Multi-Class Congestion-Aware Routing for on-demand Service

Semester 8 Project Thesis

By: Niharika Shrivastava (IIT2016501)
Supervisor: Prof. Malika Meghjani
Co-Supervisor: Prof. O.P. Vyas

Department of Information Technology
Indian Institute of Information Technology
Allahabad, India

April 1, 2020

Candidate's Declaration

We declare that the project work titled "Multi-Class Congestion-Aware Routing" done at Singapore University of Technology and Design and submitted at Indian Institute of Information Technology, Allahabad is the bonafide work of Niharika Shrivastava(IIT2016501). It is a genuine record of our study carried out from January 2020 till present under the guidance of Prof. Malika Meghjani and Prof O.P. Vyas. Due acknowledgements have been made in the text to all the materials used.

Supervisor:

....

Prof. Malika Meghjani

Co-Supervisor:

....

Prof. O.P Vyas

Date: xx/xx/xxxx

Place: Allahabad

Abstract

Recent developments in urban mobility have demonstrated the potential capacity of road networks to accommodate increasing traffic demand. This paper addresses the problem of dynamic customer routing in a congested network. We provide congestion-aware routing for mobility-on-demand services that operate using a combination of heterogeneous multi-class fleet (walking, cycling, cars, public transport). Our aim is to provide optimal transit points from one mile to another for this service within a capacity-bound transportation network, where congestion might disrupt throughput. Our solution accommodates real-time traffic congestion and is computationally more efficient and accurate compared to state-of-the-art methods. <This is summary of numerical experiments> <this is limitation>

Contents

1 Introduction

Traffic congestion on urban road networks has become increasingly problematic since the 1950s [1]. With an increase in vehicles on the road and capacity of the roads remaining approximately constant, the speed of the traffic stream slows. This results in higher travel times and increased pollution. With the increasing popularity of on-demand mobility services [2, 3, 4] like Uber, Grab, Yulu and Lyft, the traffic demand has started to reach road capacities more often, thereby enabling congestion to set in. Many works have shown that autonomous vehicles may be able to drive faster and follow other vehicles at closer distances without compromising safety, thereby effectively increasing the capacity of a road and reducing congestion [5, 6]. Others have suggested ride-sharing services such as UberPool and LyftLine, or public transport to lessen the traffic burden of the [7, 8, 9]. Rerouting empty vehicles to potential regions of higher customer requests based on historical data has also shown to lessen region-wise network congestion [10, 11].

Multi-class fleets [12] allow breaking a customer trip into 3 classes: first mile, middle mile and last mile. The customer can choose to either walk on foot on pedestrian paths, use dedicated cycle-ways, or on-demand scooters for their first and last mile. The middle mile will cover up to 90% of the customer trip using the main road networks of a city and make use of fast-speed cars or public transports as per convenience. By using all three classes in an optimal combination, customers can maneuver through crowded paths and congested roads with ease. This also gives customers flexibility in choosing their preferred mode of transport based on accessibility and cost.

In this paper, we suggest ways to integrate multi-class fleets with on-demand services and intelligently route them based on dynamic congestion-flow information for the city of Singapore.

2 Literature Review

- In [7], car-pooling with 2-3 riders per vehicle was introduced in order to serve more customers with lesser vehicles on the road. [8] presented a more general model for real-time high-capacity ride-sharing that had rider-capacity of up to 10 simultaneous customers per vehicle. Their results showed that 98% of the taxi rides currently served by over 13,000 taxis (of capacity one) could be served with just 3,000 taxis (of capacity four). This can result in greater traffic throughput for increasing demand or lesser congestion for constant demand.

- [9] computed future demands of passenger requests, based on historical data. The predictions improved the positioning of the vehicles towards satisfying future requests, reduced waiting and travel time. This lessened region-wise congestion by performing re-balancing of the network. However, seasonal changes was not taken into account while predicting future requests.

- [13] presented a network flow model of an AMoD system on a capacitated road network. It proved that it's always feasible to have optimal customer flows and re-balancing flows together without any traffic congestion. Congestion-flow information was calculated using the Bureau of Public Roads heuristic [14] for every edge of the network. Vehicles were then greedily routed event-based to least congested paths using the A* algorithm.

- In [10], the network is optimally partitioned into re-balancing regions. Real-time demand estimate for every region is determined using incoming requests based on which idle vehicles are optimally assigned to these regions. There was a reduction in the average travel delay by 86%, the average waiting time by 37%, and the amount of ignored requests by 95% compared to [8] at the expense of an increased distance travelled by the fleet. The algorithm could not incorporate the existing public transportation infrastructure.

- In [15], ride-sharing was explored in terms of how many vehicles were needed, where they should be initialized, and how they should be routed to service all the demand. There was a reduction in the fleet size by 69% and travel delay of 1.7 mins (for up to 2 customers per vehicle) and by 77% and 2.8 mins (for up to 4 passengers per vehicle). However, it's computationally expensive for online adaptation.

- In [12], multi-class fleets were introduced that serviced a request using a combination of heterogeneous vehicles (cars, buggies, scooter, walking).

Optimal fleet size and vehicle assignment was computed using the genetic algorithm. However, when the demand is much greater than the expected demand used to optimize the fleet size, the total average travel time for multi-class is worse than that of single-class.

- In [16], shortest distances in a network are found using a hybrid algorithm that uses euclidean distances as a base line. The time complexity for this is $O(n^2)$. It's highly useful in cases of large cities where total number of routes are huge.

3 Proposed Methodology

3.1 Timeline

Timeline	Tasks
15 - 31 January	Study the previous work on ride-sharing, congestion-management, fleet sizing, assignment and re-balancing.
1 - 14 February	Build a road network from existing opensource GIS (e.g. open street maps)
15 - 29 February	Develop tools for querying APIs for public transport schedules and dynamic congestion flow information.
1 - 14 March	Integrate existing tools for graph queries and path planning algorithms.
15 March - 30 June	Develop novel algorithms for congestion-aware routing of multi-class fleet.

3.2 Method Overview

3.2.1 Phase 1

1. Setting up of Singapore city's road network (GIS) consisting of pedestrian paths, cycle-ways, and main roads for driving using Open Street Maps.
2. Collection of Traffic Speed Band dataset using LTA Datamall, an API for retrieving static and dynamic network information of Singapore.

3.2.2 Phase 2

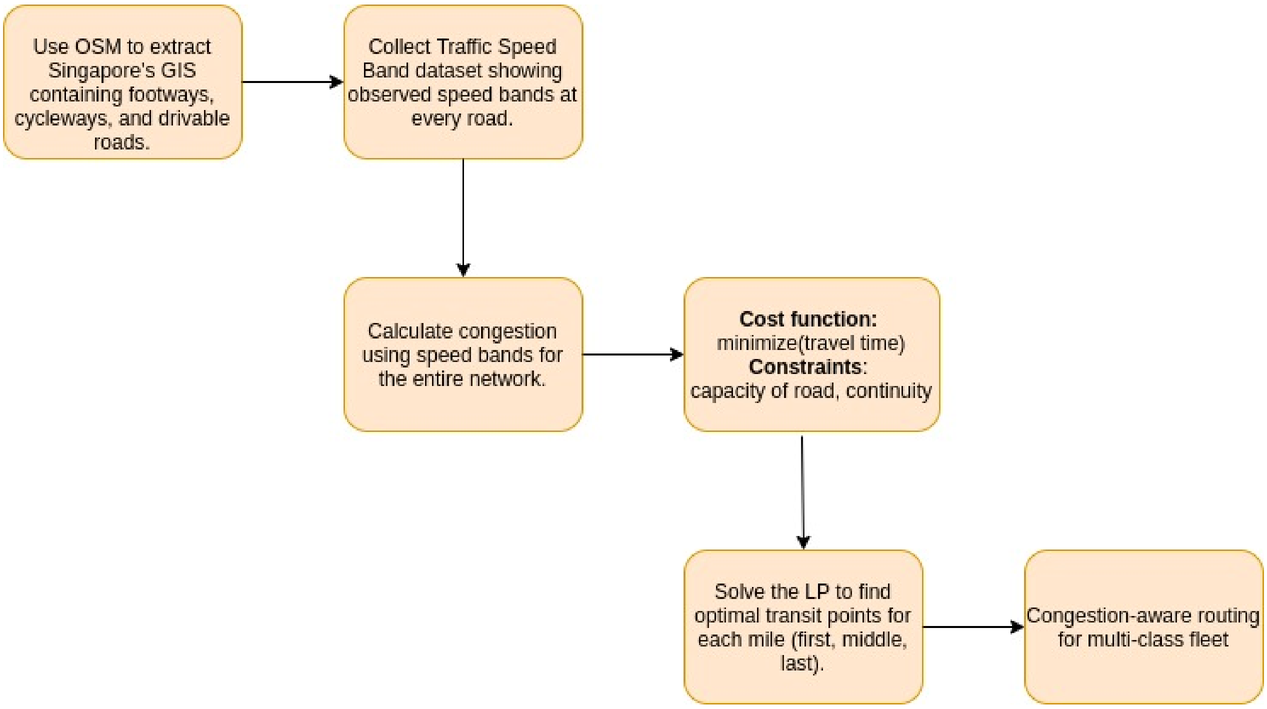
1. Integrate the traffic speed band dataset with our OSM GIS and calculate congestion information for the entire network, time-wise.
2. Formulate a linear programming problem (LP) that minimizes travel times for each customer request while satisfying capacity constraints of the road network.

3.2.3 Phase 3

1. Solve the LP to find optimal transit points for each mile (first, middle, last) for every customer request such that we avoid the onset of congestion.

2. Route the customer request on the least congested path for a multi-class fleet using these optimal transit points.

Figure 1: Project life cycle



4 Requirements

4.1 Data set

1. **Singapore's GIS:** Open Street Map (OSM) was used to extract Singapore's geographic information system (GIS) using the Overpass API.

   (a) The network is tagged with different types of roads or edges like pedestrian, cycle-ways, and drive-able routes. They constitute of corridors, cycle lanes, highways, service roads, etc.

   (b) The network is tagged with different types of nodes like bus stops, elevators, residential areas, parking lots, motorway junctions, etc.

   (c) Routing is possible from one node to another using a series of edges.

2. **Traffic Speed Band Data set:** It was collected from Land Transport Authority (LTA) Datamall owned by the Singapore Government.

   (a) It is a dynamic data set that updates itself every five minutes.

   (b) It shows speed bands (maximum and minimum observed speed) for every road link in Singapore giving real-time information about congestion on that link.

   (c) It was collected for two weeks including weekends, for time intervals of 09:00 A.M., 12:00 P.M., 03:00 P.M., 06:00 P.M., 09:00 P.M. .

4.2 Hardware and Software Requirements

1. A GPU with 12 GB RAM for fast computation .

2. **OSMnx 0.11.4:** Used to download, model, project, visualize, and analyze complex street networks from OpenStreetMap.

3. **NetworkX 2.4:** Used for path planning, routing and studying the structure and functions of Singapore's complex networks.

4. **Rtree 0.9.4:** Provided advanced spatial indexing features like Nearest neighbors search and Intersection search.

5. **IBM ILOG CPLEX or Scipy/linprog:** It solves very large linear programming problems. The Concert technology supports C++.

6. **Python 3.6:** Interpreter language on top of which all these libraries would function.

5 Implementation

1. Extract Singapore's GIS using the place name as "Singapore, Central, Singapore" and integrate it with the traffic speed band data set for one slice of time, e.g., 6:00 P.M. The entire Singapore network can be visualized as a densely-connected directed graph $G(V, E)$ with V vertices (nodes) and E edges.

2. Calculate congestion information for the entire network. Congestion information is calculated in terms of how much travel time is required to cross a road link in the network.

   (a) If vehicles present on a road link (u, v) are within the link capacity $c(u, v)$, they are assumed to be operating at free flow speed. Thus, the time taken for vehicles to cross a road link at free flow speed is called the free flow time $t(u, v)$. However, if vehicles present on a road link exceed the link capacity, congestion starts to set in. This lowers the the speed at which they are operating and hence, travel time to cross that link increases.

   (b) The travel time $t_d(u, v)$ along each edge is computed using a heuristic delay function that is related to the current volume of traffic on each edge. We use the Bureau of Public Roads (BPR) delay model [14], which computes the travel time on each edge (u, v) $\in$ edges (E) as:

   $$t_d(u, v) := t(u, v) \left(1 + \alpha \left(\frac{f(u, v)}{c(u, v)} \right)^{\beta} \right)$$

   where $f(u, v) := \sum_{m=1}^{M} f_m(u, v)$ is the total flow on edge (u, v), and α and β are usually set to 0.15 and 4 respectively.

3. In order to find least congested paths in the entire network and minimize customer travel times for each request, we formulate a linear programming problem. The objective or cost function is to minimize the travel times of every customer request $m \in M$, where $m := (start, target)$. The capacity of the roads is used as a constraint for the upper bound on number of vehicles that can be at an edge E at the same time. Hence, the LP problem formulates as:

$$
\begin{aligned}
\text{minimize:} \quad & \sum_{m\in M}\sum_{(u,v)\in E} t(u,v) f_m(u,v) & \\
\text{subject to:} \quad & \sum_{u\in V} f_m(u,s_m) + \lambda_m = \sum_{w\in V} f_m(s_m,w), & \forall m \in M \\
& \sum_{u\in V} f_m(u,t_m) = \lambda_m + \sum_{w\in V} f_m(t_m,w), & \forall m \in M \\
& \sum_{u\in V} f_m(u,v) = \sum_{w\in V} f_m(v,w), & \forall m \in M, v \in V\backslash\{s_m,t_m\} \\
& \sum_{m\in M} f_m(u,v) \le c(u,v), & \forall (u,v) \in E
\end{aligned}
$$

Constraints (2), (3) and (4) enforce continuity of each trip (i.e., flow conservation) across nodes. Finally, constraint (5) enforces the capacity constraint on each link.

4. Solve the LP to find feasible integral customer flows on congestion-free road links. It is to note that these customer flows belong solely on the drive-able network, since congestion is being monitored only for vehicles and public transport. These flows can be then decomposed into a flow routing algorithm.

5. Find optimal transit points for each class of fleet (first, middle, last) in order to get a completely congestion-free route. Each customer request is composed of multiple legs served by different transportation modes. For example, first mile cycling, middle mile in a car, and last mile walking. Thus, each customer trip would be like this:

$$\mathbf{X} \longrightarrow \mathbf{A} \longrightarrow \mathbf{B} \longrightarrow \mathbf{C} \longrightarrow \mathbf{D} \longrightarrow \mathbf{Y}$$

X: Starting coordinates (x,y) of customer request
A: Nearest known node to X, start node of the first mile
B: Start node of the middle mile
C: End node of the middle mile, start node of the last mile
D: End node of the last mile, nearest known node to Y
Y: Target coordinates (x,y) of customer request

X and **Y** are constant for every customer request. Therefore, we only need to compute **A**, **B**, **C**, and **D**. In order to exercise maximum flexibility in choosing our combination of transit points while minimizing the travel times for every mile, we first route for the middle mile, i.e., ($\mathbf{B} \longrightarrow \mathbf{C}$)

(a) To find an optimal starting **B** and ending node **C** for the middle mile, an assumption cum flexibility for the customer is made. According to it, all nodes within a 720 metre radius (euclidean distance) from the customer's starting coordinates **X**, are considered as potential sources and targets for the middle mile. This means that a customer is allowed to walk or cycle to/from their middle mile endpoints.

$$PotentialSources(B) = \{s_1, s_2, \ldots, s_n\}$$

$$PotentialTargets(C) = \{t_1, t_2, \ldots, t_m\}$$

A total of $m * n$ combinations of {source, target} pairs are made.

(b) We use euclidean lengths as base distances for comparison with real lengths. Since our parameter is travel time, we calculate the euclidean length between each {source, target} pair and divide it by the permissible average speed of vehicles in Singapore (50 km/hr).

(c) To calculate the real-time travel time of each {source, target} pair route, we use weighted Dijkstra with edge weights as the Bureau of Public Roads delay heuristic. This outputs a route between the {source, target} pair with the least travel time, given real-time congestion information.

(d) A hybrid algorithm [16] using the euclidean distances and weighted Dijkstra is used to greedily, yet efficiently compute a {source, target} pair that yields a route with the least travel time in a congested network. This {source, target} pair corresponds to **B** and **C** respectively.

To compute the first and last mile routes, only pedestrian roads and cycleways are used for routing. The customer is expected to travel these routes using either footways, cycleways, or on-demand scooters (like Yulu). These distances are calculated keeping in mind that a customer may want to (i) walk for a maximum of 10 minutes or 720 metres (assuming walking speed as 1.2 m/s), or (ii) cycle for a maximum of 15 minutes or 2 Km (assuming cycling speed as 5.4 m/s).

(a) For the first mile, we select the nearest node from **X**. This node is **A**. The customer is expected to reach from $\mathbf{X} \longrightarrow \mathbf{A}$ in order to start their first mile journey. A further route from **A** to **B** is calculated using weighted-Dijkstra using edge weights as length of the paths.

(b) For the last mile, we select the nearest node from **Y**. This node is **D**. A further route from **C** to **D** is calculated using weighted-Dijkstra using edge weights as length of the paths. The customer is expected to reach from **D** $\longrightarrow$ **Y** in order to finish their entire trip.

6 Results

1. Write how Ford-Fulkerson and A* are base algorithms used for comparison. Write SSP.

2. Variance between eucli and dijkstra. Variance between exhaustive and hybrid.

7 Conclusion

8 Future Scope

...

References

[1] R. W. Caves, "Encyclopedia of the city," *Routledge p.141*, 2004.

[2] W. Mitchell, B. Borroni, and L. Burns, "Reinventing the automobile: Personal urban mobility for the 21st century," *MIT Press*, 2010.

[3] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone, "Toward a systematic approach to the design and evaluation of autonomous mobility-on-demand systems: A case study in singapore. in: Road vehicle automation," *Springer International Publishing, Cham, Switzerland*, 2014.

[4] M. Pavone, S. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for mobility-on-demand systems," *Int Journal of Robotics Research*, 2012.

[5] J. Pérez, F. Seco, V. Milanés, A. Jiménez, J. Diaz, and T. De Pedro, "An rfid-based intelligent vehicle speed controller using active traffic signals.," *Sensors 10(6):5872–5887*, 2010.

[6] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: a queuing-theoretical perspective," *Proceedings of Robotics: Science and Systems Con- ference, Berkeley, CA*, 2014.

[7] P. Santi, "Quantifying the benefits of vehicle pooling with shareable networks," *Proc Natl Acad Sci USA 111(37):13290–13294*, 2014.

[8] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences of the United States of America*, 2017.

[9] J. Alonso-Mora, A. Wallar, and D. Rus, "Predictive routing for autonomous mobility-on-demand systems with ride-sharing," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[10] A. Wallar, J. Alonso-Mora, D. Rus, and M. van der Zee, "Vehicle rebalancing for mobility-on-demand systems with ride-sharing," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[11] K. Spieser, S. Samaranayake, W. Gruel, and E. Frazzoli, "Shared-vehicle mobility-on-demand systems: A fleet operator's guide to re-balancing empty

vehicles," *Transportation Research Board 95th Annual Meeting, January Washington, DC*, 2016.

[12] M. Meghjani, S. D. Pendleton, K. Anna, Marczuk, Y. Hong Eng, X. Shen, M. H, J. Ang, and D. Rus, "Multi-class fleet sizing and mobility on demand service," *Comples Systems Design Management Asia*, 2018.

[13] F. Rossi, R. Zhang, Y. Hindy, and M. Pavone, "Routing autonomous vehicles in congested transportation networks: structural properties and coordination algorithms," *Springer Link: Autonomous Robots (May)*, 2018.

[14] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems, Science, and Cybernetics*, 1968.

[15] A. Wallar, J. Alonso-Mora, and D. Rus, "Optimizing vehicle distributions and fleet sizes for mobility-on-demand," *International Conference on Robotics and Automation (ICRA)*, 2019.

[16] M. Meghjani, S. Manjanna, and G. Dudek, "Fast and efficient rendezvous in street networks," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.