hEART 2017 - 6th Symposium of the European Association for Research in Transportation
September 12-14, 2012, Haïfa, Israel

The Electric Autonomous Dial-a-Ride Problem:
An Optimization Framework for Routing, Scheduling, and Battery Management

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

Keywords: Dial-a-Ride Problem; Mixed-Interger-Linear Programming; Branch-and-Cut; Electric Autonomous Vehicles; Intelligent City Logistics; Autonomous Mobility On-Demand.

ABSTRACT

Ride sharing is modifying urban mobility by offering reliable and convenient on-demand door-to-door services at any time. This new buiness model is of great interest in the area of city logistics and transportation engineering, as it creates an alternative between private cars (which are expensive, experience congestion, and are polluting) and public transport (which have fixed times and higher accessibility costs). Given the constant increase in demand, ride-sharing businesses are currently planning to expand their portfolio to include on-demand door-to-door transit by the use of electric Autonomous Vehicles (AVs) ([*1*],[*2*]).

The Dial-a-Ride Problem (DARP) is a class of combinatorial optimization problems developed for the operational planning of on-demand door-to-door transportation systems. The DARP consists in defining minimum cost routes and schedules for a fleet of vehicles serving a set of customers with given pickup and dropoff locations. Typically, the objective function considers the operator's interest, although some studies have also considered users' criteria [*3*]. Common operational constraints include vehicles' capacities, maximum vehicles-route duration, and maximum user ride times. In addition, service start times at pickup and dropoff locations are usually limited by time-window constraints. Finally, static and dynamic versions of the DARP have been widely studied in the literature. In the first case, demand is fully known in advance, whereas in the second case demand is revealed online. For further details on DARP models and algorithms, the reader is referred to [*3*].

The autonomous vehicles at the focus of this study are electric, thus the planning problem is supplemented with a battery management problem. That is, to ensure continuous and safe service, the batteries' state of charge should be maintained between predefined bounds. In the vehicle routing literature, several recent studies have incorporated battery management aspects ([*4*],[*5*],[*6*],[*7*]). However, to the best of our knowledge, battery management has not been studied in the context of DARP.

In this study, we introduce the Electric Autonomous Dial-a-Ride Problem (e-ADARP). The e-ADARP objective incorporates minimizing vehicles' routing costs and users' excess ride time. We focus on the static version of the problem and assume all demands need to be satisfied. The e-ADARP process flow is given in Figure 1. Dynamic cases will be considered in a future work.

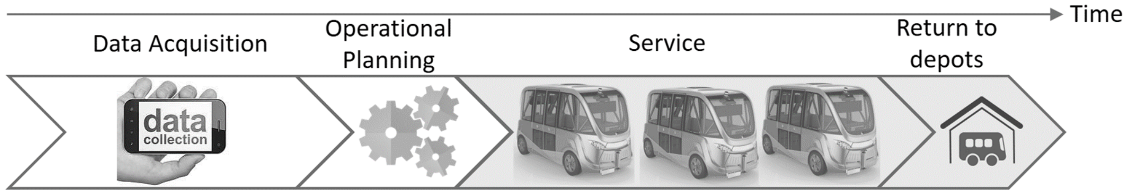

FIGURE 1 Process flow for the electric autonomous dial-a-ride transit system

As compared to classical DARP, the e-ADARP is supplemented with the following features: 1) battery management; 2) intermediate stops for vehicles' recharging; 3) vehicles are heterogeneous in terms of capacity and battery specifications; 4) vehicles may be initially located at different depots and may return to a set of optional depots; 5) no restrictions on maximum vehicle travel time are applied. For the sake of brevity, due to space limitations, a comprehensive description of the problem sets, parameters, and decision variables, as well as the model formulation is omitted from this abstract.

The e-ADARP is a generalization of the DARP and therefore is also is NP-hard, with a complexity which generally depends on the number of nodes and vehicles ([*8*], [*9*], [*10*]). To sufficiently handle large-scale instances of the static e-ADARP, we envision a framework that combines a problem decomposition method based on clustering techniques and an exact solution method, such as Branch-and-Cut. A field test under development in a Swiss city is expected to implement the framework in the near future. To date, the work has been focused on the second stage.

Branch-and-cut is a technique that has significantly improved the performance of branch-and-bound methods by implementing cutting-plane inequalities in the search tree. Such inequalities are added to the linear relaxations of integer programming problems in order to tighten their solutions [*11*]. Branch-and-Cut has been applied to solve integer and mixed-integer linear programing problems in various fields as transportation ([*12*], [*13*]) scheduling ([*14*], [*15*]), inventory planning [*16*], facility location [*17*].

At the initialization of the branch-and-cut algorithm, pre-processing is performed in order to reduce the size of the problem and strengthen the initial lower bound. This includes time-window tightening, adding symmetry breaking constraints and variable assignment. For the sake of brevity, we only present one out of the multiple pre-processing cuts that have been specifically designed from e-ADARP properties. Consider a complete directed graph $G =< V, E >$ in which the vertex set V is partitioned into sets $\{\Theta, \Delta, P, D, S\}$, where Θ is the set of origin depots, Δ is the set of destination depots, $P = \{1, ..., n\}$ and $D = \{n + i, ..., 2n\}$ are sets representing the pickup and drop-off locations of n customers, respectively, and S is the set of recharging stations along the network. Sets $\{\Theta, \Delta, S\}$ (and therefore V) are generalized to be vehicle-dependent, which we indicate through superscript k. Denoting with $x^k_{i,j}$ the set of binary decision variables such that $x^k_{i,j} = 1$ if vehicle k sequentially stops at location $i, j \in V$, and with $\beta^k_{i,j}$ the battery consumption, we may introduce arc elimination from battery considerations. In particular, arcs (i, j) and $(j, n + i)$ for $i, j \in P$ are infeasible if no vehicle has enough effective battery capacity Q^k to cover trips $\{s, i, j, n + i, n + j, s'\}$ and $\{s, i, j, n + j, n + i, s'\}$, with $s, s' \in S$ (1). That is:

$$x^k_{i,j} = 0,\;\; x^k_{j,n+i} = 0 \qquad \forall i, j \in P, s \in S^k, s' \in S^k; i! = j, s' \neq s| \qquad (1)$$
$$\beta^k_{s,i} + \beta^k_{i,j} + \beta^k_{j,n+i} + \beta^k_{n+i,n+j} + \beta^k_{n+j,s'} \geq Q^k$$
$$\beta^k_{s,i} + \beta^k_{i,j} + \beta^k_{j,n+j} + \beta^k_{n+j,n+i} + \beta^k_{n+i,s'} \geq Q^k$$

Sets of constraints with polynomial number of inequalities can also be added in the pre-processing stage. For example, a set of valid infeasible path inequalities that reflect the interaction between recharging stations and time windows are also added at the root node (2)-(3). In particular, identifying customers $i \in P$ and $j \in P; j! = i$ that violate both paths $\{n+i, s, j\}$ and $\{n+j, s, i\}$ due to time windows incompatibiity, infeasible path inequalities (2) can be added. If only one out of the two paths $\{n+i, s, j\}$ and $\{n+j, s, i\}$ is violated, then weaker infeasible path inequalities are identified (3).

$$x^k_{n+i,s} + x^k_{s,j} + x^k_{n+j,s} + x^k_{s,i} \leq 1 \quad \forall i \in P, j \in N | \{n+i, s, j\} \;\&\&\; \{n+j, s, i\} \; infeasible \tag{2}$$

$$x^k_{n+i,s} + x^k_{s,j} + x^k_{n+j,s} \leq 1 \quad \forall i \in P, j \in N | \{n+i, s, j\} \;||\; \{n+j, s, i\} \quad infeasible \tag{3}$$

The heart of the branch-and-cut algorithm lies in two components: 1) formulation of sets of valid inequalities with exponential number of constraints 2) separation heuristics devised to search these sets for violated inequalities to be added to the linear relaxations. For the second component, we design a greedy heuristics which initializes n paths and, starting from seed node $i \in P$, chooses the next node $j \in \{P \cup D \cup S\}$ such that $\sum\limits_{k \in K} x^k_{i,j}$ is maximized. The search terminates when no successive node can be found. For the sake of brevity, in this abstract we only introduce one set of valid inequalities with exponential size derived from e-ADARP properties, although we formulate more. For example, consider a path $\mathcal{P}'$ connecting the pickup location of user i and a recharging station s by a sequence of p intermediate locations $\{m_1, m_2, \ldots, m_p\}$ with $m_{1,\ldots,p} \in P \cup D \setminus n+i$. Noting that visits to recharging stations are only possible when vehicles are empty, we can select at most $p-1$ arcs from path $\mathcal{P}'$ (4). Similarly, infeasible path inequalities (5) are derived by considering a path $\mathcal{P}''$ connecting a recharging station s to a dropoff location $n+i$ by a sequence of q intermediate locations $\{m_1, m_2, \ldots, m_q\}$ with $m_{1,\ldots,q} \in D \cup P \setminus i$.

$$\sum_{k \in K} \left(x^k_{i,m_1} + \sum_{h=1}^{p-1} x^k_{m_h,m_{h+1}} + \sum_{s \in S^k} x^k_{m_p,s} \right) \leq p-1 \tag{4}$$

$$\sum_{k \in K} \left(\sum_{s \in S^k} x^k_{s,m_1} + \sum_{h=1}^{q-1} x^k_{m_h,m_{h+1}} + \sum_{s \in S^k} x^k_{m_q,n+i} \right) \leq q-1 \tag{5}$$

Preliminary results for several instances of the problem are showed in Table 1, Table 2, and Table 3. The instances presented in Table 1 are obtained by supplementing DARP benchmark instances [*12*] with four recharging stations. Vehicles have homogeneous capacities (i.e. 3 passengers/vehicle) and battery specifications. Recharging and discharging rates are set to represent 5-hours battery autonomy from a nominal capacity of 16.5 kWh. The effective battery capacity, as well as the initial battery charge is set to 14.85 kWh (i.e. vehicles are required to keep 10% of the nominal capacity at all times). The algorithm is implemented in Julia 0.5.11 using Gurobi v7.0.1 on a 3.60 GHz Intel(R) Core(TM) with 16 Gb of RAM. Current settings impose a maximum running time of 3600 seconds. Table 2 shows the results obtained by letting the commercial solver Gurobi deal with the e-ADARP problem, Table 1 shows the results obtained by introducing literature and new e-ADARP inequalities through a Branch-and-Cut framework. In conclusion, the contribution of this study is as follows: First, we introduce the e-ADARP and formulate it as a Mixed-Integer Linear Problem. Numerical results show that the proposed approach improves the algorithm performance and that it is particularly helpful when vehicles' battery management is an issue. Current work is focusing on designing more realistic instances from trips data from Uber Inc. in San Francisco and extending our framework to deal with large-scale instances.

TABLE 1 Test instances

Instance Name	Time Horizon [min]	# Customers	# Vehicles	Final Battery level [%]	Initial Battery level [%]
a2-16-14.85-0.3-FULL	480	16	2	30	100
a2-16-14.85-0.5-FULL	480	16	2	50	100
a2-16-14.85-0.6-0.7	480	16	2	60	70
a2-16-14.85-0.6-FULL	480	16	2	60	100
a2-16-14.85-0.9-0.7	480	16	2	90	70
a2-16-14.85-0.9-FULL	480	16	2	90	100
a2-20-14.85-0.5-FULL	600	20	2	50	100
a2-20-14.85-0.6-FULL	600	20	2	60	100
a2-20-14.85-0.9-0.7	600	20	2	90	70
a2-20-14.85-0.9-FULL	600	20	2	90	100
a2-24-14.85-0.3-FULL	720	24	2	30	100
a2-24-14.85-0.5-FULL	720	24	2	50	100
a2-24-14.85-0.6-0.7	720	24	2	60	70
a2-24-14.85-0.6-FULL	720	24	2	60	100
a2-24-14.85-0.9-0.7	720	24	2	90	70
a2-24-14.85-0.9-FULL	720	24	2	90	100
a3-18-14.85-0.3-FULL	360	18	3	30	100
a3-18-14.85-0.5-FULL	360	18	3	50	100
a3-18-14.85-0.6-0.7	360	18	3	60	70
a3-18-14.85-0.6-FULL	360	18	3	60	100
a3-18-14.85-0.9-0.7	360	18	3	90	70
a3-18-14.85-0.9-FULL	360	18	3	90	100

TABLE 2 Basic Model

Instance Name	Status	Solution Time [sec]	Objective Value	Gap[%]	#Constraints	#Variables	#Explored Nodes
a2-16-14.85-0.3-FULL	Optimal	231.014	303.419	0.000	15098	3180	27018
a2-16-14.85-0.5-FULL	Optimal	117.399	301.489	0.000	15098	3180	25153
a2-16-14.85-0.6-0.7	Optimal	67.516	302.781	0.000	15098	3180	24677
a2-16-14.85-0.6-FULL	Optimal	124.306	301.489	0.000	15098	3180	33488
a2-16-14.85-0.9-0.7	Optimal	106.615	301.489	0.000	15098	3180	22584
a2-16-14.85-0.9-FULL	Optimal	66.505	301.489	0.000	15098	3180	12131
a2-20-14.85-0.5-FULL	UserLimit	3600.000	366.859	5.666	21944	4584	707063
a2-20-14.85-0.6-FULL	Optimal	749.843	355.398	0.000	21944	4584	104050
a2-20-14.85-0.9-0.7	Optimal	683.335	355.398	0.000	21944	4584	65533
a2-20-14.85-0.9-FULL	Optimal	545.720	353.129	0.000	21944	4584	28434
a2-24-14.85-0.3-FULL	Optimal	1072.981	455.158	0.000	30102	6244	33998
a2-24-14.85-0.5-FULL	Optimal	420.726	450.083	0.000	30102	6244	16253
a2-24-14.85-0.6-0.7	Optimal	341.291	454.283	0.000	30102	6244	35058
a2-24-14.85-0.6-FULL	Optimal	223.010	448.435	0.000	30102	6244	14780
a2-24-14.85-0.9-0.7	Optimal	192.689	451.112	0.000	30102	6244	12476
a2-24-14.85-0.9-FULL	Optimal	239.930	448.089	0.000	30102	6244	19269
a3-18-14.85-0.3-FULL	UserLimit	3600.000	317.366	5.457	27726	5766	199680
a3-18-14.85-0.5-FULL	UserLimit	3600.000	308.184	0.508	27726	5766	248076
a3-18-14.85-0.6-0.7	UserLimit	3600.000	311.725	2.948	27726	5766	311193
a3-18-14.85-0.6-FULL	Optimal	2545.469	306.955	0.000	27726	5766	101159
a3-18-14.85-0.9-0.7	Optimal	1705.231	306.955	0.000	27726	5766	146161
a3-18-14.85-0.9-FULL	Optimal	2455.816	306.955	0.000	27726	5766	208585

TABLE 3 Branch-and-Cut framework

Instance Name	Status	Solution Time [sec]	Objective Value	Gap[%]	#Constraints	#Variables	#Explored Nodes
a2-16-14.85-0.3-FULL	Optimal	13.517	303.419	0.000	21777	3180	727
a2-16-14.85-0.5-FULL	Optimal	5.301	301.489	0.000	21777	3180	35
a2-16-14.85-0.6-0.7	Optimal	8.353	302.781	0.000	21777	3180	94
a2-16-14.85-0.6-FULL	Optimal	7.783	301.489	0.000	21777	3180	53
a2-16-14.85-0.9-0.7	Optimal	5.051	301.489	0.000	21777	3180	48
a2-16-14.85-0.9-FULL	Optimal	6.011	301.489	0.000	21777	3180	69
a2-20-14.85-0.5-FULL	Optimal	1165.742	365.211	0.000	32389	4584	89759
a2-20-14.85-0.6-FULL	Optimal	89.270	355.398	0.000	32389	4584	2847
a2-20-14.85-0.9-0.7	Optimal	126.796	355.398	0.000	32389	4584	4889
a2-20-14.85-0.9-FULL	Optimal	99.946	353.129	0.000	32389	4584	4155
a2-24-14.85-0.3-FULL	Optimal	384.137	455.158	0.000	44950	6244	15465
a2-24-14.85-0.5-FULL	Optimal	123.103	450.083	0.000	44950	6244	3548
a2-24-14.85-0.6-0.7	Optimal	196.324	454.283	0.000	44950	6244	7253
a2-24-14.85-0.6-FULL	Optimal	138.579	448.435	0.000	44950	6244	3330
a2-24-14.85-0.9-0.7	Optimal	117.333	451.112	0.000	44950	6244	2699
a2-24-14.85-0.9-FULL	Optimal	93.992	448.089	0.000	44950	6244	1801
a3-18-14.85-0.3-FULL	UserLimit	3600.000	317.366	0.736	38110	5766	138166
a3-18-14.85-0.5-FULL	Optimal	482.602	308.184	0.000	38110	5766	9222
a3-18-14.85-0.6-0.7	Optimal	1577.929	311.725	0.000	38110	5766	44074
a3-18-14.85-0.6-FULL	Optimal	579.538	306.955	0.000	38110	5766	10144
a3-18-14.85-0.9-0.7	Optimal	481.185	306.955	0.000	38110	5766	16862
a3-18-14.85-0.9-FULL	Optimal	551.245	306.955	0.000	38110	5766	8016

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. Alba, D. Uber is now letting passengers schedule rides in advance. *Wired*, September 2016. http://www.wired.com/2016/06/uber-now-letting-passengers-schedule-rides-advance/. Accessed: September 27, 2016.
2. Taylor, E., and Wolde, H. T. Uber seeking to buy self-driving carts:source. *Reuters*, March 2016. http://www.reuters.com/article/us-daimler-uber-idUSKCN0WK1C8. Accessed: June 1, 2016.
3. Cordeau, J., and Laporte, G. The dial-a-ride probelm: models and algorithms. *Annals of Operations Research*, Vol. 153, 2007, pp. 29–46.
4. Pelletier, S., Jabali, O., and Laporte, G. 50th anniversary invited article - goods distribution with electric vehicles: Review and research perspectives. *Transportation Science*, Vol. 50(1), 2016, pp. 3–22.
5. Schneider, M., Stenger, A., and Goeke, D. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, Vol. 48(4), 2014, pp. 500–520.
6. Desaulniers, G., Errico, F., Irnich, S., and M.Schneider. Exact algorithms for electric vehicle-routing problems with time windows. *Les Cahiers du GERAD*, Vol. G-2014-110, 2014.
7. Doppstadt, C., Koberstein, A., and Vigo, D. The hybrid electric vehicle-traveling salesman problem. *European Journal of Operational Research*, Vol. 253, 2016, pp. 852–842.
8. Baugh, J., Kakivaya, G., and Stone, J. Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. *Engineering Optimization*, Vol. 30, 1998, p. 2.
9. Healy, P., and Moll, R. A new extension of local search applied to the dial-a-ride problem. *European Journal of Operations Research*, Vol. 83 (1), 1995, pp. 83–104.
10. Savelsbergh, M. Local search in routing problems with time windows. *Annals of Operations Research*, Vol. 4 (1985/6), 1985, pp. 285–305.
11. Mitchell, J. E. *Branch-and-cut algorithms for combinatorial optimization problems. Handbook of applied optimization*. Handbook of applied optimization, 2002.
12. Cordeau, J. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, Vol. 54(3), 2006, pp. 573–586.
13. Ropke, S., Cordeau, J., and Laporte, G. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, Vol. 49(4), 2007, pp. 258–272.
14. Hoffman, K. L., and Padberg, M. Solving airline crew scheduling problems by branch-and-cut. *Management Science*, Vol. 39(6), 1993, pp. 657–682.
15. Caccetta, L., and Hill, S. An application of branch and cut to open pit mine scheduling. *Journal of global optimization*, Vol. 27(2-3), 2003, pp. 349–365.
16. Belvaux, G., and Wolsey, L. bc-prod: A specialized branch-and-cut system for lot-sizing problems. *Management Science*, Vol. 46(5), 2000, pp. 724–738.
17. Labbé, M., Yaman, H., and Gourdin, E. A branch and cut algorithm for hub location problems with single assignment. *Mathematical Programming*, Vol. 102(2), 2005, pp. 371–405.