Singapore Management University
School of Information Systems
2015/2016 Semester 1
IS703 – Decision Analytics & Optimization
Programming Project

Due: 27 November 2015

1 Description

In this project, you are given three datasets describing the road networks of three global cities: Pittsburgh, P.A. (PGH), Washington, D.C. (WAS), and Singapore (SIN). The three datasets can be downloaded from: `http://git.io/vcYeN`. Each road network is a graph $G = (V, E)$, where each node $v \in V$ is an intersection (or midsection of a road) and each edge $e \in E$ is a road segment (aka "link"). Each dataset is in an XML-based format specifically suited for network data called GraphML (`http://graphml.graphdrawing.org`). GraphML can be easily read using standard network analysis packages such as `igraph` (`http://igraph.org/redirect.html`) and `Gephi` (`https://gephi.github.io`). An online tutorial (`http://vietletruc.com/notes-on-programming-assignment`) has been written on how to read the road networks and perform basic shortest path routing using Python to give you a head start on the project. Each edge has attributes such as: segment length, street name, number of lanes, road type, speed distribution (mean and variance) or speed limit (called "max_speed"). Depending which city, the attributes may vary. Consult the attached `README` file for the attribute list of each city.

This project explores an important and well-studied topic in transportation planning: routing. We are given N vehicles and K ($K \geq N$) passengers. Each passenger has a given travel demand, where each demand is given by a pair of nodes (called an OD pair): origin (the passenger's current location) and destination. Imagine you are the operator of the fleet of N identical and cooperative vehicles located at their given current locations. Your job is to route the fleet of vehicles to fetch the passengers and transport them to their respective destinations. You are to design and implement a program that can perform the following functions.

(a) [All-or-nothing] Let $K = N$ and suppose that each vehicle can travel at the "free-flow" speed, i.e. the speed limit of a segment or the segment's average speed + 2 standard deviations (whichever is available in the dataset). Each vehicle serves exactly one demand. You should use a shortest path algorithm (e.g., Dijkstra's, A*, etc.) to route each vehicle from its current location to a passenger's demand location and then to their respective destination. You should output the paths (i.e. sequences of nodes/edges) taken to satisfy all demands. Details of the output format will be provided at a later point.

Our objective is to minimize the *total* travel time. For each trip, the time taken by each vehicle is sum of the time traveled to fetch its passenger plus the travel time to destination. For simplicity, we assume throughout that all demands start at the same time (say time zero), when all vehicles are currently free. Denote $T(x_j)$ as the travel time for segment $x_j \in E$ (which is the segment length divided by the segment's free-flow speed). Our objective function can be written as:

$$\min_x S(x) = \sum_{i=1}^{N} \sum_{x_j \in P_i} T(x_j), \tag{1}$$

where P_i is the sequence of edges traversed to fetch passenger i at the origin and transport them to their destination.

Your task is to solve the All-or-nothing routing problem described above, and run your program on a given set of test instances (to be revealed later).

(b) [Incremental Assignment] In reality, vehicles normally do not travel at the free-flow speed due to traffic congestion. Thus, to model congestion effects, we make use of the formula proposed by the Bureau of Public Roads (BPR) [`http://www.fhwa.dot.gov`]:

$$T(v) = T_0(1 + 0.15(\frac{v}{c})^4), \tag{2}$$

where:

- v – the volume of traffic over a particular segment. In this assignment, for simplicity, we will use the number of vehicles routed over a segment to be its volume.
- c – the segment's capacity. Our data does not capture this information, but we can approximate it using the square root of the segment's max_speed (or mean speed + 2 standard deviations) rounded to the nearest integer.
- T_0 – the segment's free-flow travel time;
- $T(v)$ – the segment's estimated travel time under traffic volume v.

Still assuming $K = N$, we will use a heuristic called the Incremental Assignment to simulate congestion and real-world route selection. The heuristic can be briefly described as follows. We randomly split the set of vehicles into 4 partitions: 40%, 30%, 20%, and 10% of the size N. Each partition represents an iteration of the algorithm. At each iteration, we perform the All-or-nothing routing for each partition as in (a). At the end of each iteration, we update the current travel time over a routed link according to the BPR formula (2) using the *cumulated* volume over the link thus far. Hence, at the first iteration, vehicles are allowed to travel at free-flow speeds. In later iterations, speeds (and travel times) are proportionally reduced over the routed links. The final iteration reflects the final travel time over a link for all the routed vehicles.

Same as part (a), your are to implement the Incremental Assignment heuristic and run experiments on the given instances. Furthermore, you are to compare the results (the average total time per passenger and the routed paths) with those obtained from part (a).

(c) [Ridesharing] In this scenario, let $K > N$ and suppose that each vehicle can transport **up to** Q ($1 \leq Q \leq 4$) passengers at a time while each passenger still has a unique OD pair. This is called the *ridesharing* scheme. Design a "reasonable" mechanism for ridesharing and justify your proposed mechanism. For example, a vehicle may pick up an additional passenger only if their origin happens to be "near" to the route that serves the current demand(s). How near is near and how close the new route should be from the original route in order to share the ride is up to you to decide and justify. Furthermore, we have an additional requirement that **all K demands are to be satisfied**. This means that a vehicle may possibly make multiple trips in order to serve all demands. In this part, we wish to optimize the objective function given by Eqn. (1) using the free-flow speeds, i.e. you need not worry about congestion here.

Compare the results obtained with those in part (a) and perform a sensitivity analysis to show the effects of varying $Q \in [1, 4]$ on the objective function. (Obviously, $Q = 1$ means no sharing.)

(d) [Starvation] Still keeping $K > N$, propose another ridesharing mechanism with the new objective to minimize the maximum *waiting time* of all the passengers. We are still making sure that all demands are served as in (c). The waiting time of a passenger is defined as the time from when their request is made (i.e. time zero) until when the passenger is picked up by a vehicle. Again, use the free-flow speeds and experiment with your proposed mechanism for the given test instances and compare the results with those in part (c).

2 Evaluation

This project has two deliverables:

(1) Presentation (10%): Each team will make a 20-minute presentation in class on 20 Nov. 2015 (during Week 14's class).

(2) Report and System (30%): Due on 27 Nov. 2015 (23:59 hours). Each team is required to submit: (a) A hardcopy of your report (limit to 10 pages). This hardcopy should be delivered either in the mailbox of Prof. Lau (in the SIS General Office) or slipped under the door of his office (if he is not around). (b) A zipped folder in the eLearn course Dropbox comprising:

- Project report (in MS Word or LaTeXformat – same content as (2. a) above);
- Presentation slides (in PowerPoint format – revised content from (1) above);
- Source code containing the model and algorithms; data files and executable system.

On Presentation: You are expected to present:

- Your algorithm design;
- A live demo of your system;
- A summary of results obtained from running your system on given test cases.

Grading criteria:

- Quality of your slides (5%)
- Your presentation and response to questions (5%)

On Report and System: The report should contain the following: problem definition, proposed algorithm and complexity analysis, implementation approach (e.g. data structures used), summary of experimental results and instructions for installing/running your system.

Grading criteria:

- Content of report (10%)

- Quality of writing (10%)
- System implementation (10%)

References

[1] Jacques A Ferland, Michael Florian, and Claude Achim. On incremental methods for traffic assignment. *Transportation Research*, 9(4):237–239, 1975.

[2] Douglas O Santos and Eduardo C Xavier. Dynamic taxi and ridesharing: A framework and heuristics for the optimization problem. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2885–2891. AAAI Press, 2013.