Nicholas Warner
C950 - Algorithms and Datastructures II
4.11.2021

Analysis of Self-Adjusting Algorithm in C950
Parcel Delivery Final Assessment

This submission uses the “Nearest Eligible Neighbor” algorithm to determine the order in which parcels are loaded and delivered for the final assessment. The “Nearest Eligible Neighbor” algorithm can be found predominantly in the load_truck method in controller.py. Assuming all packages to be delivered have been loaded into the application as a set, the pseudocode for this algorithm is as follows:

```
BEGIN

For package in allPackages
        If (package is available AND package is allowed on truck)
                Add package to eligiblePackages

For package in eligiblePackages
        If (package.deadline is within two hours)
                Add package to priorityPackages

While (truck is not full AND eligiblePackages is not empty)
        If (groupPackages is not empty)
                Load the nearest group package onto the truck
        Else If (priorityPackages is not empty)
                Load the nearest priority package onto the truck
        Else
                Load the nearest eligible package onto the truck

        If (the loaded package is part of a group)
                Add each package in the group to groupPackages

        Remove loaded package from all relevant sets

Deliver packages on truck in the order they were loaded

END
```

The algorithm was written in Notepad++ v8.1.5 using python. The space-complexity of the algorithm is

$$V(P + D^2)$$

where P is the number of packages and D is the number of destinations. The complexity of destinations is exponential because each destination uses a matrix to store its distance from every other destination. The time-complexity for the algorithm is

$$O(T * (N + N + CN + 1 + N)) = O(TN)$$

where N is the number of packages, T is the number of trucks, and C is a constant. As T is relatively low compared to N, time-complexity will scale closer to linearly than exponentially. However, the algorithm increases likelihood of meeting package deadlines by giving loading priority to packages with deadlines, so if the number of packages with deadlines greatly increases compared to an increase in the number of trucks, there is a risk some packages will be delivered late.

The software is efficient and easy to maintain because it is a simple greedy algorithm which can flexibly organize any set of packages at runtime.

The strengths of the algorithm include near-linear time complexity, and its simplicity. The disadvantages of the algorithm are the exponential time complexity and the aforementioned risk accompanying a high density of package deadlines.