Multi-class Fleet Sizing and Mobility on Demand Service

Malika Meghjani[1(✉)], Scott Drew Pendleton[2,3], Katarzyna Anna Marczuk[2,3], You Hong Eng[1], Xiaotong Shen[1], Marcelo H. Ang Jr.[2], and Daniela Rus[4]

[1] Singapore-MIT Alliance for Research and Technology, Singapore, Singapore
{malika,youhong,xiaotong}@smart.mit.edu
[2] National University of Singapore, Singapore, Singapore
{scott.pendleton01,katarzyna}@u.nus.edu, mpeangh@nus.edu.sg
[3] nuTonomy, Singapore, Singapore
[4] Massachusetts Institute of Technology, Cambridge, USA
rus@csail.mit.edu

Abstract. This paper addresses multi-class fleet sizing and vehicle assignment problem where we aim to provide Autonomous Mobility-on-Demand (AMoD) service using a fleet of heterogeneous vehicles. We present a chain of transportation with three classes of autonomous vehicles including cars, buggies and scooters. Each class of vehicle can access a subset of the network, such that, there are some links exclusive for that particular class. Our fleet management system then assigns available vehicles to trips based on the travel time for passenger pick-up and drop-off, their queue time and accessibility of the road network by the vehicle. Each assignment may consist of a set of vehicles allocated for one trip that is composed of multiple-legs served by different vehicles. For example, first mile pick-up by a scooter, middle mile on a car and last-mile trip on a buggy. We apply a genetic algorithm for heterogeneous fleet sizing and propose a hierarchical structure for travel time optimal assignment of the multi-class autonomous vehicles to passengers. We validated our approach with a range of heterogeneous fleet sizes constrained on the given budget. Our approach is more time efficient than taking a ride on a single-class autonomous vehicle for middle mile plus walking during the first and the last miles. Hence, we provide the convenience of autonomously covering the entire journey using multi-class vehicles with no additional travel or transit delays compared to single-class.

1 Introduction

Autonomous Mobility on Demand (AMoD) systems provide demand-responsive transportation services using self-driving vehicles. These vehicles generally cover partial to complete journeys of the passengers, depending upon the vehicle accessibility to the road network. Our goal is to specifically provide complete coverage from first mile to last mile using multi-class autonomous vehicles. Our fleet of multi-class vehicles, shown in Fig. 1, range from slow-drive, easily navigable,

M. A. Cardin et al. (Eds.): CSD&M 2018, AISC 878, pp. 37–49, 2019.
https://doi.org/10.1007/978-3-030-02886-2_4

personal mobility scooters for individuals, to medium speed autonomous buggies and faster speed self-driving cars with larger capacity and accessibility to larger lanes and main streets.

Fig. 1. SMART's multi-class autonomous vehicles: road car (upper left), buggy (lower left), scooter (right).

The specifications for each of the autonomous vehicles is given in Table 1. These specifications are also represented by a star diagram for easier comparison in Fig. 2, where larger values indicate better performance. The car is most ideal for travel along the road network, as it has the highest speed and is capable of longest range. However, the scooter would be more ideal for narrow passageways and crowded pedestrian environments. It is the cheapest platform, most efficient in terms of weight, most maneuverable (smallest turning radius), and small enough to fit inside building hallways and on small sidewalks. The buggies are most well suited for large pedestrian areas, such as parks, plazas, airport terminals and hospitals.

By using all three classes in combination, a greater accessibility and service coverage can be achieved such that the users can be taken not just between building pick-up and drop-off points, but even from the room of one building to a specific room in another building several kilometers away. Our proposed multi-class mobility on demand system is best applicable to elderly or disabled people who might need assisted modes of navigation. In addition, it is also very useful for passengers and/or goods transportation at facilities with long transits (e.g. airports and hospitals) or under bad weather conditions.

An ideal application scenario for multi-class AMoD service is a campus network where the shuttle buses provide transportation between the major stops

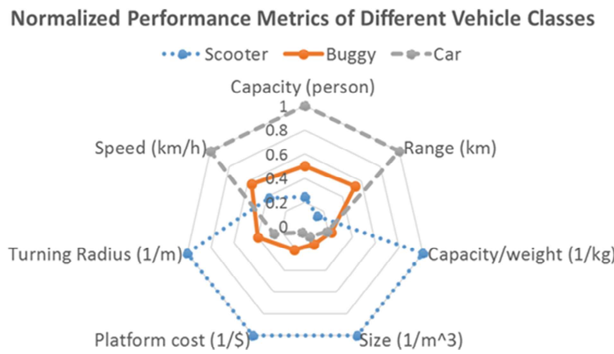


Fig. 2. Comparison of vehicle classes based on normalized values extracted from Table 1. Higher values are more ideal.

Table 1. Self-driving vehicle specifications

	Car	Buggy	Scooter
Length (mm)	3885	2525	930
Width (mm)	1515	1200	485
Height (mm)	1750	1890	2100
Empty Weight (kg)	1200	500	56
Seating Capacity (persons)	4	2	1
Operating Speed (km/h)	19	10.8	7.2
Range (km)	150	80	20
Turning Radius (m)	4.5	3.0	1.2
Platform Retail Cost (USD)	29125	7499	1599

for the middle mile and the passengers are required to walk to cover their first and last mile. Our goal is to analyze this campus scenario where the transit points are fixed and the shuttle bus schedules are almost certain. Given this scenario, we first solve the multi-class fleet sizing problem to decide on the number of vehicles in each class. We then propose a multi-class vehicle assignment algorithm to allocate combinations of road cars, buggies, and personal mobility scooters to the passengers. Our solution has the travel time efficiency better than single-class assignment at the convenience of multi-class door-to-door service. We provide a thorough case study of the campus scenario in simulation. Our proposed method is generalizable to other hierarchical transportation models involving private, shared and public transportation systems for inter and intra-city commutes. A proof of concept implementation for multi-class mobility on demand system with real autonomous vehicles (one in each category) was also illustrated in our previous work [1]. The contributions of this paper are as given below.

- A multi-class fleet sizing algorithm for deciding the number of vehicles required in each class, given an estimated demand size and expected budget.
- A generic and scalable multi-class assignment algorithm for providing door-to-door transportation.
- An empirical analysis in a real-world scenario given the fleet sizing and vehicle assignment algorithm.

Each of these contributions are highlighted in the following sections. Specifically, Sect. 3 outlines our general software architecture for multi-class management system, Sect. 4 provides algorithms for heterogeneous fleet sizing and vehicle assignment and Sect. 5 presents our simulation results.

2 Related Work

Research in the field of autonomous vehicles has matured in the past few years. Recent studies have also been conducted to model and anticipate the social impact of implementing AMoD [2,3]. The case studies have shown that MoD systems would make a more affordable and convenient access to mobility compared to traditional transportation system characterized by extensive private vehicle ownership [3].

Mobile Internet technology has created the opportunity to enable dynamic and on-demand transportation services, i.e., through e-hailing applications. These services have a potential to provide societal and environmental benefits when designed adequately. The core of the real-time e-hailing concept is the development of algorithms for optimally matching vehicles (or drivers in traditional systems) and passengers [3–5]. We have seen a growing interest in the intelligent transportation literature to address the optimization issues in the dynamic assignment for autonomous mobility on demand systems. As of today, the number of specific contributions is still small [6].

A demand-responsive personalized service to passengers is introduced in [7]. In this paper, the passengers have flexibility to choose the service type (taxi, shared-taxi and mini-bus) from a menu that is optimized in an assortment optimization framework. For operators, there is flexibility in terms of vehicle allocation to different service types, which implies that the vehicles are changing their class according to the demand pattern. This framework is tested on a small network. Similarly, [3,6] propose a personalized mobility solution with autonomous vehicles. In their work, the dispatcher assigns vehicles to trips (with single and multiple pick-ups) using Greedy and Bipartite assignment and with prediction for vehicle pre-positioning. The authors test their dispatcher in the simulation environment for the city of Singapore. However, they consider only one class of vehicle. Similarly, most of the theoretical approaches to the assignment problem usually attempt to solve it for single-class vehicles [8–10].

Autonomous driving on urban roads has seen tremendous efforts in the recent years. Google is at the forefront of these efforts, having tested its fleet of autonomous vehicles for more than 2 million miles, with expectation to soon launch a pilot MoD service project using 100 self-driving vehicles [11]. Autonomous systems have also been implemented in unstructured outdoor environments and urban pedestrian areas such as side walks and university campuses. A recent example is the Smart Wheelchair System developed at Lehigh University [12].

While the above mentioned research has shown promising results in their restricted cases, to our knowledge, this paper presents the first integrated solution that considers multiple types of vehicles for transport in both pedestrian and urban road environments. We first introduced the concept of multi-class AMoD in [13], where a self-driving buggy and a road car were used in combination. This concept was extended to three classes with the addition of a self-driving personal mobility scooter in [1]. In this paper, we present a novel hierarchical multi-class Fleet Management Service (FMS) algorithm tasked to assign large-scale multiple heterogeneous vehicles to each passenger request as deemed appropriate to the mission. We also address the multi-class fleet sizing problem which provides input to our multi-class FMS algorithm.

3 Software Overview

An overview of our AMoD system is shown in Fig. 3. The booking application within this system can be used by passengers to request a mobility service to travel from the *Pick-Up* location to the *Drop-Off* location. All the mobility requests are sent to a fleet management server which assigns Autonomous Vehicles (AVs) to passengers and generates unique verification codes corresponding to each request.

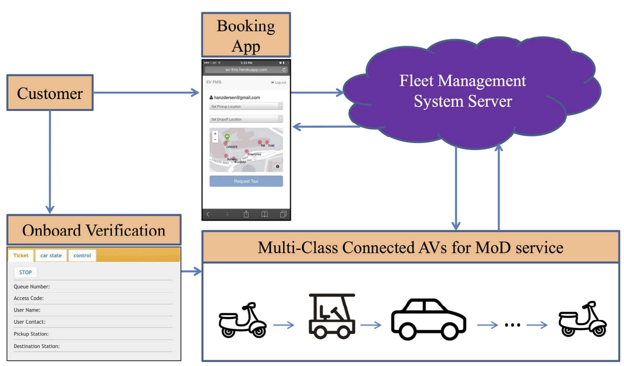


Fig. 3. Overview of our Autonomous Mobility-on-Demand (AMoD) system to provide connected transportation for mobility service [1].

Based on the traversability of the route, availability of the vehicle and the assignment cost, a suitable type of vehicle, such as a scooter, a buggy, a road car or their combination will be assigned to pick up the passenger from their origin and transit locations. Our assignment algorithm is specifically explained in Sect. 4. The assigned vehicle IDs are sent sequentially to the booking application to notify the passenger which vehicle(s) to board. Once the AV arrives at the *Pick-Up* location, the passenger can board the vehicle and key in their verification code to start the trip. The assigned AV will autonomously navigate to either the *Drop-Off* location to bring the passenger to the destination or a transfer station to allow the passenger continue the trip via another type of vehicle. Once the passenger reaches their destination or transit location, the FMS will assign the vehicle to a new user request for continuously providing the AMoD service.

4 Technical Approach

Given the network information, vehicle specifications, expected demand and the estimated budget, we first solve the fleet sizing problem. The multi-class fleet sizing problem for static demand (i.e., spatially distributed but not temporally) is similar to the bounded knapsack problem [14] where the total weight constraint of the knapsack is equivalent to our budget constraint and the items are vehicle classes with different weights and values. The weights of the items can be interpreted as the cost of the vehicle and the values of the items as the speed of the vehicle. However, we are interested in the multi-class fleet sizing problem for time-varying demand which is even more challenging than the aforementioned bounded knapsack optimization problem that is known to be NP-hard. Therefore, we focus on using approximation methods such as the genetic algorithm. The genetic algorithm [15] assesses different combinations of fleet sizes and evaluates the expected total travel time (sum of queue, assignment and travel time) per passenger using our proposed multi-class fleet assignment algorithm (Algorithm 1) as the fitness function. The components of total travel time are illustrated in Fig. 4.

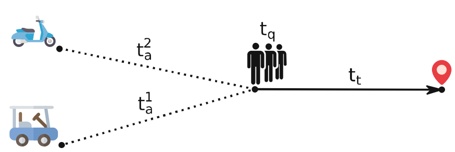


Fig. 4. Components of total travel time comprising queue time t_q, assignment time t_a and travel time to destination t_t

The multi-class fleet sizing algorithm begins with an initial guess for fleet size of each class of vehicle which needs to be provided as the input. This is used by the genetic algorithm to generate samples of fleet size combinations that satisfy the budget. These samples together form an initial population. It then creates a sequence of new combinations from the current generation of samples to create a new generation. Some of the combinations having the best fitness value (expected total travel time per passenger) from the current generation are retained and passed to the next generation. In addition, more samples of combinations are added to the next generation by either mutating or crossing over different combination sets. The sample combination corresponding to the overall best fitness value is considered as the final outcome for fleet sizing of each class of vehicle.

For the multi-class vehicle assignment problem, we propose a hierarchical approach. This process involves two stages of assignments corresponding to the first-last miles and middle mile of the journey. These assignments are performed sequentially for each passenger. We define fixed transit points to traverse from the first mile to the middle mile or from the middle mile to the last mile. We refer to these transit points as the main stations and the paths connecting them are only accessible by cars. Each of these main stations are part of a sub-network which is accessible by either scooters or buggies or both. The nodes in the sub-network are referred to as sub-stations. In our current work, we assume that all vehicles have a capacity of one passenger (i.e. no ride-sharing). An overview of our assignment algorithm is presented above in Algorithm 1.

Input: Locations of
- passengers $P = \{p_1, p_2, ..., p_q\}$
- scooters $S = \{s_1, s_2, ..., s_l\}$
- buggies $B = \{b_1, b_2, ..., b_m\}$
- cars $C = \{c_1, c_2, ..., b_n\}$
- main stations $MT = \{mt_1, mt_2, ...mt_r\}$
- network information $G(V, E)$ where $MT \in V$

while *(Not all passengers are served)* **do**
  foreach $(mt_i \in MT)$ **do**
    - Find first/last mile unassigned passengers $P' \in P$, scooters $S_u \in S$ and buggies $B_v \in B$ such that mt_i is main station for all.
    - Assign $S_u \vee B_v$ to P' s.t. min. cost, f_c.

  end
  - Find middle mile unassigned passengers $P'' \in P$ and cars $c_w \in C$ such that $mt_i == p_k$, where $p_k \in P''$.
  - Assign c_w to $p_k \in P''$ s.t. min. cost, f_c.

end

Greedy: $f_c = t_a(j) + t_t(j), j \in S \vee B \vee C$, where passengers are served based on their request time.

Bipartite: $f_c = \sum_{ij} t_a(i, j) + t_t(i, j) + t_q(i)$, $i \in P$, $j \in S \vee B \vee C$ where, passengers are served in batches.

Algorithm 1. Multi-class assignment

The input to the algorithm are the road network $G(V, E)$ with edges E represented as the Manhattan distance between the station nodes V, constantly updated location list of unserved or partially served passengers, $P = \{p_1, p_2, ..., p_q\}$, and locations of available multi-class vehicles with the set of scooters as $S = \{s_1, s_2, ..., s_l\}$, buggy set as $B = \{b_1, b_2, ..., b_m\}$ and car set as $C = \{c_1, c_2, ..., b_n\}$. For each iteration of the algorithm, at each main station $mt_i \in MT$, the first and last mile passengers, P' are identified with respect to the current time step. These passengers, P' are assigned either a buggy $b_v \in B_v$ or a scooter $s_u \in S_u$ from the set of buggies and scooters that are available within their respective sub-network. This is followed by assignment of cars $c_w \in C$ for all the middle mile passengers P'' that are at the main stations. When no vehicles are available, the passengers queue up. The assignment is based on the vehicle's accessibility to the network, the assignment time, travel time to destination and passenger's queue time. We implemented two assignment techniques namely, Greedy and Bipartite which are represented by f_c in Algorithm 1. This hierarchical process with first-last mile assignments followed by middle mile assignment is repeated until all the passengers in the system are served. The assignments are performed only when the passengers are either at the main or the sub-station nodes and not along the edges of the network.

As an extension to multi-class assignment with scooters and buggies for first and last miles, we also included walking as the third mode of commute. Ideally, the walking mode needs to be assigned when the sum of queue, travel and assignment time exceeds the walking time as given in Eq. 1.

$$min(t_w, t_q + t_a + t_t) \leq t_w \tag{1}$$

However, for Greedy assignment, this would cause a delay in mode assignment for the passengers as they would initially be required to wait until they are first in the queue and then the assignment time is calculated based on the available vehicles. Hence, we use a heuristic and assign walking mode when the expected queue time, t_q of the passenger exceeds the walking time, t_w.

5 Experimental Setup and Results

We validated our proposed fleet management system in simulation. We considered a real campus environment for testing the three classes of vehicles with different accessibility to the road network and different speeds. Specifically, the main streets on campus are accessible by the self-driving cars, the inter-building paths are navigable by the autonomous buggies and for narrow passageways we used personal mobility scooters. The accessible paths by each category of vehicle were pre-defined. An illustration of our campus map is presented in Fig. 5. The shaded nodes on the map represent the main stations and the plain nodes are sub-stations. The speeds of the vehicles were constantly set to the average speeds of the real platforms with car speed as 5.5 m/s, buggy speed as 3 m/s and scooter speed as 2 m/s. For performance comparison of single-class with

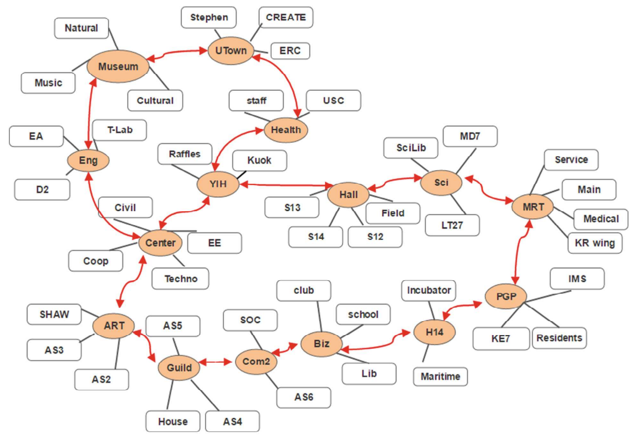


Fig. 5. Case study in National University of Singapore campus.

multi-class assignment combined with walking, we considered the average walking speed as 1.2 m/s. The details of our simulation and field experiments are further explained in the following sections.

5.1 Simulation

5.1.1 Setup

The fleet management system includes simulated passengers, vehicles and booking requests in a real network with the inter-node travel time defined based on the Manhattan distance between the nodes and average speed of the vehicle. Each booking request corresponds to a source and destination location. The source locations are sampled uniformly random from a weighted distribution, where the weights are selected based on the degree of the node in the network. Since, historically the demand is proportional to the connectivity of the node in the campus network. The demand (passenger) inter-arrival time is defined by an exponential distribution with mean based on the average travel time in the network [16]. The destination of the passengers and initial locations of the vehicles are impartially chosen in uniformly random order from the list of stations, while considering the restricted stations for each class of vehicle. The simulation scenario is run until all the passengers are served for their booking requests.

5.1.2 Results

As mentioned earlier, the first challenge for any mobility-on-demand service is to solve the fleet sizing problem. Given the expected demand distribution and the estimated budget, we apply the genetic algorithm to obtain multi-class fleet sizes. The result of applying this algorithm for a budget of approximately $11M$ USD and serving demand size of 1000 passengers in a span of about 40 min is presented in Fig. 6.

We recorded the best and the mean penalty values for each generation of the genetic algorithm. The penalty value is represented as the total travel time per passenger. The lowest best penalty value corresponding to the lowest total travel time per passenger was evaluated to be 376.982 s using 342 cars, 68 buggies and 302 scooters. The aforementioned quantities obtained correspond to about 91%, 5% and 4% of the total budget for the cost of cars, buggies and scooters, respectively.

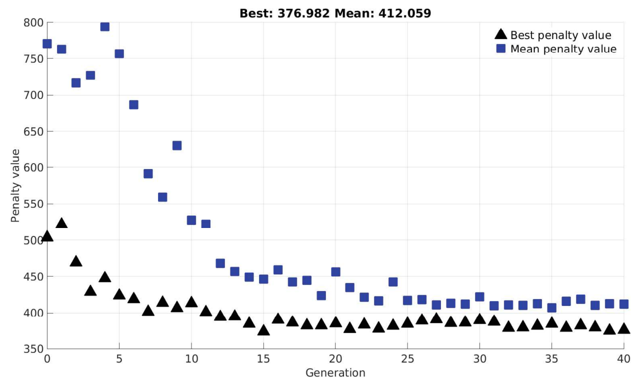


Fig. 6. Total travel time per passenger represented as penalty value corresponding to different combinations of fleet sizes illustrated as generations of the genetic algorithm.

We further analyzed the effect of change in budget (cost) on the total travel time per passenger. For three categories of budget ($6M$, $11M$ and $16M$), we observed that as the travel time decreases, the cost increases exponentially as shown in Fig. 7. This particular relation between the cost and travel time can provide meaningful insights to policy makers to decide on a convenient trade-off between the financial investment for infrastructure and expected travel time of the passengers.

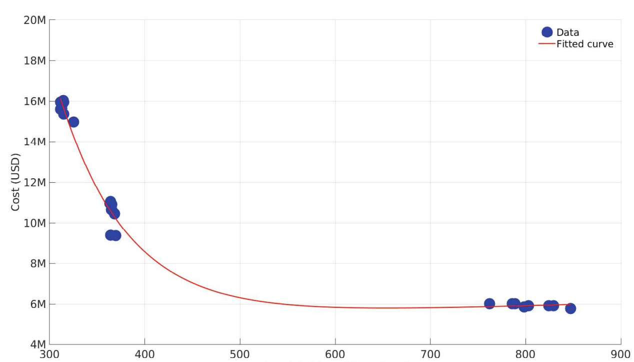
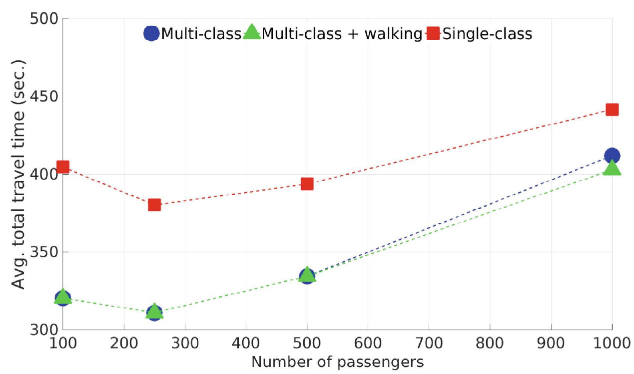
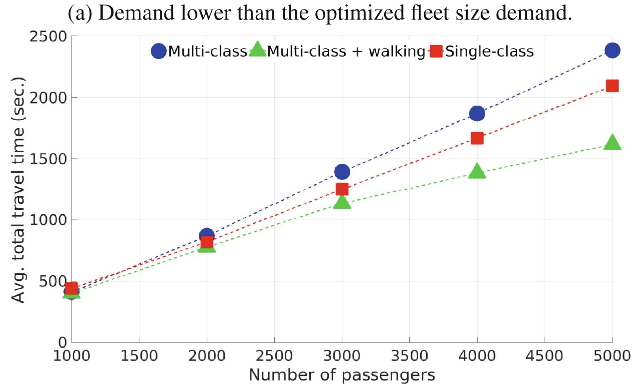


Fig. 7. The total cost of vehicles (budget) varies exponentially with respect to the total travel time per passenger.

Lastly, we compare the total travel time per passenger using our multi-class greedy assignment algorithm to single-class assignment where the passengers walk for the first and last miles. For the single-class assignment, we used the entire budget to purchase all cars resulting in more number of cars for single-class than multi-class. We then divided our analysis into two parts: (a) demand less than 1000 for which we have optimized the fleet size (Fig. 8a) and (b) demand more than 1000 for which the fleet size is insufficient (Fig. 8b). For lesser demand, analysis (a), it is clearly evident that the total travel time per passenger is significantly less for multi-class assignment when compared to single-class. However, when the demand is much greater than the expected demand that we used to optimize the fleet size, it can be observed that the total average travel time for multi-class is worse than the single-class. This is due to the fact that the fleet size is insufficient for the high demand, resulting in longer queue time t_q and hence large total travel time per passenger. In such scenarios, multi-class with walking mode which allows the passengers to walk when the expected queue time is greater than the walking time can significantly reduce the total travel time per passenger.



(a) Demand lower than the optimized fleet size demand.



(b) Demand greater than the optimized fleet size demand.

Fig. 8. Multi-class comparison with respect to the total travel time per passenger.

6 Discussion and Conclusion

In summary, we propose a multi-class mobility on-demand system which offers faster transportation than single-class system with the additional convenience of being able to provide door-to-door service and reduced total travel time per passenger. We also included walking mode in our multi-class assignment method which further improved the travel time efficiency of our AMoD system. The travel time efficiency of our multi-class AMoD system depends on the number of vehicles designated in each class. Hence, we used a genetic algorithm to optimize the heterogeneous fleet sizes given an expected budget. Lastly, we empirically analyzed the relation between the budget and total travel time per passenger which can be used for efficient demand management and traffic flow in the network.

Acknowledgment. This research was supported by the National Research Foundation, Prime Minister's Office, Singapore, under its CREATE programme, Singapore-MIT Alliance for Research and Technology (SMART) Future Urban Mobility (FM) IRG. We would also like to acknowledge the support of NVIDIA Corporation's NVAIL program.

References

1. Pendleton, S.D., Andersen, H., Shen, X., Eng, Y.H., Zhang, C., Kong, H.X., Leong, W.K., Ang Jr., M.H., Rus, D.: Multi-class autonomous vehicles for mobility-on-demand service. In: 2016 IEEE/SICE International Symposium on System Integration (SII). IEEE (2016)
2. Zhang, R., Spieser, K., Frazzoli, E., Pavone, M.: Models , Algorithms, and Evaluation for Autonomous Mobility-On-Demand Systems. In: American Control Conference, pp. 2573–2587 (2015)
3. Marczuk, K.A., Soh, H.S., Azevedo, C.M., Lee, D.H., Frazzoli, E.: Simulation framework for rebalancing of autonomous mobility on demand systems. In: International Conference on Transportation and Traffic Engineering (2016)
4. Wang, X.: Optimizing ride matches for the dynamic RDE-sharing systems. Ph.D. thesis, Georgia Institute of Technology (2013)
5. Meghjani, M., Marczuk, K.: A hybrid approach to matching taxis and customers. In: 2016 IEEE Region 10 Conference (TENCON), pp. 167–169, November 2016
6. Azevedo, C.L., Marczuk, K., Raveau, S., Soh, H., Adnan, M., Basak, K., Loganathan, H., Deshmunkh, N., Lee, D.H., Frazzoli, E., Ben-Akiva, M.: Microsimulation of demand and supply of autonomous mobility on-demand. J. Transp. Res. Board (2564), 21–30 (2016)
7. Azevedo, C.L., Marczuk, K., Raveau, S., Soh, H., Adnan, M., Basak, K., Loganathan, H., Deshmunkh, N., Lee, D.-H., Frazzoli, E., Ben-Akiva, M.: Optimizing a flexible mobility on demand system. Transp. Res. Rec. (2536), 76–85 (2015)
8. Psaraftis, H.N.: A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transp. Sci. **14**(2), 130–154 (1980)
9. Psaraftis, H.N.: An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. Transp. Sci. **17**(3), 351–357 (1983)

10. Cordeau, J.-F., Laporte, G.: A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transp. Res. Part B Methodol. **37**(6), 579–594 (2003)
11. Google Inc. Google Self-Driving Car Project
12. Montella, C., Perkins, T., Spletzer, J., Sands, M.: To the bookstore! autonomous wheelchair navigation in an urban environment. In: Field and Service Robotics (2014)
13. Pendleton, S., Chong, Z.J., Qin, B., Liu, W., Uthaicharoenpong, T., Shen, X., Fu, G.M.J., Scarnecchia, M., Kim, S.W., Ang, M.H., Frazzoli, E.: Multi-class driverless vehicle cooperation for mobility-on-demand. In: Intelligent Transportation Systems World Congress (ITSWC) (2014)
14. Dantzig, T., et al.: Number, the language of science: a critical survey written for the cultured non-mathematician (1954)
15. Goldberg, D.E.: Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley Publishing Company, Inc. (1989)
16. Allen, A.O.: Probability, Statistics, and Queueing Theory. Academic Press (2014)