Project Design

Zhiyi Li, Yao Shao, Huben Liu

November 2019

There are several ideas we came up with and have tried or will try. And these are only preliminary ideas and We will update our algorithm further later.

1 TSP Approximation and Further Optimization

First use Floyd algorithm to find the shortest path between any pair of vertices. Then let the car driving along all TAs homes. Run simulated annealing for TSP on all homes. This give the initial cost C_0.

Then we delete one home v_i from the car route, fixing other homes and find all possible alternate drop off place for the TA whose home is v_i. If we find a better solution than C_0, update the route. Then check remaining homes one by one based on the update route until all homes are checked. Then check every two adjacent homes. Then check every there. Finally get to the case for driving along no homes and letting all TAs walk home (the case driving to a place and then drop all TAs and let all of them walk home is considered in this case). Report the best solution we have ever tried.

The idea comes from two extreme cases: driving all TAs home and letting all of them walk home. First, we find an approximate solution for the case of driving all TAs home, reducing it to TSP problem. Then we try to find better solutions step by step, from letting one TA walk home (all cases) to letting all TAs walk home, leading to another extreme case. Finally, the algorithm will return the optimal case we have every tried. In addition, we may also try the inverse order, from letting all TAs walk home to driving all TAs home by adding homes to the route.

Although there may be some cases we ignored, we have checked most of the cases which have high probability to be optimal. And the simulated annealing used to find TSP performs better.

We have tried the idea and it turns out to perform well.

2 Greedy Approach

We can also use the greedy approach to get a TSP solution in part one by going to the nearest TAs home every time. It is worth trying just because it may give a better solution in case.

3 K cluster

We can assume all dropping off points as centers of the clusters. And TAs whose homes are in the same cluster have the same place to get off the car at the center of the cluster. The cost function we use is the sum of the distance between all homes and their center plus the 2/3* travel distance of the car from the starting point through all centers and bank to the starting point. We need to try different number of K because we don't know how many drop-offs the optimal solution has.

The idea comes from the intuitive understand that all those TAs living nearly may get off the car at the same place. Because we will try all possible K, it may give us solution better and better.