Congestion-Aware Routing for
Multi-Class Mobility-on-Demand Service

Niharika Shrivastava*[1] and Malika Meghjani*[2]

***Abstract*— Urban mobility solutions such as mobility-on-demand services have become prevalent given the convenience of door-to-door transport. However, a majority of these approaches are user-centric greedy solutions that cause traffic congestion. We propose a near social-optimal routing algorithm which accounts for the overall network traffic congestion. Specifically, we leverage on multi-class mobility options to dissipate traffic congestion while maintaining near social optimal travel time efficiency. We divide each route into three parts with micro-mobility options such as walking or cycling for the first and last parts and on-demand cars for the middle part of the route. In addition, we propose a computational and travel time efficient transit point search algorithm for switching between different modes of travel. We validate our approach by using a diverse set of road networks from different cities. We achieve an average of 84% increase in network utilization by using our proposed multi-class social model compared to single-class user-centric approach. Our proposed transit point search algorithm is on average 68% more computationally efficient with an insignificant maximum average travel time delay of less than 5 seconds compared to an optimal exhaustive routing solution.**

I. INTRODUCTION

Traffic congestion on urban road networks has been a persistent problem since the 1950s [1]. The growing popularity of on-demand mobility services like Uber, Grab, and Lyft [2], [3], [4], have further aggravated the traffic by reaching road capacities faster. A potential reason is that the passengers inherently opt for the shortest route in the network. Thus, resulting in congested roads [5] and under-utilization of the city's road networks.

Many countries counter traffic congestion by pedestrianization of streets. For example, in the UK, they reduced congestion by creating "out of town" commercial areas and pedestrianizing of city centres, thereby reducing drive-able routes in the most populated areas [6]. The pedestrianization solutions are further augmented with multi-class mobility-on-demand solutions for better utilization of network resources. The concept of multi-class fleets introduced in [7] allowed customer journeys to be broken down into separate parts for greater accessibility and increased service coverage. Specifically, the customers are recommended to either walk or use on-demand scooters on dedicated cycle-ways [8] for the first and last parts of their routes, to minimize their overall travel time. The middle part of the route covers a majority of the customer trip which uses the main road network of a city and makes use of fast-speed cars or public transport. However, this approach provides user-centric shortest path routes without accounting for congestion.

*Authors contributed equally.

[1]Niharika is affiliated with Indian Institute of Information Technology, Allahabad, India iit2016501@iiita.ac.in

[2]Malika is affiliated with Singapore University of Technology and Design, Singapore malika_meghjani@sutd.edu.sg

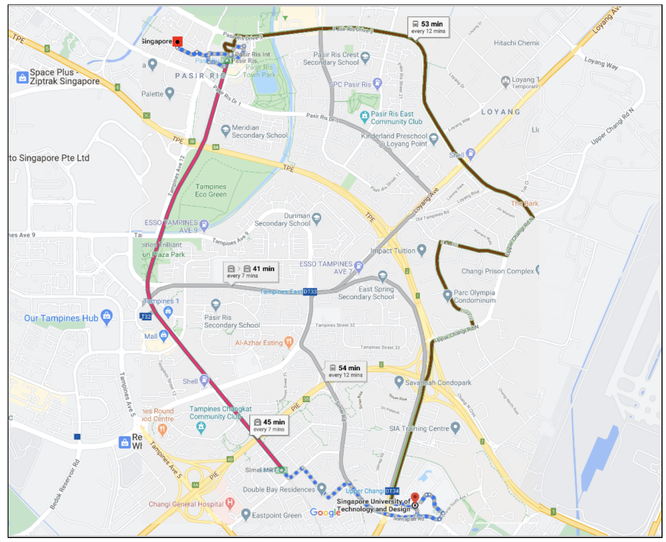

Fig. 1: Multi-class routing solution obtained using our system-centric congestion-aware algorithm (left) and Google Maps © (right). The latter suggests 21 minutes of walking with the total trip time as 53 minutes. Whereas, the path recommended by our congestion-aware multi-class approach suggests a reduction in the total travel time to 45 minutes at the cost of increasing the walking option to 33 minutes.

We use this concept of multi-class fleets to disperse traffic congestion across the network by using different modes of travel in a near travel time social optimal combination. Specifically, we propose a system-centric routing algorithm which accounts for the estimated traffic congestion to provide multi-class mobility-on-demand solution. In addition, we propose a computational and travel time efficient transit point search algorithm to seamlessly switch between different modes of travel. Even though a multi-class route would not often be a passenger's first choice due to the added overhead of walking or changing transport modes, our approach validates its benefits in dissipating congestion and even sometimes reducing the total travel time when such inter-class recommendations are followed. For example, as illustrated in Fig. 1, our approach reduces the overall travel time of a trip with a trade-off of increased usage of micro-mobility options. Our solution also inherently provides flexibility to the passengers in choosing their preferred mode of transport based on accessibility and cost.

Our contributions in this paper are (a) a system-centric congestion-aware routing framework for serving all trip demands using mobility-on-demand vehicles; (b) a computational and travel-time efficient dynamic transit point selection algorithm between different modes of travel that minimizes the overall travel time in a capacity-bound transportation network; (c) a robust validation of our proposed approach using road network data from a geographically diverse set of cities. In the following sections we highlight our aforementioned contributions.

II. RELATED WORK

Our proposed approach provides the key desirable capabilities of a multi-class mobility-on-demand system which includes congestion-free routes along with easily accessible transit points between modes in real-time. In contrast, the existing routing solutions are either single-class with congestion-aware routing or multi-class user-centric routing that do not consider network congestion. We present the relevant approaches below.

A. Single-Class Routing

The problem of congestion-aware routing was addressed by using a centralized network of single-class Autonomous Mobility-on-Demand (AMoD) vehicles in [9]. The intelligent routing and re-balancing of each AMoD vehicle on capacitated roads did not increase congestion for small demand sizes. However, the trips were sub-optimally routed in event-based batches using a greedy A* approach which resulted in a Wardrop User equilibrium [10]. Congestion-aware route-planning policy for AMoD system in mixed traffic participants (e.g. private vehicles) scenarios was proposed in [11]. The AMoD vehicles and their re-balancing flows were routed using a system-centric approach. However, for high levels of demand, pure AMoD travel was known to be detrimental due to the additional traffic stemming from its re-balancing flows. In addition, since non-AMoD vehicles are generally routed in a user-centric way, the resulting system-wide routing was sub-optimal. Therefore, our proposed approach provides a non-greedy system-centric solution for any mobility-on-demand vehicle while accounting for overall traffic patterns arising from all traffic participants.

B. Multi-Class Routing

Multi-class fleets were introduced in [7] to serve requests using a combination of heterogeneous vehicles (cars, buggies, scooter, walking). The optimal fleet size for individual class of vehicles were obtained using a genetic algorithm which was constrained on the overall budget of the multi-class AMoD system. Multi-class vehicles were assigned to each customer sequentially with the objective of minimizing the travel time across first, middle and last parts of individual customer journey. The authors report that when the demand was much greater than the expected demand used to optimize the fleet size, the total average travel time for multi-class was worse than that of single-class. Hence the authors introduce, multi-class with walking mode where the customers are suggested to walk for the first and/or last parts of their journey to avoid waiting time in queue during unexpected demand periods. Similarly, in [11], the multi-class AMoD system was implemented with moderate amount of walking, resulting in overall system performance to be improved by 50%. Thus indicating the benefit of multi-class mobility-on-demand (including walking) in addressing high demands and potentially dispersing congestion. Hence, in our work we propose multi-class mobility on demand solution including walking mode while additionally considering the traffic congestion.

C. Computation Efficiency

One of the major challenge for both single-class and multi-class congestion-aware routing algorithms is the computational efficiency of finding real-time solutions. For the single-class system-centric routing, there is a significant amount of work demonstrating new computationally efficient ways to replace the traditional process of the Traffic Assignment Problem [12]. For example, Frank Wolfe optimization [13] and Contraction Hierarchies [14] were used in [15] to achieve a system-optimal solution for congestion-aware routing problem. The authors showcased a 20% improvement in computation time over the traditional TAP assignments. Furthermore, the convergence rate of Frank Wolfe can potentially be made faster with parallel computing. In [11], various cost functions such as the Bureau of Public roads (BPR), 2-line approximation, 3-line approximation, and Davidson's heuristics were compared. In case of convex problems, BPR was an effective choice. Since we are not accounting for re-balancing flows, our problem remains convex. Hence, we use a dynamic BPR heuristic to quantify and account for congestion within our framework by integrating the traffic speed bands into its capacity term [16]. In addition we obtain real-time transit points for switching between different modes of travel by using the algorithm proposed in [17]. This algorithm was originally proposed for obtaining optimal rendezvous points by using Euclidean distances as the shortest path heuristic. The time complexity for their proposed method is $O(n^2)$ which is highly useful in large combinatorial problems such as our transit point search problem.

In summary, we draw inspiration from the aforementioned papers and propose improvements by (a) implementing a Wardrop *system equilibrium* [18] with real-time congestion information, (b) incorporating *multi-class fleets* to dissipate congestion, and (c) apply a *computational and travel time efficient* algorithm to obtain real-time and near-optimal dynamic transit points between different modes of transport.

III. PROBLEM DESCRIPTION

Given a graph G representing street network of a city with vertices V as potential transit points between different modes of transport, and edges E as the highways or city streets, we would like to solve the customer routing problem in a congested network using multi-class vehicles and micro-mobility options. The problem is particularly challenging given an assortment of edges comprising of vehicle-only roads as well as mixed traffic streets and walkways (e.g. pedestrian walkways and by-lanes) for a capacity bound transportation network. The trip is distributed into three parts in order to decrease the overall travel time of the customer. Specifically, we propose incorporating micro-mobility options such as walking, cycling or biking for the first and last parts of the trip along with on-demand cars for the middle part. The transit points between different modes of transport are selected such that the first and the last parts of the trip can be easily completed using the walking mode. In addition, all customer requests are assigned paths based on a social model in order to get the least travel time for the entire system. An overview of our proposed solution is illustrated in Fig. 2, where we first receive the origin-destination pair for each customer. Next, we obtain and apply the traffic data to the route map. Then the transit points are selected followed by providing system-centric socially optimal routes.

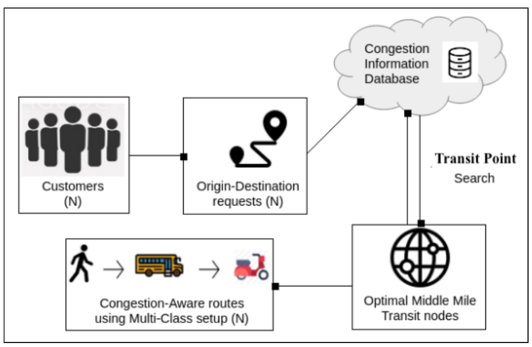

Fig. 2: Multi-class congestion-aware routing framework

We apply the following assumptions in our framework: (a) our proposed congestion-aware routing framework is applicable to any mobility-on-demand vehicle; (b) while routing a customer, there is always a vehicle available on demand at the transit points; and (b) the vehicles in the system follow multi-class recommendations given by our framework in order to achieve a system equilibrium. (d) the maximum expected walking time is based on average acceptable walking distance [19]. In order to focus on solving the main problem of efficient congestion-aware routing, we do not address problems such as vehicle to customer assignment, fleet sizing and re-balancing in this work.

IV. PROPOSED APPROACH

Given that a majority of customers in a mobility-on-demand network inherently opt for a single and fast-speed vehicle along the shortest-distance route, the resulting traffic flow in the network becomes congested in certain parts while being under-utilized in others. Moreover, such myopic, user-centric routing, results in sub-optimal solution for the overall system. Our multi-class system-centric framework divides a customer journey into three parts with first and last parts through pedestrian walkways in order to dissipate traffic and reduce congestion on main roadway network.

A central challenge for multi-class congestion-aware routing problem is to dynamically select transit points in order to get optimal entry/exit into/from the main middle part of the network. These transit points are constrained by the maximum acceptable walking distances for the first and last parts of the route while minimizing the total travel time for the entire journey. In addition, these transit points need to be available at the intersection of pedestrian and drive-only networks so that the continuity of the entire journey is maintained. We consider these constraints and obtain near-optimal transit points for congestion-aware, system-centric routing.

A. Computing middle transit nodes

In order to compute the transit nodes, we first divide a customer trip into three parts as given below.

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

Where, **X:** Origin of request or start node of first part
A: End node of first part or start node of middle part
B: End node of middle part or start node of last part
Y: End node of last part or destination of request

For quantifying congestion information, we only use estimated travel time for the middle part of the trip, as congestion for first and last parts is not applicable. Specifically, we calculate traffic travel time for the middle part of the trip using *Contraction Hierarchies* with the Bureau of Public Roads heuristic as the edge cost. We first route for the middle part ($\mathbf{A} \longrightarrow \mathbf{B}$) as it allows to attain the maximum flexibility in choosing transit nodes while minimizing overall travel time. Assuming that customers are willing to walk a maximum total of $(2D)$ units of first and last parts of their trip distance, we consider all nodes within D units radius from **X** and **Y** as potential candidate transit points for the middle part, as given below.

$$\textit{Candidate Sources (A)} = s_1, s_2, \ldots, s_n$$
$$\textit{Candidate Targets (B)} = d_1, d_2, \ldots, d_m$$

Since the total number of candidate paths obtained using n candidate sources and m candidate targets is quite large $(n x m)$, we reduce the computation cost for finding optimal transit points by proposing **Transit Point Search** algorithm (see Algorithm 1). Our proposed transit point search algorithm is based on the Hybrid algorithm presented in [17]. The Hybrid algorithm finds potential rendezvous points in a network without potentially requiring to explore all combinations of paths. The algorithm introduces an early stopping criteria based on a guaranteed heuristic. Specifically, it uses a Euclidean distance heuristic for fast comparison of candidate paths to find the shortest travel time path. The algorithm obtains the time required to travel the Euclidean path length for all candidate paths. These paths are sorted in ascending order of their Euclidean path travel time. Each Euclidean path travel time is replaced in order with its corresponding actual traffic travel time and resorted in the list of candidate paths. This process is repeated until the traffic travel time of the path being replaced is lesser than the Euclidean path travel time of the path that is next in order. The algorithm then returns the least traffic travel time path from the sorted list of paths. This path is guaranteed to be optimal. In the worst case scenario, the Hybrid algorithm replaces all the candidate Euclidean path travel time paths with traffic travel time, resulting in no reduction of computation cost.

The underlying challenge is to maximize the probability of selecting optimal transit points that provide the shortest-time path by analyzing only a subset of paths, and without querying all $(n\text{x}m)$ path combinations, as experienced in the worst-case for the Hybrid algorithm. Hence, we update the stopping criteria used in the Hybrid algorithm to further improve the computation cost by imposing a hard-cutoff based on a modified version of the *Secretary problem* [20]. We derive an analogy that each candidate path is analogous to a candidate secretary considered for hiring. In our problem formulation, we have the advantage of selecting a previously considered path. According to the proof provided by *Odds algorithm* [21] for the Secretary problem, the near-optimal winning probability is at least within the first $(n\text{x}m/e)$ candidate paths. So we select the least traffic travel time path from the sorted list of first $(n\text{x}m/e)$ paths obtained from the Hybrid algorithm. The end points of the selected path are our proposed near-optimal transit points to/from the middle part of the route. Since the total number of path combinations for selecting optimal transit points can be in the order of 10^4 for dense networks, having a near-optimal stopping policy can significantly reduce the computation cost.

B. Compute system-optimal flows for the middle part

Once the transit points are selected, our goal is to compute flows for the entire network such that there is no congestion and all requests are served in minimum travel time. Given a capacitated, symmetric network $G(V, E)$, a set of transportation requests $R := (s, d)$ between source s and destination d, time to travel on an edge between nodes u and v as $t(u, v)$, and capacity of the edge as $c(u, v)$, our objective is to minimize the network congestion.

$$\text{minimize:} \quad \sum_{r\in R}\sum_{(u,v)\in E} t(u,v) f_r(u,v) \tag{1}$$

$$\text{subject to:} \quad \sum_{u\in V} f_r(u, s_r) = \sum_{w\in V} f_r(s_r, w), \quad \forall r \in R \tag{2}$$

$$\sum_{u\in V} f_r(u, d_r) = \sum_{w\in V} f_r(d_r, w), \quad \forall r \in R \tag{3}$$

$$\sum_{u\in V} f_r(u, v) = \sum_{w\in V} f_r(v, w), \quad \forall r \in R, v \in V\backslash\{s_r, d_r\} \tag{4}$$

$$\sum_{r\in R} f_r(u, v) \le c(u, v), \quad \forall (u, v) \in E \tag{5}$$

We denote a feasible traffic flow as $f_r(u, v)$ that satisfies Equations (2), (3), (4) and (5). Constraints (2), (3) and (4) enforce flow conservation of source, destination and transit nodes, respectively, along with continuity of each trip. Constraint (5) enforces the capacity constraint on each link. We obtain this LPP formulation as an instance of the fractional multi-commodity flow problem in [22]. While we have defined the LPP in terms of fractional flows, an integer-valued counterpart can be defined and (approximately) solved to find routes for each individual customer trip. Hence we use a conditional gradient descent algorithm, Frank-Wolfe Optimizer [13], to find system-optimal flows for all requests. These flows are further decomposed into routes using Dijkstra algorithm [23] with an additional non-zero capacity check on all paths. In case an edge capacity is

Algorithm 1: Transit Point Search

Data: Dictionary X with Key:(Node A, Node B); Value: Time to travel Euclidean distance (A,B), sorted in increasing order of value
Result: Nodes (A,B) with minimum travel time

```
Cutoff ← Floor(Length(X)/e);
(Source, Dest) ← (X[1][Source], X[1][Dest]);
minTime ← ∞;
for i ← 1 to Length(X) do
    travelTime ←
     ContractionHierarchies(Source, Dest);

    if i > Cutoff then
        if travelTime < minTime then
            Return current (Source, Dest)
             as its optimal;
        else
            Return (Source, Dest) optimal
             till now;
        end
    else
        if travelTime < minTime then
            Current (Source, Dest) is
             optimal till now;
            minTime ← travelTime;
            if travelTime < X[i+1][Euclidean
             Travel Time] then
                Return current (Source, Dest)
                 as its optimal;
            else
                (Source, Dest) ← (X[i+1][Source],
                 X[i+1][Dest]);
            end
        else
            if minTime < X[i+1][Euclidean Travel
             Time] then
                Return (Source, Dest) optimal
                 till now;
            else
                (Source, Dest) ← (X[i+1][Source],
                 X[i+1][Dest]);
            end
        end
    end
end
```

(a) Singapore (b) New York (c) Paris

Fig. 3: Road networks obtained from OpenStreetMap

full and no alternate edge or a path is able to accommodate the request then we consider the entire path for that request infeasible. The system-centric flow is re-computed for all the identified infeasible requests.

V. EXPERIMENTAL RESULTS

A. Setup

We validate our proposed algorithm on road networks from three diverse cities: Singapore, New York, and Paris. The network information is extracted using OpenStreetMap software. We get the drive-only network by providing network type as *"drive"*, and a pedestrian/cycle-way network with network type as *"walk"* or *"bike"*. The entire network can be visualized as a densely-connected graph as shown in Fig. 3. It is to note that New York's road network is a cross-section of Manhattan distances, whereas Singapore and Paris have curvy lanes. Hence, our heuristics (Euclidean distances) for Transit Point Search is expected to work better for New York while giving realistic estimates for Singapore and Paris.

We apply the following assumptions to obtain the numerical results. First, the speed limits for cars and bikes are considered according to the government regulations of the subject city [24], [25], [26]. A list of average speeds that we considered is categorized by vehicle type and city in Table I. Second, all transit points chosen for middle route are within a $500m$ radius from the source and destination (assuming reasonable walking distances of up to $1km$ for first and last parts of the route).

	Walk (m/s)	Bike (m/s)	Car (m/s)
Singapore	1.78	3.7	13.89
New York	1.78	6.71	11.18
Paris	1.78	5.55	22.22

TABLE I: Speed limits for different parts of the route

In order to validate the robustness of our approach for different traffic patterns, we use three instances of time: $12:00$ A.M., $3:00$ P.M., and $6:00$ P.M; corresponding to off, moderate and high peak times, respectively. For real-time traffic information, we use Traffic SpeedBand Dataset [27] collected for the duration February-March 2020 incorporated within the BPR heuristic in its capacity term for Singapore city. For New York and Paris, we lack real-time traffic information and so we estimate it using a Gaussian random variable, similar to the approach in [17] and reduce the capacity of roads by 25% and 50% for moderate and high peak times respectively. Congestion information is obtained for the entire drive-only network in terms of time required to cross a road link. For a quantitative analysis of our proposed approach, we consider 5 batches of 500 customers each (totaling to 2500 origin-destination (OD) trips), for downtown Singapore, Manhattan, and Paris for a given time frame.

B. Multi-Class Congestion-aware Routing

We compare our proposed congestion-aware multi-class routing algorithm with single-class user-centric approach. We evaluate the performance using the following cost metrics.

1) Flow-Time Cost: We define flow-time as the time taken (in hours) to serve a given number of requests. Fig. 4 shows that the average reduction in the overall flow-time cost for multi-class system-equilibrium over single-class user-equilibrium is $\mathbf{12.13\%}$, $\mathbf{48.45\%}$, $\mathbf{53.04\%}$ in Singapore, New York, and Paris respectively. This shows a drastic decrease in potential network congestion by using micro-mobility options for a very small percentage of the entire trip. It is interesting to note that the average flow-time cost reduction is least in Singapore potentially due to the accurate Traffic SpeedBand Dataset.

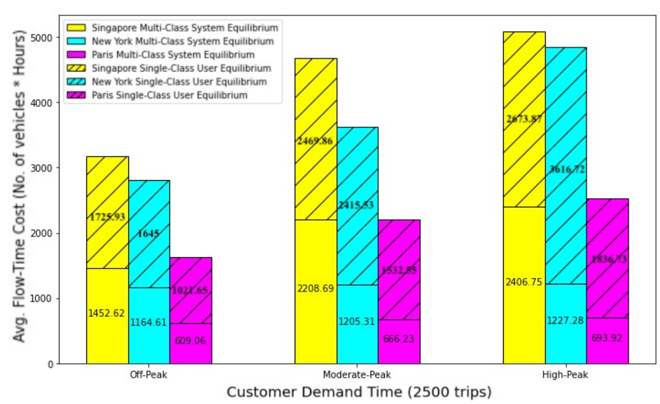

Fig. 4: Average Flow-Time Cost

2) Network Utilization: We evaluate congestion in terms of the infeasible trips which are caused when the edges along all potential paths of the trip reach their maximum capacities.

Fig. 5 shows that there are much greater infeasible trips causing increase in waiting queue size for single-class user-equilibrium than multi-class system-equilibrium. An explanation can be that edge capacities are reached faster in a user-centric approach. For the network utilization analysis, we infer it based on the number of infeasible trips where an increase in the number of infeasible trips is interpreted as a decrease in network utilization. Our framework increases network utilization while serving more requests in a given time interval by $\mathbf{74.26\%}$, $\mathbf{83.78\%}$, $\mathbf{92.96\%}$ Singapore, New York, and Paris respectively.

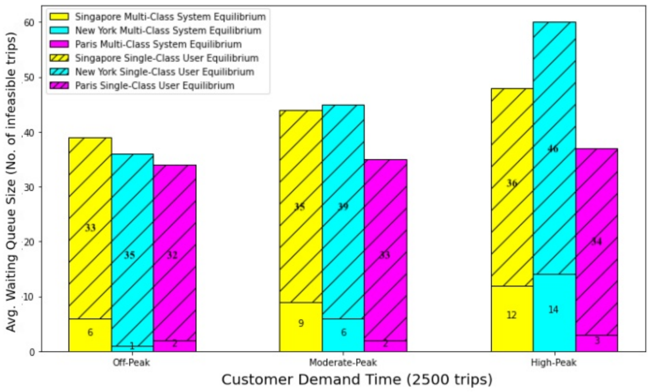

Fig. 5: Average Network Utilization

3) Travel-Time: We compare the overall travel time between a multi-class system setup and single-class setup. Fig. 6 shows $\mathbf{1.72\%}$ increase in travel-time for our framework in Singapore. This is because of the usage of micro-mobility options in the first and last parts of the routes which take longer travel-time than cars. However, this increase is insignificant at the cost of serving a higher number of trips. Since Paris and New York's modelled travel times depend on real path lengths rather than number of vehicles on the road, we see a decrease in the travel time for our framework by $\mathbf{67.58\%}$, $\mathbf{49.97\%}$ in New York and Paris, respectively.

4) Computational Efficiency: A majority of the computational cost for our proposed multi-class congestion-aware approach is associated with calculating the transit point for the middle part of the route in real-time. We measure the real-time computational efficiency of finding the transit points in terms of (a) the travel time for the middle part of the route and (b) the number of queries required for obtaining

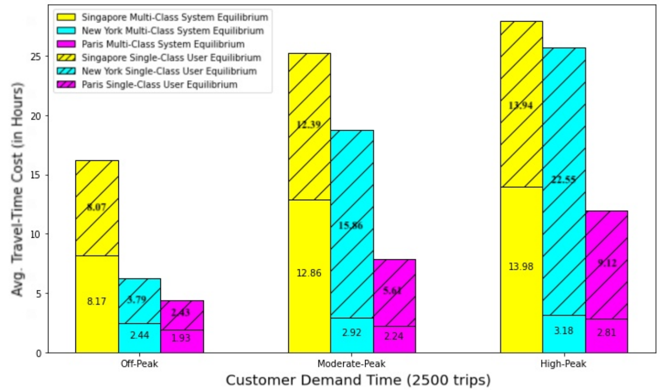

Fig. 6: Average Travel-Time

the route. We compare our sub-optimal transit point search algorithm against the Hybrid algorithm from [17] and an optimal exhaustive search algorithm.

(i) Estimated Travel-Time: Tables II, III show that both Hybrid and exhaustive search are optimal as they provide minimum travel time. However, our proposed Transit Point Search does not output optimal travel times. The average deviation in travel times for our approach with respect to the two baselines is $\mathbf{4.67}$, $\mathbf{1}$, $\mathbf{1}$ secs in Singapore, Paris, and New York, respectively. Since, this deviation is insignificant with respect to the total travel time, our proposed Transit Point Search can be considered as near-optimal.

(ii) Number of queries: The number of queries made by each of the three search techniques is tabulated in Table II, III, where, obtaining the distance between between two nodes for a path combination, constitutes a query. We observe a significant reduction in the computation cost for our proposed Transit Point Search algorithm over the Hybrid and the Exhaustive search algorithms. Thus, our proposed Transit Point Search algorithm proves to be highly computationally efficient with an average speedup of $\mathbf{68\%}$ and a negligible maximum average travel time delay ($\mathbf{5}$ sec) in comparison to the Exhaustive and the Hybrid Search algorithms for the presented experimental results.

Therefore, our multi-class system-centric framework is travel time cost-effective, results in better network utilization, is computationally efficient and is a social model for reducing congestion compared to the single-class user-centric framework.

	Travel Time (secs)				No. of queries			
	Transit Point	Hybrid	Exhaustive	**Deviation**	Transit Point	Hybrid	Exhaustive	**% Reduction**
Off Peak	847	843	843	**4**	925	2081	3756	**75.37**
Moderate Peak	928	924	924	**4**	1178	2909	3756	**68.34**
High Peak	961	955	955	**6**	1270	3208	3756	**66.19**
Average	**912**	**907.33**	**907.33**	**4.67**	**1124.3**	**2732.67**	**3756**	**70.07**

TABLE II: Comparison between different transit point search algorithms with real-time traffic data: Singapore

	Travel Time (secs)				No. of queries			
	Transit Point	Hybrid	Exhaustive	**Deviation**	Transit Point	Hybrid	Exhaustive	**% Reduction**
New York	785	784	784	**1**	1117	2753	3245	**65.58**
Paris	557	556	556	**1**	1740	4653	5106	**65.92**
Average	**671**	**670**	**670**	**1**	**1428.5**	**3703**	**4175.5**	**65.75**

TABLE III: Comparison between different transit point search algorithm with estimated travel time.

VI. DISCUSSION AND CONCLUSION

In this paper, we studied the achievable benefits of using a centrally-controlled multi-class mobility-on-demand system for providing congestion-aware routes to customers. We proposed a fast and efficient Transit Point Search algorithm to find transit nodes between different modes of travel that accounts for reasonable walking/cycling distances. Our algorithm reduces the number of path queries by 68%, minimizes the overall flow-time cost of the system, and provides shortest possible routes in a capacity-bound network. Our experimental results support our hypothesis that with a small increase in the expected travel time, we can maximize the network-utilization on an average by 84%. In conclusion, we successfully built a social framework that performs congestion-aware routing for multi-class mobility-on-demand services and proves computationally better than the state-of-the-art approaches.

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] M. Barnard, "Autonomous cars likely to increase congestion," *cleantechnica.com/2016/01/17/autonomous-cars-likely-increase-congestion*, 2016.

[6] I. UK, "A car-free future? how uk cities are moving towards a pedestrian age," *www.independent.co.uk/environment/car-free-cities-pedestrianisation-cycling-driverless-vehicles-york-oslo-birmingham-a9299856.html*, 2020.

[7] 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.

[8] D. Kondor, X. Zhang, M. Meghjani, P. Santi, J. Zhao, and C. Ratti, "Estimating the potential for shared autonomous scooters," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[9] 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.

[10] J. G. Wardrop, "Some theoretical aspects of road traffic research," *Proceedings of the Institution of Civil Engineers*, 1952.

[11] S. Wollenstein-Betech, A. Houshmand, M. Salazar, M. Pavone, C. Cassandras, and I. Paschalidis, "Congestion-aware routing and rebalancing of autonomous mobility-on-demand systems in mixed traffic," *arxiv.org/abs/2003.04335*, 2020.

[12] A. Krylatov, V. Zakharov, and T. Tuovinen, "Optimization models and methods for equilibrium traffic assignment," 01 2020.

[13] M. Frank and P. Wolfe, "Frank-wolfe algorithm," *Frank-Wolfe algorithm Wikipedia*, 1956.

[14] J. Dibbelt, B. Strasser, and D. Wagner, "Customizable contraction hierarchies," *Journal of Experimental Algorithmics*, 2016.

[15] K. Solovey, M. Salazar, and M. Pavone, "Scalable and congestion-aware routing for autonomous mobility-on-demand via frank-wolfe optimization," *arxiv.org/abs/1903.03697*, 2019.

[16] Z. John, B. James A, D. Richard G, R. Paul, V. Mark, K. Wayne, R. Nagui M, S. Bastian J, H. Ali, A. Behzad, C. Thomas, S. Soheil, M. Richard A, E. Lily, and S. Cambridge, "Incorporating travel time reliability into the highway capacity manual," 01 2014.

[17] 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.

[18] Wikipedia, "Wardrop equilibria," *en.wikipedia.org/wiki/John-Glen-Wardrop*, 1952.

[19] P. Olszewski and S. Wibowo, "Using equivalent walking distance to assess pedestrian accessibility to transit stations in singapore," *Transportation Research Record Journal of the Transportation Research Board*, vol. 1927, pp. 38–45, 01 2005.

[20] P. Freeman, "The secretary problem and its extensions: A review," *International Statistical Review / Revue Internationale de Statistique*, 1983.

[21] F. T. Bruss, "Sum the odds to one and stop"," *The Annals of Probability*, 2000.

[22] O. J. Ahuja RK, Magnanti T, "Network flows: Theory, algorithms and applications.," *Prentice Hall*, 1993.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Section 24.3: Dijkstra's algorithm". introduction to algorithms (second ed.)," *MIT Press and McGraw–Hill. pp. 595–601. ISBN 0-262-03293-7*, 2001.

[24] About-France.com, "Driving rules for france," 2019.

[25] A. S. Council, "Speed limits in new york," 2014.

[26] L. T. A. Singapore, "Driving rules and regulations," 2021.

[27] S. Govt, "Lta datamall dynamic datasets," *mytransport.sg/content/mytransport/home/dataMall/dynamic-data.html*, 2020.