README.md - Grip

Training Instances and Grouth Truths

These instructions are pertinent to the **programming project** of IS421: Enterprise Analytics for Decision Support, Semester 2, 2016.

This folder contains the following:

- Set of **training instances**: 5 for part (a) and 5 for part (b);
- Set of ground truth **output paths** for each instance, for each part;
- Set of ground truth **objective values**: minimum total time for part (a) and average wait time for part (b).

For the IO format of the input instances and output paths, refer to the online side deck.

Training instances

Each instance is an input text file that starts with the prefix `sin_train_` and followed by the number of vehicles (taxis) and then the number of OD pairs. For part (a), the number of vehicles is equal to the number of OD pairs. For part (b), the number of vehicles is strictly less than the number of OD pairs. For example, input file `sin_train_5_5.txt` is for part (a) because there are equal number (5) of vehicles and OD pairs. Whereas, the input file `sin_train_5_6.txt` is meant for part (b) since there are more OD pairs (6) than vehicles (5).

Output paths

Similar to the training instances, each output CSV file starts with the prefix `path_` and then followed by the number of vehicles and OD pairs. Therefore, distinguishing between the output of part (a) and (b) is exactly the same as above (for training instances). Notice that each CSV file contains no headers, and each is a sequence of edges taken to fetch and transport the passengers. These output paths are to be used for the Routing App v2 for visualization and evaluation purposes.

Objective values

The objective values are contained in two CSV files: `results_a.csv` for part (a) and `results_b.csv` for part (b). For part (a), our objective is to minimize the total (wait + travel) time of all passengers; while for part (b), the objective is to minimize the average wait of the passengers. Each CSV file contains the values for each of the respective 5 instances.

For part (a), the values represent the absolute optimal objective values obtained from optimization models. These serve as ground truths for you to benchmark against your implementations. Therefore, your obtained objective values should ideally be close to these numbers, and should not be significantly lower (since the minima are absolute) or higher. That having said, due to rounding errors and various implementation quirks of the shortest-path algorithm, and precise formulations of the optimization models, your obtained objective values may be lower (or larger) than these, but should not by significant margins if your implementations are correct. In other words, these values serve as the *lower bounds*.

For part (b), the heuristic used is *random assignment*, which is a pretty naive heuristic. Thus, the objective values here are served as the *upper bounds*, and your proposed heuristic should ideally improve on these. In other words, your proposed heuristic should produce objective values not higher than these, and should ideally be significantly lower.