

Electric Vehicle Routing Problem with Time Dependent Waiting Times at Recharging Stations

Merve Keskin

Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey
CIRRELT, Montréal, Canada

Gilbert Laporte

CIRRELT and Department of Management Sciences, HEC, Montréal, Canada

Bülent Çatay

Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey
Email: catay@sabanciuniv.edu

1 Introduction and Problem Description

The Electric Vehicle Routing Problem with Time Windows (EVRPTW) was introduced by [1] as an extension to the Green Vehicle Routing Problem of [2]. It is a variant of the Vehicle Routing Problem with Time Windows where a fleet of electric vehicles (EVs) is used instead of internal combustion engine vehicles. Since the EV has a limited driving range it may need to have its battery recharged at a station en route. In [1], it is assumed that the battery is fully recharged at any state of charge (SoC) and the duration is linearly proportional to the amount of energy transferred. The full recharge restriction was later relaxed in [3]. Furthermore, it is shown that the recharge duration is a concave function of the energy transferred and charging slows down when the energy level reaches 85% of the battery capacity [4-6].

The existing literature assumes that recharging at the station starts as soon as the EV arrives at the station. However, in practice the number of chargers in a station is limited and they may not be available at the time of arrival. So, the EV may have to queue before recharging and this waiting time may affect the routing decisions. The waiting time may vary depending on the location of the station and the time of the visit. Some variations in the waiting time are difficult to predict. For instance, if a charger is out of order, the EVs should queue in at another charger, if available, or go to a nearby station. On the other hand, some waiting time can be foreseen, for example when there are more vehicles on the roads during the rush hours, which translates in increased demand for recharging. In this study, we extend the EVRPTW by considering waiting times at the

recharging stations. We assume an M/M/1 queueing system and use expected waiting times to predict the queue lengths. We also use a nonlinear charging function and allow late arrivals at the customers and the depot with penalties. Moreover, we assume that the battery is operated between 10% and 90% of its capacity since it degrades faster beyond these limits [5]. Recently, [7] has addressed a similar environment where the stations have limited number of chargers (1, 2 or 3) and an EV may need to wait for service if the chargers are busy recharging other EVs in the fleet. In this problem, the use of the chargers depends on the routing and charging decisions made, whereas in our setting the queue lengths are independent from our fleet, such as is the case at public stations.

In this context, we split the planning horizon (usually a day) into a predetermined number of time intervals such as morning, afternoon, evening, and night, and the average queue lengths at the stations vary according to the arrival time. We assume different EV arrival rates depending on the time of the day, e.g. the stations are less busy in the morning since the EVs depart with a full battery whereas the energy on the battery is usually consumed in the afternoon or evening, hence the EVs need recharging to continue their routes. The routing decisions are then made according to these time-dependent waiting times at the recharging stations. The objective is to minimize a total cost function which includes the energy cost, cost of vehicles, driver wages, and penalties associated with time window violations at the customers and the depot.

2 Solution Methodology

We assume that each recharging station is equipped with a single charger. The arrivals are Poisson with mean λ , and the service times follow an exponential distribution with parameter μ . We apply a *first come first served* queue discipline where the first in first out property holds, i.e. the vehicles leave the station in the order they have arrived.

Our solution approach is a matheuristic which integrates the Adaptive Large Neighborhood Search (ALNS) method with an exact method. ALNS is a metaheuristic framework that iteratively destroys a solution and then repairs it using a greedy heuristic [8]. It has been successfully applied for solving various VRP variants in the recent literature. It uses several operators to destroy and repair the solution which are selected adaptively based on their past performances in improving the solution. In this study, we use the customer removal and insertion operators introduced in [3]. The repaired solution is accepted according to a simulated annealing criterion which allows the acceptance of worse solutions with a certain probability. Furthermore, we optimize the charging station selection and recharge quantity decisions of the vehicles by solving a mixed integer program using a commercial solver every after a predetermined number of iterations Δ . Basically, the best solution obtained in the last Δ iterations is further enhanced by optimizing charging de-

cisions while preserving the sequence of the customers. To solve this fixed-route problem fast, we formulate an effective mathematical model.

3 Experimental Results

To validate the performance of the proposed approach, we perform experiments using 36 small instances generated by [1]. We compare our results with those obtained with CPLEX. The service rate μ is calculated as explained in Section 1. We assume $\lambda = \mu/3$ for the least busy time interval and set the arrival rate λ_t in time interval t randomly by multiplying λ with a positive scalar. All stations are assumed to have the same arrival rates for each time interval. Finally, the average waiting time at a station in time interval t is calculated as $w_t = \lambda_t / (\mu - \lambda_t)$. Δ is set to 200 and matheuristic terminates after 10,000 iterations. All experiments are conducted on an Intel Xeon E5 3.30 GHz processor and 32 GB of RAM. The mathematical model and the matheuristic are coded in Java and solved by CPLEX 12.6.2 with default settings.

Table 1: Results on Small Size Instances

Instance	CPLEX		Matheuristic		Instance	CPLEX		Matheuristic	
	Cost	Time	Cost	Time		Cost	Time	Cost	Time
C101C10	3562.18	2233	3562.18	10	C103C15	3589.52	7200	3583.61	26
C104C10	2348.00	7200	2347.63	31	C106C15	3476.10	7200	3476.10	30
C202C10	3904.82	7200	3904.82	38	C202C15	4576.78	7200	3988.89	118
C205C10	3417.17	160	3417.17	25	C208C15	3753.73	7200	3753.73	125
R102C10	1611.83	7200	1608.37	12	R102C15	2313.14	7200	2313.14	20
R103C10	1271.70	7200	1271.70	11	R105C15	2185.12	7200	2185.05	16
R201C10	1289.27	6272	1289.27	48	R202C15	1738.07	7200	1660.70	110
R203C10	1319.45	7200	1319.45	55	R209C15	1494.19	7200	1487.39	130
RC102C10	2211.27	7200	2211.27	8	RC103C15	2342.45	7200	2342.45	10
RC108C10	2073.69	7200	2073.69	9	RC108C15	2147.50	7200	2147.50	14
RC201C10	1605.90	7200	1605.90	29	RC202C15	1943.72	7200	1934.67	86
RC205C10	1534.29	7200	1534.29	26	RC204C15	1409.21	7200	1368.43	140

The results are reported in Table 1. The computation times are given in seconds. We set the time limit for CPLEX to 7200 seconds. So, if the computation time of CPLEX is less than this limit, the solution is optimal; otherwise, it is the best upper bound found. The matheuristic solutions are the best of 10 runs whereas the computational times are average of 10 runs. Since both CPLEX and our method found the optimal solution in all 5-customer instances, we did not include them in the table. On average, the optimal solutions are attained in 206.8 and 10.7 seconds by CPLEX and the matheuristic, respectively. In 10-customer instances, the matheuristic provides better solutions in two instances which are highlighted in bold. In the remaining 10 instances, CPLEX and the matheuristic find the same solutions but CPLEX run times are significantly more and reach the time limit

in most of the cases. In 15-customer instances the matheuristic achieves better solutions in 7 out of 12 instances. Furthermore, the run time is significantly better than CPLEX for all cases. Overall, we can claim that the proposed method outperforms CPLEX both in solution quality and computational time.

In conclusion, the proposed approach is effective in solving all small size instances. Our future work will concentrate on testing its performance on large instances under various scenarios of queue characteristics.

References

- [1] M. Schneider, A. Stenger and G. Goeke, “The electric vehicle-routing problem with time windows and recharging stations”, *Transportation Science*, 48(4), 500-520, 2014.
- [2] S. Erdoğan, E. Miller-Hooks, “A green vehicle routing problem”, *Transportation Research Part E: Logistics and Transportation Review*, 48(1), 100-114, 2012.
- [3] M. Keskin, B. Çatay, “Partial recharge strategies for the electric vehicle routing problem with time windows”, *Transportation Research Part C: Emerging Technologies*, 65: 111-127, 2016.
- [4] T.M. Sweda, I.S. Dolinskaya and D. Klabjan, “Optimal recharging policies for electric vehicles”, *Transportation Science*, 51(2), 457-479, 2017.
- [5] S. Pelletier, O. Jabali, G. Laporte, M. Veneroni, “Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models”, *Transportation Research Part B: Methodological*, 103, 158-187, 2017.
- [6] A. Montoya, C. Guéret, J.E. Mendoza, J.G. Villegas, “The electric vehicle routing problem with nonlinear charging function”, *Transportation Research Part B: Methodological*, 103, 87-110, 2017.
- [7] A. Froger, J.E. Mendoza, O. Jabali, G. Laporte, “A Matheuristic for the Electric Vehicle Routing Problem with Capacitated Charging Stations”, Technical Report 2017-31, CIRRELT, 2017.
- [8] S. Ropke, D. Pisinger, D. “An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows”, *Transportation Science*, 40(4), 455-472, 2006.