hEART 2019 – Symposium of the European Association for Research in Transportation
September 4–6, 2019, Budapest University of Technology and Economics, Hungary

Providing real-time operational solutions for the on-demand capacitated ride sharing problem

Zahra Ghandeharioun* Anastasios Kouvelas

Institute for Transport Planning and Systems (IVT), Department of Civil ,Environmental and Geomatic Engineering Eidgenössische Technische Hochschule Zürich (ETH), Switzerland
{zahra.ghandeharioun,anastasios.kouvelas}@ivt.baug.ethz.ch
*Corresponding author

Keywords: Taxi ride sharing; mobility on-demand; optimization; event-based simulation.

Abstract

Urban mobility is facing a paradigm shift towards providing more convenient, environmentally friendly and on-demand services. In the recent years the growth of smartphone technologies and inexpensive cellular communications have led to a more individualized transport in urban areas; companies like Uber, Lyft, Via, and Cab-with-me have risen that focus on developing demand responsive services, known as Mobility-on-Demand (MoD). Furthermore, considering the ride sharing potential and benefits, these companies have also adjusted their services with sharing options. On the other hand, with the parallel rising of automated driving technologies, it seems that semi- or fully-automated ride sharing services would be an attractive option in the near future. Satisfying customer needs in a cost-efficient way has been the goal of many ride-sharing systems. Taxi ride sharing is considered nowadays an effective service for reducing traffic congestion and pollution; however, the operational strategies that can be used to optimize on-demand ride sharing have not been well investigated in the literature. A better understanding of the complex ride sharing problem would allow for more effective system deployments. The challenges arise from the fact that multiple stakeholders are involved with contradicting interests. If the objective of a private taxi company is to increase its revenue by offering more rides, this is in contradiction with the interests of governments (e.g. less traffic congestion and pollution). At the same time, usually a customer's objective is to travel from point A to B in the fastest and most inexpensive way. As a result, different policies need to be designed and studied in order to come up with mutual collaborative solutions and provide an efficient service. Nevertheless, only a few studies in the literature provide reliable insights and results about capacitated ride-sharing systems.

Optimization of ride sharing services has attracted a lot of research interest recently. This problem can be formulated mathematically is a similar way as the vehicle-routing and the dynamic pick-up and delivery problems. The general definition for pick-up and delivery problems is how to optimally transport objects or people from an origin to a destination. Given that all the input data are available before the determination of the routes, the problem is classified as a static optimization problem. On the other hand, in the dynamic version of the problem, some of the input data are communicated during the time horizon of the operational process (e.g. customer requests). Hence the solutions we are seeking are known as strategies to decide the real-time operations as a new request is received. Another classification presented in [1] is based on whether the information received by a request is certainly known (deterministic) or still undetermined and subject to changes (stochastic). Regarding

this classification, in the current work, we focus on the dynamic deterministic pick-up and delivery problem.

According to literature, there are only a few works investigating the dynamic pick-up and delivery problem [2]. Early approaches focus on the transportation of elderly and handicapped people, which is known as the dial-a-ride problem (DARP). In [3] a solution is proposed for a demand responsive service for individuals who are not able to use public transport. Early solutions of DARP have mainly utilized heuristic methods, e.g., the solution provided by MIT for the DAR service in Rochester, New York, USA [4]. Psaraftis in [5] introduced the first model with a dynamic programming approach for both the static and dynamic version of DARP. In the past 40 years, there has been a steady growth of different methods applied to solve DARP ([6,7]). In the current work, we mainly review the most recent approaches that address the dynamic deterministic DARP.

Ho et al. [6] classify the solutions provided by different researchers for dynamic deterministic DARP into theoretical and experimental approaches. The theoretical solutions include (a) an online algorithm, which has proven to provide better results versus its offline counterpart, or (b) a methodology to compute a lower bound which is tighter compared to the previously introduced lower bounds [8]. On the other hand, experimental approaches mostly develop simulation engines or other dynamic models. In these approaches, a new input (event) triggers the simulation engine or the model to make decisions in a short period of time. As stated in [6], a passenger request is in most of the cases the simulation trigger (event) for rescheduling the vehicles' routes (e.g [2,9]). The aim of such approaches is to serve the new request in an optimal way. In most of the studied cases, there is a penalty when a request is rejected [6]. Ho et al. [6] recommend to consider other triggers (events) for rescheduling, such as vehicle breakdowns, and unexpected events, in order to have a more realistic representation of the dynamic deterministic DARP (see e.g. [10]).

Researchers have applied different methods to solve different types of DARPs. Since DARP is an NP-hard problem, most of the proposed techniques include classical heuristics or metaheuristics that can be applied to large-scale problems. Implementation of construction heuristics can be beneficial when a feasible solution is required fast, e.g. in dynamic DARPs (see [11,12]). Moreover, Wong et al. in [11] have evaluated different strategies by application of construction heuristics. Most of these heuristics are based on the concept of greedy insertion heuristics, in which the vehicle with the cheapest insertion criterion in their route is chosen to accommodate the request [13]. Finally, Diana and Dessouky in [14] have studied various insertion strategies.

Tabu Search (TS) is a well-known and commonly used metaheuristic to solve static DARPs. In TS a tabu list is created and the local search algorithm avoids revisiting previously evaluated solutions. As stated in Ho et al. [6], Cordeau and Laptore [15] were among the first who applied the TS method to solve static DARPs. However, for a dynamic DARP a shorter response time is required; Attanasio et al. [16] provided several parallel computation methods for implementing TS processes to solve dynamic DARPs. There are several other metaheuristics developed to solve different classifications of DARPs, e.g., simulated annealing, variable and large neighborhood search, generic algorithms, and hybrid methods. Interested readers are referred to review surveys (e.g. [6,7]) for a detailed summary of previous works.

In the current work, we focus on solving the on-demand ride sharing service in a real-time framework, considering different optimization techniques. More precisely, by investigating different decision variables and cost functions, we evaluate various management strategies. Moreover, we study the sensitivity of the solutions to different ride sharing capacities. Within this framework, we develop an event-based simulation engine that can be utilized in order to propose and evaluate a real-time taxi ride sharing search algorithm. The aim of the algorithm is to decide in real-time (i.e. within few seconds) among competing shuttle candidates that satisfy both the user inquiries and the problem constraints. This framework enables us to study different realistic scenarios regarding various cost functions, in which we can study preferences of different stakeholders (e.g., maximum number of ride sharing per vehicle, maximum waiting times, quality of service, etc.). Furthermore, in this approach the state of the system is updated not only when a request arrives as an input, but also whenever a drop-off or change in traffic conditions takes place; all these are considered events and trigger the procedure of updating the system state. Finally, by utilizing millions of real trip data from the New York City taxi database, we evaluate the feasibility and real-time applicability of the proposed framework, and evaluate the results for different strategies and optimization techniques. It should be stated that in the current work we have applied a similar framework as the one described in [17]; however we define a cost function and implement different methods from operations research.

The main components of the simulation engine explained as follows: The fleet consists of shuttles and each of them is considered as an individual object. In principle, they may have different attributes (e.g. capacities, maximum number of shared trips, battery autonomy in case they are electric). Table 1 presents the list with the fleet variables. Moreover, each shuttle contains information about its current location, speed, occupancy, and a sorted list with all the drop-offs updated in the last simulation step.

Commuters are also considered as objects with attributes that defined when the travel request is produced: e.g. pick-up location, drop-off location, maximum tolerated delay time, ride sharing willingness, etc. All the parameters that are associated with commuters are listed in Table 2. The scheduler starts the operation as a request triggers the system; simply speaking the scheduler assigns a commuter to a shuttle, in an optimal manner, considering the commuter specified constraints and the cost function optimization policy. The algorithm that the scheduler follows will be described in the full paper.

The test case network that is utilized in the proposed framework is an urban network. Links (L) and nodes (N) represent roads and intersections respectively. The network constitutes a graph

Vehicle Status	Description
O	Current Occupancy
V	Velocity
$S_i, \ldots, S_n$	Stop sequence
Ride-Sharing Constraints	**Description**
C	Capacity of a vehicle
n_{share}	Number of trips that can be shared, 0 if the commuter
t_{delay}	Maximum time that a commuter will tolerate a delay in pick-up
t_{extra}	Maximum time that will be added by a ride sharing
Stop Info	**Description**
l_n	Location of stop S_n
o_n	Number of commuters connected to this stop

Table 1: Fleet specifications: vehicle status, ride-sharing variables, stop information.

Commuter Ride-Sharing Constraints	Description
o_{p}	Number of commuters
t_{pick}	time request for pick-up
L_{pick}	pick-up location
L_{drop}	drop-off location
n_{share}	maximum number of trips to be shared
t_{delay}	Maximum time that a commuter will tolerate a delay in pick-up
t_{extra}	Maximum time that will be added by a ride sharing

Table 2: Ride-sharing commuter variables.

$G(L, N)$. If a road is a two-way street, there will be two links defined for that segment. We denote with T_i the time required to travel link L_i . We assume that each trip starts/ends at a node and if a pick-up/drop-off location is in the middle of the link, it is projected to the nearest node. Using this framework, we utilize real-time information about the traffic conditions, as a weight for each link, and this can be updated to the network dynamically over time.

the simulation engine is evaluated using millions of trip data from New York City (NYC) taxi database, and different scenarios are studied. Each trip includes pick-up and drop-off time, the corresponding latitudes and longitudes, and the number of passengers. In this framework, we model shuttles and trips as separate objects and the scheduler assigns the trips to the vehicles of the fleet. The NYC dataset provides a real distribution of trip requests, which is a good basis for evaluating different ride-sharing policies. Table 3 presents some simulation parameters which are not included in the components of the vehicles or passengers.

The simulation engine can derive the best ride sharing scenario based on the parameters in an event-based approach. The requests are based on historical trip data of NYC. As a request is received, the scheduler reads the current state of all the fleet: location, capacity, next stop and tolerable delay or extra time. The scheduler computes the cost for each candidate shuttle and assigns the trip to the vehicle with the lowest cost (given that it satisfies the ride sharing constraints). If there is no shuttle available to accommodate this request, the request will be rejected; this is modeled with a large penalty parameter in the cost function. In the current work, we focus on the real time ride sharing approach, therefore our approach should consider online simulations and react immediately as a request is received. The operator needs to provide a response to the customer weather the request is accepted or not, and the status of all the fleet must be updated. This approach is a new contribution compared to the type of work in which all the requests are known in advance, which is not adequate for online ride sharing systems. The details considering the cost function and optimization algorithm are omitted here due to space limitations, and will be provided it in the full paper.

In order to have an insight about the input data to our simulation engine the following figures are illustrated: In the first stages of simulation engine development, we consider the data of one of the boroughs of NYC – Manhattan in one month in 2011. The data of Manhattan area in 2011 has been studied by many researchers in the literature and has the benefit that our results can be compared to existing benchmarks. Figures 1 and 2 present the distribution of the trip lengths for all the taxi trips that happened in the whole month of February 2011 (5,886,154 trips). In Figures 3, the spatial distribution of requests per zone is depicted for the same month, where we can clearly see the hot spots for this area.

Input Parameters	Description
m	Number of taxis
C	Default taxi capacity
n_{share}	Default number of allowable ride sharing the commuter n_{share} over writes this variable
t_{delay}	Default wait time tolerable by a commuter
t_{extra}	Default extra time added due to ride sharing tolerable by a commuter
$F(r, c)$	Cost function- cost for a taxi c to accommodate request r

Table 3: Simulation parameters.

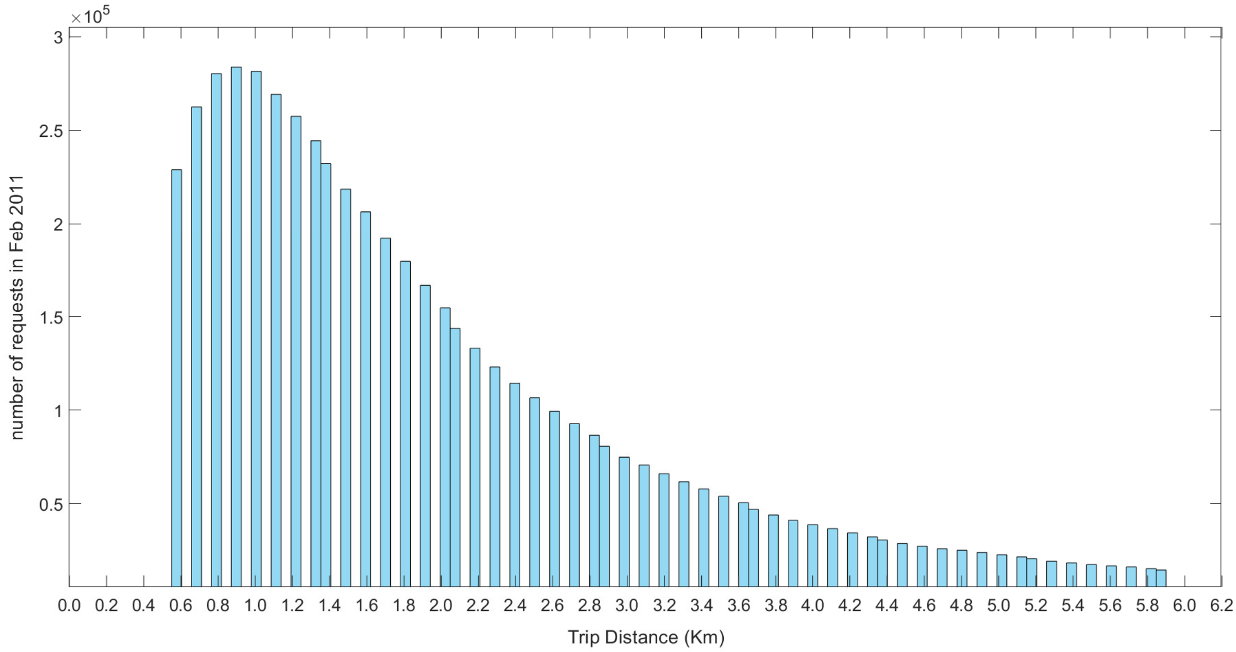

Figure 1: Distribution of trip lengths in Manhattan (February 2011).

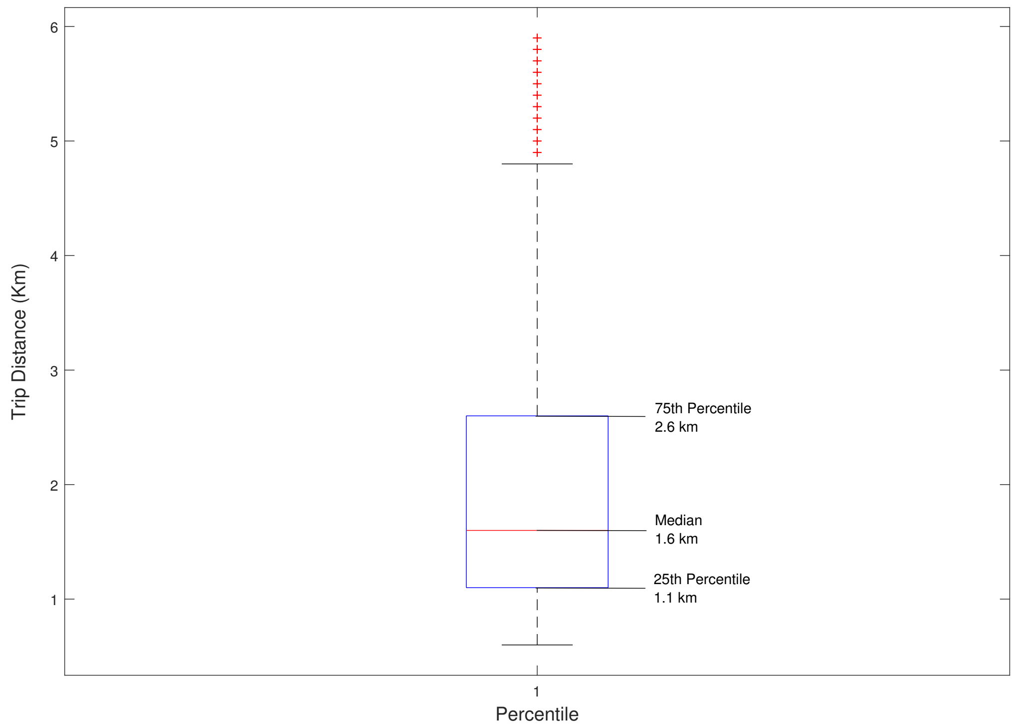

Figure 2: Boxplot of trip lengths distribution in Manhattan (February 2011).

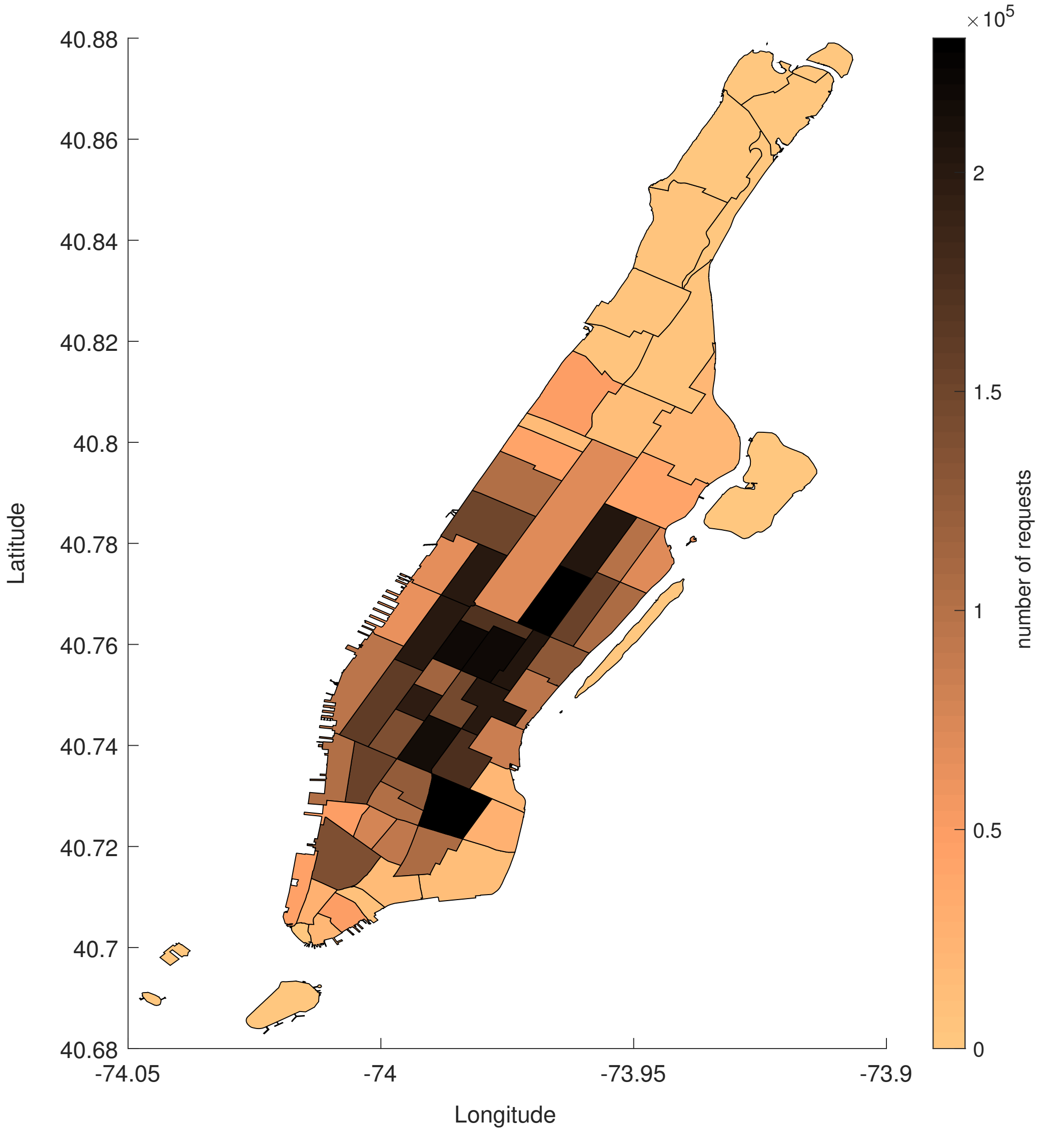

Figure 3: Number of passenger requests per zone in Manhattan area.

References

[1] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013.

[2] G. Berbeglia, J.-F. Cordeau, and G. Laporte, "A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem," *INFORMS Journal on Computing*, vol. 24, no. 3, pp. 343–355, 2012.

[3] P. Oxley, "Dial/a/ride: a review," *Transportation Planning and Technology*, vol. 6, no. 3, pp. 141–148, 1980.

[4] N. H. Wilson, J. Sussman, H.-K. Wong, and T. Higonnet, *Scheduling algorithms for a dial-a-ride system.* Massachusetts Institute of Technology. Urban Systems Laboratory, 1971.

[5] H. N. Psaraftis, “A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem,” *Transportation Science*, vol. 14, no. 2, pp. 130–154, 1980.

[6] S. C. Ho, W. Szeto, Y.-H. Kuo, J. M. Leung, M. Petering, and T. W. Tou, “A survey of dial-a-ride problems: Literature review and recent developments,” *Transportation Research Part B: Methodological*, vol. 111, pp. 395–421, 2018.

[7] J.-F. Cordeau and G. Laporte, “The dial-a-ride problem: models and algorithms,” *Annals of operations research*, vol. 153, no. 1, pp. 29–46, 2007.

[8] H. A. Waisanen, D. Shah, and M. A. Dahleh, “A dynamic pickup and delivery problem in mobile networks under information constraints,” *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1419–1433, 2008.

[9] C. H. Häll, J. T. Lundgren, and S. Voß, “Evaluating the performance of a dial-a-ride service using simulation,” *Public Transport*, vol. 7, no. 2, pp. 139–157, 2015.

[10] A. Beaudry, G. Laporte, T. Melo, and S. Nickel, “Dynamic transportation of patients in hospitals,” *OR spectrum*, vol. 32, no. 1, pp. 77–107, 2010.

[11] K. Wong, A. Han, and C. Yuen, “On dynamic demand responsive transport services with degree of dynamism,” *Transportmetrica A: Transport Science*, vol. 10, no. 1, pp. 55–73, 2014.

[12] N. Marković, R. Nair, P. Schonfeld, E. Miller-Hooks, and M. Mohebbi, “Optimizing dial-a-ride services in maryland: benefits of computerized routing and scheduling,” *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 156–165, 2015.

[13] J.-J. Jaw, A. R. Odoni, H. N. Psaraftis, and N. H. Wilson, “A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows,” *Transportation Research Part B: Methodological*, vol. 20, no. 3, pp. 243–257, 1986.

[14] M. Diana and M. M. Dessouky, “A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows,” *Transportation Research Part B: Methodological*, vol. 38, no. 6, pp. 539–557, 2004.

[15] J.-F. Cordeau and G. Laporte, “A tabu search heuristic for the static multi-vehicle dial-a-ride problem,” *Transportation Research Part B: Methodological*, vol. 37, no. 6, pp. 579–594, 2003.

[16] A. Attanasio, J.-F. Cordeau, G. Ghiani, and G. Laporte, “Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem,” *Parallel Computing*, vol. 30, no. 3, pp. 377–387, 2004.

[17] M. Ota, H. Vo, C. Silva, and J. Freire, “Stars: Simulating taxi ride sharing at scale,” *IEEE Transactions on Big Data*, vol. 3, no. 3, pp. 349–361, 2017.