CS 221 P3 Progress Report

Jaclyn Chen

Mavis Li

Zahan Malkani

Problem Setup

Framework

Given a group of n people at a starting location s_i and the car capacity c_i (number of seats they are providing, i.e. 4 or 0), and the destination location Goal.

Objectives

Minimize total commute time

Output

Give assignment of cars to people and time to meet at starting location

Assumptions

- Everyone wants to go to the same destination
- Car capacity is set at 4, else the person does not have a car

---

Implementation

We are using Python, with the NetworkX library for graph data storage and manipulation.

Data Generation

We randomly generate our data to test our algorithms by assuming an m x m grid in the world. We sample from those k+1 of those grid points uniformly, taking the first k points as the locations of people and the last grid point as the goal location. From those k locations, we select c of them to have cars. The c we choose will be greater than k/4 and less than or equal to k.

The input to our algorithms will also include distances between all k+1 locations, which is just be the euclidean distance between locations on our grid.

Baseline Algorithm

For simplicity, we assume everyone with a car is driving. We will try to minimize the sum of total distances traveled by all cars. First we use an exhaustive search to get the absolute minimum. This will provide a basis for other heuristic algorithms we try later. The algorithm consists of two steps. In the first step, we assign people to cars, with car capacities as a constraint. Then, we calculate the minimum cost path of starting at where the car is, picking up all assigned people, and then going to the goal state. We consider all possible assignment of people to cars and look for the minimum total distance.

Other Algorithms (Domain Specific)

Since the permutation of all possible assignments of people to cars and the calculation of the minimum cost path is very computationally expensive, we will explore some heuristic algorithms on the same datasets to see how they perform.

On the Way Projection

We will draw the path of the starting location of a car to the goal, and determine the orthogonal projections of all non-car locations onto the path. Picking someone up can be considered a detour along the straight line path for the car. For non-car people who are not really along the way (no projection exists), we use the distance from their location to the car as the projection distance. Then we assign people to a car starting with the shortest projection distances until a car is filled up.

Agglomerative approach

Then we will try an approach which involves pairing the closest locations together. At each step we will pair together a non-car location with a set of points containing a car location until the car is full.
