Assignment 4

Erik Norlin, CID: norliner

October 8, 2023

Problem 8

Let i be the house the vehicle starts at and visits first, S be the set containing all houses to visit (including house i), t the total time taken to travel between houses in S; initially zero at house i. Then $OPT(i, S, t)$ is the shortest time taken to visit the houses in S such that all houses are visited before their respective deadlines, assuming there exists such a solution (or solutions). Defining the following recursive formula,

$$OPT(i, S, t) = \begin{cases} \infty \text{ if } t > d_i \\ \min_{j \in S, j \neq i}\{t_{ij} + OPT(j,\ S \backslash \{i\},\ t + t_{ij})\} \text{ if } t \leq d_i \\ 0 \text{ if } S = \{i\} \text{ (only house } i \text{ left in the subset) and } t \leq d_i \end{cases}$$

let j be the next house to travel to from i, t_{ij} the time it takes to travel from house i to j, i.e., $|s_i - s_j|$, and d_i the deadline of house i. If t exceeds d_i we set $OPT(i, S, t)$ to a positive infinite time as penalty because no lateness is acceptable. This makes sense because we are dealing with a minimization problem, thus paths with infinite time will never be optimal solutions. On the other hand, if t does not exceed the deadline d_i we continue to search for an optimal solution but now recursively computing the travel time of paths starting from each $j \in S$ such that $j \neq i$. When only one house i is left in the subset and t meets d_i, the optimal travel time is set to zero to indicate a valid solution. When the algorithm is done with all recursive calls, any solution that that equals to zero is a valid solution that visits all houses before their respective deadlines, and any solution that equals to infinity is an invalid solution.

As for the time complexity analysis, memoization is used to avoid computing the same values more than once. There are 2^{n-1} unique subsets given a starting point i and there are n houses to start from, so in total there are $n2^{n-1}$ unique subsets. However, for each unique subset S there can be $|S|!$ different paths and it is not obvious that all of those paths take the same time to traverse. In the worst case, roughly all possible combinations of paths are unique sub-problems and results therefore in $n!$ number of sub-problems, so the upper bound is in $O(n!)$. On the other hand, if we make a simplification and say that all paths in a unique subset take the same amount of time there would be $n2^{n-1}$ unique sub-problems. For each sub-problem n houses have to be visited, so the lower bound is in $\Omega(n^2 2^n)$.