimal so far”. In the “greedy-stays-ahead” argument we do induction on the number of “pieces” that make up the optimal solution. In the “promising set” argument we do induction on the number of iterations that the greedy algorithm performs.

The “exchange” (or “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 “exchange” (or “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 its 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.
The promising set argument bears some resemblance to the exchange argument in that the optimal solution that extends the partial solution may need to be changed in some iterations, and the change involves replacing (i.e., exchanging) some elements of an optimal solution with some elements of the solution constructed by the greedy algorithm. For instance, in the preceding proof, we had to change $A_i^*$ to $A_{i+1}^*$ by replacing $j^*$ with $j$. 