*hEART 2018 - 7th Symposium of the European Association for Research in Transportation*
*September 5-7, 2018, Athens, Greece*

# A Two Phase Heuristic Approach For The Dynamic Electric Autonomous Dial-a-Ride Problem

Claudia Bongiovanni [*] Mor Kaspi [*] Nikolas Geroliminis [*]
[*]Urban Transport Systems Laboratory, School of Architecture, Civil & Environmental Engineering
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{claudia.bongiovanni,mor.kaspi,nikolas.geroliminis}@epfl.ch

## ABSTRACT

In the Dial-a-Ride-Problem (DARP) a fleet of vehicles provide shared-ride services to users specifying their origin, destination, and preferred arrival time. In the dynamic version of the DARP, some trips are booked in advance while others come in real-time. In this work, a two-phase heuristic algorithm is designed for the dynamic DARP with the use of electric autonomous vehicles (the electric Autonomous Dial-a-Ride Problem, e-ADARP). In addition to classic constraints from the DARP (i.e. time-windows, origin-destination precedence, users' maximum ride-times, vehicles' maximum route durations), the e-ADARP includes online battery management, decisions regarding detours to charging stations, partial vehicles' recharging, and selection of destination depots. The two-phase heuristic approach includes an insertion stage, in which online requests are introduced in existing routes, and an improvement stage, in which local search heuristics are applied in order to re-optimize the vehicles' plans after a number of successful insertions. An event-based simulation environment is designed to perform computational experiments on adapted benchmark instances from DARP literature and instances based on real data from Uber Technologies Inc.

**Keywords:** Dynamic Vehicle Routing, Online Dial-a-Ride Problem, Electric Autonomous Vehicles, Autonomous Mobility On-Demand, Heuristics.

## INTRODUCTION

The development of autonomous vehicles has rapidly evolved in recent years. Examples of involved stakeholders include major car manufacturers (Tesla, Mercedes), ridesharing providers (UBER, Lyft) and technology companies (Google, Intel). Full driving autonomy creates opportunities for resource sharing, as idle vehicles can relocate to serve other customers and, as a result, increase vehicle utilization. Thus, the usage of autonomous vehicles to provide ride-sharing services, and ultimately public transportation services, is gaining major attention. In this study, we consider the deployment of electric autonomous mini shuttles to provide shared on-demand transportation.

The electric Autonomous Dial-a-Ride Problem (e-ADARP) is class of the well-studied Dial-a-Ride Problem (DARP). Typically, the DARP includes operational constraints related to time-windows at customer pickup and dropoff locations, vehicles' capacities, maximum vehicles-route duration, and maximum customer ride times [*1*]. The e-ADARP integrates classic DARP with the following features: battery management, decisions regarding detours to charging stations, recharging times, and selection of destination depots. The static version of e-ADARP, i.e. when demands are fully known in advance, was defined and studied in [*2*]. Problem-specific valid inequalities were developed and incorporated in a branch-and-cut algorithm devised to exactly solve medium size instances. This study focuses on the dynamic version of the problem, that is, the case in which demand is revealed online. For a review on dynamic DARP studies, the reader is referred to Section 6 in [*3*].

The operation of electric autonomous vehicles introduces new challenges and opportunities that should be taken into account in real-time planning processes. First, autonomous vehicles offer more flexibility to modify vehicles' plans in real-time according to changing conditions. Such changes do not only correspond to the arrival of new transportation requests but also to unexpected increases in traffic conditions and modified availabilities at recharging facilities. Second, autonomous vehicles can operate non-stop and are not required to return to remote depots, hence, reducing vehicles' deadhead miles. In this work, we propose a two-phase heuristic approach to solve the dynamic e-ADARP. The first phase includes an insertion algorithm that is designed to efficiently incorporate new requests in existing vehicle routes by modifying the vehicles' schedules. Such modifications also include rerouting to recharging facilities and adaptation of recharging times. In the second phase, improvement heuristics are utilized to re-optimize vehicles' plans.

## PROBLEM DEFINITION

The input of the dynamic e-ADARP includes information regarding the transportation network, the electric autonomous vehicle fleet and the customer requests. The transportation network is represented by a complete directed graph where the nodes of the network represent all possible vehicle stops and the connecting links represent the distance between neighboring stops. A subset of these stops represent dedicated recharging facilities where idle vehicles may wait for long periods of time and recharge. Each recharging facility is characterized by a recharging rate.

The vehicle fleet is assumed to be heterogeneous. Each vehicle is characterized by a passenger capacity, battery capacity, discharging rates, average travel speed and a set of recharging facilities in which it may recharge. For each vehicle, we are given its currently planned route and schedule. A vehicle plan consists of the vehicle's current location, its planned final destination and, possibly, a set of intermediate locations. Such locations may include pickup and dropoff locations of already scheduled customers and planned visits to recharging facilities. The vehicle final destination is typically a recharging facility where the vehicle is allowed to park when idle. Note that the final destination may be modified dynamically due to newly accepted requests.

Customer requests are represented by desired pickup locations, dropoff locations and time-windows around the desired service times at these locations. In addition, each request includes a booking time, that is, the time in which the customer informs the system about her request. To avoid situations in which passengers spend too much time in the system after pickup, a maximum ride time per customer is enforced.

The main goal of the dynamic setting is to accept as many customer requests as possible. Given the accepted requests, a secondary goal is to optimize a weighted bi-objective function consisting of vehicles' total travel time (operational costs) and users' total excess time (quality of service).

The acceptance of new requests or modification of existing plans must satisfy the following conditions: 1) The time windows of all accepted customers are respected and maximal customer ride time is not exceeded. 2) The pickup and dropoff locations of each customer are visited by the same vehicle and their precedence is maintained 3) The battery level of each vehicle may not decrease below a predefined value at any point of time. 4) Vehicle capacities are not exceeded.

The output of the e-ADARP consists of a planned route and schedule for each vehicle. Namely, the sequence of locations to be visited, the service start time at each visited location and for the case of recharging facilities, the recharging duration.

## SOLUTION APPROACH

The heuristic approach proposed in this study consists of two phases, namely, insertion of new requests and improvement of existing plans. In the following, we provide a brief overview of the two phases. For each new request, as an initial screening, we only consider insertion in vehicles that can reach the pickup location of the request before its latest desired service start time. Afterwards, the insertion algorithm searches in each candidate vehicles for segments of the given plan in which the pickup and dropoff can be inserted without violating time window constraints. In order to accommodate such insertions, the service start time at some planned stops may need to be modified. For this purpose, we calculate for each stop in the given plan, the maximal time by which the service start time can be delayed/advanced without violating time-windows constraints, see [*4*]. The set of feasible segments for insertion is then further restricted by applying passenger capacity and maximum ride time considerations.

Next, the feasibility of each candidate insertion is verified by ensuring that battery restrictions are not violated. For a selected segment, the deviation to serve the new request implies an increase in battery consumption. To ensure feasibility, it is sufficient to verify that battery levels at any of the following recharging facilities do not decrease below the allowed level due to the increase in consumption. If battery constraints are violated, three strategies are applied. First, the algorithm checks whether recharging times can be prolonged at preceding visits to recharging facilities. Second, a shift backward in the following visits to recharging facilities is considered. Third, an insertion of a new visit to a recharging facility is considered. Finally, if multiple feasible insertions are identified, the algorithm selects the one that optimizes the bi-objective function, otherwise, the new request is denied.

The insertion phase described above is executed each time a new request appears. After a number of insertions, an improvement phase is executed in order to re-optimize the vehicles' plans. Local search heuristics are used to explore neighborhood solutions of the vehicles' paths. Specifically, intra-route and inter-route modifications are considered in search for better plans for the accepted customers. The frequency of execution of the improvement phase depends on the arrival rate of requests and the operational policy, that is, the processing time allowed before notifying customers about acceptance or rejection. In addition, the improvement phase includes a module that attempts to increase battery levels by shifting vehicle wait times from customer stops to recharging facilities.

## PRELIMINARY RESULTS

The proposed algorithm is implemented in Julia v.0.6.6 on a 3.60 GHz Intel(R) Core(TM) computer with 16 Gb of RAM. Numerical experiments are derived from ride-sharing data from Uber Technologies Inc. in 2011 [*2*]. The demands comprise 24'400 trips from one week in San Francisco. Two-hours aggregated demands are reported in Figure 1.

To test the proposed solution approach in a dynamic environment, we have developed an

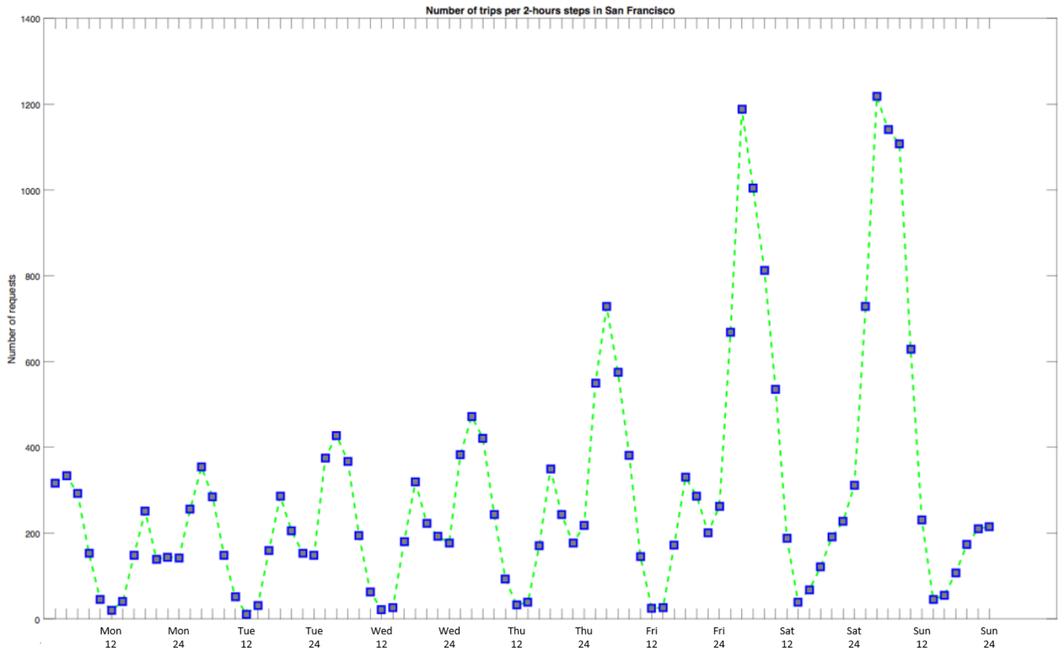


**FIGURE 1 Data used for the numerical experiment: Uber weekly demands in San Francisco**

event-based simulation. The events in the simulation include vehicle events (arrivals, departures) and customer events (generation, pickups and dropoffs). A list of customer requests is generated in advance and is given as an input to the simulation. The events list is initialized with all customer requests and the following event of each vehicle, sorted in ascending order of booking time.

The experiment design includes testing the effects of system parameters such as fleet size, vehicles' capacities, initial battery inventory and customers' booking times and maximum ride times. In the following, we present some preliminary results obtained by executing only the insertion heuristic (first phase). Specifically, we test the effect of the fleet size on the number of accepted requests, average passengers excess time, and CPU time per insertion. The 2'025 dynamic requests from the Uber data on Monday are considered for the test.Ten draws from an exponential distribution with rate parameter 5 minutes were computed to derive the requests booking times. The vehicle initial solutions do not contain any booked request. For every vehicle, the origin and destination depot is randomly selected as well as ten 15-minutes visits to recharging stations. Vehicles' capacities are homogeneous and are set to 15 seats. Users' maximum ride times are homogeneous and set to be 30 minutes. The results of some initial tests are reported in Figures 2-4.

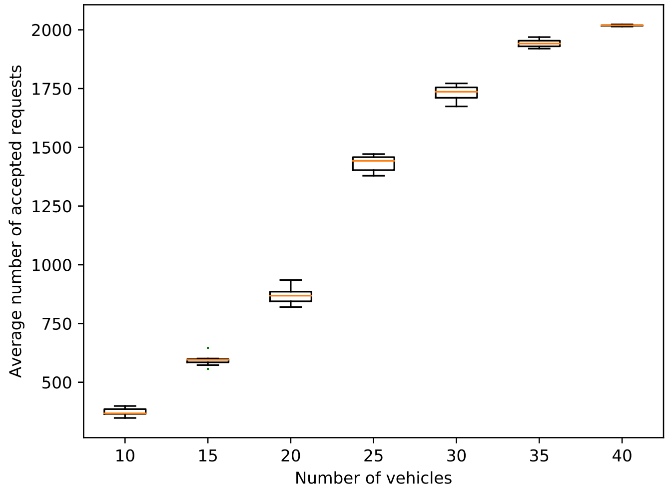


**FIGURE 2 Average number of accepted requests as a function of the fleet size**

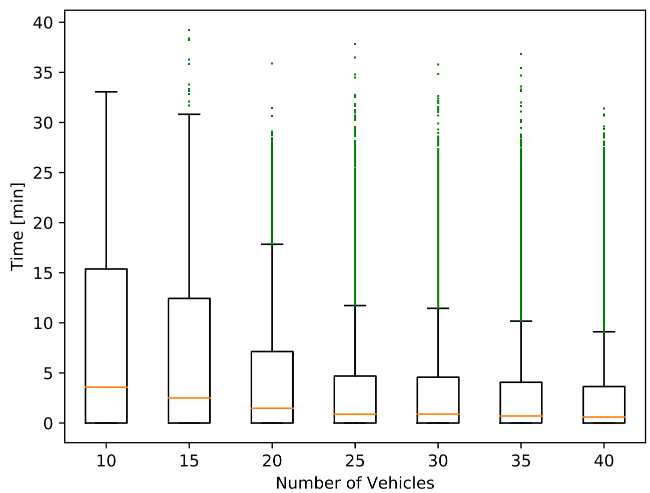


**FIGURE 3 Average excess ride time per customer as a function of the fleet size**

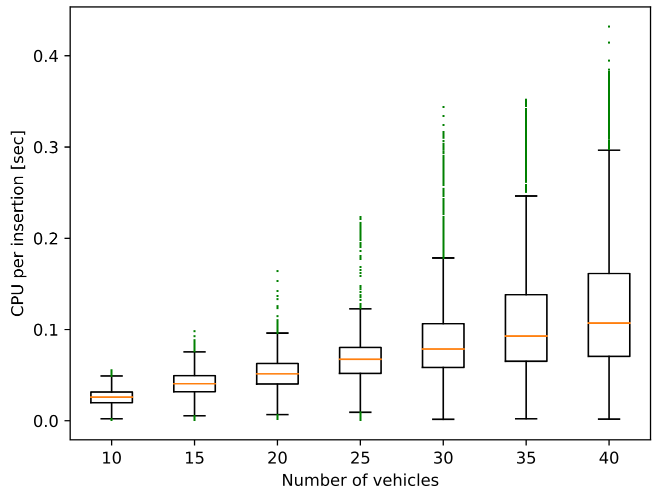


**FIGURE 4 Average CPU time per insertion as a function of the fleet size**

The average number of accepted requests as a function of the fleet size is showed in Figure 2. As it can be noticed, the average acceptance rate is non-linear and ranges from about 12.5% with 10 vehicles to 100% with 40 vehicles. Nevertheless, we observe economics of scale up to 20 vehicles. Figure 3 shows the average excess ride time per customer as a function of the fleet size. Note that the average travel time in the proposed system, being calculated as eucledian distance, is of about 7 minutes. Figure 4 shows the average CPU time per insertion as a function of the fleet size. Note that, the average CPU time per insertion increases quasi-linearly in the fleet size. However, we note that the insertion to different vehicles can be examined in parallel and thus reduce significantly the CPU time. Current efforts are focused in this direction and in the improvement of the obtained vehicle plans using local search heuristics.

## ACKNOWLEDGEMENTS

This research is supported by the Commission for Technology and Innovation (CTI) Grant # 18045.1 PFES-ES *Innovative fleet management algorithms for planning and operation of autonomous vehicles*. The authors are grateful to the CTI and the industrial partners from BestMile who provided insights on the challenges and practices of autonomous mini-buses.

## REFERENCES

1. Cordeau, J., and Laporte, G. The dial-a-ride probelm: models and algorithms. *Annals of Operations Research*, Vol. 153, 2007, pp. 29–46.
2. C.Bongiovanni, Kaspi, M., and Geroliminis, N. The electric autonomous dial-a-ride problem. *Transportation Research Part B (paper provisionally accepted)*, 2017.
3. Berbeglia, G., Cordeau, J., and Laporte, G. Dynamic pickup and delivery problems. *European Journal of Operational Research*, Vol. 202, 2010, pp. 8–15.
4. Diana, M., and Dessouky, M. M. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B: Methodological*, Vol. 38(6), 2004, pp. 539–557.