CS170 Project Final Report

The approach that we used is an approximation algorithm to solve this problem. Inspired by the lecture on 11/10/2019, we think that K-cluster approximation is a good approach to DTH (Drive TAs Home) problem.

There are 3 steps in this algorithm:

1. The first step is to find the cluster centers. We first pick a random home index from the home indices list we get as the first cluster center. Then we implement the cluster approximation method: The second cluster center is the one that is farthest from the first, and the third cluster center should be chosen in a way such that it is the “farthest” from the first two centers (actually it is a max-min process: find the center that has the maximum minimum distance from those first two centers). Then we continue the process using the same logic until all the centers are found.
2. The second step is to find the cluster that each home belongs to. The logic is just to iterate through all the homes and then for each home find the cluster center that is nearest (in terms of shortest path distance) to it. Then we will drop the TAs at the cluster centers that are closest to their homes and let them walk home via the shortest path.
3. The third step is to find a cycle that go through all the cluster centers in the graph. Here we employ a greedy approach. We first find the cluster center that is the nearest from the starting location, then find the next center that is nearest to the last one selected. Then we use networkx package to find the path between cluster centers. In the end, we add the starting location to the end of the path. A cycle is formed.

We initially set k as a fixed number. To improve the algorithm, we decided to tune the number of cluster centers that we want to use. The approach we take is to iterate through the list [1, 9, 17, 25, ..., len(list_of_homes)] and then select the k value with the smallest total cost (The function used for calculating cost is from student_utils.py). In this iterating approach, we select 8 as the step size to maintain a reasonable runtime and performance balance.

Our approximation algorithms achieved reasonably good performance on the input files. One advantage of using this algorithm is that it’s very efficient and requires little computing resources. For example, it took us only several hours to run 949 inputs files using a Mac laptop. One of the drawbacks is that we use a greedy approach to find the cycle that traveling through all the cluster centers, which may not be optimal. Also, we observed that as k increases, we have less total cost, which means that our approximation algorithm tends to travel to as many homes as possible.

If we have more time, we’ll try to find the cycle that traveling through all the cluster centers using some TSP solvers instead of taking a greedy approach.