Proving Correctness of Greedy Algorithms

COMP3121/9101 21T3

October 7, 2021

This document presents two approaches to prove the correctness of the greedy algorithm presented in lecture 6 for the Activity Selection Problem. These two proofs are written in full detail, using notation to be as precise as possible. Assignment submissions can use worded arguments instead, as long as they are logically sound, clear and convincing.

Problem

Suppose you have a list of n activities, the ith starting at time s_i and finishing at time f_i. A set of activities is said to be *compatible* if no two of them overlap. Find a compatible set of activities which is of maximum size.

Algorithm

Sort the activities in increasing order of finishing time. Then traverse the sorted list, maintaining the finishing time of the most recently selected activity. An activity i is selected if and only if it does not overlap with the most recently selected activity j, i.e. its starting time s_i is later than the finishing time f_j.

Proofs of Correctness

Both proofs will begin by relabelling the activities in order of increasing endpoints, so that $f_1 \leq f_2 \leq \ldots \leq f_n$.

Proof 1: Exchange Argument

Suppose our greedy algorithm takes a set of activities

$$A = \{a_1, a_2, \ldots, a_k\} \text{ where } a_1 < a_2 < \ldots < a_k.$$

Assume for a contradiction that the maximum size of a compatible set is $\ell > k$, and therefore let

$$B = \{b_1, b_2, \ldots, b_\ell\} \text{ where } b_1 < b_2 < \ldots < b_\ell$$

be a compatible set.

Suppose the jth smallest entries of sets A and B are the first place in which they differ, i.e. $a_1 = b_1$, $a_2 = b_2$, $\ldots$, $a_{j-1} = b_{j-1}$ and $a_j \neq b_j$.

Consider $B' = B \cup \{a_j\} \setminus \{b_j\}$. We claim that B' is also a compatible set of size ℓ.

First, we find the size of B'. B' is formed by removing an element from B and replacing it with a_j. To ensure that $|B'| = |B|$ we only need to confirm that $a_j \notin B$. If a_j were to be an element of B, then it must appear after b_j. But then the greedy strategy would have picked b_j as its jth choice before getting to a_j, giving a contradiction.

Now we prove that B' is compatible. We know there are no conflicts in B, so any conflicts in B' would have to be between an existing element b_i and the new element a_j.

Case 1: $i < j$

> This case is trivial; we know that $b_i = a_i$ (as the first $j - 1$ entries of both sets matched), and a_i does not conflict with a_j (since they are both part of the greedy selection A) so b_i and a_j do not conflict.

Case 2: $i > j$

> Activities b_j and b_i do not conflict (since they are both part of the compatible set B) so b_j must finish before b_i starts, i.e. $f_{b_j} < s_{b_i}$.
>
> However, the greedy algorithm chose activity a_j instead of b_j, so a_j must finish no later than b_j. Then $f_{a_j} \leq f_{b_j}$.
>
> Therefore $f_{a_j} < s_{b_i}$ and hence activities a_j and b_i do not conflict.

Therefore B' is a compatible set of size ℓ. We can now compare A to B', in which at least the j smallest entries agree.

Continuing in this way, we can resolve any differences between the greedy selection and an optimal solution, until all k choices made by the greedy selection A also appear as the first k choices of the optimal solution B. Since $|B| = \ell > k$, there is still at least one activity $j = b_{k+1} \in B$ unaccounted for. However, we have one final contradiction; this activity does not conflict with anything in A, so our greedy algorithm would have selected it. Therefore the maximum size of a compatible set is indeed k, i.e. the greedy algorithm is optimal.

Proof 2: Greedy Stays Ahead

Let $m(k)$ be the maximum size of a compatible subset of activities $\{1, 2, \ldots, k\}$. We will prove by induction that our greedy algorithm achieves $m(k)$ for all $k = 1..n$.

Clearly $m(1) = 1$, and the greedy algorithm selects the earliest finishing activity, so the required statement is true for $k = 1$.

Suppose the greedy algorithm achieves $m(1), m(2), \ldots, m(k)$. We will now prove that it also achieves $m(k + 1)$.

First, we show that $m(k + 1)$ equals either $m(k)$ or $m(k) + 1$. Clearly, $m(k+1) \geq m(k)$. It is also true that $m(k+1) \leq m(k)+1$, as a selection from the first $k + 1$ activities can take at most $m(k)$ of the first k and potentially one more (activity $k + 1$).

In the first case, the correctness is self-evident; our greedy algorithm picked $m(k)$ of the first k activities, so the same selection takes the most possible activities out of the first $k+1$. Hereafter we assume that $m(k+1) = m(k)+1$.

<u>Case 1: the greedy algorithm takes activity $k + 1$</u>

> By the assumption, the greedy algorithm took $m(k)$ of the first k activities, so if it also takes activity $k + 1$, we now have $m(k) + 1 = m(k + 1)$ of the first $k + 1$ activities, as required. So in this case, the greedy algorithm successfully achieves $m(k + 1)$.

<u>Case 2: the greedy algorithm does not take activity $k + 1$</u>

> In this case, the greedy algorithm would only take $m(k)$ of the first $k + 1$ activities, rather than $m(k + 1) = m(k) + 1$, so it would not be optimal. Therefore our task is to prove that this case never arises.
>
> Suppose that from the first k activities, the greedy algorithm takes a

set

$$A = \{a_1, a_2, \ldots, a_{m(k)}\} \text{ where } a_1 < a_2 < \ldots < a_{m(k)},$$

and let the last chosen activity be j $(= a_{m(k)})$. We assume in this case that activity $k+1$ was not chosen, so it must overlap with the last chosen activity j, i.e. $f_j \geq s_{k+1}$.

Assume for a contradiction that there is a compatible set of $m(k+1) = m(k)+1$ of the first $k+1$ activities. From the proof that $m(k+1) \leq m(k)+1$, we know that such a set must include activity $k+1$. Therefore let

$$B = \{b_1, b_2, \ldots, b_{m(k)}, b_{m(k)+1}\} \text{ where } b_1 < b_2 < \ldots < b_{m(k)} < b_{m(k)+1}$$

be such a set, and let the last activity $b_{m(k)+1}$ be $k+1$ and the penultimate activity $b_{m(k)}$ be i.

Since set B is compatible, we know that activities $b_{m(k)}$ and $b_{m(k)+1}$ do not conflict, i.e. $f_i < s_k + 1$. We therefore have that $f_i < f_j$, so $i < j$ and hence $i \leq j-1$. If we now consider only the first $j-1$ activities, we see that the greedy strategy took $m(k)-1$ of them, whereas $B \setminus \{k+1\}$ is a compatible set consisting of $m(k)$ of the first $j-1$ activities. But we assumed (in the induction hypothesis) that the greedy algorithm achieved the maximum size of a compatible subset of the first $j-1$ activities. This is a contradiction!

Therefore case 2 never arises; i.e. for all k where $m(k+1) = m(k)+1$ (rather than $m(k)$), the greedy algorithm does indeed take activity $k+1$.

By induction, we see that the greedy algorithm achieves $m(k)$ for all $k \leq n$. In particular, for $k = n$, the greedy algorithm takes the maximum size compatible set out of all activities, i.e. it is optimal.

Proof 3: Greedy Stays Ahead

Algorithm Design by Kleinberg and Tardos includes an alternative proof using the "greedy stays ahead" idea; they prove that the greedy algorithm minimises the finishing time of the kth selected activity for all k.