The electric on-demand bus routing problem with partial charging and nonlinear function

Reference:
Lian Ying, Lucas Flavien, Sörensen Kenneth.- The electric on-demand bus routing problem with partial charging and nonlinear function
Transportation research : part C : emerging technologies - ISSN 1879-2359 - 157(2023), 104368
Full text (Publisher's DOI): https://doi.org/10.1016/J.TRC.2023.104368
To cite this reference: https://hdl.handle.net/10067/2012400151162165141

The Electric On-Demand Bus Routing Problem with Partial Charging and Nonlinear Function

Ying Lian[a,*], Flavien Lucas[b], Kenneth Sörensen[a]

[a] *ANT/OR - Operations Research Group, Department of Engineering Management, University of Antwerp*
[b] *IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems, F-59000 Lille, France*

Abstract

The electric vehicle routing problems (EVRPs) with recharging policy consider the limited range of electric vehicles and thus include intermediate visits to charging stations (CSs). In general, minimizing the resultant charging costs such as charging duration or charging amount is also part of the objective of the EVRP. Accordingly, the EVRP has received considerable attention over the past years. Nevertheless, this type of problem in the domain of passenger transportation, a VRP variant, has been rarely studied in the literature, especially with time windows, a realistic nonlinear charging function or a partial charging policy. Hence this research extends the existing work on the EVRP to the on-demand bus routing problem (ODBRP) which transports passengers with the bus station assignment (BSA). The resultant problem is the electric ODBRP (EODBRP). Specifically, each passenger can have more than one stations to board or alight, and they are assigned to the ones with the smallest increase in the total user ride time (URT). In the EODBRP, frequent intermediate visits to CSs are considered. Moreover, a nonlinear charging function is used and the partial charging strategy is applied. To solve the EODBRP, a greedy insertion method with a 'charging first, routing second' strategy is developed, followed by a large neighborhood search (LNS) which consists of local search (LS) operators to further improve the solution quality. Experimental data are generated by a realistic instance generator based on a real city map, and the corresponding results show that the proposed heuristic algorithm performs well in solving the EODBRP. Finally, sensitivity analyses with divergent parameters such as the temporal distributions of passengers and bus ranges may provide practical guidance.

Keywords: on-demand bus routing problem, electric vehicle, nonlinear charging function, partial charging

1. Introduction

Electric vehicles (EVs) have steady increase in the automotive market in recent years, and one of the motivations is to reduce greenhouse gas emissions from land transportation (Zhou et al., 2015). In parallel, the study of EVs has also emerged in the vehicle routing problem (e.g. Schneider et al., 2014; Hof et al., 2017; Schiffer and Walther, 2017), which is referred to as '*electric vehicle*

*Corresponding author.
Email addresses: ying.lian@uantwerpen.be (Ying Lian), flavien.lucas@imt-nord-europe.fr (Flavien Lucas), kenneth.sorensen@uantwerpen.be (Kenneth Sörensen)

routing problem' (EVRP), where the EV-related technological constraints are taken into account. In particular, compared with gasoline or diesel-powered vehicles, the driving range of EVs is limited, as a result, battery charging (or swapping) is usually necessary in route planning. Thus the main objective of the EVRP is an overall efficient routing of visits to customers and charging or swapping stations. Especially, the charging process of lithium-ion batteries still generates greenhouse gases significantly (Onn et al., 2018), which adds to the importance to investigate efficient charging solutions. Specifically, the charging process of these batteries is non-linear, and research has shown that ignoring nonlinearity can cause the solutions to be inefficient or even infeasible (Pelletier et al., 2017; Montoya et al., 2017).

Despite research on the EVRP, considering EVs in the context of passenger transportation such as the dial-a-ride problem (DARP) or other variants, however, has not been widely studied. To the best of our knowledge, only a few papers (Masmoudi et al., 2018; Bongiovanni et al., 2019; Sayarshad et al., 2020; Ma et al., 2021; Zhang et al., 2022) have investigated the EDARP with a focus on detouring to battery charging or swapping stations, with small vehicles of six seats or electric autonomous cars. While in practice, given the development of the fast charging technologies of electric buses, a minibus or even a regular bus can be quickly charged from empty to 80% full within an hour or fully charged in around two hours, which provides the possibility to apply EVs in passenger transportation, both in real life and in the research field. In fact, there are cities that have largely replaced conventional buses with pure electric or hybrid buses, benefiting from the technologies of fast charging and enlarged battery capacity (He et al., 2020). Hence, this work investigates the on-demand bus routing problem (ODBRP) integrated with intermediate visits to charging stations (CSs). Specifically, the partial charging policy and a nonlinear charging function are adopted in order to make the model more realistic.

In this work, we propose a heuristic approach to solve the electric ODBRP (EODBRP). To start with, the term 'leg' is used to refer to the part of a bus route between two consecutive visits to charging stations. Then each request is inserted in the leg with the minimum increase in the total user ride time (URT). Related modifications are thus routing among stations, as well as adapting CS, including which CS to visit and how much to charge. Calculating the recharging times is mathematically modeled as a subproblem and solved by CPLEX if the number of legs within a route is larger than or equal to three (for fewer legs the optimal recharging time can be calculated without CPLEX). In this way, the optimal recharging times can be achieved. In the second phase, a large neighborhood search (LNS) with a local search (LS) operator is devised to improve the solution quality. Particularly, two parameters within the LNS framework which control how many requests to destroy, and the number of candidate solutions to choose the most promising one from for LS. The performance of the LNS framework is tested.

In terms of experiments, a real city map with given stations and distance matrix is used, and requests are generated by a realistic instance generator from the literature. Then the performance of the proposed algorithm is tested, including the parameter setting of the LNS. Next, the features of the best-known solutions (BKS) are analyzed and presented. In addition, the effect of the bus station assignment (BSA) is tested among various instance sizes, which distinguishes the ODBRP from the DARP and as a result, improves the overall efficiency of the solution. Moreover, the experiments that compare the performance with conventional buses are conducted, i.e., the number of buses needed, among instances with various temporal distributions of passengers, as well as the use of mini or regular buses. Experimental results show that intermediate charging plays an important role in applying electric buses; the BSA not only improves the total URT but also reduces the total charging times and thus contributes both to the service quality and the environment; both the mini and regular electric buses can be qualified substitutes of conventional buses with robustness, among various passengers' sizes and temporal distributions.

The contributions of this work can be listed as follows: It extends the use of EVs in the context of passenger transportation, with realistic charging functions and charging policy. Subsequently, it proposes and verifies a possible division of the integrated problem, as well as solves the subproblem to optimality and the entire problem efficiently. Moreover, the parameter setting of the LNS framework achieves a tradeoff between exploring more neighborhoods and maintaining the possibly favorable attributes of the best-known solution, the solution quality is thus improved. Last but not least, it conducts extensive numerical experiments with realistic artificial instances, in order to assess the performance of the proposed method as well as the feasibility of applying electric buses with recharging in the ODBRP.

The remainder of this paper is organized as follows. Section 2 presents the related work in the research field. Section 3 formally introduces the EODBRP. Next, Section 4 describes the solution method, i.e., the heuristic and the subproblem, and Section 5 presents the computational experiments. Finally, Section 6 concludes the paper and discusses future research.

2. Literature review

This Section provides a brief review of recent and related literature. First, the literature on the EVRP is introduced. Next, since this study involves partial charging and nonlinear charging functions, the relevant literature is reviewed. Subsequently, as a variant of '*Fixed Route Vehicle Charging Problem*'(FRVCP) is the subproblem of this study for charging times, we revise the papers including the FRVCP. Finally, the relevant research on the EDARP is presented.

Research on EVRPs originally started with the green vehicle routing problem (Green VRP) proposed by Erdoğan and Miller-Hooks, 2012. In the Green VRP, vehicles with alternative fuels will be fully charged at refill stations. Then Schneider et al., 2014 extended the Green VRP to the EVRP. More specifically, in their paper, an EVRP with time windows was solved using a hybrid heuristic of variable neighborhood search and tabu search.

In the EVRP, the charging policy particularly can be divided into three categories: full recharging with a linear recharging function (e.g. Goeke and Schneider, 2015; Hiermann et al., 2016), partial recharging with a linear function (Ángel Felipe et al., 2014; Schiffer and Walther, 2017; Desaulniers et al., 2016) or partial recharging with a nonlinear function (Montoya et al., 2017). In this study, a partial charging policy with a nonlinear function is adopted. Thus in the following, we briefly review the literature that deals with partial charging and a nonlinear function, while for more comprehensive and recent reviews of the EVRP, we refer to Kucukoglu et al., 2021; Erdelić and Carić, 2019.

2.1. Partial charging policy

Full recharging was considered initially in most EVRP literature. However, this can be time-consuming and not eco-friendly, as full charging can be a waste of energy and the charging process still generates greenhouse gases. Moreover, full charging decreases the operational duration of buses for passengers, therefore, more buses have to be purchased. On the contrary, partial charging can enable the feasibility of serving more passengers with narrow time windows, since vehicles can have a wide range of charging times, from minutes to hours, depending on the State-of-Charge (SoC) level, charging technology, and battery capacity (Martínez-Lao et al., 2017). In partial recharging, it is natural to only charge the amount that is enough to cover the remaining route.

The concept of allowing partial charging was considered in Ángel Felipe et al., 2014; Bruglieri et al., 2015. Then Desaulniers et al., 2016 adopted partial charging in an EVRP with time windows. In addition, Keskin and Çatay, 2016 adopted partial charging in an EVRP with time windows and provided the mathematical model. This problem was solved by an adaptive large neighborhood search algorithm. Compared with the full charge strategy, they concluded partial charge can save the total costs substantially. In terms of the EVRP with time windows, Desaulniers et al., 2016 also solved a problem in this domain and concluded that allowing multiple visits to CSs as well as partial charging both help reduce routing costs and the number of vehicles compared with a single visit and with full charging. Exact branch-price-and-cut algorithms were used to solve this problem. Similarly, Schiffer and Walther, 2017 concluded the benefits of partial charging in an electric location routing problem with time windows. Specifically, a partial charging strategy can reduce the total travel distance and the number of visits to CSs.

2.2. Nonlinear charging function

The charging procedure was initially modeled as a linear function. While in fact, lithium-ion batteries are often charged in constant-current constant voltage (CC-CV) phases: in the CC phase, SoC increases linearly, while in the CV phase nonlinearly, and the charging time is prolonged due to the drop of the current (Pelletier et al., 2017). The standard way to deal with the nonlinear function is to transform it into a piece-wise linear function.

Based on the piecewise linear and concave charging functions, Zündorf, 2014 developed a propagating algorithm to compute a battery-constrained route. Later, Montoya et al., 2017 fitted piecewise linear functions according to real-world data, and showed ignoring nonlinearity can lead to poor or even infeasible solutions. A hybrid metaheuristic combining an iterated local search and a method named 'heuristic concentration' was developed to solve the EVRP with nonlinear charging functions. Later, based on the nonlinear charging function in Montoya et al., 2017, Froger et al., 2019 proposed an arc-based model tracking the time and the state of charge. Besides, they also proposed a path-based model. The experimental results showed the two models outperform the node-based model.

2.3. Fixed route vehicle charging problem

One of the most challenging issues when solving an EVRP and its variants is the charging decisions, especially with the partial charging policy and a nonlinear function, it is critical to decide when and how much to charge, since the charging decisions influence significantly the feasibility and quality of the solutions. Therefore, in terms of solution methods to position a CS in a route, some authors have studied the FRVCP as a subproblem of the EVRP. In such problems, the customer sequence in a route is fixed while the positions of CSs and the amount to charge are adjusted. The FRVCP is a variant of the Fixed Route Vehicle Refueling Problem (FRVRP) which aims to minimize the refueling cost for a fixed route. The FRVRP is NP-hard (Suzuki, 2014), thus the FRVCP is also NP-hard, and various solution approaches have been proposed in the literature.

Montoya et al., 2016 solved an FRVCP with full charging and a constant charging time. In their work, the FRVCP was formulated as a constrained shortest path problem and solved by the Pulse algorithm (Lozano and Medaglia, 2013). Similar approaches were also applied in the EVRP with time windows (namely EVRPTW) (Keskin and Çatay, 2018; Hiermann et al., 2019), electric location routing problem (Schiffer and Walther, 2018a,b). Next, some literature assumed an EV can visit at most one CS between any pair of non-CS vertices. Among them, an FRVCP within an EVRPTW with single charging technology and a linear charging function were solved in Hiermann et al., 2016; Schiffer and Walther, 2018a; Hiermann et al., 2019. The difference is whether the full charging policy (Hiermann et al., 2016) or the partial charging policy (Schiffer and Walther, 2018a; Hiermann et al., 2019) was adopted. Exact algorithms were proposed to solve these problems. Meanwhile, Montoya et al., 2017 solved an FRVCP as a sub-problem of an EVRP with nonlinear charging functions, heterogeneous charging stations, and a partial charging policy. A mixed integer linear programming (MILP) model was proposed and the problem was solved by a hybrid metaheuristic combining an iterated local search and a heuristic concentration. Based on the specific MILP formulation and the metaheuristic, Koç et al., 2019 solved an FRVCP in an EVRP with shared CS and nonlinear charging. They proposed a multi-start heuristic performing an adaptive large neighborhood search, together with the solution of mixed integer linear programs. Moreover, Baum et al., 2019 solved an FRVCP for EV with realistic and heterogeneous CS and battery swapping stations. A specific algorithm that combines different algorithmic techniques was developed and can solve this problem optimally on realistic inputs.

2.4. Extension in the EDARP

A typical DARP includes operational constraints related to time windows, vehicles' capacities, and maximum ride durations. Then the EDARP integrates the classic DARP with the following features: routing to charging facilities, selecting charging stations, determining recharging times, as well as adapting the arrival and departure time at passengers' stations. In literature, Masmoudi et al., 2018 proposed an EDARP with battery swapping stations and a realistic energy consumption function. Three enhanced evolutionary variable neighborhood search algorithms were devised to solve this problem.

Besides, there are papers that adopt electric autonomous cars which are more flexible to modify vehicles' routes, especially in real time. Among them, Bongiovanni et al., 2019 proposed an electric autonomous DARP which covers detours to charging stations and recharge times. This problem was formulated as a 2-index and 3-index MILP, and solved by a Branch-and-Cut algorithm with new valid inequalities, with the objective of minimizing a weighted sum of the total travel time and excess URT. Later, Sayarshad et al., 2020 devised a non-myopic dynamic routing problem of electric taxis with battery swapping stations and proposed a formulation of the traveling salesman problem with pickup and drop-off, including the battery capacity constraints. Then Ma et al., 2021 proposed a location routing problem for the car-sharing system with autonomous electric vehicles, and solved it separately by the genetic algorithm and GAMS. Recently, Zhang et al., 2022 proposed a routing problem of shared autonomous electric vehicles under uncertain travel time and uncertain service time. A branch-and-price algorithm was used to solve this problem.

To summarize, the EVRP which takes charging decisions into account has drawn interest from academia. On the other hand, the EDARP and its variants have not been widely studied. Thus, applying EVs as well as considering charging decisions with the partial charging strategy and a nonlinear charging function distinguish this study from the literature. In addition, this study proposes a unique variant of the FRVCP with a 'charging first, routing second' strategy. Details will be explained in Section 3 and Section 4.

3. Problem description

In this paper, we investigate the ODBRP using electric vehicles with a nonlinear charging function and the partial charging policy (EODBRP). Mathematically, the EODBRP can be defined with a mixed integer programming (MIP) formulation, which consists of two parts: the ODBRP with conventional vehicles introduced by Melis and Sörensen, 2022; in addition, the constraints concerning electric vehicles with a nonlinear charging function and the partial charging policy are from the EVRP introduced by Montoya et al., 2017.

3.1. The electric on-demand bus routing problem

A formal description of the EODBRP is as follows.

Let S denote the set of predefined stations for passengers to board and alight. Each station node $s \in S$ can be visited more than once (even by the same bus) or never. Let F be the set of CS each with an unlimited capacity where vehicles can be fully or partially charged, then let F' be the set of β copies of F. The use of F' is for the modeling convenience as each node $\in F'$ can be restricted to be visited at most once. Next, let V be the set of nodes, i.e., $V = S \cup F'$. Let $G = (V, A)$ be a complete graph, where A is the set of arcs connecting any pair of vertices $\in V$. For any arc $(i, j) \in A$, the travel time is constant.

Passengers are geographically dispersed within a service area, and they send in their travel requests in advance, possibly via a mobile application or website. The requests are assumed static and deterministic. For simplicity, we set each request corresponds to one passenger. Besides, passengers also specify their locations of origin and destination. Thus, let the set of bus stations available for each passenger p's pickup or drop-off be noted a^u_{ps} or a^o_{ps}. Then for each passenger, the stations within a predefined walking distance to the origin or destination are stored in a^u_{ps} or a^o_{ps}. In our problem setting, at least one station is guaranteed for each passenger to respectively board or alight. If there are two or more stations, then the bus station assignment (BSA), namely, which station is selected to board or alight among the stations in the set, is a decision to be made by the algorithm.

An illustrative example is in Fig 1. A bus departs from a CS, then visits the passengers' stations as well as an intermediate CS, and finally ends at a CS. Compared with the dotted lines where passengers are only picked up and dropped off at the closest stations to their origins and destinations, the BSA reduces the distances both between passengers' stations and the distances to/from CSs. The BSA also allows for the pooling of passengers at stations, so that the bus does not have to stop as frequently.

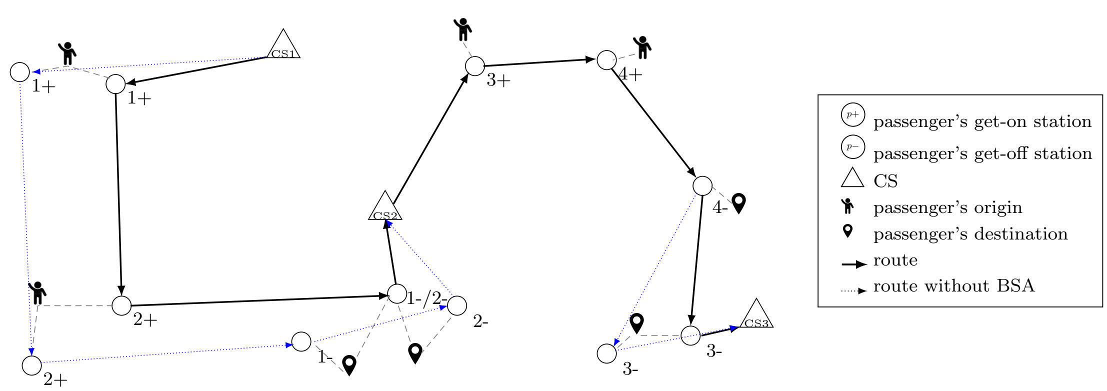


Figure 1: Illustrative example of the EODBRP

Passengers also indicate either their desired departure time or arrival time, then we implicitly calculate a hard time window for each passenger. In literature such as Cordeau and Laporte, 2003, a general formulation of time windows distinguishes inbound or outbound requests, as well as has separate time windows for pickup and drop-off. Besides, there is an additional constraint for the maximum ride time. However, in our model, each passenger p has one single hard time window of the earliest departure and the latest arrival $[e_p, l_p]$, thus the passenger has to be picked up and dropped off within $[e_p, l_p]$. The duration of each time window is set to $f \times t_{dir}$, i.e., $l_p - e_p = f \times t_{dir}$ where f is a constant, and t_{dir} is the direct travel time from the origin to the destination. The parameter f controls how strict the time windows are. A small value leads to a strict time window and vice versa. In this way, the maximum ride time for each passenger is restricted as well.

A homogeneous fleet of electric buses is dispatched, each with a finite capacity Q^{cap}, i.e., the maximum number of passengers that are on board simultaneously. All the buses have a battery of capacity Q (expressed in km). It is assumed that the EVs are equipped with a fully charged battery at the beginning of their route. Then feasible solutions should satisfy that the battery level when an EV arrives at and departs from any vertex is between 0 and Q. The details of the charging function will be explained separately in Section 3.2.

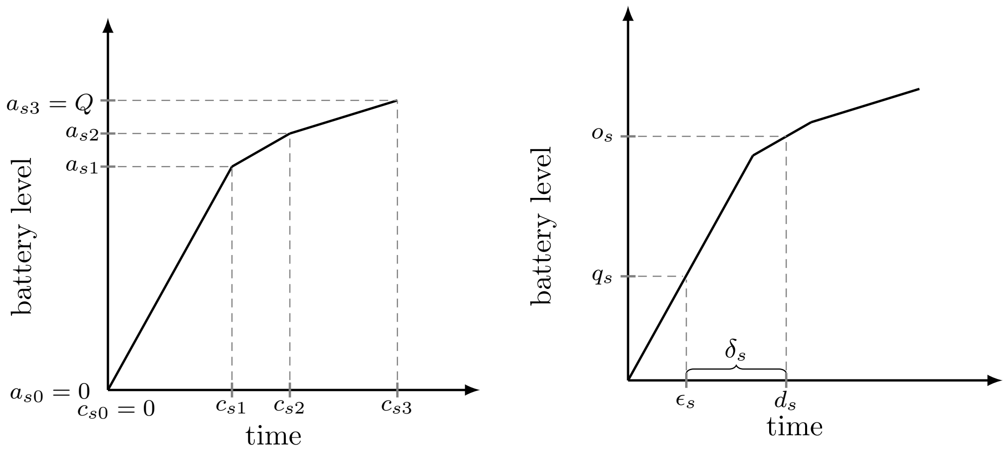
Travel between any two nodes $\in V$ involves a travel time and consumption of electricity. The travel time is proportional to the distance, given the travel speed is constant; similarly, the electricity consumption is assumed to be only proportional to the distance as well, regardless of the load, road slope, etc. Thus, the triangular inequality holds for both travel time and electricity consumption.

The station sequence of each bus which records the stations (including CSs) that the bus has to visit, is one of the decisions made by the algorithm. Each station is visited only when needed, i.e., at least one passenger needs to get on/off, or the bus charges a positive amount of electricity at a CS. A visit to a charging station CS $\in F'$ is only allowed when no passengers are on board.

Each bus b's arrival or departure time at the n-th station of its route is respectively denoted as t^a_{nb} and t^d_{nb}. For a visited CS $s \in F'$, $t^d_{nb} = t^a_{nb} + \delta_s$, where δ_s denotes the charging time at $s \in F'$. Namely, the departure time from a CS is equal to the arrival time plus the charging time. For a passenger station $s \in S$, t^d_{nb} may be later than the arrival time t^a_{nb} as well, that is to say, buses are allowed to wait at station $s \in S$, even if there are passengers on board, under the condition that at least one passenger p is boarding at s and the arrival t^a_{nb} is prior to the earliest departure e_p. So the departure time t^d_{nb} may be later than the arrival time t^a_{nb}. The reason to allow waiting is that it is a way to decrease the total URT, and thus indirectly increase the possibility to serve more passengers. Although it increases the URT of passengers aboard, it can reduce others'. On the other hand, if the holding policy does not apply, we assume passengers board and alight immediately when the bus arrives, and this service duration is negligible, namely t^d_{nb} is simply equal to t^a_{nb}.

The exact MIP formulation of the EODBRP is in Appendix A. The aim of the EODBRP is to transport all passengers with high service quality. Thus the objective function is to minimize the total URT. As for the service quality regarding the waiting time before passengers get on board, it is considered guaranteed as long as the time windows are satisfied. Similarly, the service quality in terms of the walking distance is fulfilled implicitly when calculating a^u_{ps} and a^o_{ps} for every passenger p. In addition to the total URT, in the subproblem FRVCP, the total charging time is minimized for a given bus route, in order to maximize the duration that can be spent on serving passengers, as well as minimize the impact on the environment, given the charging amount monotonically increases with charging time.

The decisions to be made in this EODBRP are summarized as follows: first, the passenger-bus assignment has to be performed, namely which bus serves which passenger; second, the BSA, i.e.,



(a) Breakpoints of the function

(b) Variables indicating the charging time and battery level

Figure 2: Piecewise linear charging function

s; finally, the station sequence of each bus as well as the arrival and departure time at each station, especially when, where and how much to charge.

3.2. Modeling of the nonlinear charging function

For the charging-related formulation, we adopt the model introduced by Montoya et al., 2017, where they proposed an EVRP with nonlinear charging functions which were further approximated by piecewise linear functions.

In this study, we assume the CS are identical, i.e., with the same charging function. Specifically, each CS $s \in F'$ is associated with a piecewise linear concave charging function $g_s(q_s, \delta_s)$. This function maps the charging level q_s and the charging time δ_s. More precisely, when a vehicle arrives at a CS s, the electricity level is represented by the variable q_s, then the vehicle leaves s with the electricity level o_s, and the corresponding charging duration is δ_s as explained before this subsection. Moreover, in order to obtain the value of δ_s, let $\hat{g_s}(\delta_s)$ be the charging function when $q_s = 0$ and the battery is charged for δ_s time units. Therefore the generalized δ_s when $q_s \geq 0$ and $o_s \geq q_s$ is calculated by the values of the inverse function as $\delta_s = \hat{g_s}^{-1}(o_s) - \hat{g_s}^{-1}(q_s)$.

As $g_s(q_s, \delta_s)$ is piecewise linear, the breakpoint of each piece is represented by $k \in \Gamma$, where $\Gamma = \{0, 1, .., \gamma\}$ is the set of breakpoints of the piecewise linear function. For instance, the first linear piece is bounded with breakpoints 0 and 1, while the second piece is bounded with 1 and 2, etc. Then c_{sk} and a_{sk} represent the charging time and the charging level for the breakpoint $k \in \Gamma$ of the CS $s \in F'$, Figure 2 is an illustration of $g_s(q_s, \delta_s)$ with breakpoints. ϵ_s and d_s are the mapped charging times according to the function. In other words, the electricity level q_s corresponds to the charging time point ϵ_s, while o_s corresponds to d_s. Naturally, variable $\delta_s = d_s - \epsilon_s$ represents the time spent at CS $s \in F'$. Variables z_{sk} and w_{sk} are equal to 1 if the charge level is between $a_{s,k-1}$ and a_{sk}, with $k \in \Gamma \setminus \{0\}$, when the EV arrives at and departs from the CS $s \in F'$ respectively. Finally, variables α_{sk} and λ_{sk} are the coefficients of the breakpoint $k \in \Gamma$ in the piecewise linear approximation, when the EV arrives at and departs from CS $s \in F'$ respectively. These variables and the corresponding constraints are included in the exact MIP formulation of the EODBRP in Appendix A.

4. Solution method

Healy and Moll, 1995 demonstrated that the standard DARP is NP-hard. Since the ODBRP is a generalization of the DARP, we deduce that it is also NP-hard. Furthermore, the EODBRP can be considered as a generalization of the ODBRP, because the ODBRP is the EODBRP in the special case where the battery has enough capacity thus charging is not needed. Equivalently, the EODBRP is NP-hard.

In order to develop an algorithm to solve the EODBRP efficiently, let us first qualitatively analyze the problem. In the EODBRP, briefly, five main decisions need to be made. First, passenger-bus assignment needs to be made; second, the alternative bus stations of each passenger have to be selected; third, the sequence of the passengers to be picked up and dropped off has to be determined for each bus; fourth, the schedule of each bus when to arrive at and depart from each station has to be settled; finally, in each route, whether charging is necessary has to be examined, if so, when, where, how many times to charge, as well as how much to charge have to be determined.

The routing and charging decisions can be made either simultaneously or sequentially. As the inter-dependency increases the complexity of the problem, we propose the 'charging first, routing second' policy. In particular, multiple legs in each route are created, that is, each leg is bounded by two consecutive visits to the CSs. Subsequently, to respectively construct the initial solution and further improve the solution quality, our approach consists of a greedy insertion and a subsequent heuristic, like numerous heuristics for the VRPs and the variants. More specifically, for the second stage, a large neighborhood search (LNS) combined with a local search (LS) operator is developed. Besides, two parameters K and σ are introduced in the proposed LNS algorithm, trying to find better solutions more efficiently. In both stages, the algorithm makes routing and charging decisions. The former determines the sequence of the stations where each passenger is picked up as well as dropped off, while the latter decides where and how much to charge. To make these charging decisions, we solve the FRVCP that decides which CS to visit and how much to charge at this visit. Algorithm 1 outlines the structure of the solution method summarized above. Thus in the following subsections, we respectively describe in detail the greedy insertion (Subsection 4.1), the FRVCP (Subsection 4.2), and the LNS (Subsection 4.3).

Algorithm 1: Solution method

Notations: χ —best-known solution, χ_0 —initial solution, χ', χ'' —neighborhood solutions of χ
Input: requests' locations and time windows, distance matrix, etc
Requests are ranked by e_p;
Empty legs are created as placeholders;
for *passenger* p *= 1,2,...,P* **do**
  for *positions in nonempty legs* **do**
    Assume p is inserted;
    Calculate the charging amount directly or by the FRVCP;
  If at least a feasible position exists:
  Then insert p at the position w.r.t the least increase in the total URT;
  Else:
  for *empty legs* **do**
    Assume p is inserted;
    Calculate the charging amount directly or by the FRVCP;
  If at least a feasible position exists:
  Then insert p at the position w.r.t the least increase in the total URT;
  Else:
  Assign p to a new bus;
$\chi \leftarrow \chi_0$;
while *stopping criterion is not met* **do**
  LNS:
  for *round = 1,...,*σ **do**
    Remove randomly K requests from χ;
    Repair operator generates χ'';
    If $f(\chi'') < f(\chi')$:
    Then $\chi' \leftarrow \chi''$, i.e., pick the best solution χ' from neighborhood $N(\chi)$;
  Apply LS on χ';
  Move or not:
  If $f(\chi') < f(\chi)$ **Then** $\chi \leftarrow \chi'$;
Output: best-known solution

4.1. Construct the initial solution

For the initialization phase, a greedy insertion method is used. The procedure to construct the initial solution is as follows. To start with, the requests are ranked in ascending order of their earliest departure time. In other words, the request with the smallest earliest departure is the first one on the request list. Next, a certain number of empty legs as placeholders are created for each bus route. The number of legs is large enough, and once all the requests are inserted, the empty legs will be erased. Then each request is removed from the request list and inserted in one of the non-empty legs of all the buses, in the position with the least increase in the total URT that respects time windows, bus capacity, and the battery level at each station of the route. Besides, the station pair among its alternative get-on/off choices with the least increase in the total URT is chosen. If p cannot be inserted in any non-empty legs due to any violation of these constraints, p is inserted into an empty leg. The bus to serve p is the one with the smallest distance between p's boarding station and its previous passenger station. Similarly, if p cannot be inserted in any existing bus routes, an empty bus is dispatched to serve this request. Again, the station pair with the least increase in the total URT is chosen. This insertion procedure is repeated until all requests are served.

If there is only one leg in a bus route, in other words, the bus only charges at the beginning and the end of its route, then the route is feasible as long as the battery level upon arrival at the CS at the end is nonnegative. The charging at the beginning is omitted, as we simply assume the battery has been fully charged overnight. The CS at the beginning of the route is the one closest to the first passenger station of the route. Similarly, the CS at the end is the one closest to the last passenger station.

If there are two legs, in other words, besides the CS at the beginning and the end of the route mentioned before, the bus pays an intermediate visit to a CS. In this case, an additional requirement is, given the battery level upon arrival at the intermediate CS, and the maximum allowed charging time (i.e., the duration that can be postponed without any violation of the time windows), the battery level upon leaving the intermediate CS is enough to reach the CS at the end. Same as the one-leg case, the CS at the beginning and the end are simply the closest ones to the first or last passenger stations. However, for the CS in the middle, it cannot be simply chosen as the one with the smallest sum of the distance between its previous and next passenger stations, since this may cause one of the legs' range to exceed the battery level. Instead, all the CSs are ranked according to the sum of distance, then the first one that respects the battery levels of the two legs is chosen as the mid-route CS.

If there are three or more legs, the optimal total charging time of this bus route is not as straightforward as in the former two cases. To solve this case, the rule proposed by Ángel Felipe et al., 2014 is usually adopted in literature within heuristic methods: when visiting a CS, charge the minimum amount of energy needed to make the route energy-feasible. Specifically, if it is the last CS before returning to the depot, the minimum amount is thus to cover the remaining route to the depot. Or if there is at least one CS downstream, charge the amount needed to reach the next CS. If reaching the next CS (or the depot) is impossible, the move is deemed infeasible.

However, the optimal charging time or amount may not be obtained by the rule mentioned above. Hence in this study, the exact mathematical model is proposed and CPLEX is called to solve the subproblem to optimality, and these will be explained in Section 4.2.

4.2. Mathematical formulation of the FRVCP

The formulation of the FRVCP is explained in Montoya et al., 2017. However, the main difference is that the positions of the intermediate CSs in the routes need to be decided in the FRVCP of their work, i.e., when to visit CSs, while these are fixed in the FRVCP of this study. Nevertheless, which CS to visit each time still needs to be determined, since it cannot be simply set as the CS with the smallest total distance between its preceding and succeeding station (will be explained in this subsection). Despite the difference, we adopt the variables and related constraints used in their formulation, with the details as follows.

The parameters of the mathematical model are explained below. Let $i \in \{0, 1, ..., n_r\}$ be the index of the CSs in a specific bus route, where 0 and n_r respectively correspond to the CS at the beginning and the end of the route, while $1, 2, ..., n_r - 1$ correspond to the intermediate CSs. As explained in Section 3.2, all the CSs $\in F'$ have the same piecewise linear function, where each piece is defined by the breakpoint set Γ. Specifically, the piece k is between the breakpoint $k - 1$ and k where $k \in \Gamma \setminus \{0\}$, with a slope denoted as ρ_k. Besides, each piece k is bounded by the battery levels $a_{j,k-1}$ and a_{jk}, where $j \in F$ is to indicate the CSs. In addition, since the positions of the CSs are determined, together with the CSs to visit (which CSs are chosen will be explained in the last paragraph of this subsection), the distance to travel within each leg (including the distance from the CS prior to the leg and the CS after the leg) is simply a constant that can be easily calculated. Let $d^p_{i,i+1}$ where $i \in \{0, 1, ..., n_r - 1\}$ denote the energy consumption of the route that only consists of the passengers' stations between the CSs with index i and $i + 1$. Similarly, let d_{ij} where $i \in \{0, 1, ..., n_r - 1\}$ denote the energy consumption between CS $j \in F$ and its succeeding passengers' station of leg i. Naturally, $d_{i-1,j}$ where $i \in \{1, 2, ..., n_r - 1\}$ denote the energy consumption between the CS $j \in F$ and its preceding passengers' station of leg $i - 1$. Thus the energy consumption is a constant for each route. Finally, let the buffer at the beginning of each leg (indexed by i) be divided into two components, denoted as w_i and b_i, respectively representing the slack that can be consumed without/with postponing the start time of the succeeding leg. The formulation of the buffer is the same as the concept of forward time slack in literature Savelsbergh, 1992; Cordeau and Laporte, 2003.

In terms of the variables, let ϕ_i denote the battery level upon arrival at the CS with index i. Then let δ_{ijk} denote the charging amount when the charging process finishes on the piece between breakpoints $k - 1$ and k, at CS j with index i. Further, let μ_{ijk} denote the battery level when the charging process finishes on the piece between breakpoints $k - 1$ and k, at CS j with index i. Moreover, let ϵ_{ij} equal to 1 if CS j is at position i. Next, let θ_{ijk} equal to 1 if CS j is at position i, and the battery is charged within the segment bounded by $k - 1$ and k. Finally, we introduce the variables x_i to indicate whether the start time of the succeeding leg is affected. Then the mixed integer programming (MIP) formulation is as follows.

$$\min \sum_{i=\{1,2,...n_r-1\}} \sum_{j\in F} \sum_{k\in\Gamma\setminus\{0\}} \frac{\delta_{ijk}}{\rho_k} \tag{1}$$

s.t.

$$\phi_1 = Q - d^p_{01} - \sum_{j\in F} \epsilon_{0j} d_{0j} - \sum_{j\in F} \epsilon_{1j} d_{1j} \qquad i = 1 \tag{2}$$

$$\phi_i = \mu_{i-1,j\gamma} - d^p_{i-1,i} - \sum_{j\in F} \epsilon_{i-1,j} d_{i-1,j} - \sum_{j\in F} \epsilon_{ij} d_{ij} \qquad \forall i \in \{2, 3, ..., n_r\} \tag{3}$$

$$\mu_{ij1} = \phi_i + \delta_{ij1} \qquad \forall i \in \{1, 2, ..., n_r - 1\}, \forall j \in F \tag{4}$$

$$\mu_{ijk} = \mu_{ij,k-1} + \delta_{ijk} \qquad \forall i \in \{1, 2, ..., n_r - 1\}, \forall j \in F, \forall k \in \Gamma \setminus \{0, 1\} \tag{5}$$

$$\mu_{ijk} \geq a_{j,k-1}\theta_{ijk} \qquad \forall i \in \{1, 2, ..., n_r - 1\}, \forall j \in F, \forall k \in \Gamma \setminus \{0, 1\} \tag{6}$$

$$\mu_{ijk} \leq a_{jk}\theta_{ijk} + (1 - \theta_{ijk})Q \qquad \forall i \in \{1, 2, ..., n_r - 1\}, \forall j \in F, \forall k \in \Gamma \setminus \{0\} \tag{7}$$

$$\delta_{ijk} \leq \theta_{ijk} Q \qquad \forall i \in \{1, 2, ..., n_r - 1\}, \forall j \in F, \forall k \in \Gamma \setminus \{0\} \tag{8}$$

$$\theta_{ijk} \leq \epsilon_{ij} \qquad \forall i \in \{1, 2, ..., n_r - 1\}, \forall j \in F, \forall k \in \Gamma \setminus \{0\} \tag{9}$$

$$\sum_{j\in F} \epsilon_{ij} \leq 1 \qquad \forall i \in \{1, 2, ..., n_r\} \tag{10}$$

$$x_i \geq \sum_{j\in F} \sum_{k\in\Gamma\setminus\{0\}} \frac{\delta_{ijk}}{\rho_k} - w_{i+1} \qquad \forall i \in \{1, 2, ..., n_r - 1\} \tag{11}$$

$$\sum_{j\in F} \sum_{k\in\Gamma\setminus\{0\}} \frac{\delta_{ijk}}{\rho_k} \leq b_i - x_{i-1} \qquad \forall i \in \{1, 2, ..., n_r - 1\} \tag{12}$$

$$\phi_i \geq 0 \qquad \forall i \in \{1, 2, ..., n_r\} \tag{13}$$

$$\delta_{ijk} \geq 0 \qquad \forall i \in \{1, 2, ..., n_r\}, \forall j \in F, \forall k \in \Gamma \setminus \{0\} \tag{14}$$

$$\mu_{ijk} \geq 0 \qquad \forall i \in \{1, 2, ..., n_r - 1\}, \forall j \in F, \forall k \in \Gamma \setminus \{0\} \tag{15}$$

$$\theta_{ijk} \in \{0, 1\} \qquad \forall i \in \{1, 2, ..., n_r - 1\}, \forall j \in F, \forall k \in \Gamma \setminus \{0\} \tag{16}$$

$$\epsilon_{ij} \in \{0, 1\} \qquad \forall i \in \{1, 2, ..., n_r\}, \forall j \in F \tag{17}$$

$$x_i \geq 0 \qquad \forall i \in \{1, 2, ..., n_r - 1\} \tag{18}$$

The objective function (1) seeks to minimize the total charging time of all the visits to the intermediate CSs. Constraints (2) and (3) are the consistency of the battery level upon arrival at each CS, namely the battery level when the bus arrives at this CS is equal to the battery level after charging at the last CS minus the energy consumption (simply the distance). Constraints (4) and (5) impose that the battery level after charging equals the battery level before charging plus the total amount charged within all the segments of the charging function. Constraints (6) and (7) impose that if the battery is charged within a segment, the battery level after charging is within the bounded values $a_{j,k-1}$ and a_{jk}. Constraints (8) ensure that a battery can be only charged at the selected segments of the charging function. Constraints (9) impose that charging is allowed only if the CS is visited. Constraints (10) forbid a bus from visiting two or more CSs simultaneously. Constraints (11) use variables x_i to represent whether the charging time affects the succeeding legs. If the charging time at i is smaller or equal to w_i, then the succeeding leg's maximum charging time is not affected; otherwise, they are deducted by x_i. Then constraints (12) impose the charging time at i does not exceed the allowed slack minus the deducted values by the previous legs. Finally, constraints (13) - (18) set the range of the decision variables.

In terms of choosing the CSs to visit, naturally, the CSs at the beginning and the end are still the closest ones as before. For the CSs in the middle, the principle is similar to the two-leg case while more complicated. Instead of the rank of one single CS, all the combinations of the mid-route CSs are ranked according to the total distance from and to the CSs. Correspondingly, the first combination that respects the battery levels is chosen. Apart from this, the CSs can be chosen by CPLEX as well, as in the mathematical model j is the index of CS, and which CS to visit is to be determined.

To accelerate the procedure of the FRVCP, certain infeasible solutions can be discarded before sending them to CPLEX, given the premise that the piecewise linear charging function is concave (i.e., $\rho_{k-1} \geq \rho_k, \forall k \in \Gamma \backslash \{0, \gamma\}$), in other words, the first segment has the fastest charging rate. Besides, since the maximum buffer regardless of the impact of the former legs at each position is known, i.e. b_i in constraints (12), if $\frac{b_i}{\rho_0}$ is less than the travel distance of leg i, this solution can be discarded directly. Nevertheless, this is only a rough strategy, and more sophisticated strategies can be developed in future work.

4.3. Large neighborhood search

In order to solve the EODBRP, a large neighborhood search (LNS) (Pisinger and Ropke, 2019) is developed. In the LNS framework, the best-known solution is partially destroyed and reconstructed in each iteration. In recent years, the LNS has shown its effectiveness in solving the DARPs (Li et al., 2016; Belhaiza, 2019; Gschwind and Drexl, 2019).

4.3.1. Destroy operator

In the LNS framework, solutions need to be destroyed and then rebuilt, in order to escape local optima. In this regard, a destroy operator is needed. In our algorithm, in each iteration, the destroy operator removes a certain number (represented by K) of requests from the best-known solution. The requests are randomly chosen while the total number K is predetermined and constant. More specifically, when requests are removed and not served by any bus, the buses can possibly have fewer stations to visit and larger buffers, thus can leave room for new solutions. With regard to K, it can be varied strategically. If K is too large, the preferable properties resulting in the best-known solution can be significantly eliminated, then the new solutions will be likely of low quality; on the other hand, a too small K can result in a too small neighborhood and the search procedure can be thus slow. Hence in Section 5.2 different values of K are tested.

4.3.2. Repair operator

The principle of the repair operator is similar to the greedy initial solution algorithm, i.e., inserting the unassigned requests into the current solution, each at the best position with the minimum increase of the objective value, that respects the constraints of time windows, bus capacity and charging. Same as in the initial stage, in the repair operator, the charging decisions are adjusted once a route has been modified. More specifically, the same procedure of solving the FRVCP in the initial stage is triggered in order to check whether this modification results in a feasible charging solution, and calculate the optimal charging duration for each route.

However, there are several differences with the initial solution algorithm. First, the requests in the repair operator are inserted in the inverse order of being removed, namely, the request removed first is the last one to be inserted. In this way a new solution can be possibly generated; otherwise, the original solution is likely to be reached again. In addition, the destroy-repair operator is repeated σ times resulting in maximum σ different candidate solutions, and the best one in terms of the objective value is chosen as the current solution for the LS to further improve. In such manner, a more promising current solution is possibly made. Divergent values of σ are tested with the combinations of different values of K in the destroy operator in Section 5.2.

In the destroy-repair cycle, infeasible solutions can occur. If a new solution fails at serving all the passengers due to any violation of the time windows, bus capacity or charging, this solution will not be considered as a candidate but simply discarded. Thus only feasible solutions will be kept as candidates for further possible improvement by the LS. Note that due to infeasible solutions, the actual size of the candidate pool can be smaller than σ.

4.3.3. Local search

An LS operator is included in the LNS framework and applied after a new feasible constructive solution is found by the repair operator, thus the quality of this new solution can be possibly improved by making small changes. This is done by a reinsert operator. Specifically, each time the repair operator generates the best new feasible solution among σ candidates, the reinsert operator is applied to this solution by trying to remove and reinsert each request sequentially. Each request is inserted into the position with the minimum objective value, if none of the constraints is violated. If there is no better position among all buses in terms of the objective value, the request remains in its original position. The LS repeats until no improvement is found within the last round where it examines all the requests in sequence.

4.3.4. Acceptance criterion

In each iteration of the LNS framework, each time a new feasible solution is generated, its objective value is compared to that of the best-known solution. If the value improves, the new solution is accepted and it replaces the best-known solution immediately.

4.3.5. Stopping criterion

After the initial solution is computed, the loop of the LNS algorithm starts. The algorithm terminates if a predetermined runtime, or a round limit without improvement (will be specified in Section 5) is reached.

5. Computational experiments

In this Section, the experimental data and computational results are presented. Initially, the instance generation is explained in Subsection 5.1, then the parameters in the LNS framework are tested in Subsection 5.2. Next, the properties of the BKS are analyzed in Subsection 5.5, followed by the analysis of the BSA in Subsection 5.6. Then whether the LS operator contributes to the solution quality is investigated in Subsection 5.7. Last but not least, the comparisons with conventional buses are presented in Subsection 5.8.

5.1. Instance generation

The instances are generated based on the map of Antwerp, Belgium. The city's area is 204.51 km^2 with 10 randomly located clusters, each has a maximum area $4km^2(2km \times 2km)$. Figure 3 shows the city map and the road network with clusters (in red). In order to make the instances more realistic, real stations on the map within the clusters are used for passengers to board and alight. The total number of stations is 215, with the average distance between neighbors around 500 meters. In addition, passengers' trips can be either intra- or inter-clusters. For the detailed instances' generation, we refer to Queiroz et al., 2022.

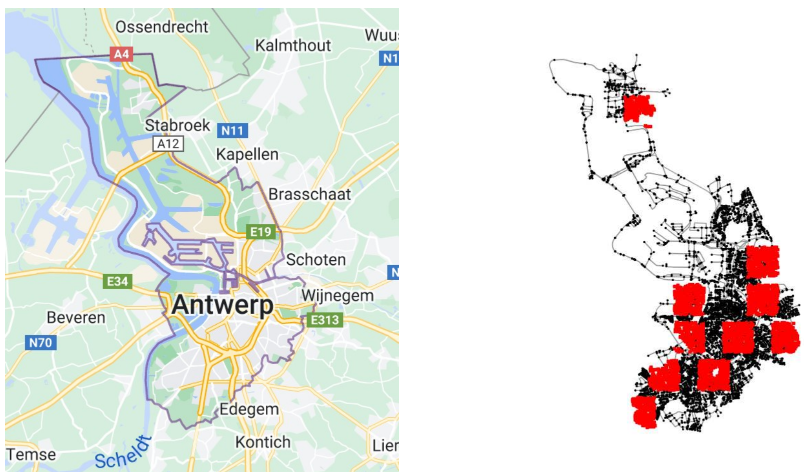


Figure 3: Map and clusters' locations

The number of passengers ranges from small, medium to large, specifically, 100, 200, . . . , 900, 1000, 1500, 2000, 3000. There is one instance of each and there are in total 13 instances. The topology of the clusters and stations, as well as the generation principle of passengers, remain the same in all instances.

The travel speed is 30km/h and assumed constant everywhere for simplicity. The operation time is from 6 am to 6 pm (12 hours). Passengers' time windows are evenly distributed within the operation time. The time window for each passenger is set to 1.5 $\times$ direct ride time, where 1.5 is chosen intuitively, thus the time windows are neither too strict nor too loose, which is in line with our definition of the on-demand bus: a tradeoff between the public transport and a direct ride.

Each request can have at most two stations respectively to board and alight.

Three different types of buses are used in this study for the variant experiments. We choose buses on the market for the sake of realistic formulations of the charging functions and the battery capacities. They are the electric minibus Karsan e-Jest with single and double batteries (`https://`

www.karsan.com/en/jest-electric-highlights), as well as a transit bus BYD K7MER (https://en.byd.com/bus/k7mer). The detailed parameters of the bus seats and maximum ranges are listed in Table 1. The capacities of the buses are equal to the numbers of passenger seats written in the brochures, as standing is not considered in this work, due to the consideration of service quality.

On the other hand, the exact charging functions with the fast charging mode are not specified in the brochures. Instead, we calculate them based on the regulation presented by Montoya et al., 2017, i.e., the charging rate differs among the three segments of a piecewise linear charging function (see Fig. 2): in the first segment, 0.7 time units per 1 unit of charge from 0 to 85% (excluding) of the battery capacity; in the second segment, 1.5 time units per 1 unit of charge from 85% (including) to 95% (excluding) of the battery capacity; in the third segment, 5 time units per 1 unit charge from 95 percent (including) to 100 percent (including) of the battery capacity. The piecewise linear charging function is merely an approximation of the real-world charging function, with relative absolute errors ranging from 0.9% to 1.9% (Montoya et al., 2017), thus it can be considered as a precise approximation.

Based on the calculation above, the detailed parameters of the approximated charging functions are also provided in Table 1. Note the battery capacity is usually represented with kWh, while in this study, as we assume the energy consumption is proportional to the traveled distance, we replace kWh with the maximum range (km) given in the brochures, and the regulation described above still holds. However, in practice, the relationship between the consumed energy and the travel distance is more complicated.

Table 1: Bus parameters

bus	seats	break points of time (min)			break points of amount (km)		
		c_{s1}	c_{s2}	c_{s3}	a_{s1}	a_{s2}	a_{s3}
mini (single)	12	58.44	73.17	97.72	89.25	99.75	105
mini (double)	12	69.06	86.375	115.2	178.5	199.5	210
regular	20	72	90	120	267.75	299.25	315

All code is written in C++ (Visual Studio 2017) and all tests are performed on a personal computer with an Intel® Core^{TM} i7-8850H 2.60Ghz processor, 16GB RAM, and Windows 10 system.

5.2. Parameter setting

The first experiment evaluates the impact of divergent K and σ on the performance of the LNS, as illustrated in Section 4.3. The double-battery minibus is used for this experiment. The values of K are 2, 5, and 10 requests despite the instances' sizes. The minimum is set to 2 since 1 request at a time is identical to the LS operator in our algorithm. Similarly, σ is chosen from four values: 1, 5, 10, and 15. The minimum value of 1 means without the elite selection. Since each time the destroy operator chooses the requests randomly, the best-known solution varies as well. To obtain an adequate estimate of the performance of the LNS, each instance with each combination of K and σ is run 10 times, and the resultant average value, standard deviation, min, max, and quartiles are computed. Each time the LNS algorithm is forced to terminate when the duration reaches 0.1 $\times$ instance size seconds. For example, the instance with 100 passengers is allowed to run for 10 seconds, while with 1000 passengers the algorithm stops once the duration reaches 100 seconds.

The runtime is limited due to the total number of combinations, also because the combination that achieves the steepest descent within a short time is preferred. For each instance, the combination that performs the best is listed in Table 2. Additionally, for each combination, the frequency that it performs the best is counted. Then the results are shown in Fig 4. The combinations that do not appear on the x-axis correspond to the fact that their frequency equals to zero.

Despite the limited number of instances and randomness, there are several observations. First, none of the tested values of K or σ is optimal for all instances. However, a smaller K usually performs better, K = 2 more frequently reaches a lower total URT than K = 5 or 10. The reason is that the BKS is destroyed less thus the preferable features are more likely to be maintained if K is small. A similar conclusion is reached in Galarza Montenegro et al., 2021. Next, although the frequencies of reaching the lowest total URT when $\sigma = 5$, 10, or 15 are comparable, $\sigma = 1$, i.e., without selection of the LNS solutions to apply the LS, is less likely to be optimal.

In addition, a method for testing the robustness of the combinations is proposed, in order to prevent selecting a combination that performs the best for certain instances but poorly for others. The rank of each combination among all the instances is recorded. In particular, if a combination outperforms the others for one instance, it receives a score of 1. On the other hand, it receives a score of 12 if it performs the worst out of the 12 combinations, and so on. The results are shown in Fig 5. Despite the fact that no single combination dominates the others, the previous findings are confirmed: a smaller K is preferable, and $\sigma = 1$ is the worst for various K values, while the disparity when σ varies among 5, 10 and 15 is not obvious.

Table 2: The combination of K and σ that performs the best for each instance

instance	100	200	300	400	500	600	700	800	900	1000	1500	2000	3000
K	10	5	2	2	2	2	2	2	5	2	2	5	2
σ	15	15	10	15	5	5	5	15	10	5	10	10	1

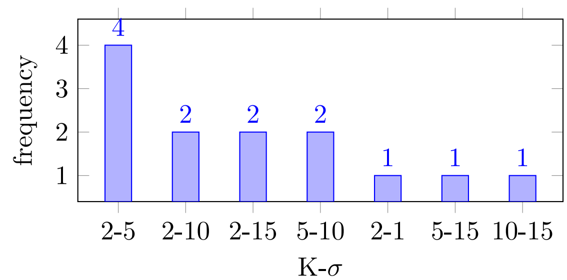


Figure 4: The frequency of each K and σ combination that outperforms the others

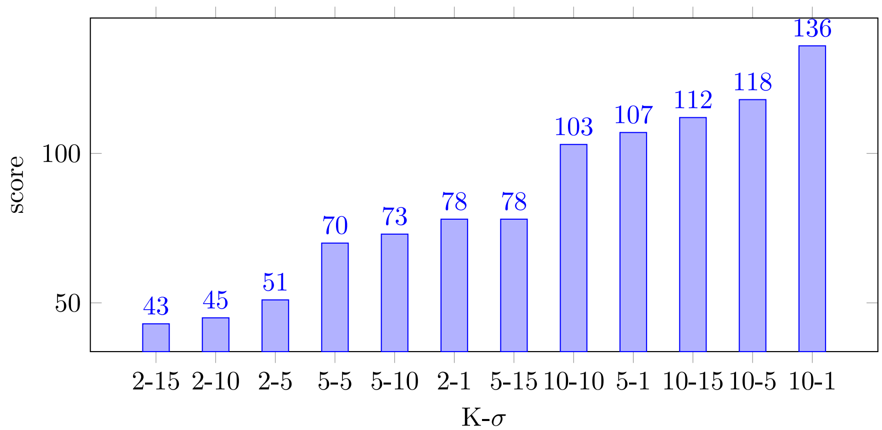


Figure 5: The overall score of each combination of K and σ

The former experiments are conducted with the runtime equal to 0.1 $\times$ instance size seconds. However, it is worth investigation whether the favorable values of K and σ change if the runtime is prolonged. Therefore, we allow the LNS algorithm to run for 0.5 $\times$ instance size seconds, and the results are analyzed in the same manner in Table 3 as well as Figures 6 and 7. An interesting observation can be drawn that, destroying the best known solution to a larger extent is more likely to result in a better solution, i.e., K equal to 5 or 10. Therefore, if a good solution is needed as soon as possible, a smaller K of the destroy operator is preferable; otherwise, a larger K can perform better. Regarding the values of σ, there is still no notable difference among 5, 10 and 15. Nevertheless, $\sigma = 1$, namely without selection, still performs the worst.

Table 3: The combination of K and σ that performs the best for each instance, prolonged runtime

instance	100	200	300	400	500	600	700	800	900	1000	1500	2000	3000
K	10	10	5	10	5	2	5	5	10	10	10	5	10
σ	5	10	15	15	15	5	5	10	5	5	10	15	5

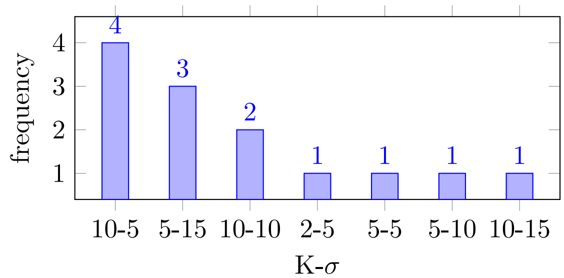


Figure 6: The frequency of each K and σ combination that outperforms the others, prolonged runtime

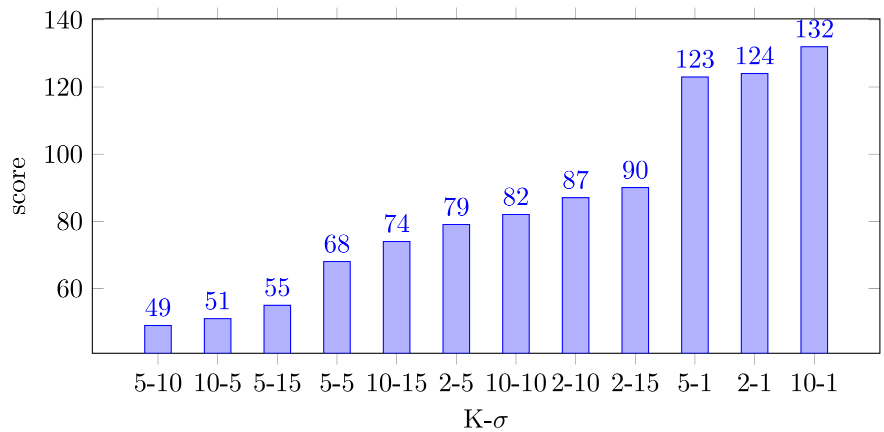


Figure 7: The overall score of each combination of K and σ, prolonged runtime

5.3. Compare the algorithm with LocalSolver

In this paper, we deal with a problem with too many constraints to be solved by exact methods. Instead, we compare our heuristic algorithm with LocalSolver (version 11.5) that is an optimization solver combining exact and heuristic methods. The mathematical model in Appendix A is formulated in LocalSolver. We limit the execution time of LocalSolver to 4 hours (14400 seconds). LocalSolver and the proposed heuristic are run three times for each instance, and the average values are taken. We test four tiny instances, each contains, respectively, 6 passengers and 4 stations, 7 passengers and 5 stations, 8 passengers and 5 stations, as well as 10 passengers and 6 stations. In all the four instances, the number of buses is fixed to 1, each with the capacity equal to 8. There is only one CS, and the bus needs to charge exactly once in midroute. The optimal solution is to charge the bus partially.

The results are summarized in Table 4. The first column lists the sizes of the instances, "p" represents passengers and "s" abbreviates stations. The next columns present the gap between the objective function values (the total URT) of the found solutions and the global optima. The tested instances are tiny and straightforward, and the optimal solutions can even be calculated by hand. It happens that the first feasible solutions found by both LocalSolver and the proposed method are already the optimal solutions. In addition to the gap, this table lists the average CPU time per instance required to yield the solutions.

As shown by the runtimes in the table, LocalSolver spends respectively seconds and minutes to solve the instance with 6 passengers and 4 stations, as well as the instance with 7 passengers and 5 stations. However, for the instance with only one more passenger, LocalSolver spends almost three hours to yield a feasible solution. Finally, for the instance with 10 passengers, LocalSolver is unable to find a feasible solution within 4 hours.

Thus, only tiny instances of the EODBRP can be solved by LocalSolver within an acceptable time limit. In literature (Melis and Sörensen, 2022; Lian et al., 2022), LocalSolver has been applied in solving the ODBRP with conventional vehicles. Correspondingly, all the constraints related to electricity consumption and charging process are excluded in the mathematical model (Melis and Sörensen, 2022). In this case, LocalSolver was able to find feasible solutions for instances with hundreds of passengers. Thus, the constraints related to electricity consumption and the charging process make it substantially more difficult to solve the EODBRP.

Despite the fact that the greedy insertion already reaches the optimal solutions, in practice, it is usually unknown whether the global optimal solution is obtained by a heuristic algorithm or not. Therefore, we still implement the LNS algorithm and record the runtimes to reach the stopping criterion (1000 consecutive rounds without implement). The LNS algorithm takes less than 0.2 seconds to terminate as shown in Table 4. To summarize, the first and optimal solutions are found in less than 0.2 seconds by the greedy insertion and the LNS algorithm, while they take drastically longer time to be obtained by LocalSolver. This proves the necessity of a fast solution method for the EODBRP and the reliability of the proposed algorithm.

We continue to examine the performance of the LNS algorithm on larger instances in the next subsection.

instance	first and best solutions					
	objective function value gap			**runtime**		
	LocalSolver	**greedy**	**LNS**	**LocalSolver (s)**	**greedy (ms)**	**LNS (s)**
6p/4s	0	0	0	28	0.04	0.12
7p/5s	0	0	0	499	0.05	0.13
8p/5s	0	0	0	10125	0.07	0.15
10p/6s	-	0	0	-	0.09	0.18

Table 4: Comparison between the proposed method and LocalSolver

5.4. Performance of the LNS algorithm

As the comparison with LocalSolver is not feasible for large instances, in this subsection, we present the runtimes of the LNS algorithm on the former generated instances in Subsection 5.1. Additionally, we calculate the percentages of improvement over the initial solutions created by greedy insertion, in order to demonstrate that the algorithm can yield good solutions. The double-battery minibus is used for this experiment. Table 5 presents the statistics of runtimes of the LNS algorithm, including the average, minimum, maximum values and the standard deviation, as well as the percentages of improvement.

A few observations can be drawn from the table. First, the runtime varies drastically among instances, and it is essentially dependent on the size of the instance. For instances up to 800 passengers, the LNS algorithm terminates within several minutes. While for the largest instance, an individual run with the given stopping criterion can take more than two hours. Second, the standard deviation of the runtime is large, due to the fact that the heuristic algorithm tends to explore divergent neighborhoods in each run of the algorithm. Despite of the variance in runtime, the best known solutions found by the individual runs are actually quite close in the objective values, as the deviations shown in the Subsection 5.5.1. Finally, the LNS algorithm can drastically improve the objective function value, as the average URT is reduced by 15% to 32%, on average 25%, compared to the initial solutions. Meanwhile, the more efficient solutions result in less charging, as the charging amount on average is reduced by 23%.

In order to examine how the solutions are improved during the execution, let us further inspect the progress of an individual run of the LNS algorithm. Figure 8 shows this progress for the instance with 100 passengers. Each black circle represents a newly found solution with a lower feasible objective function value.

As anticipated, the LNS algorithm makes the most progress at the beginning. The objective function value drastically decreases from 19.36 to 15.03 minutes within the first 0.6 seconds. On

instance	runtime (s)				URT (%)	char (%)
	ave	**std**	**min**	**max**		
100	13.74	3.88	10.17	19.20	28.92	28.32
200	58.61	26.04	26.86	89.13	32.17	30.86
300	65.96	23.21	49.47	100.23	27.43	19.15
400	90.90	58.97	28.69	166.56	28.00	26.83
500	144.87	121.55	68.05	325.11	25.95	18.10
600	351.34	137.01	184.83	509.00	20.86	25.25
700	426.48	138.04	224.94	538.23	21.03	26.91
800	543.43	146.54	394.91	729.05	20.85	6.73
900	1076.98	469.23	640.19	1621.11	30.88	23.09
1000	1212.20	336.81	985.61	1700.77	27.85	26.36
1500	2992.16	488.92	2557.08	3619.47	26.34	27.66
2000	5026.45	2193.54	3712.80	6195.59	18.86	10.01
3000	7861.79	1632.78	6146.22	9998.50	15.66	29.28
ave					24.98	22.97

Table 5: The runtime, the reduced URT and charging amount of the instances, double-battery bus

the other hand, reaching the LNS algorithm's stopping criterion, i.e., 1000 iterations without an improvement, takes up the majority of the runtime. As shown in the figure, the algorithm already finds the local optima after 1.5 seconds, while it takes 10 more seconds to terminate. Therefore, it can be inferred that even when the number of rounds without improvement in the stopping criterion is smaller, the LNS algorithm may possibly still produce good results.

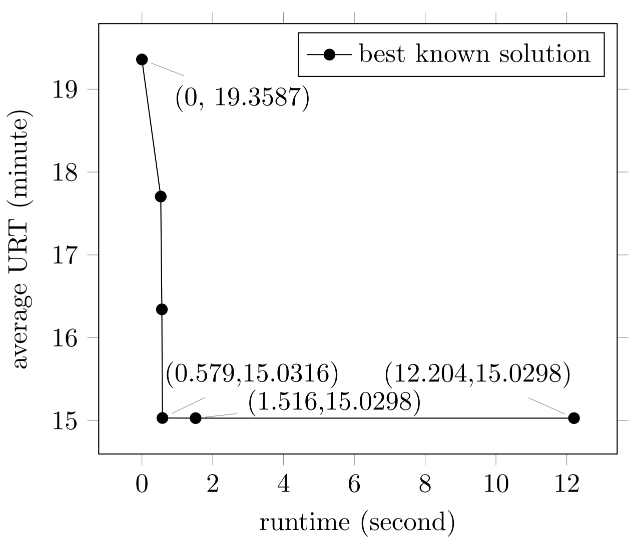


Figure 8: Progress of the LNS algorithm, 100 passengers

As a comparison, we then inspect an individual run for the instance with 3000 passengers, which is shown in Figure 9. Similarly, the LNS algorithm improves the objective function value substantially within the first 100 seconds. More specifically, the average URT decreases from 21.01 to 18.47 minutes, around 12%. While the algorithm spends 4803 more seconds ($\tilde{8}0$ minutes) only to improve the average URT to 16%. Naturally, when the objective function value is already lower, it generally takes longer and longer time to find a possibly better solution. Finally, reaching the stopping criterion takes roughly 22 minutes. Thus for larger instances, the algorithm has to spend considerable time both in making slight improvements and in terminating.

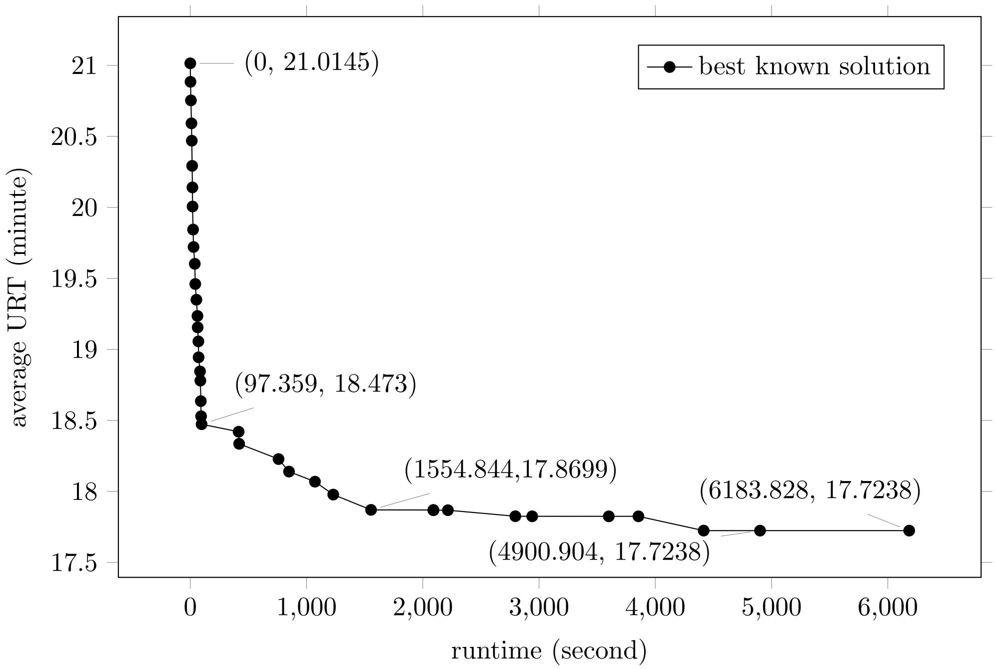


Figure 9: Progress of the LNS algorithm, 3000 passengers

5.5. Properties of the best-known solutions

The properties of the BKS found in our study are analyzed in this subsection, in order to provide readers with possibly useful information in developing new solution methods for the EODBRP with intermediate charging. The single - and double - battery mini bus are in use for the experiment in this subsection. First, once the combination of K and σ (K = 2 and σ = 15) that performs the best on average under the shorter runtime is found in the last subsection, it is fixed for all the experiments. In other words, the best combination given by the double-battery mini bus are in use for all the instances with the single-battery bus directly, despite the combination do not necessarily hold for every case. However, the LNS does not stop when the runtime reaches 0.1 $\times$ instance size, but if the BKS has not been improved after 1000 consecutive rounds. The motivation for changing the stopping criteria is to possibly reach better solutions.

Subsequently, the corresponding solutions are analyzed below. The solutions with the double-battery bus are presented first in Subsection 5.5.1, followed by the single-battery case in Subsection 5.5.2.

5.5.1. Best-known solutions with the double-battery buses

In this subsection, the BKS of the EODBRP with the double-battery buses is presented and analyzed. Specifically, the average URT of each passenger, as well as the average charging time and amount of each charged bus, are presented in Table 6 and Table 7. Same as in the last subsection, all the results are the average of ten separate solutions.

The results of the average URT in Table 6 show that the URT does not have an obvious increasing trend with instance sizes, either with the electric or conventional buses. Next, the last column of Table 6 is the increased URT compared with conventional buses. A distinct gap exists between electric and conventional buses, with the average value of all the instances equal to 26.11%. This is due to EVs needing to make detours to the CSs and spending excessive time being charged, thus the time for serving passengers is less. Despite the fact that the detours are done with empty buses, the detours still occupy considerably the operation time. This adds further constraints to our problem. In particular, having EVs instead of conventional vehicles is in some ways equivalent to having tighter time windows for the passengers: this indirectly affects the solution quality.

	URT (minute)				
instance	**ave**	**std**	**min**	**max**	**conv (%)**
100	13.76	0.02	12.99	15.16	32.72
200	13.22	0.29	12.56	13.84	15.22
300	14.45	0.17	13.88	15.65	31.68
400	13.93	0.09	13.62	14.49	23.76
500	15.32	0.13	14.33	16.12	29.51
600	15.74	0.01	15.67	15.84	37.92
700	16.24	0.09	15.10	16.73	32.71
800	16.14	0.05	15.47	16.85	38.60
900	14.59	0.05	13.32	14.90	17.75
1000	14.31	0.04	13.20	14.83	23.76
1500	15.17	0.01	15.03	15.66	20.37
2000	16.68	0.00	16.58	16.79	25.66
3000	17.72	0.00	17.72	17.73	9.82

Table 6: The average URT of each passenger, double batteries

instance	charging time (minute)				charging amount (km)				slope
	ave	std	min	max	ave	std	min	max	
100	11.61	0.89	10.18	13.11	30.01	2.31	26.30	33.89	2.58
200	12.97	1.62	10.50	14.48	33.52	4.20	27.14	37.43	2.58
300	15.38	1.13	13.75	16.89	39.75	2.88	35.53	43.66	2.58
400	23.95	1.54	23.69	24.06	61.91	3.96	61.23	62.20	2.58
500	16.34	0.85	15.66	17.75	42.22	2.15	40.48	45.87	2.58
600	20.20	1.04	18.41	21.06	52.21	2.72	47.58	54.44	2.58
700	15.63	0.78	14.93	16.83	40.40	2.01	38.58	43.51	2.58
800	22.93	1.02	22.35	24.51	59.28	2.59	57.77	63.34	2.58
900	19.45	0.58	18.72	20.33	50.26	1.61	48.38	52.53	2.58
1000	16.99	0.48	16.22	17.36	43.92	1.20	41.93	44.88	2.58
1500	17.48	0.16	17.47	17.51	45.18	0.31	45.16	45.25	2.58
2000	13.44	0.42	12.90	13.82	34.74	1.05	33.34	35.72	2.58
3000	24.87	0.27	24.57	24.97	64.28	0.55	63.51	64.54	2.58

Table 7: The average charging time and amount of the charged buses, double batteries

Next, our BKS are made up of 355 routes in total, and the fraction of which that include an intermediate charging are presented in Fig. 10. The data show that 73.2% of the routes in the BKS contain an intermediate CS. The percentage varies among instances, with a slight increasing trend when the sizes of the instances are larger. The results show intermediate charging and its solution quality are essential for decent EODBRP solutions, especially in each of these instances the number of buses is set as the smallest value found by the initial solution (which will be explained in Section 5.8).

Thirdly, the times of intermediate charging per route is calculated. Despite the fact that charging more than once is allowed, in the solutions with the double-battery buses, the 73.2% buses that charge in mid-route only charge once. Given the data in Table 7, each bus on average spends 17.79 minutes to charge the amount covering 45.98 km's ride.

Finally, we analyze the segment(s) within which the energy is recovered through mid-route charging. According to the calculated slope (the charging amount divided by charging time) in Table 7, most buses only charge in the first segment, i.e., the one with the fastest charging rate. This is reasonable since when visiting CSs, the objective is to minimize the charging time, and only until the battery level is as low as it is in the first segment the CSs are visited. In addition, whether each bus is fully or partially charged is recorded as well, and it shows all the mid-route charges are partial, this proves the efficiency and importance of the partial charging policy compared with full charging.

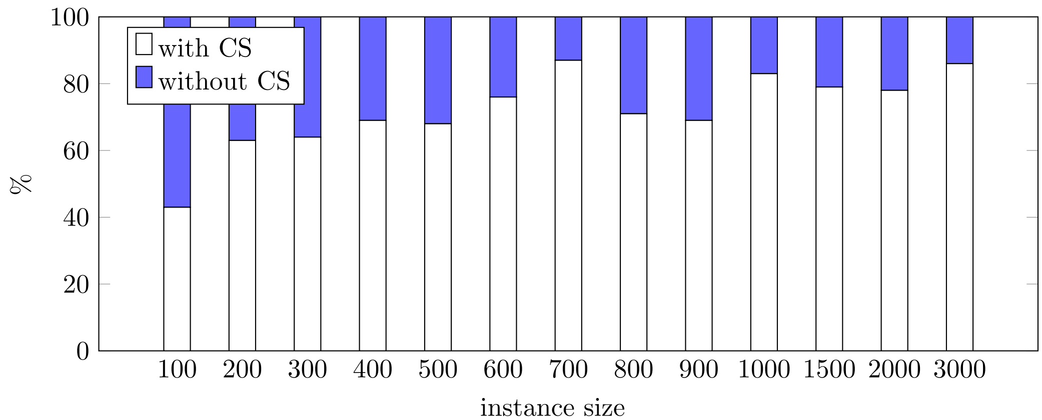


Figure 10: Percentage of the routes with/without an intermediate CS by instance size, double-battery bus

5.5.2. Best-known solutions with the single-battery buses

In this subsection, the BKS of the EODBRP with the single-battery buses are presented and analyzed. Same as the last subsection, the average URT of each passenger, as well as the average charging time and amount of each charged bus, are presented in Table 8 and 9. All the results are the average of ten separate solutions as well.

The results of the average URT in Table 8 show that despite the URT is not obviously correlated to the instance size, the average URT of the EODBRP with single-battery buses is generally even larger compared to the double-battery buses. Specifically, the average gap between the electric and conventional buses of all instances is 39.32%. Correspondingly, the charging time and amount drastically increase, i.e., each bus on average spends 57.36 minutes to charge the amount covering 82.17 km's ride. The reasons are that the range of the single-battery bus is only half of the double-battery one, and the charging rates are substantially lower as well. In particular, the charging rate of the first segment of the double-battery bus is $\frac{178.5}{69.06} \approx 2.58$, while the one of the single-battery bus is only $\frac{89.25}{58.44} \approx 1.53$.

The BKS consist 459 routes in total, and the fraction of which that include intermediate charging are presented in Fig. 11. In the double-battery case, certain buses charge once while the others do not charge. However, in the single-battery case, all buses charge at least once in mid-route. In particular, 88.9% of the routes in the BKS contain an intermediate CS, while the 11.1% contain two. The results once again show intermediate charging and its solution quality are essential for the EODBRP.

Finally, the calculated slope in Table 9 indicates the segment(s) within which the energy is recovered through mid-route charging. Compared with the double-battery case, it is more common that buses not only charge in the first segment, but also the slower segment(s). Since if the frequency of mid-route charging increases, the problem is more complicated to solve, while the positions to insert CSs are not a decision of the FRVCP in this work. In other words, optimizing the positions to insert CSs is not explicitly included in the objective function, thus the charging solution may not be optimal, despite the fact that the charging time calculated by CPLEX with given CSs is optimal. Thus it is one limitation of this work. Nevertheless, none of the buses is fully charged in the result, thus it can be once again concluded that partial charging provides more flexibility and efficiency than full charging.

URT (minute)					
instance	**ave**	**std**	**min**	**max**	**conv (%)**
100	17.77	0.98	16.77	18.29	71.40
200	13.66	0.95	12.57	14.81	19.05
300	16.13	1.13	13.97	17.68	46.98
400	14.42	0.24	13.05	15.25	28.13
500	18.44	0.04	18.17	18.63	55.88
600	17.33	0.19	16.59	17.57	51.85
700	18.72	0.06	18.39	19.87	52.98
800	16.74	0.18	15.23	17.96	43.75
900	15.60	0.10	14.34	15.89	25.90
1000	15.29	0.04	14.83	15.51	32.23
1500	16.76	0.01	16.54	16.79	32.98
2000	17.24	0.02	16.87	17.48	29.90
3000	19.39	0.01	19.13	19.60	20.15

Table 8: Average URT of each passenger, single-battery bus

instance	**charging time (minute)**				**charging amount (km)**				**slope**
	ave	**std**	**min**	**max**	**ave**	**std**	**min**	**max**	
100	42.35	0.27	40.93	43.96	64.80	0.15	62.62	65.83	1.53
200	50.85	0.61	44.22	54.88	74.39	0.19	67.65	76.24	1.46
300	61.91	0.07	60.44	63.11	84.99	0.09	83.07	86.54	1.37
400	49.49	0.07	48.82	50.30	75.72	0.04	74.33	76.02	1.53
500	66.29	0.02	65.53	66.55	99.75	0.08	97.37	101.18	1.50
600	60.40	0.17	55.31	63.49	88.12	0.13	81.20	92.02	1.46
700	59.80	0.03	58.87	60.56	86.02	0.06	83.88	87.45	1.44
800	68.52	0.00	68.50	68.54	93.85	0.00	93.85	93.86	1.37
900	68.73	0.03	68.29	69.83	92.75	0.00	92.71	92.81	1.35
1000	52.32	0.02	50.89	52.96	78.07	0.05	76.15	80.17	1.49
1500	54.88	0.03	53.20	56.77	72.04	0.04	69.28	74.04	1.31
2000	46.76	0.01	45.77	47.55	71.39	0.02	69.89	72.61	1.53
3000	63.37	0.00	63.17	63.52	86.34	0.00	86.04	86.60	1.36

Table 9: Average charging time and amount of the charged buses, single-battery bus

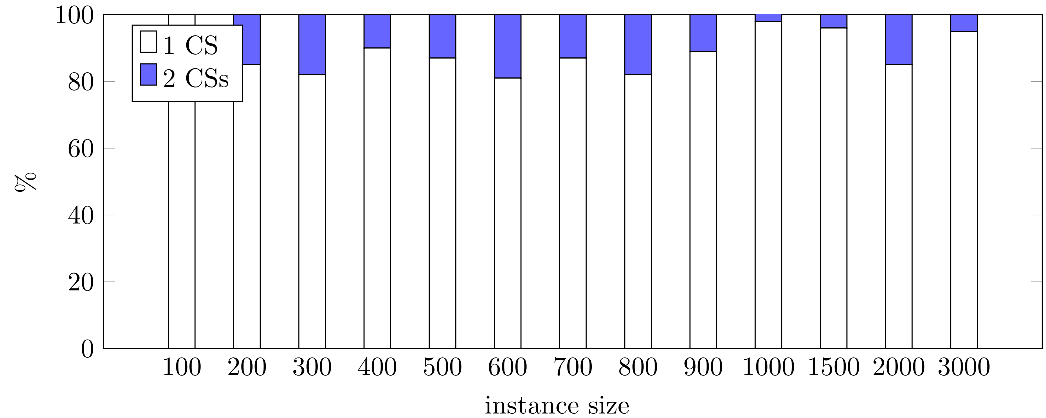


Figure 11: Percentage of the routes charging once or twice in mid-route by instance size

5.6. Effect of the bus station assignment

Since the inclusion of the BSA distinguishes the ODBRP from the DARP in essence, the effect of the BSA is investigated in this subsection. As explained in Section 3, when the BSA is considered, passengers can be assigned to either stations for boarding and alighting (since the maximum number of alternative stations is limited to 2 in these experiments). On the contrary, when the BSA is excluded, the standard way in the DARP literature is assigning passengers to the closest stations, or even transporting passengers door-to-door. Since the locations of stations are predefined and buses are only allowed to dwell at the stations in the ODBRP, when the BSA is excluded, passengers are simply assigned to the closest stations to their origins or destinations.

According to the experimental results in Melis and Sörensen, 2022, excluding the BSA leads to a significant increase in the objective value (the total URT) when conventional buses are used. Thus in this study, apart from the average URT, we further compare the impact of the BSA on the total charging time and amount. In addition, we vary in each instance the percentage of passengers who have alternative stations, namely 0, 50%, and 90% respectively. The maximum is set to 90%, as we assume in real life it might not be the case that every passenger or every station has alternates nearby. Then we remove alternative stations from the 90% case to build the 50% case. That is to say, we randomly choose 40% passengers to remove their alternative stations. Finally for case 0, simply no passenger has alternative stations. To analyze the effect of the BSA, the instances with 1000, 2000, and 3000 passengers are chosen for experiments. The double-battery buses are used. Same as before, each instance is run ten times with the 1000-round stopping criterion. The average URT, total charging time, and amount is respectively in Fig 12 (Note the y-axis starts approximately at 6 instead of 0), 13 and 14. The observations are, first, the BSA can not only reduce the URT but also the charging time and amount, thus can benefit the environment. Next, the more passengers with alternative stations, the more effective the BSA is. Nevertheless, each passenger has at most two stations to board or alight in this study, more stations can be worth investigating for future research.

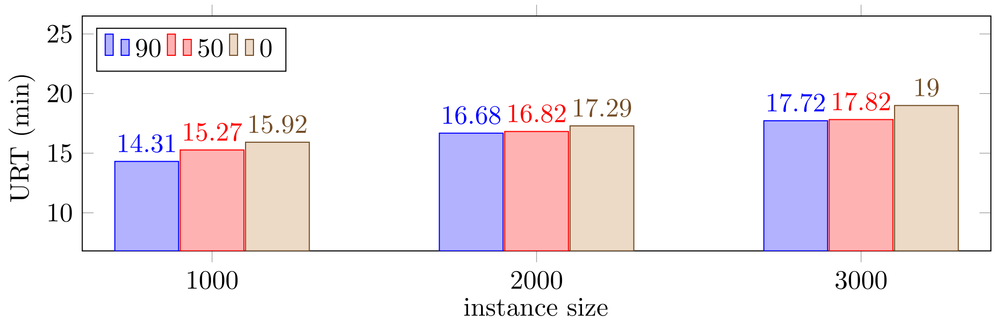


Figure 12: BSA - average URT (min)

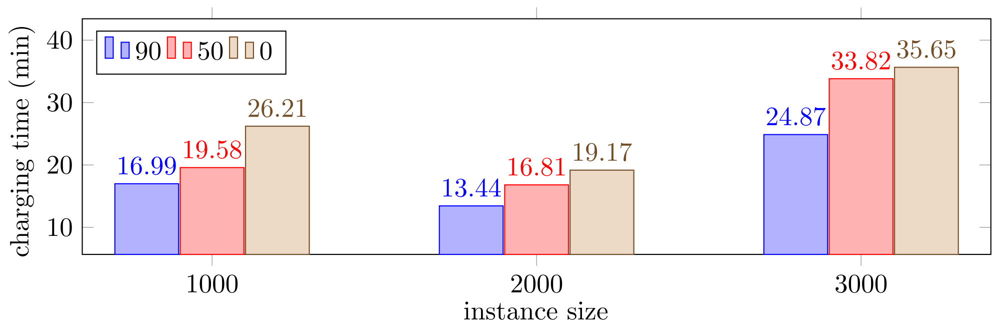


Figure 13: BSA - average charging time per charged bus (min)

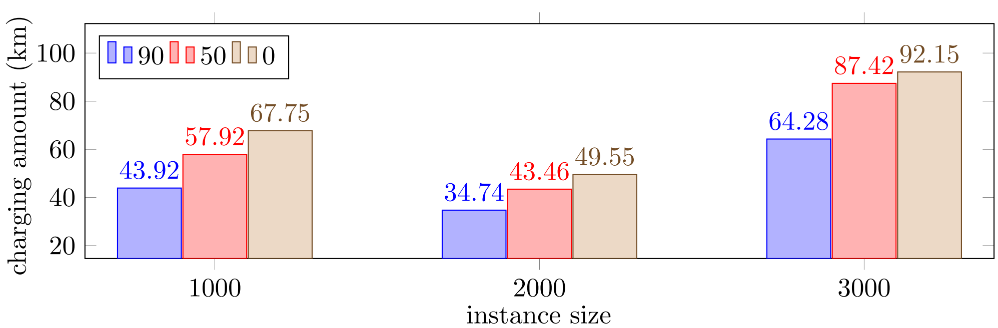


Figure 14: BSA - average charging amount per charged bus (km)

5.7. Influence of the local search operator

The LNS algorithm is combined with a local search operator which sequentially removes and reinserts each request, and the operator's influence on the solution quality is investigated in this subsection. For a fair comparison, we set the same runtime for the two algorithms with or without the LS operator. Particularly, the average runtime of the LNS with the LS operator when the BKS has not been improved within 1000 consecutive rounds is set for the LNS algorithm without the LS operator. Three instances are chosen for this test: 1000, 2000, and 3000 with the double-battery buses. Each instance is run ten times and the average URT, charging time, and amount are calculated and presented in Fig 15, where the values are the percentage relative to the solution solved by the LNS with the LS operator. For example, when the LS operator is excluded, the passengers on average have to spend around 38%, 22%, or 18% longer time on the buses, depending on the total number of passengers. It can be seen that the LS operator benefits the solution quality by reducing not only the average URT but also the charging time and amount. This efficacy holds for various instance sizes.

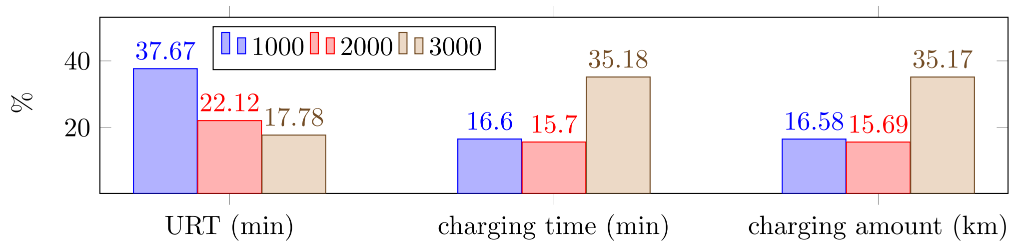


Figure 15: The LNS without the LS operator results in additional average URT, charging time and amount over the LNS with LS

5.8. Comparison with conventional buses

In this subsection, we explicitly compare the number of needed buses with conventional vehicles. For the case of conventional buses, since the upper bound of the travel distance is 360 km given our operation time (12 hours) and the constant speed (30 km/h), refueling is simply not considered. Thus the number of buses needed is only constrained by the time windows and the bus capacity.

The number of either conventional or electric buses is calculated solely by constructing the initial solution, while the LNS does not contribute to reducing the number of buses. This is a limitation of the study, as a more sophisticated algorithm can lead to more accurate numbers of buses.

Once we obtain the number of needed conventional buses, the same number is set for electric buses. If all the passengers are feasibly inserted, then the number of electric buses needed is found; otherwise, an empty bus is added and the procedure repeats.

In order to perform sensitivity analyses of the number of buses, the passengers' temporal distribution and the bus's maximum range are varied, and the results are presented in Section 5.8.1 and 5.8.2 respectively.

5.8.1. Temporal distribution of passengers

In the former instances, the passengers' earliest departure time and the time windows are temporally evenly distributed. However, in this subsection, the temporal distribution is changed to be more realistic. In particular, the durations from 7 am to 9 am, and from 4 pm to 6 pm are set as peak hours where more passengers belong to. In the experiments, we intuitively set that 70% passengers belong to the peak hours, while the rest 30% passengers are still evenly scattered in the other hours of the operation time. Therefore, for each instance, it has two temporal distributions, thus in total 26 instances. For each instance size, we compare the two passenger distributions. Note for the two temporal distributions, for each passenger, only the time window is (possibly) different, while the stations to get on and get off remain the same. The results of the numbers of double-battery minibuses with even distribution or peak hours are shown in Fig 16 and 17, where the numbers correspond to electric vehicles. The results show the number of electric buses is the same as the conventional buses in most instances, while for certain instances one more bus is needed. In addition, the distribution with peak hours needs significantly more buses, which also influences the gap between conventional and electric buses, that is, no more electric bus is needed, while for the case of even distribution, 5 out of 13 instances require one additional electric bus.

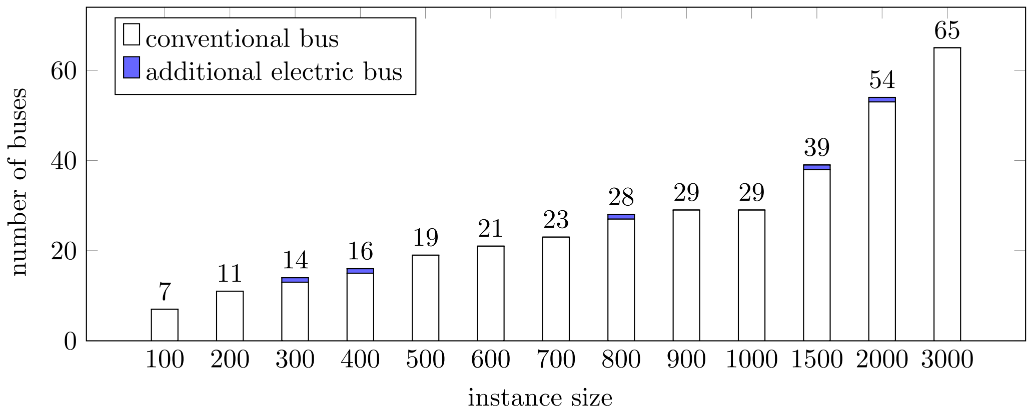


Figure 16: Number of conventional and electric buses

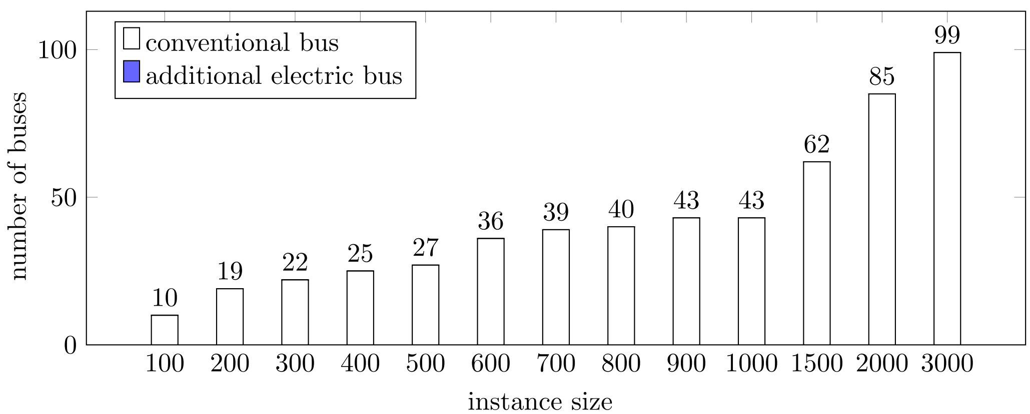


Figure 17: Number of conventional and electric buses with peak hours

5.8.2. Bus range

In this subsection, we investigate the impact of bus range on the number of needed buses. As the results of the double-battery mini bus are presented in Subsection 5.8.1, in this subsection only the results of the single-battery mini bus and the regular bus are listed, in order to evaluate the relationship of the bus range and the gap between the electric and conventional buses.

The additional single-battery mini buses or regular buses with the two temporal distributions are shown in Table 10. Comparing the single- and double-battery buses, the numbers of single-battery bus increase significantly, and more additional buses are needed relative to the conventional buses, while the numbers of the double-battery buses are generally the same as the conventional buses. Similar to the reason for the surge in the charging time and amount explained in Section 5.5.2, the increased number of buses is also due to the smaller range of the single-battery bus, in addition to the lower charging rates.

For the regular bus, despite the increased bus capacity compared with the mini one, the number of buses do not decrease, due to the small density of requests, this is one of the limitations of this study. Nevertheless, since the maximum range is significantly larger, the number of the electric buses are the same as conventional buses for all the instances with the two temporal distributions.

instances	conv	mini	reg	conv, ph	mini, ph	reg, ph
100	7	1	0	10	0	0
200	11	2	0	19	0	0
300	13	4	0	22	0	0
400	15	5	0	25	0	0
500	19	4	0	27	1	0
600	21	5	0	36	0	0
700	23	7	0	39	0	0
800	27	7	0	40	1	0
900	29	8	0	43	0	0
1000	29	11	0	43	1	0
1500	38	14	0	62	0	0
2000	53	14	0	85	0	0
3000	65	27	0	99	3	0

Table 10: The number of additional single-battery mini buses and regular buses over conventional buses

6. Conclusion

In this work, we investigate the on-demand bus routing problem with detours to charging stations. The charging functions are modeled as piecewise linear, and partial charging is allowed at the CS. To solve the problem we adopt a 'charging first, routing second' strategy, then propose a hybrid heuristic with an LNS algorithm and a subproblem FRVCP that can be solved to optimality by CPLEX. The LNS algorithm consists of parameter settings and an LS operator, and both contribute to the solution quality. The subproblem FRVCP optimizes the charging decision (how much to charge) given the fixed sequence of stations, including both the passenger stations and the CSs. Realistic instances based on a real city map are generated. The experimental results show the proposed algorithm is able to deliver high-quality solutions. Then the analyses of the solutions conclude that good solutions tend to partially charge in mid-route, and employ the segment with the fastest charging rate of the charging function as much as possible. Subsequently, the bus station assignment of the ODBRP not only reduces the URT but also the total charging time and charging amount. Finally, our results conclude both the electric mini and regular buses can substitute conventional buses under various passenger distributions.

Possible future research directions include a more realistic energy consumption function that may consider road slope, speed, temperature, or the number of passengers aboard. Another interesting direction could be to develop more sophisticated heuristics or exact methods, together with speedup techniques that can exclude quickly infeasible solutions. Furthermore, capacitated charging stations with queuing is worth studying. Last but not least, the comparison between battery charging and swapping stations as well as their location routing problem could also be an interesting research topic.

Acknowledgements

The authors would like to thank the Bijzonder Onderzoeksfonds of the University of Antwerp for funding the main author of this research.

CRediT authorship contribution statement

Ying Lian: Conceptualization, Methodology, Software, Formal analysis, Investigation, Drafting, Writing - review and editing, Visualization. **Flavien Lucas**: Conceptualization, Validation, Writing - review and editing, Formal analysis, Visualization, Supervision. **Kenneth Sörensen**: Conceptualization, Methodology, Validation, Resources, Writing - review and editing, Project administration, Funding acquisition, Supervision.

References

Baum, M., Dibbelt, J., Gemsa, A., Wagner, D., and Zündorf, T. (2019). Shortest feasible paths with charging stops for battery electric vehicles. *Transportation Science*, 53(6):1627–1655.

Belhaiza, S. (2019). A hybrid adaptive large neighborhood heuristic for a real-life dial-a-ride problem. *Algorithms*, 12(2).

Bongiovanni, C., Kaspi, M., and Geroliminis, N. (2019). The electric autonomous dial-a-ride problem. *Transportation Research Part B: Methodological*, 122:436–456.

Bruglieri, M., Pezzella, F., Pisacane, O., and Suraci, S. (2015). A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 47:221–228. The 3rd International Conference on Variable Neighborhood Search (VNS'14).

Cordeau, J.-F. and Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579–594.

Desaulniers, G., Errico, F., Irnich, S., and Schneider, M. (2016). Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405.

Erdelić, T. and Carić, T. (2019). A survey on the electric vehicle routing problem: variants and solution approaches. *Journal of Advanced Transportation*, 2019:Article ID 5075671, 48 pages.

Erdoğan, S. and Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114.

Froger, A., Mendoza, J. E., Jabali, O., and Laporte, G. (2019). Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. *Computers & Operations Research*, 104:256–294.

Galarza Montenegro, B. D., Sörensen, K., and Vansteenwegen, P. (2021). A large neighborhood search algorithm to optimize a demand-responsive feeder service. *Transportation Research Part C: Emerging Technologies*, 127:103102.

Goeke, D. and Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1):81–99.

Gschwind, T. and Drexl, M. (2019). Adaptive large neighborhood search with a constant-time feasibility test for the dial-a-ride problem. *Transportation Science*, 53(2):480–491.

He, Y., Liu, Z., and Song, Z. (2020). Optimal charging scheduling and management for a fast-charging battery electric bus system. *Transportation Research Part E: Logistics and Transportation Review*, 142:102056.

Healy, P. and Moll, R. (1995). A new extension of local search applied to the dial-a-ride problem. *European Journal of Operational Research*, 83(1):83–104.

Hiermann, G., Hartl, R. F., Puchinger, J., and Vidal, T. (2019). Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*, 272(1):235–248.

Hiermann, G., Puchinger, J., Ropke, S., and Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3):995–1018.

Hof, J., Schneider, M., and Goeke, D. (2017). Solving the battery swap station location-routing problem with capacitated electric vehicles using an avns algorithm for vehicle-routing problems with intermediate stops. *Transportation Research Part B: Methodological*, 97:102–112.

Keskin, M. and Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65:111–127.

Keskin, M. and Çatay, B. (2018). A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Computers & Operations Research*, 100:172–188.

Koç, Ç., Jabali, O., Mendoza, J. E., and Laporte, G. (2019). The electric vehicle routing problem with shared charging stations. *International Transactions in Operational Research*, 26(4):1211–1243.

Kucukoglu, I., Dewil, R., and Cattrysse, D. (2021). The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, 161:107650.

Li, B., Krushinsky, D., Van Woensel, T., and Reijers, H. A. (2016). An adaptive large neighborhood search heuristic for the share-a-ride problem. *Computers & Operations Research*, 66:170–180.

Lian, Y., Lucas, F., and Sörensen, K. (2022). The on-demand bus routing problem with real-time traffic information. Working paper, University of Antwerp.

Lozano, L. and Medaglia, A. L. (2013). On an exact method for the constrained shortest path problem. *Computers & Operations Research*, 40(1):378–384.

Ma, B., Hu, D., and Wu, X. (2021). The location routing problem of the car-sharing system with autonomous electric vehicles. *KSCE Journal of Civil Engineering*, 25(8):3107–3120.

Martínez-Lao, J., Montoya, F. G., Montoya, M. G., and Manzano-Agugliaro, F. (2017). Electric vehicles in spain: an overview of charging systems. *Renewable and Sustainable Energy Reviews*, 77:970–983.

Masmoudi, M. A., Hosny, M., Demir, E., Genikomsakis, K. N., and Cheikhrouhou, N. (2018). The dial-a-ride problem with electric vehicles and battery swapping stations. *Transportation Research Part E: Logistics and Transportation Review*, 118:392–420.

Melis, L. and Sörensen, K. (2022). The static on-demand bus routing problem: large neighborhood search for a dial-a-ride problem with bus station assignment. *International Transactions in Operational Research*, 29(3):1417–1453.

Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2016). A multi-space sampling heuristic for the green vehicle routing problem. *Transportation Research Part C: Emerging Technologies*, 70:113–128.

Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G. (2017). The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological*, 103:87–110. Green Urban Transportation.

Onn, C. C., Mohd, N. S., Yuen, C. W., Loo, S. C., Koting, S., Abd Rashid, A. F., Karim, M. R., and Yusoff, S. (2018). Greenhouse gas emissions associated with electric vehicle charging: The impact of electricity generation mix in a developing country. *Transportation Research Part D: Transport and Environment*, 64:15–22.

Pelletier, S., Jabali, O., Laporte, G., and Veneroni, M. (2017). Battery degradation and behaviour for electric vehicles: review and numerical analyses of several models. *Transportation Research Part B: Methodological*, 103:158–187. Green Urban Transportation.

Pisinger, D. and Ropke, S. (2019). Large neighborhood search. In *Handbook of Metaheuristics*, pages 99–127. Springer International Publishing.

Queiroz, M., Lucas, F., and Sörensen, K. (2022). Instance generation tool for on-demand transportation problems. Working paper, University of Antwerp.

Savelsbergh, M. W. P. (1992). The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 4(2):146–154.

Sayarshad, H. R., Mahmoodian, V., and Gao, H. O. (2020). Non-myopic dynamic routing of electric taxis with battery swapping stations. *Sustainable Cities and Society*, 57:102113.

Schiffer, M. and Walther, G. (2017). The electric location routing problem with time windows and partial recharging. *European Journal of Operational Research*, 260(3):995–1013.

Schiffer, M. and Walther, G. (2018a). An adaptive large neighborhood search for the location-routing problem with intra-route facilities. *Transportation Science*, 52(2):331–352.

Schiffer, M. and Walther, G. (2018b). Strategic planning of electric logistics fleet networks: A robust location-routing approach. *Omega*, 80:31–42.

Schneider, M., Stenger, A., and Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520.

Suzuki, Y. (2014). A variable-reduction technique for the fixed-route vehicle-refueling problem. *Computers and Industrial Engineering*, 67(1):204–215.

Zhang, L., Liu, Z., Yu, L., Fang, K., Yao, B., and Yu, B. (2022). Routing optimization of shared autonomous electric vehicles under uncertain travel time and uncertain service time. *Transportation Research Part E: Logistics and Transportation Review*, 157:102548.

Zhou, Y., Wang, M., Hao, H., Johnson, L., Wang, H., and Hao, H. (2015). Plug-in electric vehicle market penetration and incentives: a global review. *Mitigation and Adaptation Strategies for Global Change*, 20(5):777–795.

Zündorf, T. (2014). Electric vehicle routing with realistic recharging models. *Unpublished Master's thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany.*

Ángel Felipe, Ortuño, M. T., Righini, G., and Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71:111–128.

Appendix A. Mathematical model

Table A.11: Variables and parameters of the EODBRP

x_{snb}	1 if the n-th station of bus b is station s and 0 otherwise, where $n \in N$, $b \in B$ and $s \in V$
y^u_{pnb}	1 if passenger p is picked up at the n-th station of bus b and 0 otherwise, where $p \in P$, $n \in N$ and $b \in B$
y^o_{pnb}	1 if passenger p is dropped off at the n-th station of bus b and 0 otherwise, where $p \in P$, $n \in N$ and $b \in B$
q^{cap}_{nb}	net number of passengers picked up (or dropped off) at the n-th station of bus b, where $n \in N$ and $b \in B$
t^a_{nb}	arrival time of bus b at its n-th station, where $n \in N$ and $b \in B$
t^d_{nb}	departure time of bus b at its n-th station, where $n \in N$ and $b \in B$
T_p	user ride time of passenger $p \in P$
B	the fleet of buses
P	the set of transportation requests, $\|P\|$ denotes the number of requests
S	the set of bus stations
F'	the set of charging stations
V	the set of all the stations, i.e., bus stations and charging stations
Q^{cap}	capacity of bus
a^u_{ps}	1 if passenger $p \in P$ can be assigned to station $s \in S$ for pick-up
a^o_{ps}	1 if passenger $p \in P$ can be assigned to station $s \in S$ for drop-off
e_p	the earliest departure time for passenger $p \in P$
l_p	the latest arrival time for passenger $p \in P$
$TT_{ss'}$	travel time between station s and station s', where $s, s' \in V$
u_{nb}	battery level at the n-th station of bus b, where $n \in N$ and $b \in B$
$e_{ss'}$	electricity consumption from s to s', where $s, s' \in V$
Q	battery capacity
q_s	battery level upon arriving at $s \in F'$
o_s	battery level upon departing from $s \in F'$
ϵ_s	the corresponding charging time of q_s, where $s \in F'$
d_s	the corresponding charging time of o_s, where $s \in F'$
δ_s	the time spent at charging station $s \in F'$
$z_{s\gamma}$	equal to 1 if the battery level is between $a_{s,\gamma-1}$ and $a_{s,\gamma}$, where $\gamma \in \Gamma\backslash\{0\}$, upon arriving at charging station $s \in F'$
$w_{s\gamma}$	equal to 1 if the battery level is between $a_{s,\gamma-1}$ and $a_{s,\gamma}$, where $\gamma \in \Gamma\backslash\{0\}$, upon departing from charging station $s \in F'$
$\alpha_{s\gamma}$	coefficients of the break point $\gamma \in \Gamma\backslash\{0\}$ upon arriving at $s \in F'$
$\lambda_{s\gamma}$	coefficients of the break point $\gamma \in \Gamma\backslash\{0\}$ upon departing from $s \in F'$

The objective function is to minimize the total URT. Constraints (A.2) enforce that a bus can only stop at one station at the same time. Constraints (A.3) make sure that the indices used in the bus route are used consecutively and the sequence is started at the first position. Constraints (A.4) and (A.5) respectively enforce a bus to stop at one station if and only if at least one passenger uses it either to board or alight. Constraints (A.6) and (A.7) respectively impose that a station is designated to a passenger to board/alight only if the station belongs to the passenger, i.e., it is within the predefined walking distance. Constraints (A.8) impose for any two consecutive stations, the arrival time at the latter station is (larger than or) equal to the departure time at the previous one plus the travel time. Constraints (A.9) guarantee the departure time at a passenger's pickup station is greater than or equal to the earliest allowed value. Correspondingly, constraints (A.10) guarantee the arrival time at a passenger's drop-off station is smaller or equal to the latest allowed value. Constraints (A.11) impose that the pickup station precedes the corresponding drop-off one for any passengers. Constraints (A.12) enforce that each passenger gets on and gets off the same bus. Constraints (A.13) make sure each passenger is served at most once. Together with constraints (A.14), every request is served once and only once. Constraints (A.15) forbid two consecutive stations from being the same. Constraints (A.16) calculate the net capacity at each station, which is equal to the number of passengers getting on minus the one of getting off. Consequently, constraints (A.17) forbid the violation of bus capacity. Constraints (A.18) calculate the URT of each passenger, i.e., the arrival time at the get-off station minus the departure time at the get-on station. Constraints (A.19) and (A.20) make the electricity level equal to o_s when the bus departs from CS s. Constraints (A.21) compute the electricity level at the first passenger station. Constraints (A.22) - (A.27) track the battery level of each node. Constraints (A.28) define the relationship of the battery levels when arriving at and departing from a CS. Constraints (A.29) - (A.35) define the battery level and its corresponding charging time from the charging function, when a bus arrives at a CS. Similarly, constraints (A.36) - (A.42) define the counterparts when a bus departs from a CS. Constraints (A.43) calculate the time spent at each CS. Constraints (A.44) and (A.45) impose the relationship between the departure and arrival time at a node. Constraints (A.46) forbid consecutive visits to CSs. Constraints (A.47) and (A.48) ensure the copies of each CS are visited in order. Constraints (A.49) - (A.50) enforce that a bus can visit a CS only if no passengers are on board. Constraints (A.51) - (A.58) define the range for each variable.

$$\min \text{URT} = \sum_{p\in P} T_p \tag{A.1}$$

s.t.

$$\sum_{s\in V} x_{snb} \le 1 \qquad \forall n\in N, b\in B \tag{A.2}$$

$$\sum_{s\in V} (x_{snb} - x_{s(n+1)b}) \ge 0 \qquad \forall n\in N, b\in B \tag{A.3}$$

$$M\sum_{s\in S} x_{snb} - \sum_{p\in P} (y^u_{pnb} + y^o_{pnb}) \ge 0 \qquad \forall n\in N, b\in B \tag{A.4}$$

$$\sum_{s\in S} x_{snb} - \sum_{p\in P} (y^u_{pnb} + y^o_{pnb}) \le 0 \qquad \forall n\in N, b\in B \tag{A.5}$$

$$x_{snb} + y^u_{pnb} - a^u_{ps} \le 1 \qquad \forall s\in S, n\in N, p\in P, b\in B \tag{A.6}$$

$$x_{snb} + y^o_{pnb} - a^o_{ps} \le 1 \qquad \forall s\in S, n\in N, p\in P, b\in B \tag{A.7}$$

$$t^a_{(n+1)b} - t^d_{nb} - TT_{ss'} + (x_{snb} + x_{s'(n+1)b} - 2)(-M) \ge 0 \qquad \forall s, s'\in V \mid s\ne s', n\in N, b\in B \tag{A.8}$$

$$t^d_{nb} - e_p + (y^u_{pnb} - 1)(-M) \ge 0 \qquad \forall p\in P, n\in N, b\in B \tag{A.9}$$

$$t^a_{nb} - l_p + (y^o_{pnb} - 1)M \le 0 \qquad \forall p\in P, n\in N, b\in B \tag{A.10}$$

$$\sum_{n\in N} (ny^u_{pnb} - ny^o_{pnb}) \le 0 \qquad \forall p\in P, b\in B \tag{A.11}$$

$$\sum_{n\in N} (y^u_{pnb} - y^o_{pnb}) = 0 \qquad \forall p\in P, b\in B \tag{A.12}$$

$$\sum_{b\in B}\sum_{n\in N} y^u_{pnb} \le 1 \qquad \forall p\in P \tag{A.13}$$

$$\sum_{p\in P}\sum_{b\in B}\sum_{n\in N} y^u_{pnb} = |P| \tag{A.14}$$

$$x_{snb} + x_{s(n+1)b} \le 1 \qquad \forall s\in V, n\in N, b\in B \tag{A.15}$$

$$\sum_{p\in P} (y^u_{pnb} - y^o_{pnb}) - q^{cap}_{nb} = 0 \qquad \forall n\in N, b\in B \tag{A.16}$$

$$\sum_{n'\le n} q^{cap}_{nb} \le Q^{cap} \qquad \forall n, n'\in N \mid n\ge n', b\in B \tag{A.17}$$

$$T_p + (2 - y^o_{pn'b} - y^u_{pnb})M - t^a_{n'b} + t^d_{nb} \ge 0 \qquad \forall p\in P, n, n'\in N \mid n' > n, b\in B \tag{A.18}$$

$$u_{nb} \ge o_s + (\sum_{s\in F'} x_{snb} - 1)M \qquad \forall s\in F', n\in N, b\in B \tag{A.19}$$

$$u_{nb} \le o_s + (1 - \sum_{s\in F'} x_{snb})M \qquad \forall s\in F', n\in N, b\in B \tag{A.20}$$

$$u_{0b} = Q - \sum_{s\in S} e_{0s}x_{s0b} \qquad \forall b\in B \tag{A.21}$$

$$u_{nb} - u_{(n+1)b} \le (e_{ss'} - Q)(x_{snb} + x_{s'(n+1)b} - 1) + Q \qquad \forall s\in V, s'\in S, s\ne s', n\in N, b\in B \tag{A.22}$$

$$u_{nb} - u_{(n+1)b} \ge (e_{ss'} + Q)(x_{snb} + x_{s'(n+1)b} - 1) - Q \qquad \forall s\in V, s'\in S, s\ne s', n\in N, b\in B \tag{A.23}$$

$$u_s - q_{s'} \le (e_{ss'} - Q)(x_{snb} + x_{s'(n+1)b} - 1) + Q \qquad \forall s\in V, s'\in F', s\ne s', n\in N, b\in B \tag{A.24}$$

$$u_s - q_{s'} \ge (e_{ss'} + Q)(x_{snb} + x_{s'(n+1)b} - 1) - Q \qquad \forall s\in V, s'\in F', s\ne s', n\in N, b\in B \tag{A.25}$$

$$u_{nb} \le u_s + (x_{snb} - 1) * (-Q) \qquad \forall s\in V, s'\in S, s\ne s', n\in N, b\in B \tag{A.26}$$

$$u_{nb} \ge u_s + (x_{snb} - 1) * Q \qquad \forall s\in V, s'\in S, s\ne s', n\in N, b\in B \tag{A.27}$$

$$q_s \le o_s \qquad \forall s\in F' \tag{A.28}$$

$$q_s = \sum_{k\in\Gamma} \alpha_{sk}a_{sk} \qquad \forall s\in F' \tag{A.29}$$

$$\epsilon_s = \sum_{k\in\Gamma} \alpha_{sk}c_{sk} \qquad \forall s\in F' \tag{A.30}$$

$$\sum_{k\in\Gamma} \alpha_{sk} = \sum_{k\in\Gamma\setminus\{0\}} z_{sk} \qquad \forall s\in F' \tag{A.31}$$

$$\sum_{k\in\Gamma\setminus\{0\}} z_{sk} = \sum_{n\in N}\sum_{b\in B} x_{snb} \qquad \forall s\in F' \tag{A.32}$$

$$\alpha_{s0} \le z_{s1} \qquad \forall s\in F' \tag{A.33}$$

$$\alpha_{sk} \le z_{sk} + z_{s(k+1)} \qquad \forall s\in F', k\in\Gamma\setminus\{0,\gamma\} \tag{A.34}$$

$$\alpha_{s\gamma} \le z_{s\gamma} \qquad \forall s\in F' \tag{A.35}$$

$$o_s = \sum\nolimits_{k\in\Gamma} \lambda_{sk} a_{sk} \qquad \forall s \in F' \tag{A.36}$$
$$d_s = \sum\nolimits_{k\in\Gamma} \lambda_{sk} c_{sk} \qquad \forall s \in F' \tag{A.37}$$
$$\sum\nolimits_{k\in\Gamma} \lambda_{sk} = \sum\nolimits_{k\in\Gamma\backslash\{0\}} w_{sk} \qquad \forall s \in F' \tag{A.38}$$
$$\sum\nolimits_{k\in\Gamma\backslash\{0\}} w_{sk} = \sum\nolimits_{n\in N} \sum\nolimits_{b\in B} x_{snb} \qquad \forall s \in F' \tag{A.39}$$
$$\lambda_{s0} \leq w_{s1} \qquad \forall s \in F' \tag{A.40}$$
$$\lambda_{sk} \leq w_{sk} + w_{s(k+1)} \qquad \forall s \in F', k \in \Gamma\backslash\{0, \gamma\} \tag{A.41}$$
$$\lambda_{s\gamma} \leq w_{s\gamma} \qquad \forall s \in F' \tag{A.42}$$
$$\delta_s = d_s - \epsilon_s \qquad \forall s \in F' \tag{A.43}$$
$$t^a_{nb} \geq t^d_{nb} \qquad \forall n \in N, b \in B \tag{A.44}$$
$$t^a_{nb} \geq t^d_{nb} + \delta_s + (x_{snb} - 1)M \qquad \forall s \in F', n \in N, b \in B \tag{A.45}$$
$$x_{snb} + x_{s'(n+1)b} \leq 1 \qquad \forall s \in F', n \in N, b \in B \tag{A.46}$$
$$t^a_{nb} \sum\nolimits_{n\in N} \sum\nolimits_{b\in B} x_{s'nb} \geq t^a_{nb} \sum\nolimits_{n\in N} \sum\nolimits_{b\in B} x_{snb} \qquad \forall s, s' \in F', s \leq s' \tag{A.47}$$
$$\sum\nolimits_{n\in N} \sum\nolimits_{b\in B} x_{snb} \geq \sum\nolimits_{n\in N} \sum\nolimits_{b\in B} x_{s'nb} \qquad \forall s, s' \in F', s \leq s' \tag{A.48}$$
$$q^{cap}_{nb} \geq \sum\nolimits_{s\in F'} (x_{snb} - 1)M \qquad \forall n \in N, b \in B \tag{A.49}$$
$$q^{cap}_{nb} \leq \sum\nolimits_{s\in F'} (x_{snb} - 1)(-M) \qquad \forall n \in N, b \in B \tag{A.50}$$
$$x_{snb} \in \{0, 1\} \qquad \forall s \in S, n \in N, b \in B \tag{A.51}$$
$$y^u_{pnb} \in \{0, 1\} \qquad \forall p \in P, n \in N, b \in B \tag{A.52}$$
$$y^o_{pnb} \in \{0, 1\} \qquad \forall p \in P, n \in N, b \in B \tag{A.53}$$
$$q^{cap}_{nb} \in \mathbb{Z} \qquad \forall n \in N, b \in B \tag{A.54}$$
$$u_{nb} \geq 0 \qquad \forall n \in N, b \in B \tag{A.55}$$
$$z_{sk} \in \{0, 1\}, w_{sk} \in \{0, 1\} \qquad \forall s \in F', k \in \Gamma\backslash\{0\} \tag{A.56}$$
$$\alpha_{sk} \geq 0, \lambda_{sk} \geq 0 \qquad \forall s \in F', k \in \Gamma \tag{A.57}$$
$$q_s \geq 0, u_s \geq 0, o_s \geq 0, \epsilon_s \geq 0, d_s \geq 0, \delta_s \geq 0 \qquad \forall s \in F' \tag{A.58}$$