



HAL
open science

Extending JSprit to solve electric vehicle routing problems with recharging

Ayman Hassan Mahmoud, Tarek Chouaki, Sebastian Hörl, Jakob Puchinger

► To cite this version:

Ayman Hassan Mahmoud, Tarek Chouaki, Sebastian Hörl, Jakob Puchinger. Extending JSprit to solve electric vehicle routing problems with recharging. The 13th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 5th International Conference on Emerging Data and Industry 4.0 (EDI40), 2022, Porto, Portugal. pp.289-295, 10.1016/j.procs.2022.03.039 . hal-03655329

HAL Id: hal-03655329

<https://hal.science/hal-03655329>

Submitted on 22 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License



The 13th International Conference on Ambient Systems, Networks and Technologies (ANT)
March 22 - 25, 2022, Porto, Portugal

Extending JSprit to solve electric vehicle routing problems with recharging

Ayman Mahmoud^{a,*}, Tarek Chouaki^{a,b}, Sebastian Hörl^b, Jakob Puchinger^{a,b}

^aLaboratoire Genie Industriel, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette 91190, France

^bInstitut de Recherche Technologique SystemX, Palaiseau 91120, France

Abstract

In this article, we present our adaptation of the Ruin-and-Recreate (R&R) algorithm to solve the electrical vehicle routing problem with time windows and multiple trips. We implement this adaptation in JSprit, an open-source vehicle routing problem solver. We implement the framework for a case study of Lyon, France. We assess the efficiency impact of adding charging constraints to a simulation of a fleet of autonomous delivery robots.

© 2020 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the Conference Program Chairs.

Keywords: electric, vehicle, routing, vrp, jsprit, optimization

1. Introduction

With the evolution of technology, policy-makers are working towards making urban areas less polluted and carbon-free. One of the main factors affecting the quality of air and CO₂ emissions are freight flows. There is great interest in transforming the existing logistics fleet by integrating battery electrified vehicles (BEV). Especially, a large potential is seen for last-mile logistics, where distances are short and the flows of goods may be consolidated [12], [11].

Clearly, with electrification come operational challenges related to vehicle autonomy and the necessity to recharge. Hence, a range of research endeavors is focused on solving problems related to the specific characteristics of BEVs, such as the problem of locating recharging stations [18] or the electric vehicle routing problem [14, 5].

For the latter class of problems, the present paper introduces distance constraints and recharging activities to JSprit, an open-source solver for common vehicle routing problems. In Section 2, we present the theoretical background and a literature review on existing use cases of JSprit, while Section 3 covers the implementational aspects of extending

* Corresponding author.

E-mail address: ayman.mahmoud@centralesupelec.fr

the framework. Our adaptations are exemplified with a case study for Lyon, France, in Section 4. We conclude with a discussion of the proposed approach in Section 5.

2. Background

Generally, Vehicle Routing Problems (VRP) have the aim to arrange a given set of transport jobs such as delivery pick-ups and drop-offs, or depot visits into routes for a given (or variable) vehicle fleet. Choices to be made by algorithms solving a VRP are which job to assign to which vehicle, and how to temporally order the jobs along a vehicle's route. Furthermore, specific problem formulations can be derived by introducing constraints such as requiring that for a specific delivery, a pick-up job comes before a drop-off job, that every vehicle can only cover a limited distance before returning to a depot, that certain jobs need to be performed within predefined time windows, and many more. The objective is often to minimize the total distance travelled, but more complex cost formulations exist considering the number of vehicles used, or the time spent. Daniele Vigo and Paolo Toth [4] provide an overview of VRP variants and their mathematical formulation.

2.1. *Ruin & Recreate heuristics*

Schrimpf et al. [15] present the Ruin & Recreate principle, which is a large neighborhood search that combines elements of simulated annealing and threshold-accepting algorithms. Highlighting its performance for complex optimization problems with discontinuous search spaces, and problems with many constraints, the researchers have tested their meta-heuristic on a wide range of VRP formulations.

In general, the Ruin & Recreate heuristic follows three steps: First, an initial, feasible solution needs to be constructed. Such a feasible solution is an ensemble of routes (sequences of jobs) depending on the VRP formulation for each vehicle that fulfill a predefined set of constraints. After that, multiple iterations of executing the Ruin & Recreate steps are performed. The Ruin step selects a set of jobs to ruin based on a variety of existing strategies [15, 3]. These jobs are then removed from the existing job sequences. Following the Ruin step, the Recreate step finds a new configuration by re-inserting some or all of the removed jobs into the schedules such that, once again, a feasible solution is obtained. The Ruin & Recreate heuristic, hence, yields a feasible solution of the problem after every iteration, but the quality may vary. For that purpose, a cost function is defined which can be based, for instance, on the total distance or time driven by the vehicles. The cost of a route is calculated given the sequence of jobs and, usually, based on cost matrices indicating the cost of moving between their respective locations. Finally, the algorithm decides whether to keep the previous solution or to continue with the new one, based on threshold acceptance, temporarily allowing a worsened objective to find even better solutions in future steps.

2.2. *The JSprit framework*

JSprit is an open-source toolkit written in Java that implements the Ruin & Recreate principle in a modular and extensible way. It solves a wide range of VRP variants including Capacitated VRP (CVRP), Multiple Depot VRP, VRP with Time Windows (VRP-W), VRP with backhauls, VRP with Pickup and Delivery (VRP-PD), VRP with heterogeneous fleet, the Traveling Salesman Problem (TSP), and the Dial-a-Ride Problem (DARP)¹.

The open-source solver has been used in a range of research activities, for instance, by Villanueva [17] to address real-life waste collection problems as part of the ODL (Open Door Logistics) framework, or by Skalna Karkula [16], who presents a comparative study on open-source tools for solving capacitated vehicle routing problems with time windows.

JSprit can be integrated directly with the agent- and activity-based transport simulation framework MATSim [10] with applications on a wide range of topics. Martins-Turner et al. [9] use JSprit to generate tour plans for urban last mile supply of shops, for a case study of Berlin. For the same urban context, Schlenther et al. [13] use the tool in a study on multi-use autonomous taxis for the transport of passengers and goods. Bean and Joubert [2] present a study

¹ <https://github.com/graphhopper/jsprit>

on receiver agents with autonomous decision-making in MATSim. The emerging logistics flows are optimized using JSprit.

So far, JSprit has not yet been used to solve electric VRP instances with recharging. However, Ewert et al. [6] use the tool to solve parcel delivery problems with autonomy constraints. They introduce a vehicle type-specific range constraint and demonstrate their model on real data in retail food distribution. The consumption of electric energy is based on travel time and defined by vehicle type. Contrary to the present work, no recharging stations are considered.

2.3. Contribution

The contribution of the present paper is the introduction of (1) a flexibly adjustable energy consumption model into JSprit, (2) autonomy constraints, and (3) explicit charging activities at distinct locations (charging stations) to the implementation of Ruin & Recreate in JSprit. Furthermore, experimental results on a custom case study are presented.

3. Implementation

The implementation of the Ruin & Recreate heuristic in JSprit has been adjusted to consider energy consumption and recharging at distinct charging stations with specifiable consumption and charging patterns. For that purpose, three major components have been added to the framework: (1) functionality to calculate the state of charge (SoC) of a vehicle at any point along a route, (2) a new insertion strategy that takes into account charging stations, and (3) a Ruin strategy that is aware of charging constraints. These components are described in the following.

3.1. Calculating the state of charge

In general, the cost of a route in JSprit is derived from following the movements and jobs along a route and summing up costs. Such costs can come from a predefined matrix of routed distances, but they can also be context-aware, e.g. taking into account the current time of day or the load of a vehicle at any point along the route. The latter can be expressed as a state variable in JSprit, which can be updated along a route.

To consider the consumption of battery power, we introduce a new state variable to each vehicle, which represents the remaining energy left in each vehicle's battery. Traversing each vehicle's route, we decrease the energy level following the formulation of energy consumption used by Schneider et al. [14], which incorporates speed, gradients, and load.

Whenever a vehicle visits a charging station, battery power will be recharged. Spending time at a depot that contains a charging station is different from visiting the charging station. The charging duration is calculated based on predefined recharging rate r per vehicle type. Figure 1 shows an example of visualizing the SoC along a vehicle route with multiple stops at charging stations.

3.2. Inserting charging stations

JSprit already comes with strategies for inserting new jobs into existing vehicle routes. These insertion strategies are used to obtain an initial solution, where JSprit iteratively inserts all existing jobs into a set of initially empty vehicle routes. Later, during the Recreate phase, these strategies are used to re-insert jobs that have been selected during the Ruin phase.

The insertion heuristics consider a set of constraints that need to be fulfilled. For the purpose of working with electric vehicles, an autonomy constraint has been added which makes sure that no insertion leading to a route with negative SoC at any point along the route is accepted. This constraint is checked during all job insertions for the creation of the initial solution and the Recreate step.

To facilitate the use of charging stations, the insertion strategies have been modified. Whenever a job is to be inserted, it is first tested whether the autonomy constraint would be violated. If this is the case, the modified insertion strategy tries to insert a combination of a charging job, and the planned job to insert. To do so, the algorithm, first, finds the closest M charging stations based on the location of the preceding job. Then, it evaluates the global insertion cost after inserting the job and a charging activity using any of the closest stations at *any* point along the route (before

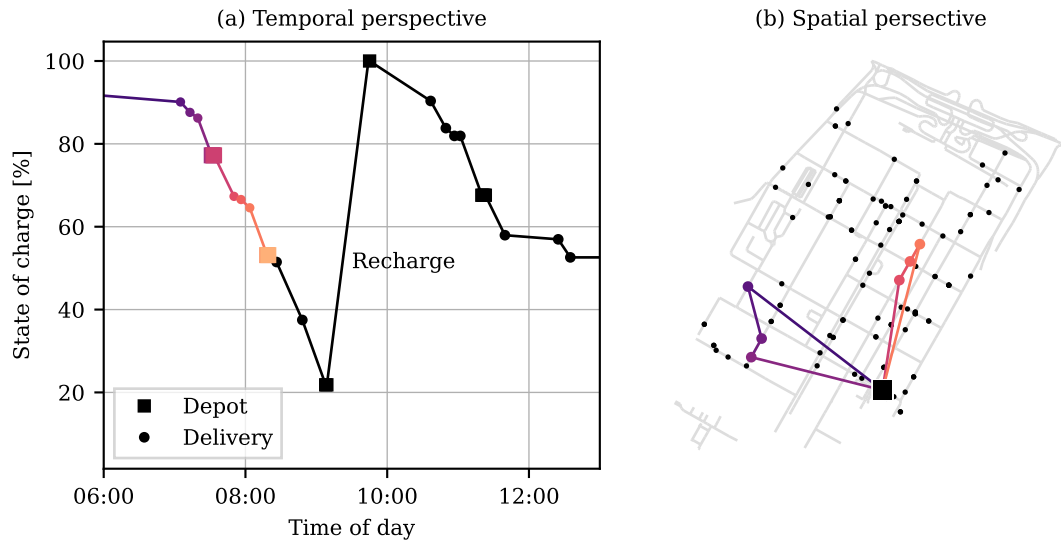


Fig. 1. Example for a temporal and spatial analysis of the SoC and the routes generated by the algorithm for the time between 06:00 at 13:00

any activity already added to the route). If any of the temporary configurations leads to a feasible solution, the best one is accepted and both the charging job and the stop job are inserted. Otherwise, the proposed insertion is tagged infeasible, and the new job will need to be added at a different location proposed by the standard procedure.

3.3. Ruining existing solutions

By default, JSprit would use the existing Ruin strategies to remove any activity along the route, including those for recharging. However, this may lead to infeasible solutions which can not be repaired by the simple insertion process as described above (see discussion (5) below).

To counteract the emergence of infeasible solutions, the existing Ruin strategies have been modified. Whenever a recharging activity is to be deleted from a route, it is first checked whether this would lead to the violation of the vehicle's energy autonomy constraint. If this is the case, the charging activity is left inside the route.

4. Case Study

We test the new functionality that has been added to the JSprit framework on a case study for parcel deliveries in Lyon. The experiments are inspired by existing studies that propose the use of delivery robots for sustainable last-mile logistics [1, 19].

To set up the case, a synthetic population for Lyon is created, based on the open-source and open data methodology presented in [8]. Such a synthetic population is an individual-based artificial representation of the real population of the city, including sociodemographic attributes on the personal and household-level. Based on these sociodemographic and statistical information on the average number of parcel deliveries of household strata as reported by [7], a discrete number of parcels for an average day is generated for every household. For the case study, households with addresses in the Confluence peninsula South of the old town of Lyon, France are chosen, leading to about 128 parcels to be delivered (see Figure 1). For each parcel, multiple delivery windows are defined, based on the presence of household members at any time during the day.

In the test case, the parcels are delivered by automated electric delivery robots from an Urban Consolidation Center (UCC) located at the Eastern shore of the study area. Each vehicle can carry a maximum of 4 parcels at a time and perform multiple tours during the day. The objective is to minimize the total distance driven by the entire fleet during one day. Distances are calculated based on the shortest path through the road network, and travel times are derived

Table 1. Optimization results with sensitivity against increased demand.

Scaling factor	1.0	1.3	1.6	2.0	eVRP
Parcels	128	165	198	256	
Vehicles	4	4	4	5	
	7	8	9	13	✓
Distance [km]	55.19	63.26	71.74	95.23	
	59.29	62.21	71.89	90.49	✓
Charging activities	4	4	5	4	✓

using a speed of 3km/h . For each vehicle of the variable size fleet, a battery capacity of $8000\text{mAh LiPo } 18.5\text{V}$ is assumed with a consumption of 24.67 Wh/km based on data obtained from the *Starship* delivery robot². The initial SoC is set to 100%.

To summarize, we present an electric Multi-Trip Vehicle Routing Problem with Time Windows (e-MTVRPTW) demonstrating the interaction of existing functionality in JSprit with our electric components, and allowing the delivery robot to return to depot to pick up shipments or to recharge.

The pickup and delivery problem is optimized using JSprit, with and without charging constraints. In each case, 1,000 iterations are performed. Furthermore, to test the sensitivity of the solutions, the experiments are repeated with 1.3, 1.6 and two times the baseline demand.

As an example of the detailed information that can be derived from the JSprit runs, Figure 1a shows the SoC at and in between depots, delivery and recharging activities, for one vehicle route. Figure 1b shows the spatial trace of a vehicle. Using this detailed information, aggregated metrics can be derived for each scenario.

Table 1 shows the optimization results in terms of problem size, the number of vehicles needed and the distance driven, as well as the number of charging activities. The last column indicates whether the problem has been solved with or without charging and autonomy constraints.

The results show that for the case without electric consumption constraints, four vehicles are needed for three demand levels, only for the highest demand scenario, five vehicles are necessary to deliver all parcels within the constraints given by the delivery time windows. In the eVRP case, the number of vehicles needed doubles in the solutions found by JSprit and even increases to 13 vehicles for the highest demand scenario. Regarding charging activities, the results show that about every second vehicle is obliged to perform a charging activity in these scenarios. In terms of distance, the results stay very similar, as the charging station is located directly at the depot.

5. Discussion and Conclusion

Several design decisions have been taken when implementing charging activities into JSprit, mainly in terms of the Ruin strategy. The way that Ruin & Recreate is used in existing literature is to maintain feasible solutions after each step. Also, the existing JSprit modules are implemented such that this condition holds implicitly after applying a Ruin strategy. No additional checks for constraint violations are performed after. However, in the case of removing a recharging activity from a vehicle's route, the energy consumption constraint may become violated.

Two options have been considered to solve the issue: As one option, a Recreate strategy may have been introduced that is able to repair any ruined solution. While it would be the more flexible option in the sense of the conceptual structure of the Ruin & Recreate approach, it would require inserting charging activities independently of the jobs, which, in turn, would lead to considerable changes in the core implementation of JSprit, where insertions are job-based.

Hence, our chosen approach is to introduce customized versions of the existing Ruin strategies to avoid creating ruined solutions with autonomy constraint violations. This has the advantage of being less invasive to the existing code base and following closer the implicit policy of existing Ruin strategies. However, the approach prevents the

² <https://www.starship.xyz/>

removal of recharging activities that are relevant for the ruined route, but not necessarily for a follow-up solution. This adds more complexity to the algorithm and may also restrict the search space or slow down convergence. While we have verified our chosen approach to be robust, the topic provides potential for future research.

For now, our implementation has been tested with up to 500 jobs. Beyond that limit, the run time of the extensive search for potential charging activities along a route becomes prohibitive. To remedy the problem, more intelligent or heuristic strategies for inserting charging activities may be introduced in the future. Additionally, the existing multi-threading functionality of JSprit may be used. While it is available for standard problem formulations, it has not been adapted to be compatible with our eVRP extensions yet.

Apart from the case study above, which shows the practical relevance of our approach, the extension has been tested on existing problems, such as the eVRP-TW instances presented by Schneider et al. [14]. While they allow minor constraint violations in their solutions obtained using a VNS-TS algorithm to reach a feasible one by corrective iteration eventually, the Ruin & Recreate approach always yields a feasible solution, if any. Furthermore, we have run test instances for the Green Vehicle Routing Problem introduced by Erdoğan and Miller-Hooks [5]. The first benchmark results, which show promising and competitive results, will be analyzed systematically in future work.

Acknowledgements

The code related to the experiments presented in this paper can be obtained from the authors and will be published as open source after publication.

This paper presents work developed at IRT SystemX in the scope of the project LEAD, which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 861598. The content of this paper does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in this paper lies entirely with the authors.

References

- [1] Bakach, I., Campbell, A.M., Ehmke, J.F., 2021. A two-tier urban delivery network with robot-based deliveries. *Networks* 78, 461–483. doi:[10.1002/net.22024](https://doi.org/10.1002/net.22024).
- [2] Bean, W.L., Joubert, J.W., 2019. Modelling receiver logistics behaviour. *Procedia Computer Science* 151, 763–768. doi:[10.1016/j.procs.2019.04.103](https://doi.org/10.1016/j.procs.2019.04.103).
- [3] Christiaens, J., Vanden Berghe, G., 2020. Slack Induction by String Removals for Vehicle Routing Problems. *Transportation Science* 54, 417–433. doi:[10.1287/trsc.2019.0914](https://doi.org/10.1287/trsc.2019.0914).
- [4] Daniele Vigo, Paolo Toth (Eds.), 2014. *Vehicle Routing: Problems, Methods, and Application*. MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics. doi:[10.1137/1.9781611973594](https://doi.org/10.1137/1.9781611973594).
- [5] Erdoğan, S., Miller-Hooks, E., 2012. A Green Vehicle Routing Problem. *Transportation Research Part E: Logistics and Transportation Review* 48, 100–114. doi:[10.1016/j.tre.2011.08.001](https://doi.org/10.1016/j.tre.2011.08.001).
- [6] Ewert, R., Martins-Turner, K., Thaller, C., Nagel, K., 2021. Using a Route-based and Vehicle Type specific Range Constraint for Improving Vehicle Routing Problems with Electric Vehicles. *Transportation Research Procedia* 52, 517–524. doi:[10.1016/j.trpro.2021.01.061](https://doi.org/10.1016/j.trpro.2021.01.061).
- [7] Gardrat, M., 2019. *Méthodologie d'enquête: le découplage de l'achat et de la récupération des marchandises par les ménages*. Research report. LAET. Lyon, France.
- [8] Hörll, S., Balac, M., 2021. Synthetic population and travel demand for Paris and Île-de-France based on open and publicly available data. *Transportation Research Part C: Emerging Technologies* 130, 103291. doi:[10.1016/j.trc.2021.103291](https://doi.org/10.1016/j.trc.2021.103291).
- [9] Martins-Turner, K., Grahle, A., Nagel, K., Göhlich, D., 2020. Electrification of Urban Freight Transport - a Case Study of the Food Retailing Industry. *Procedia Computer Science* 170, 757–763. doi:[10.1016/j.procs.2020.03.159](https://doi.org/10.1016/j.procs.2020.03.159).
- [10] Martins-Turner, K., Nagel, K., Zilske, M., 2019. Agent-based Modelling and Simulation of Tour Planning in Urban Freight Traffic. *Transportation Research Procedia* 41, 328–332. doi:[10.1016/j.trpro.2019.09.054](https://doi.org/10.1016/j.trpro.2019.09.054).
- [11] Mucowska, M., 2021. Trends of Environmentally Sustainable Solutions of Urban Last-Mile Deliveries on the E-Commerce Market—A Literature Review. *Sustainability* 13, 5894. doi:[10.3390/su13115894](https://doi.org/10.3390/su13115894).
- [12] Patella, S.M., Grazieschi, G., Gatta, V., Marcucci, E., Carrese, S., 2021. The Adoption of Green Vehicles in Last Mile Logistics: A Systematic Review. *Sustainability* 13, 6. doi:[10.3390/su13010006](https://doi.org/10.3390/su13010006).
- [13] Schlenther, T., Martins-Turner, K., Bischoff, J.F., Nagel, K., 2020. Potential of Private Autonomous Vehicles for Parcel Delivery. *Transportation Research Record* 2674, 520–531. doi:[10.1177/0361198120949878](https://doi.org/10.1177/0361198120949878).
- [14] Schneider, M., Stenger, A., Goetze, D., 2014. The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations. *Transportation Science* 48, 500–520. doi:[10.1287/trsc.2013.0490](https://doi.org/10.1287/trsc.2013.0490).

- [15] Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., Dueck, G., 2000. Record Breaking Optimization Results Using the Ruin and Recreate Principle. *Journal of Computational Physics* 159, 139–171. doi:[10.1006/jcph.1999.6413](https://doi.org/10.1006/jcph.1999.6413).
- [16] Skalna Karkula, D., 2019. Comparison of capabilities of recent open-source tools for solving capacitated vehicle routing problems WITH TIME WINDOWS - Carpathian Logistics Conference, Zakopane, Poland, pages 72-77, December 2019. .
- [17] Villanueva, R.S., 2020. A pragmatic approach to improve the efficiency of the waste management system in Stockholm through the use of Big Data, Heuristics and open source VRP solvers : A real life waste collection problem;Stockholm’s waste collection system and inherent vehicle routing problem, VRP.
- [18] Vosooghi, R., Puchinger, J., Bischoff, J., Jankovic, M., Vouillon, A., 2020. Shared autonomous electric vehicle service performance: Assessing the impact of charging infrastructure. *Transportation Research Part D: Transport and Environment* 81, 102283. doi:[10.1016/j.trd.2020.102283](https://doi.org/10.1016/j.trd.2020.102283).
- [19] Yu, S., Puchinger, J., Sun, S., 2020. Two-echelon urban deliveries using autonomous vehicles. *Transportation Research Part E: Logistics and Transportation Review* 141, 102018. doi:[10.1016/j.tre.2020.102018](https://doi.org/10.1016/j.tre.2020.102018).