COMP3121: Assignment 3 – Q4

Gerald Huang

$z5209342$

Updated: July 16, 2020

Given a set of jobs, deadlines and profits, our goal is to construct an algorithm that produces the greatest profit in $O(n^2)$. Since we're attempting to maximise profit, sort the set of jobs by profit keeping note of its deadlines and construct a deadline array that marks all of the times in which we will perform each job. Let a particular job with deadline d_i and profit p_i be denoted as i. Since the job can be completed no later than d_i, we shall look for index i in our deadline array and see if there is already a job there. One of two events may occur:

- **Case 1**: There is a vacant spot. If there is a vacant spot, then we shall simply place the job there.
- **Case 2**: There is already a job at there. If there is already a job at index i, then we shall look for the next best option which is the index $i - 1$. If there is already a job there, then we shall keep looking until we reach the end of the array or a vacant spot. If we reach the end of the array, then we simply ignore the job and continue along the set of jobs.

In each iteration, we check *at most* $(n - 1)$ spots in our deadline array. As such, we'd have to run the deadline check at most $1 + 2 + 3 + \cdots + (n - 1) = \dfrac{n(n-1)}{2} = O\left(n^2\right)$ times. As a result, the algorithm takes $O(n^2)$.

Proof of optimality

We shall prove that this strategy is optimal with a proof by contradiction. That is, assume that there *is* a strategy O that produces more profit than our strategy X does. Further, define jobs i and j with deadlines d_i, d_j and profits $p_i < p_j$ such that job i appears in X and job j appears in O at some point k.

Since j has a greater profit, then it must have been chosen at some point before i. However, assuming that it did not appear in X, then its deadline must not have been successfully found in the deadline array. We consider the three different scenarios between i and j.

- **Case 1**: $d_j < d_i$. Since the deadline of j is less than i and the fact that j does not appear in, this is an immediate contradiction since if job i is chosen by X and job j is discarded by X, that implies that job i must have been considered before j. However, this is not possible since $p_i < p_j$.
- **Case 2**: $d_j = d_i$. Again, we arise at a contradiction since, if the two deadlines are equal, then j is searched first before i. But since j does not appear in X and i appears in X, this, again, implies that X searched for a vacancy for job i first which is not possible.
- **Case 3**: $d_j > d_i$. Since j is searched before i and j does not appear in X, then more profitable jobs have already been placed before j which could have been done before j. Thus, we keep traversing down the array until we arrive at the case where $d_j = d_i$. Since $d_i < d_j$ and i is chosen by X, then this means that there were vacant positions in positions $\leq d_i$. But since job j was searched for first, then strategy X must have found a position for job j. However, since job j is discarded by strategy X, we arise at a contradiction.

Hence, strategy O is *only* as good as strategy X and as such, our strategy is optimal which completes the proof.