Transportation Research Part B 103 (2017) 87–110

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

The electric vehicle routing problem with nonlinear charging function

Alejandro Montoya[a,b], Christelle Guéret[a], Jorge E. Mendoza[c,d], Juan G. Villegas[e,*]

[a] Université d'Angers, LARIS (EA 7315), 62 avenue Notre Dame du Lac, 49000 Angers, France
[b] Departamento de Ingeniería de Producción, Universidad EAFIT, Carrera 49 No. 7 Sur - 50, Medellín, Colombia
[c] Université François-Rabelais de Tours, CNRS, LI (EA 6300), OC (ERL CNRS 6305), 64 avenue Jean Portalis, 37200 Tours, France
[d] Centre de Recherches Mathématiques (UMI 3457 CNRS), Montréal, Canada
[e] Departamento de Ingeniería Industrial, Facultad de Ingeniería, Universidad de Antioquia, Calle 70 No. 52-21, Medellín, Colombia

ARTICLE INFO

Article history:
Received 1 July 2016
Revised 4 February 2017
Accepted 8 February 2017
Available online 1 March 2017

Keywords:
Vehicle routing problem
Electric vehicle routing problem with nonlinear charging function
Iterated local search (ILS)
Matheuristic

ABSTRACT

Electric vehicle routing problems (E-VRPs) extend classical routing problems to consider the limited driving range of electric vehicles. In general, this limitation is overcome by introducing planned detours to battery charging stations. Most existing E-VRP models assume that the battery-charge level is a linear function of the charging time, but in reality the function is nonlinear. In this paper we extend current E-VRP models to consider nonlinear charging functions. We propose a hybrid metaheuristic that combines simple components from the literature and components specifically designed for this problem. To assess the importance of nonlinear charging functions, we present a computational study comparing our assumptions with those commonly made in the literature. Our results suggest that neglecting nonlinear charging may lead to infeasible or overly expensive solutions. Furthermore, to test our hybrid metaheuristic we propose a new 120-instance testbed. The results show that our method performs well on these instances.

1. Introduction

In the last few years several companies have started to use electric vehicles (EVs) in their operations. For example, La Poste operates at least 250 EVs and has signed orders for an additional 10,000 (Kleindorfer et al., 2012); and the French electricity distribution company ENEDIS runs 2000 EVs, accounting for 10% of their fleet in 2016.[1] Despite these developments, the large-scale adoption of EVs for service and distribution operations is still hampered by technical constraints such as battery charging times and limited battery capacity. For the most common EVs used in service operations, the minimum charging time is 0.5 h and the battery capacity is around 22 kWh. The latter leads to a nominal driving range of 142 km (Pelletier et al., 2014). In reality, the driving range could be significantly lower because the energy consumption increases with the slope of the road, the speed, and the use of peripherals (De Cauwer et al., 2015). For instance, Restrepo et al. (2014) documented that the heating and air conditioning respectively reduce the driving range of an EV by about 30% and 8% per hour of use.

* Corresponding author.
E-mail addresses: jmonto36@eafit.edu.co (A. Montoya), christelle.gueret@univ-angers.fr (C. Guéret), jorge.mendoza@univ-tours.fr (J.E. Mendoza), juan.villegas@udea.edu.co (J.G. Villegas).

[1] http://www.avere-france.org/Site/Article/?article_id=5644. Last accessed 11/16/2016.

http://dx.doi.org/10.1016/j.trb.2017.02.004
0191-2615/

Automakers and battery manufacturers are investing significant amounts of capital and effort into the development of new technology to improve EV autonomy and charging time. For instance, General Motors (GM) reinvested USD 20 million into the GM Global Battery Systems Lab to help the company developing new battery technology for their vehicles (Marcacci, 2013). The results of these efforts, however, are transferred only slowly to commercially available EVs. In the meantime, companies using EVs in their daily operations need fleet management tools that can take into account limited driving ranges and slow charging times (Felipe et al., 2014). To respond to this challenge, around 2012 the operations research community started to study a new family of vehicle routing problems (VRPs): the so-called electric VRPs (E-VRPs) (Afroditi et al., 2014; Pelletier et al., 2016). These problems consider the technical limitations of EVs. Because of the short driving range, E-VRP solutions frequently include routes with planned detours to charging stations (CSs). The need to detour usually arises in rural and semi-urban operations, where the distance covered by the routes on a single day is often higher than the driving range.

As has been the case for other optimization problems inspired by practical applications, research in E-VRPs started with primarily theoretical variants and is slowly moving toward problems that better capture reality. In general, E-VRP models make assumptions about the EV energy consumption, the charging infrastructure ownership, the capacity of the CSs, and the battery charging process. Most E-VRPs assume that energy consumption is directly and exclusively related to the traveled distance. However, as mentioned before, the consumption depends on a number of additional factors. To the best of our knowledge only Goeke and Schneider (2015) and Lin et al. (2016) use consumptions computed over actual road networks taking into account the EV parameters and their loads.

Similarly, most E-VRP models implicitly assume that the charging infrastructure is private. In this context, the decision-maker controls access to the CSs, so they are always available. However, in reality, mid-route charging is often performed at public stations and so the availability is uncertain. To our knowledge only Sweda et al. (2015) and Kullman et al. (2016) deal with public infrastructure and consider uncertainty in CS availability.

CS capacity is another area in which current E-VRP models are still a step behind reality. All existing E-VRP research that we are aware of assumes that the CSs can simultaneously handle an unlimited number of EVs. In practice, each CS is usually equipped with only a few chargers. In some settings this assumption may be mild (e.g., a few geographically distant routes and private CSs). However, in most practical applications CS capacity plays a restrictive role.

Finally, in terms of the battery charging process, E-VRP models make assumptions about the charging policy and the charging function approximation. The former defines how much of the battery capacity can (or must) be restored when an EV visits a CS, and the latter models the relationship between battery charging time and battery level. In this paper, we focus on these assumptions.

In terms of the charging policies, the E-VRP literature can be classified into two groups: studies assuming *full* and *partial* charging policies. As the name suggests, in full charging policies, the battery capacity is fully restored every time an EV reaches a CS. Some studies in this group assume that the charging time is constant (Conrad and Figliozzi, 2011; Erdoğan and Miller-Hooks, 2012; Adler and Mirchandani, 2014; Montoya et al., 2015; Hof et al., 2017). This is a plausible assumption in applications where the CSs replace a (partially) depleted battery with a fully charged one. Other researchers, including Schneider et al. (2014), Goeke and Schneider (2015), Schneider et al. (2015), Desaulniers et al. (2016), Hiermann et al. (2016), Lin et al. (2016), and Szeto and Cheng (2016), consider full charging policies with a linear charging function approximation (i.e., the battery level is assumed to be a linear function of the charging time). In their models, the time spent at each CS depends on the battery level when the EV arrives and on the (constant) charging rate of the CS. In partial charging policies, the level of charge (and thus the time spent at each CS) is a decision variable. To the best of our knowledge, all existing E-VRP models with partial charging consider linear function approximations (Felipe et al., 2014; Sassi et al., 2015; Bruglieri et al., 2015; Schiffer and Walther, 2017; Desaulniers et al., 2016; Keskin and Cătay, 2016).

In general, the charging functions are nonlinear, because the terminal voltage and current change during the charging process. This process is divided into two phases. In the first phase, the charging current is held constant, and thus the battery level increases linearly with time. The first charging phase continues until the battery's terminal voltage increases to a specific maximum value (see Fig. 1). In the second phase, the current decreases exponentially and the terminal voltage is held constant to avoid battery damage. The battery level then increases concavely with time (Pelletier et al., 2017).

Although the shape of the charging functions is known, devising analytical expressions to model them is complex because they depend on factors such as current, voltage, self-recovery, and temperature (Wang et al., 2013). The battery level is then described by differential equations. Since such equations are difficult to incorporate into E-VRP models, researchers rely on approximations of the actual charging functions. Bruglieri et al. (2014) use a linear approximation that considers only the linear segment of the charging function, i.e., between 0 and (around) $0.8Q$, where Q represents the battery capacity. This approximation avoids dealing with the nonlinear segment of the charging function (i.e., from (around) $0.8Q$ to Q). Henceforth we refer to this approximation as first segment (FS). Felipe et al. (2014); Sassi et al. (2014); Bruglieri et al. (2015); Desaulniers et al. (2016); Schiffer and Walther (2017), and Keskin and Cătay (2016) approximate the whole charging function using a linear expression. They do not explain how the approximation is calculated, but two options can be considered. In the first (L1) the charging rate of the function corresponds to the slope of its linear segment (see Fig. 2b). This approximation is optimistic: it assumes that batteries charge to the level Q faster than they do in reality. In the second approximation (L2) the charging rate is the slope of the line connecting the first and last observations (see Fig. 2c) of the charging curve. This approximation tends to be pessimistic: over a large portion of the curve, the charging rate is slower than in reality.

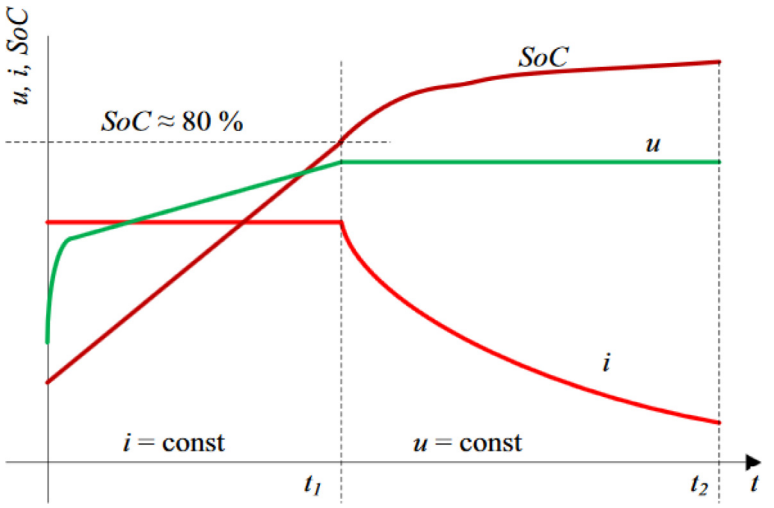


Fig. 1. Typical charging curve, where *i*, *u*, and SoC represent the current, terminal voltage, and state of charge respectively. The SoC is equivalent to the battery level. (Source: Hõimoja et al., 2012).

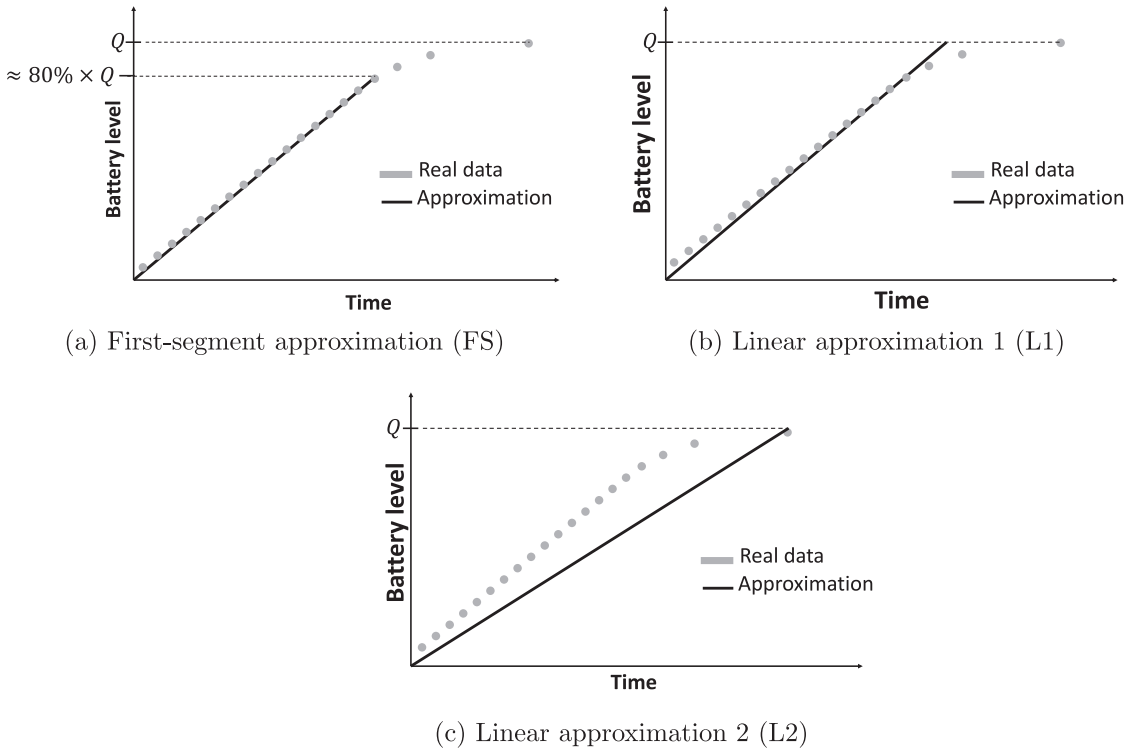


Fig. 2. Linear approximations in the literature vs. real data provided by Uhrig et al. (2015). (Uhrig et al., 2015 conducted experiments to estimate the charging time for different charge levels with two types of EVs and three types of CSs.).

In this paper, we study a new E-VRP that captures the nonlinear behavior of the charging process using a piecewise linear approximation. The main contributions of this research are fivefold. First, we introduce the electric vehicle routing problem with nonlinear charging functions (E-VRP-NL). Second, we propose a hybrid metaheuristic, which combines simple components from the literature and components specifically designed for this new problem. Third, we propose a set of realistic and publicly available instances. Fourth, we demonstrate through extensive computational experiments the importance of better approximating the actual battery charging function. Fifth, we analyze our solutions and provide some insight into the characteristics of good E-VRP-NL solutions.

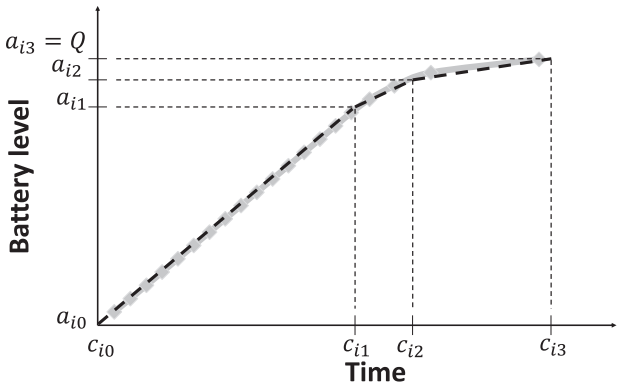


Fig. 3. Real data vs. piecewise linear approximation for a CS of 22 kW charging a battery of 16 kWh.

The remainder of this paper is organized as follows. Section 2 formally introduces the E-VRP-NL. Section 3 describes our hybrid metaheuristic, and Section 4 presents the computational experiments. Finally, Section 5 concludes the paper and discusses future research.

2. Electric vehicle routing problem with nonlinear charging function

2.1. Problem description

Let I be the set of nodes representing the customers, F the set of CSs, and 0 a node representing the depot. Each customer $i \in I$ has a service time p_i. The E-VRP-NL is defined on a directed and complete graph $G = (V, A)$, where $V = \{0\} \cup I \cup F'$ and F' contains the set F and β copies of each CS (i.e., $|F'| = |F| \times (1 + \beta)$). The value of $1 + \beta$ corresponds to the number of times that each CS can be visited. Let $A = \{(i, j) : i, j \in V, i \neq j\}$ be the set of arcs connecting vertices of V. Each arc (i, j) has two associated nonnegative values: a travel time t_{ij} and an energy consumption e_{ij}. The customers are served using an unlimited and homogeneous fleet of EVs. All the EVs have a battery of capacity Q (expressed in kWh) and a maximum tour duration $T_{\max}$. It is assumed that the EVs leave the depot with a fully charged battery, and that all the CSs can handle an unlimited number of EVs simultaneously. Feasible solutions to the E-VRP-NL satisfy the following conditions: each customer is visited exactly once; each route satisfies the maximum-duration limit; each route starts and ends at the depot; and the battery level when an EV arrives at and departs from any vertex is between 0 and Q.

Since the traveled distance is directly related to the energy consumption, most work on E-VRPs with a homogeneous fleet focuses on minimizing the total distance (Schneider et al., 2014; Desaulniers et al., 2016; Hiermann et al., 2016; Keskin and Cătay, 2016). However, this objective function neglects the impact of charging operations. This may lead to solutions that charge the batteries more than needed or charge them even when their level is high. These decisions directly affect the battery's long-term degradation cost (which according to Becker et al., 2009 can be three times the energy cost) and the charging fees at CSs (Bansal, 2015). To better capture the impact of charging operations, in the E-VRP-NL we minimize the total travel and charging time. This objective function was studied by Zündorf (2014) and Liao et al. (2016) for related routing problems.

2.2. Modeling of battery charging functions

Each CS $i \in F'$ has a charging mode (e.g., slow, moderate, fast) that is associated with a charging function $g_i(q_i, \Delta_i)$. This function maps the charge level when the vehicle arrives at i (q_i) and the time spent charging at i (Δ_i) to the charge level when the vehicle leaves i. To avoid handling a two-dimensional function, we use the transformation proposed by Zündorf (2014). Let $\hat{g}_i(l)$ be the charging function when $q_i = 0$ and the battery is charged for l time units; $g_i(q_i, \Delta_i)$ is estimated as $\hat{g}_i(\Delta_i + \hat{g}_i^{-1}(q_i))$. Note that $\Delta_i = \hat{g}_i^{-1}(o_i) - \hat{g}_i^{-1}(q_i)$, where o_i is the charge level when the vehicle leaves i.

The function $\hat{g}_i(l)$ is concave (Bruglieri et al., 2014; Hõimoja et al., 2012; Pelletier et al., 2017) with an asymptote at Q. Similarly to Zündorf (2014), we argue that $\hat{g}_i(l)$ can be accurately approximated using piecewise linear functions. We support our claim using the data provided by Uhrig et al. (2015). We fit piecewise linear functions to their data and obtain approximations with an average relative absolute error of 0.90%, 1.24%, and 1.90% for CSs of 11, 22, and 44 kW, respectively. Fig. 3 shows the piecewise linear approximation for a CS i of 22 kW charging a vehicle equipped with a battery of 16 kWh. In the plot, c_{ik} and a_{ik} represent the charging time and the charge level for the breakpoint $k \in B$ of the CS $i \in F'$, where $B = \{0, .., b\}$ is the set of breakpoints of the piecewise linear approximation.

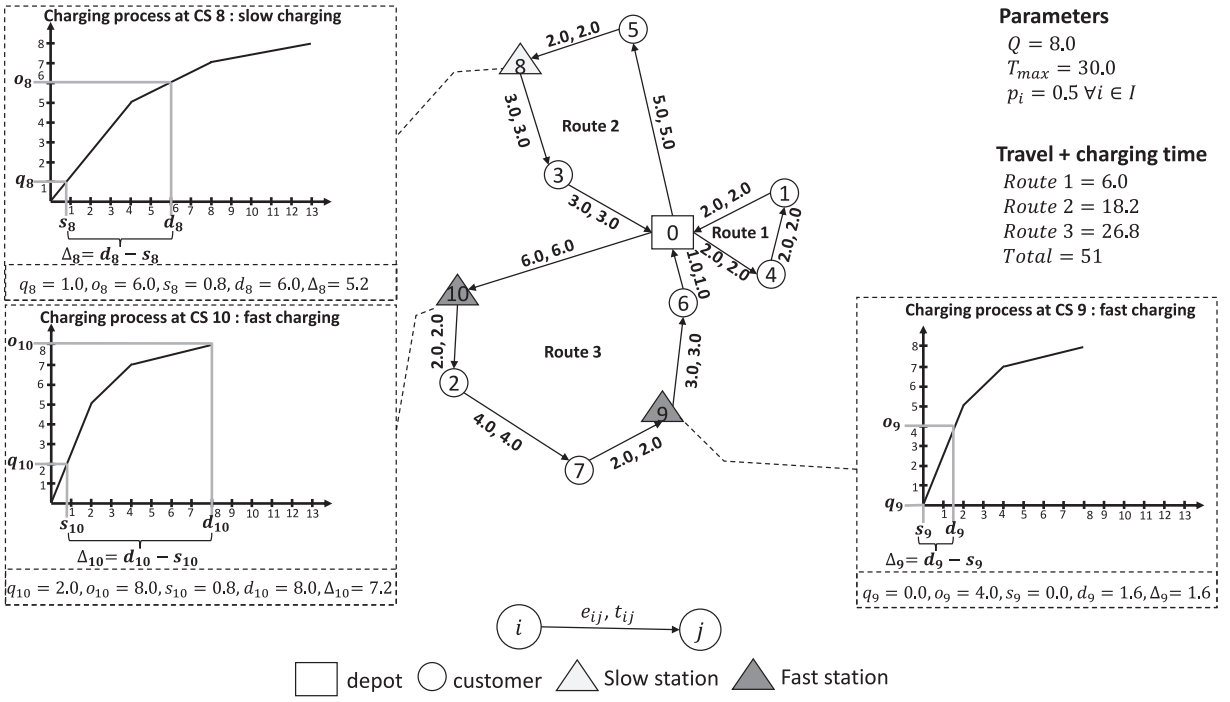


Fig. 4. Example of a feasible E-VRP-NL solution.

2.3. Illustrative example

Fig. 4 presents a numerical example illustrating the E-VRP-NL. The figure depicts a solution to an instance with 7 customers and 3 CSs. The CSs have different technologies (slow and fast), and each technology has a specific piecewise-linear charging function. The charging function maps the battery levels q_i and o_i to the charging times s_i and d_i to estimate the time spent at the CS $i \in F$ (Δ_i). In this example, Route 1 does not visit any CS, because its total energy consumption is less than the battery capacity. Route 2 visits CS 8: the EV arrives at the CS with a battery level $q_8 = 1.0$, and it charges the battery to a level $o_8 = 6.0$. To estimate the time spent at the CS, we use the piecewise-linear charging function: the charging times associated with q_8 and o_8 are $s_i = 0.8$ and $d_i = 6.0$, so the time spent at CS 8 is $\Delta_8 = 6.0 - 0.8 = 5.2$. The duration of Route 2 is the sum of the travel time (13.0), the charging time (5.2), and the service time (1.0), i.e., 19.2, which is less than T_{max}. The cost of this route is 18.2 (travel time + charging time). Finally, Route 3 visits CSs 10 and 9, and it spends $\Delta_{10} = 7.2$ and $\Delta_9 = 1.6$ time units charging in these CSs, respectively.

2.4. Mixed-integer linear programming formulation

To help the reader understand the E-VRP-NL, we now provide a mixed integer linear programming (MILP) formulation of the problem. The MILP uses the following decision variables: variable x_{ij} is equal to 1 if an EV travels from vertex i to j, and 0 otherwise. Variables τ_j and y_j track the time and charge level when the EV departs from vertex $j \in V$. Variables q_i and o_i specify the charge levels when an EV arrives at and departs from CS $i \in F'$, and s_i and d_i are the associated charging times. Variable $\Delta_i = d_i - s_i$ represents the time spent at CS $i \in F'$. Variables z_{ik} and w_{ik} are equal to 1 if the charge level is between $a_{i,k-1}$ and a_{ik}, with $k \in B \setminus \{0\}$, when the EV arrives at and departs from CS $i \in F'$ respectively. Finally, variables α_{ik} and λ_{ik} are the coefficients of the breakpoint $k \in B$ in the piecewise linear approximation, when the EV arrives at and departs from CS $i \in F'$ respectively. The MILP formulation follows:

$$\min \sum_{i,j \in V} t_{ij} x_{ij} + \sum_{i \in F'} \Delta_i \tag{1}$$

subject to

$$\sum_{j \in V, i \neq j} x_{ij} = 1, \qquad \forall i \in I \tag{2}$$

$$\sum_{j \in V, i \neq j} x_{ij} \leq 1, \qquad \forall i \in F' \tag{3}$$

$$\sum_{j\in V, i\neq j} x_{ji} - \sum_{j\in V, i\neq j} x_{ij} = 0, \qquad \forall i \in V \tag{4}$$

$$e_{ij}x_{ij} - (1-x_{ij})Q \leq y_i - y_j \leq e_{ij}x_{ij} + (1-x_{ij})Q, \qquad \forall i \in V, \forall j \in I, i \neq j \tag{5}$$

$$e_{ij}x_{ij} - (1-x_{ij})Q \leq y_i - q_j \leq e_{ij}x_{ij} + (1-x_{ij})Q, \qquad \forall i \in V, \forall j \in F', i \neq j \tag{6}$$

$$y_i \geq e_{i0}x_{i0}, \qquad \forall i \in V \tag{7}$$

$$y_i = o_i, \qquad \forall i \in F' \tag{8}$$

$$y_0 = Q \tag{9}$$

$$q_i \leq o_i, \qquad \forall i \in F' \tag{10}$$

$$q_i = \sum_{k\in B} \alpha_{ik}a_{ik}, \qquad \forall i \in F' \tag{11}$$

$$s_i = \sum_{k\in B} \alpha_{ik}c_{ik}, \qquad \forall i \in F' \tag{12}$$

$$\sum_{k\in B} \alpha_{ik} = \sum_{k\in B\setminus\{0\}} z_{ik}, \qquad \forall i \in F' \tag{13}$$

$$\sum_{k\in B\setminus\{0\}} z_{ik} = \sum_{j\in V} x_{ij}, \qquad \forall i \in F' \tag{14}$$

$$\alpha_{i0} \leq z_{i1}, \qquad \forall i \in F', \tag{15}$$

$$\alpha_{ik} \leq z_{ik} + z_{i,k+1}, \qquad \forall i \in F', \forall k \in B\setminus\{0, b\} \tag{16}$$

$$\alpha_{ib} \leq z_{ib}, \qquad \forall i \in F' \tag{17}$$

$$o_i = \sum_{k\in B} \lambda_{ik}a_{ik}, \qquad \forall i \in F' \tag{18}$$

$$d_i = \sum_{k\in B} \lambda_{ik}c_{ik}, \qquad \forall i \in F' \tag{19}$$

$$\sum_{k\in B} \lambda_{ik} = \sum_{k\in B\setminus\{0\}} w_{ik}, \qquad \forall i \in F' \tag{20}$$

$$\sum_{k\in B\setminus\{0\}} w_{ik} = \sum_{j\in V} x_{ij}, \qquad \forall i \in F' \tag{21}$$

$$\lambda_{i0} \leq w_{i1}, \qquad \forall i \in F' \tag{22}$$

$$\lambda_{ik} \leq w_{ik} + w_{i,k+1}, \qquad \forall i \in F', \forall k \in B\setminus\{0, b\} \tag{23}$$

$$\lambda_{ib} \leq w_{ib}, \qquad \forall i \in F' \tag{24}$$

$$\Delta_i = d_i - s_i, \qquad \forall i \in F' \tag{25}$$

$$\tau_i + (t_{ij} + p_j)x_{ij} - T_{max}(1 - x_{ij}) \leq \tau_j, \qquad \forall i \in V, \forall j \in I, i \neq j \tag{26}$$

$$\tau_i + \Delta_j + t_{ij}x_{ij} - (S_{max} + T_{max})(1 - x_{ij}) \leq \tau_j, \qquad \forall i \in V, \forall j \in F', i \neq j \tag{27}$$

$$\tau_j + t_{j0} \leq T_{max}, \qquad \forall j \in V \tag{28}$$

$$\tau_0 \leq T_{max} \tag{29}$$

$$x_{ij} = 0, \qquad \forall i, j \in F' : m_{ij} = 1 \tag{30}$$

$$\tau_i \geq \tau_j, \qquad \forall i, j \in F' : m_{ij} = 1, j \leq i \tag{31}$$

$$\tau_j \leq T_{max} \sum_{i \in V} x_{ij}, \qquad \forall j \in F' \tag{32}$$

$$\sum_{i \in V} x_{ih} \geq \sum_{j \in V} x_{jf}, \qquad \forall h, f \in F' : m_{hf} = 1, h \leq f \tag{33}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i, j \in V, i \neq j \tag{34}$$

$$\tau_i \geq 0, y_i \geq 0 \qquad \forall i \in V \tag{35}$$

$$z_{ik} \in \{0, 1\}, w_{ik} \in \{0, 1\}, \qquad \forall i \in F', \forall k \in B \setminus \{0\} \tag{36}$$

$$\alpha_{ik} \geq 0, \lambda_{ik} \geq 0, \qquad \forall i \in F', \forall k \in B \tag{37}$$

$$q_i \geq 0, o_i \geq 0, s_i \geq 0, d_i \geq 0, \Delta_i \geq 0, \qquad \forall i \in F' \tag{38}$$

The objective function (1) seeks to minimize the total time (travel times plus charging times). Constraints (2) ensure that each customer is visited once. Constraints (3) ensure that each CS copy is visited at most once. Constraints (4) impose the flow conservation. Constraints (5) and (6) track the battery charge level at each vertex. Constraints (7) ensure that if the EV travels between a vertex and the depot, it has sufficient energy to reach its destination. Constraints (8) reset the battery tracking to o_i upon departure from CS $i \in F'$. Constraint (9) ensures that the battery charge level is Q at the depot. Constraints (10) couple the charge levels when an EV arrives at and departs from any CS. Constraints (11)–(17) define the charge level (and its corresponding charging time) when an EV arrives at CS $i \in F'$ (based on the piecewise linear approximation of the charging function). Similarly, constraints (18)–(24) define the charge level (and its corresponding charging time) when an EV departs from CS $i \in F'$. Constraints (25) define the time spent at any CS. Constraints (26) and (27) track the departure time at each vertex, where $S_{\max} = max_{i \in F}\{c_{ib}\}$. Constraints (28) and (29) ensure that the EVs return to the depot no later than $T_{\max}$. Constraints (30)–(33) help to avoid the symmetry generated by the copies of the CSs. Parameter m_{ij} is equal to 1 if i and $j \in F'$ represent the same CS. Finally, constraints (34)–(38) define the domain of the decision variables.

3. Solving the E-VRP-NL

Lenstra and Rinnooy Kan (1981) demonstrated that the classical VRP, commonly known as the CVRP, is NP-hard. Since the CVRP is a special case of our E-VRP-NL, the latter is also NP-hard. We therefore propose a metaheuristic approach. Like many metaheuristics for other VRPs, our approach explores new solutions by building new routes or applying changes (moves) to existing ones. In this process, the algorithm makes sequencing and charging decisions. The former fix the order in which the route visits its assigned customers, while the latter determine where and how much to charge the EV serving the route. Sequencing and charging decisions can be made either simultaneously or in two phases (sequencing first, charging second). We use the latter option. To make charging decisions, we solve what we call the *fixed route vehicle charging problem*, or simply FRVCP. In a nutshell, the problem consists in (i) inserting charging stations into a fixed sequence of customers and (ii) deciding how much to charge at each inserted station. Since the FRVCP plays a key role in our approach, we discuss it in the next subsection before introducing our metaheuristic.

3.1. The fixed-route vehicle-charging problem

The FRVCP is a variant of the well-known fixed-route vehicle-refueling problem (FRVRP). The FRVRP seeks the minimum-cost refueling policy (which fuel stations to visit and the refueling quantity at each station) for a given origin-destination route (Suzuki, 2014). Most of the research into the FRVRP and its variants applies only to internal combustion vehicles (which have negligible refueling times), but a few extensions to EVs have been reported. Most of these extensions assume full charging policies (Montoya et al., 2015; Hiermann et al., 2016; Liao et al., 2016). To our knowledge, only Sweda et al. (2014) assume a partial charging policy. Their problem differs from ours in three fundamental ways: (i) they do not take into account the charging times (because their objective is to minimize the energy and degradation costs), (ii) they do not deal with maximum route duration constraints, and (iii) their CSs are already included in the fixed route and no detours are to be planned.

In the FRVCP the objective is to find the charging decisions (where and how much to charge) that minimize the sum of the charging times and detour times while satisfying the following conditions: the level of the battery when the EV arrives at any vertex is nonnegative; the charge in the battery does not exceed its capacity; and the route satisfies the maximum-duration limit. Since the FRVRP is NP-hard (Suzuki, 2014) and the FRVCP generalizes the FRVRP, we can conclude that the FRVCP is also NP-hard.

Let $\Pi = \{\pi(0), \pi(1), \ldots, \pi(i), \ldots, \pi(j), \ldots \pi(n_r - 1), \pi(n_r)\}$ be the fixed route, where $\pi(0)$ and $\pi(n_r)$ represent the depot and $\pi(1), \ldots, \pi(i), \ldots, \pi(j), \ldots, \pi(n_r - 1)$ the customers. The fixed route has a total time $\bar{t}$, which is the sum of the travel times plus the service times. Note that by definition (i) Π does not visit CSs, and (ii) it is *energy infeasible* (i.e., requires more than one full battery to complete).[2] The feasibility of Π may be restored by inserting visits to CSs. As mentioned in Section 3, each CS $j \in F$ has a piecewise-linear charging function defined by a set of breakpoints B. Each segment of the piecewise linear function is defined between breakpoints $k-1$ and $k \in B \setminus \{0\}$, has a slope ρ_{jk} (representing a charging rate), and is bounded by the battery levels a_{jk-1} and a_{jk}. The values $e_{\pi(i-1)\pi(i)}$ and $t_{\pi(i-1)\pi(i)}$ represent the energy consumption and the travel time between vertices $\pi(i-1)$ and $\pi(i) \in \Pi$. Similarly, $e_{\pi(i-1)j}$ and $t_{\pi(i-1)j}$ represent the energy consumption and the travel time between vertex $\pi(i-1) \in \Pi$ and CS $j \in F$, and $e_{j\pi(i)}$ and $t_{j\pi(i)}$ represent the energy consumption and the travel time between CS $j \in F$ and vertex $\pi(i) \in \Pi$.

Fig. 5 presents an example illustrating the FRVCP using the fixed route corresponding to Route 3 in the example introduced in Section 2. Fig. 5a shows the fixed route with 3 customers. Fig. 5b shows all the possible CS insertions into the fixed route (the arcs in bold correspond to the FRVCP solution). Fig. 5c shows the energy-feasible route resulting from solving the FRVCP.

We formulate the FRVCP as an MILP using the following decision variables: variable $\varepsilon_{\pi(i)j}$ is equal to 1 if the EV charges at CS $j \in F$ before visiting vertex $\pi(i) \in \Pi$. Variable $\phi_{\pi(i)}$ tracks the battery level. If $\varepsilon_{\pi(i)j} = 0$, $\phi_{\pi(i)}$ is the battery level when the EV arrives at vertex $\pi(i)$. If $\varepsilon_{\pi(i)j} = 1$, $\phi_{\pi(i)}$ is the battery level when the EV arrives at CS $j \in F$ immediately before visiting vertex $\pi(i)$. Variable $\theta_{\pi(i)jk}$ is equal to 1 if the EV charges on the segment defined by breakpoints $k-1$ and $k \in B \setminus \{0\}$ at CS $j \in F$ before visiting vertex $\pi(i) \in \Pi$. Finally, variables $\delta_{\pi(i)jk}$ and $\mu_{\pi(i)jk}$ are (respectively) the amount of energy charged and the battery level when the charging process finishes on the segment between breakpoints $k-1$ and $k \in B \setminus \{0\}$ at CS $j \in F$ before the visit to vertex $\pi(i) \in \Pi$. The MILP formulation of the FRVCP follows:

$$\min \sum_{\pi(i) \in \Pi \setminus \{\pi(0)\}} \sum_{j \in F} \sum_{k \in B \setminus \{0\}} \frac{\delta_{\pi(i)jk}}{\rho_{jk}} + \sum_{\pi(i) \in \Pi \setminus \{\pi(0)\}} \sum_{j \in F} \varepsilon_{\pi(i)j} (t_{\pi(i-1)j} + t_{j\pi(i)} - t_{\pi(i-1)\pi(i)}) \tag{39}$$

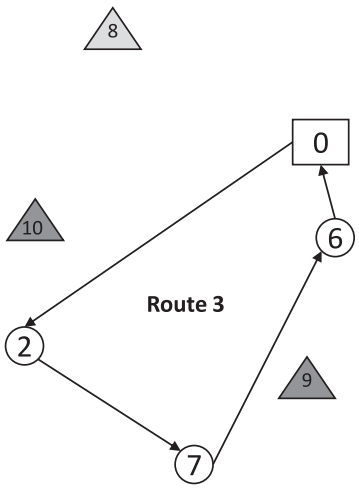
subject to

$$\phi_{\pi(1)} = Q - \sum_{j \in F} \varepsilon_{\pi(1)j} e_{\pi(0)j} - e_{\pi(0)\pi(1)} \left(1 - \sum_{j \in F} \varepsilon_{\pi(1)j}\right) \tag{40}$$

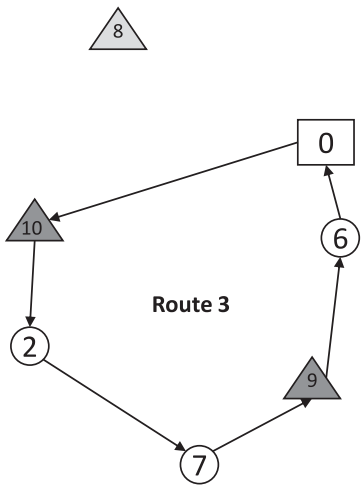
$$\begin{aligned} \phi_{\pi(i)} = \phi_{\pi(i-1)} &+ \sum_{j \in F} \sum_{k \in B \setminus \{0\}} \delta_{\pi(i-1)jk} - \sum_{j \in F} \varepsilon_{\pi(i-1)j} e_{j\pi(i-1)} \\ &- \sum_{j \in F} \varepsilon_{\pi(i)j} e_{\pi(i-1)j} - e_{\pi(i-1)\pi(i)} \left(1 - \sum_{j \in F} \varepsilon_{\pi(i)j}\right) \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0), \pi(1), \pi(n_r)\} \end{aligned} \tag{41}$$

$$\begin{aligned} \phi_{\pi(n_r)} = \phi_{\pi(n_r-1)} &+ \sum_{j \in F} \sum_{k \in B \setminus \{0\}} \delta_{\pi(n_r-1)jk} + \sum_{j \in F} \sum_{k \in B \setminus \{0\}} \delta_{\pi(n_r)jk} - \sum_{j \in F} \varepsilon_{\pi(n_r-1)j} e_{j\pi(n_r-1)} \\ &- \sum_{j \in F} \varepsilon_{\pi(n_r)j} (e_{\pi(n_r-1)j} + e_{j\pi(n_r)}) - e_{\pi(n_r-1)\pi(n_r)} \left(1 - \sum_{j \in F} \varepsilon_{\pi(n_r)j}\right) \end{aligned} \tag{42}$$

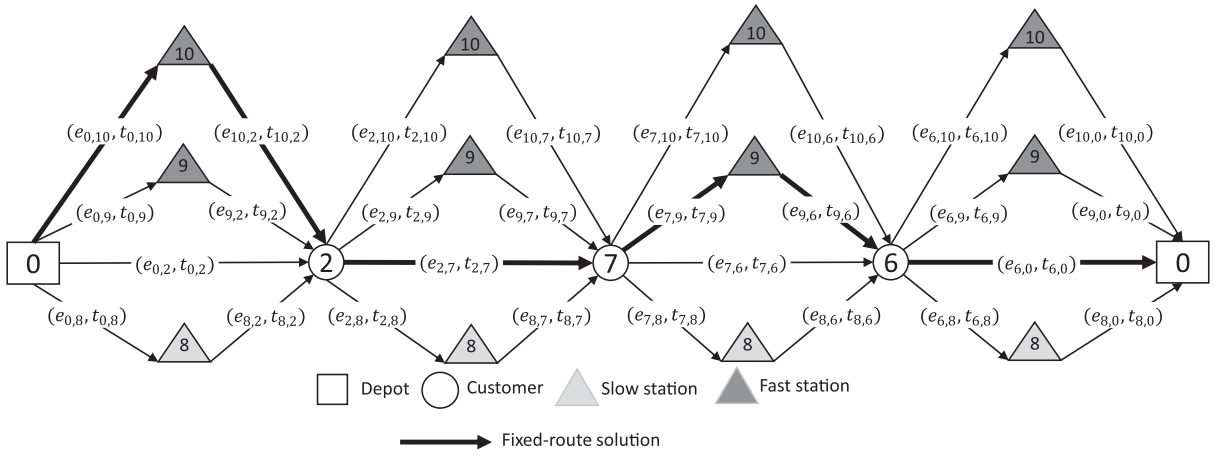
[2] The optimal solution to an FRVCP solved over an energy-feasible fixed route is trivial: the original fixed route itself.



(a) Fixed route



(c) Energy-feasible route with visits to CSs



(b) Fixed-route and possible charging detours

Fig. 5. Example of a fixed-route vehicle-charging problem.

$$\phi_{\pi(n_r-1)} + \sum_{j\in F}\sum_{k\in B\setminus\{0\}} \delta_{\pi(n_r-1)jk} - \sum_{j\in F} e_{j\pi(n_r-1)}\varepsilon_{\pi(n_r-1)j} - \sum_{j\in F} e_{\pi(n_r-1)j}\varepsilon_{\pi(n_r)j}\Big) \geq 0 \tag{43}$$

$$\mu_{\pi(i)j1} = \phi_{\pi(i)} + \delta_{\pi(i)j1} \qquad \forall \pi(i) \in \Pi\setminus\{\pi(0)\}, \forall j \in F \tag{44}$$

$$\mu_{\pi(n_r)j1} = \phi_{\pi(n_r-1)} + \sum_{l\in F}\sum_{k\in B\setminus\{0\}} \delta_{\pi(n_r-1)lk} - \sum_{l\in F} e_{l\pi(n_r-1)}\varepsilon_{\pi(n_r-1)l} - \varepsilon_{\pi(n_r)j}e_{\pi(n_r-1)j} + \delta_{\pi(n_r)j1} \qquad \forall j \in F \tag{45}$$

$$\mu_{\pi(i)jk} = \mu_{\pi(i)j,k-1} + \delta_{\pi(i)jk} \qquad \forall \pi(i) \in \Pi\setminus\{\pi(0)\}, \forall j \in F, \forall k \in B\setminus\{0,1\} \tag{46}$$

$$\mu_{\pi(i)jk} \geq a_{jk-1}\theta_{\pi(i)jk} \qquad \forall \pi(i) \in \Pi\setminus\{\pi(0)\}, \forall j \in F, \forall k \in B\setminus\{0,1\} \tag{47}$$

$$\mu_{\pi(i)jk} \leq a_{jk}\theta_{\pi(i)jk} + (1-\theta_{\pi(i)jk})Q \qquad \forall \pi(i) \in \Pi\setminus\{\pi(0)\}, \forall j \in F, \forall k \in B\setminus\{0\} \tag{48}$$

$$\sum_{j\in F} \varepsilon_{\pi(i)j} \leq 1, \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\} \tag{49}$$

$$\theta_{\pi(i)jk} \leq \varepsilon_{\pi(i)j} \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \tag{50}$$

$$\delta_{\pi(i)jk} \leq \theta_{\pi(i)jk} Q \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \tag{51}$$

$$\bar{t} + \sum_{\pi(i)\in\Pi\setminus\{\pi(0)\}} \sum_{j\in F} \sum_{k\in B\setminus\{0\}} \frac{\delta_{\pi(i)jk}}{\rho_{jk}} + \sum_{\pi(i)\in\Pi\setminus\{\pi(0)\}} \sum_{j\in F} \varepsilon_{\pi(i)j}(t_{\pi(i-1)j} + t_{j\pi(i)} - t_{\pi(i-1)\pi(i)}) \leq T_{\max} \tag{52}$$

$$\phi_{\pi(i)} \geq 0, \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\} \tag{53}$$

$$\varepsilon_{\pi(i)j} \in \{0,1\}, \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F \tag{54}$$

$$\theta_{\pi(i)jk} \in \{0,1\} \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \tag{55}$$

$$\delta_{\pi(i)jk} \geq 0 \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \tag{56}$$

$$\mu_{\pi(i)jk} \geq 0 \qquad \forall \pi(i) \in \Pi \setminus \{\pi(0)\}, \forall j \in F, \forall k \in B \setminus \{0\} \tag{57}$$

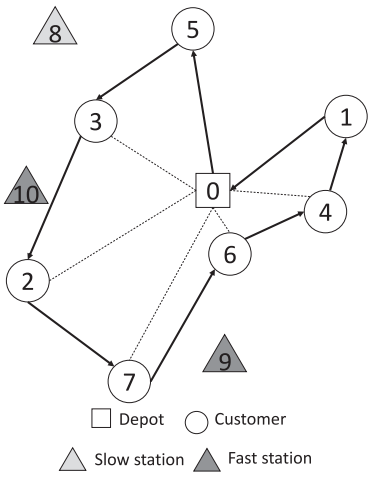
The objective function (39) seeks to minimize the total route time (including charging and detour times). Constraints (40)–(43) define the battery level when the EV arrives at vertex $\pi(i) \in \Pi$ if $\varepsilon_{\pi(i)j} = 0$; or at CS $j \in F$ before visiting vertex $\pi(i) \in \Pi$, if $\varepsilon_{\pi(i)j} = 1$. Constraints (44)–(46) define the battery level when the EV finishes charging at CS $j \in F$ in the segment between breakpoints $k-1$ and $k \in B \setminus \{0\}$ before visiting vertex $\pi(i) \in \Pi$. Constraints (47)–(48) ensure that if the EV charges on a given segment, the battery level is between the values of its corresponding break points ($a_{j,k-1}$ and a_{jk}). Constraints (49) state that only one CS is visited between any two vertices of the fixed route. Constraints (50) ensure that the EV uses only segments of the visited CSs. Likewise, constraints (51) ensure that the EV charges only at the selected segments of the visited CSs. Constraint (52) represents the duration constraint of the route. Finally, constraints (53)–(57) define the domain of the decision variables.

Our metaheuristic for the E-VRP-NL solves the FRVCP at various steps. It uses two different approaches: a commercial solver running on the MILP formulation introduced above (boosted by tailored preprocessing strategies) and a greedy heuristic. For the sake of brevity, these approaches are not discussed in the main body of the paper; full details can be found in Appendix A.

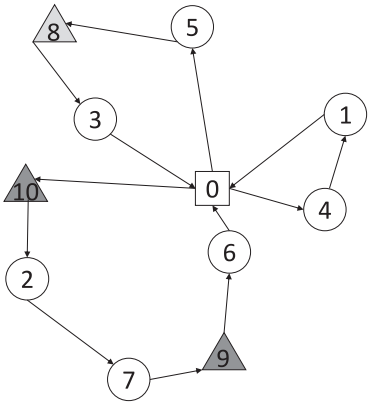
3.2. Hybrid metaheuristic

To solve the E-VRP-NL we developed a hybrid metaheuristic combining an iterated local search (ILS) and a heuristic concentration (HC). The former is a metaheuristic that starts by generating an initial solution (with a constructive heuristic). This solution is then improved by a local search procedure. At each iteration of the ILS, the best current solution is perturbed, and a new ILS iteration starts from the perturbed solution. More details of the ILS can be found in Lourenço et al. (2010). The HC is an approach that tries to build a global optimum using parts of the local optima found during a heuristic search procedure (Rosing and ReVelle, 1997).

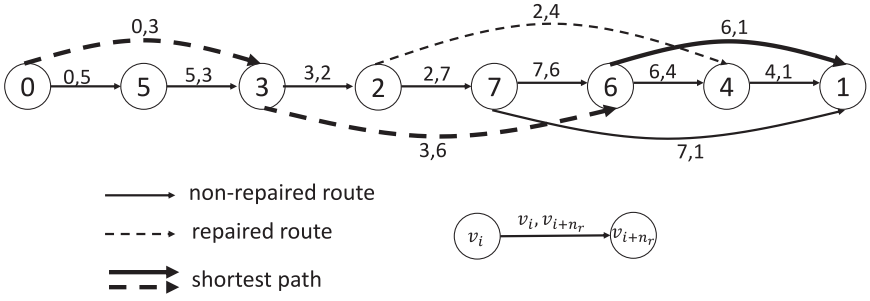
Our ILS+HC starts from an initial solution generated using a sequence-first split-second approach. The latter uses a nearest-neighbor heuristic (Rosenkrantz et al., 1977) to build a TSP tour visiting all the customers and a splitting procedure to find an E-VRP-NL solution. Then, at each iteration of the ILS we improve the current solution using a variable neighborhood descent (VND; Mladenović and Hansen, 1997) with three local search operators: relocate, 2-Opt, and global charging improvement (GCI). At the end of each ILS iteration, we update the best solution and add the routes of the local optimum to a pool of routes $\Omega \subset R$, where R is the set of all feasible routes. To diversify the search, we concatenate the routes of the local optimum to build a new TSP tour, and then perturb the new TSP tour. We start a new ILS iteration by splitting the perturbed TSP tour. After K iterations the ILS ends, and we carry out the HC. In this phase, we solve a set partitioning problem over the set of routes Ω to obtain an E-VRP-NL solution. In the remainder of this section, we describe the main components of our method.



(a) TSP tour



(c) E-VRP-NL solution



(b) Auxiliary graph

Fig. 6. Splitting a TSP tour into an E-VRP-NL solution.

3.3. Split

To extract a feasible solution from a TSP tour, we use an adaptation of the splitting procedure introduced by Prins (2004). The splitting procedure builds a directed acyclic graph $G^* = (V^*, A^*)$ composed of the ordered vertex set $V^* = (v_0, v_1, \ldots, v_i, \ldots, v_n)$ and the arc set A^*. Vertex $v_0 = 0$ is an auxiliary vertex, and each vertex v_i represents the customer in the ith position of the TSP tour. Arc $(v_i, v_{i+n_r}) \in A^*$ represents a feasible route $r_{v_i, v_{i+n_r}}$ with a travel time $t_{r_{v_i, v_{i+n_r}}}$, starting and ending at the depot and visiting customers in the sequence v_{i+1} to v_{i+n_r}.

Note that since the TSP tour includes only customers, route $r_{v_i, v_{i+n_r}}$ may be energy-infeasible; in that case, we try to repair it by solving an FRVCP. If inserting CSs into $r_{v_i, v_{i+n_r}}$ increases the duration of the route beyond $T_{\max}$, we do not include the arc associated with the route in A^*. Finally, to obtain a feasible E-VRP-NL solution, the splitting procedure finds the set of arcs (i.e., routes) along the shortest path connecting 0 and v_n in G^*.

Fig. 6 illustrates the tour splitting procedure using the example from Section 2. Fig. 6a shows the TSP tour. Fig. 6b shows the auxiliary graph G^*, where the arcs in bold correspond to the shortest path. Fig. 6c shows the solution found by the splitting procedure.

3.4. Variable neighborhood descent

To improve the solution generated by the splitting procedure we use a VND based on three local search operators. The first two operators, namely, relocate and 2-Opt,[3] focus on the sequencing decisions. In other words, these two operators alter only the sequence of customers and do not insert, remove, or change the position of CSs. To update the charging times after a relocate or 2-Opt move we use the rule proposed by Felipe et al. (2014): when visiting a CS, charge the strict minimum amount of energy needed to continue to the next CS (or the depot if there is no CS downstream). If reaching the next CS (or

[3] In our implementation we use intra-route and inter-route versions with best-improvement selection.

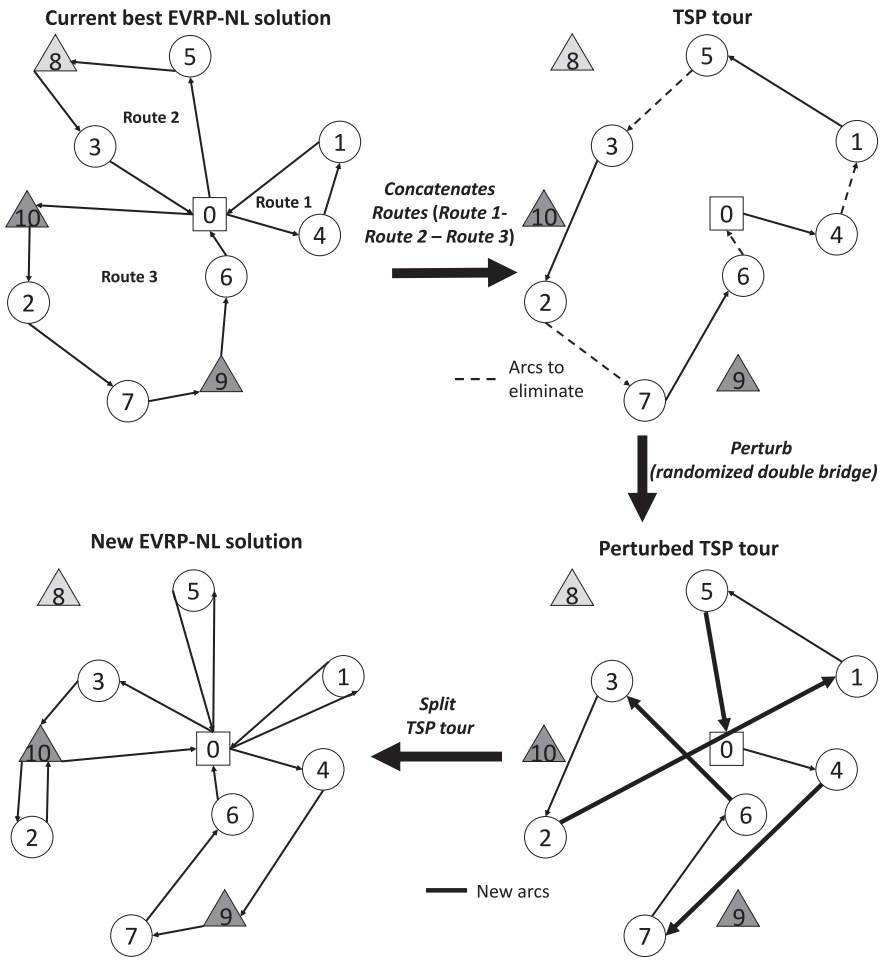


Fig. 7. Perturbation procedure example using double bridge operator.

the depot) is impossible, even with a fully charged battery, the move is deemed infeasible. Similarly, if after updating the charging times the resulting route is infeasible in terms of the maximum-duration limit, the move is discarded. It is worth noting that the Felipe et al. (2014) rule is optimal when all the CSs are homogeneous; however, this is not the case in our E-VRP-NL.

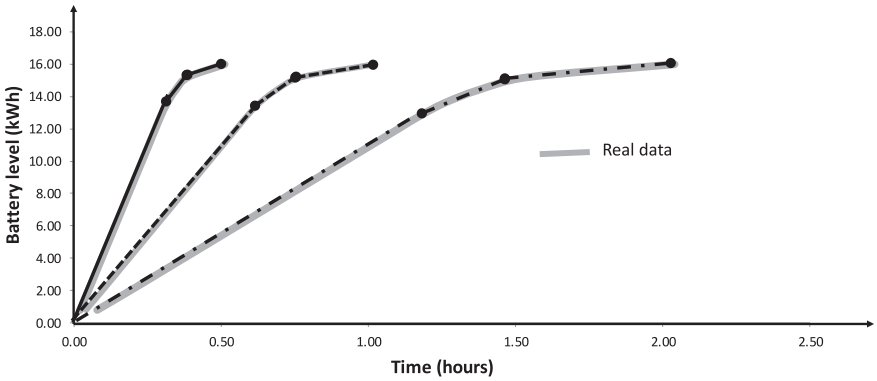
As its name suggests, the third operator, GCI, focuses on the charging decisions. GCI is applied to every route visiting at least one CS. First, it removes from the route all CS visits. If the resulting route is energy-feasible, it stops. If the route is energy-infeasible, it solves an FRVCP to optimize the charging decisions for that route.

3.5. Perturb

To diversify the search, we concatenate the routes of the current best solution in lexicographical order to build a TSP tour. Then, we perturb the resulting tour with a randomized double bridge operator (Lourenço et al., 2010) and apply the split procedure to obtain a new E-VRP-NL solution. The randomized double bridge operator cuts four arcs and introduces four new ones. Fig. 7 illustrates the steps of the perturbation procedure.

3.6. Heuristic concentration

Finally, the HC component solves a set partitioning formulation over the pool of routes Ω: $min_{\overline{R}\subseteq\Omega}\{\sum_{r\in\overline{R}} t_r : \cup_{r\in\overline{R}} = V; r_i \cap r_j = 0\ \forall r_i, r_j \in \overline{R}\}$. The objective is to select the best subset of routes from Ω to build the set of routes $\overline{R}$ (i.e., the final solution) guaranteeing that each customer will be visited by exactly one route.



Type of CSs	Slow	Moderate	Fast
Charging power (kWh/h)	11	22	44
Piecewise linear approximation	— · —	- - - - - -	———

Fig. 8. Piecewise linear approximation for different types of CS charging an EV with a battery of 16 kWh.

4. Computational experiments

In this section, we present three computational studies. The first study assesses the benefits of better approximating the battery charging function. The second study evaluates the performance of our ILS+HC. The third study analyzes the characteristics of the best known solutions (BKSs) found by our ILS+HC. The goal of this analysis is to provide researchers with insight that may be useful in the design of new solution methods for the E-VRP-NL.

4.1. Test instances for the E-VRP-NL

We generated a new 120-instance testbed built using real data for EV configuration and battery charging functions. To ensure feasibility, we opted to generate our instances instead of adapting existing datasets from the literature. To build the instances we first generated 30 sets of customer locations with {10, 20, 40, 80, 160, 320} customers. For each instance size, we generated 5 sets of customer locations. We located the customers in a geographic space of 120 × 120 km using either a random uniform distribution, a random clustered distribution, or a mixture of both. For each of the 30 sets of locations we chose the customer location strategy using a uniform probability distribution. Our main motivation for choosing the 120 × 120 km area was to build instances representing a semi-urban operation.

For each of the 30 sets of locations we built 4 instances varying the level of charging infrastructure availability and the strategy used to locate the CSs. We considered two levels of charging infrastructure availability: low and high. To favor feasibility, for each combination of number of customers and infrastructure availability level we handpicked the number of CSs as a proportion of the number of customers. We located the CSs either randomly or using a simple p-median heuristic. Our p-median heuristic starts from a set of randomly generated CS locations and iteratively moves those locations trying to minimize the total distance between the CSs and the customers. We included three types of CSs: slow, moderate, and fast. For each CS we randomly selected the type using a uniform probability distribution.

The EVs in our instances are Peugeot iOns. This EV has a consumption rate of 0.125 kWh/km, and a battery of 16 kWh. As mentioned in Section 1, the energy consumption on an arc (e_{ij}) depends on various factors. For simplicity we followed the classical approach in the literature and assumed that this consumption is simply the EV's consumption rate multiplied by the arc's length. To generate the charging functions we fit piecewise linear functions to the real charging data provided by Uhrig et al. (2015). Fig. 8 depicts our piecewise linear approximations. Finally, we set the maximum route duration for every instance to 10 h. Our 120 instances are publicly available at www.vrp-rep.org (Mendoza et al., 2014). The dataset reference is VRP-REP:2016-0020.

4.2. Benefits of better approximating the charging function

To assess the value of a charging function approximation that captures the nonlinear behavior of the process, we conducted an experiment comparing our approximation with those commonly used in the literature. The experiment consists in solving a subset of instances with four charging function approximations (i.e., FS, L1, L2, and our piecewise linear – hereafter called PL) and comparing the solutions in terms of objective function and feasibility. Since PL generalizes FS, L1, and L2, any method for the E-VRP-NL can be adapted to work with the other three approximations by a manipulation of the input data. To avoid the bias introduced by the solution method, we compare only optimal solutions delivered by the MILP

Table 1
Comparison of our charging function approximation (piecewise linear approximation) with charging function approximations from the literature.

Instance	PL			FS				L1							L2						
								Solution			Evaluation				Solution			Evaluation			
	of	***r***	***β***	***of***	***G*(%)**	***r***	***β***	***of***	***r***	***β***	***of***	***G*(%)**	***r***	***β***	***of***	***r***	***β***	***of***	***G*(%)**	***r***	***β***
tc0c10s2cf1	**19.75**	3	2	NFS	NFS	NFS	NFS	19.61	3	2	NFE	NFE	NFE	NFE	20.50	3	2	20.22	2.38	3	2
tc0c10s2ct1	**12.30**	2	0	12.61	2.52	3	0	12.22	2	0	12.42	0.98	2	0	12.46	2	0	**12.30**	0.00	2	0
tc0c10s3cf1	**19.75**	3	2	NFS	NFS	NFS	NFS	19.61	3	2	NFE	NFE	NFE	NFE	20.50	3	2	20.22	2.38	3	2
tc0c10s3ct1	**10.80**	2	0	**10.80**	0.00	2	0	10.79	2	0	11.03	2.13	2	0	10.97	2	0	**10.80**	0.00	2	0
tc1c10s2cf2	**9.03**	3	0	**9.03**	0.00	3	0	9.03	3	0	9.12	1.00	3	0	9.14	3	0	**9.03**	0.00	3	0
tc1c10s2cf3	**16.37**	3	2	NFS	NFS	NFS	NFS	15.99	3	1	NFE	NFE	NFE	NFE	16.89	3	2	**16.37**	0.00	3	2
tc1c10s2cf4	**16.10**	3	2	NFS	NFS	NFS	NFS	15.66	3	2	NFE	NFE	NFE	NFE	16.43	3	2	16.23	0.81	3	2
tc1c10s2ct2	**10.75**	3	1	**10.75**	0.00	3	1	10.75	3	0	10.76	0.09	3	0	10.94	3	0	10.78	0.28	3	0
tc1c10s2ct3	**13.17**	2	2	15.98	21.34	3	2	13.06	2	2	NFE	NFE	NFE	NFE	13.60	2	2	**13.17**	0.00	2	2
tc1c10s2ct4	**13.83**	2	1	NFS	NFS	NFS	NFS	13.34	2	1	NFE	NFE	NFE	NFE	14.17	2	1	14.17	2.46	2	1
tc1c10s3cf2	**9.03**	3	0	**9.03**	0.00	3	0	9.03	3	0	9.12	1.00	3	0	9.14	3	0	**9.03**	0.00	3	0
tc1c10s3cf3	**16.37**	3	1	NFS	NFS	NFS	NFS	15.99	3	1	NFE	NFE	NFE	NFE	16.89	3	2	**16.37**	0.00	3	2
tc1c10s3cf4	**14.90**	3	1	NFS	NFS	NFS	NFS	14.56	2	1	NFE	NFE	NFE	NFE	15.18	3	0	15.18	1.88	3	0
tc1c10s3ct2	**9.20**	3	0	**9.20**	0.00	3	0	9.19	3	0	NFE	NFE	NFE	NFE	10.80	3	0	10.57	14.89	3	0
tc1c10s3ct3	**13.02**	2	0	13.07	0.38	2	1	12.98	2	0	13.16	1.08	2	0	13.60	2	0	**13.02**	0.00	2	0
tc1c10s3ct4	**13.21**	2	0	13.58	2.80	3	1	12.92	2	1	NFE	NFE	NFE	NFE	13.71	2	0	**13.21**	0.00	2	0
tc2c10s2cf0	**21.77**	3	3	NFS	NFS	NFS	NFS	14.53	2	2	NFE	NFE	NFE	NFE	22.78	4	4	22.15	1.75	4	4
tc2c10s2ct0	**12.45**	3	2	**12.45**	0.00	3	2	12.44	3	3	NFE	NFE	NFE	NFE	12.93	3	2	**12.45**	0.00	3	2
tc2c10s3cf0	**21.77**	3	2	NFS	NFS	NFS	NFS	14.53	2	2	NFE	NFE	NFE	NFE	23.02	4	3	22.20	1.98	4	3
tc2c10s3ct0	**11.51**	3	0	**11.51**	0.00	3	0	11.50	3	0	NFE	NFE	NFE	NFE	11.92	3	0	11.54	0.26	3	0
Avg. Difference (%)					2.70							1.04							1.45		
Max. Difference (%)					21.34							2.13							14.89		
Solutions with larger fleet						3							0							2	
Infeasible solutions				9							14							0			

NFS: Infeasible solution, NFE: Infeasible evaluation
$G(\%) = (of - of_{PL})/of_{PL} \times 100$

introduced in Section 2 (running on Gurobi 5.6). This choice restricted the size of the instances used in the experiment to 10 customers and 3 CSs. We believe, however, that our conclusions hold for any instance size. All the experiments were run on a computing cluster with 2.33 GHz Intel Xeon E5410 processors with 16GB of RAM running under Linux Rocks 6.1.1.

As mentioned in Section 2, the MILP formulation uses β copies of the CSs to model multiple visits to the same CS. Although several authors have used this strategy (Conrad and Figliozzi, 2011; Erdoğan and Miller-Hooks, 2012; Schneider et al., 2014; Sassi et al., 2014; Goeke and Schneider, 2015; Hiermann et al., 2016), they do not explain how the value of β is set. It is worth noting that β plays an important role in the definition of the solution space, and therefore it restricts the optimal solution of the model. For instance, an optimal solution found with $\beta = 3$ may not be optimal for $\beta = 4$. In practice, there is no restriction on the number of times that a CS can be visited, but large values of β result in models that are computationally intractable. To overcome this difficulty, we designed an iterative procedure to solve the MILP formulation for increasing values of β. Starting with $\beta = 0$, at each iteration our procedure (i) tries to solve the MILP formulation to optimality with a time limit of 100 h, and (ii) sets $\beta = \beta + 1$. The procedure stops when the time limit is reached or an iteration ends with a solution s_β satisfying $f(s_\beta) = f(s^*_{\beta-1})$, where $f(\cdot)$ denotes the objective function and * an optimal solution.

Table 1 presents the results. Since the PL approximation is the closest to reality, the results obtained using the other approximations are compared with reference to the results of the PL. For each charging function approximation, we give the objective function value (*of*), the percentage gap between *of* and the PL solution (G), the number of routes in the solution (r), and the value of β. Since in practice the charging time is controlled by the nonlinear charging function, we evaluate the charging decisions of the L1 and L2 solutions a posteriori using the PL approximation. The last rows of Table 1 summarize the results. We present, for each approximation, the average and maximum percentage gap, the number of solutions employing more EVs than in the PL solution, and the number of infeasible solutions.

In the FS approximation EVs can charge their batteries to only around 80% of the actual capacity. Artificially constraining the capacity may force the EVs to detour to CSs more often than necessary when traveling to distant customers. Because the maximum route duration is limited, the time spent detouring and recharging the battery reduces the number of customers that can be visited. Consequently, more routes may be needed to service the same number of customers. Our results confirm this intuition: in 3 of the 20 instances the FS approximation increases the number of routes. Furthermore, in practice some distant customers may not be included in routes unless the EVs can fully use their battery capacity. In our experiments, 9 instances become infeasible with the FS approximation. In conclusion, although the FS approximation simplifies the problem (avoiding the nonlinear segment of the charging function) it may lead to solutions that are infeasible or have larger fleets and are (on average) 2.70% more expensive.

Table 2
Values of the parameters evaluated in the experiment.

Parameter	Values
K	{40, 60, 80, 100, 120}
Split(·)	{S, H}
GCI (·)	{S, H}

S stands for commercial solver (see Section A.1) H stands for greedy heuristic (see Section A.2)

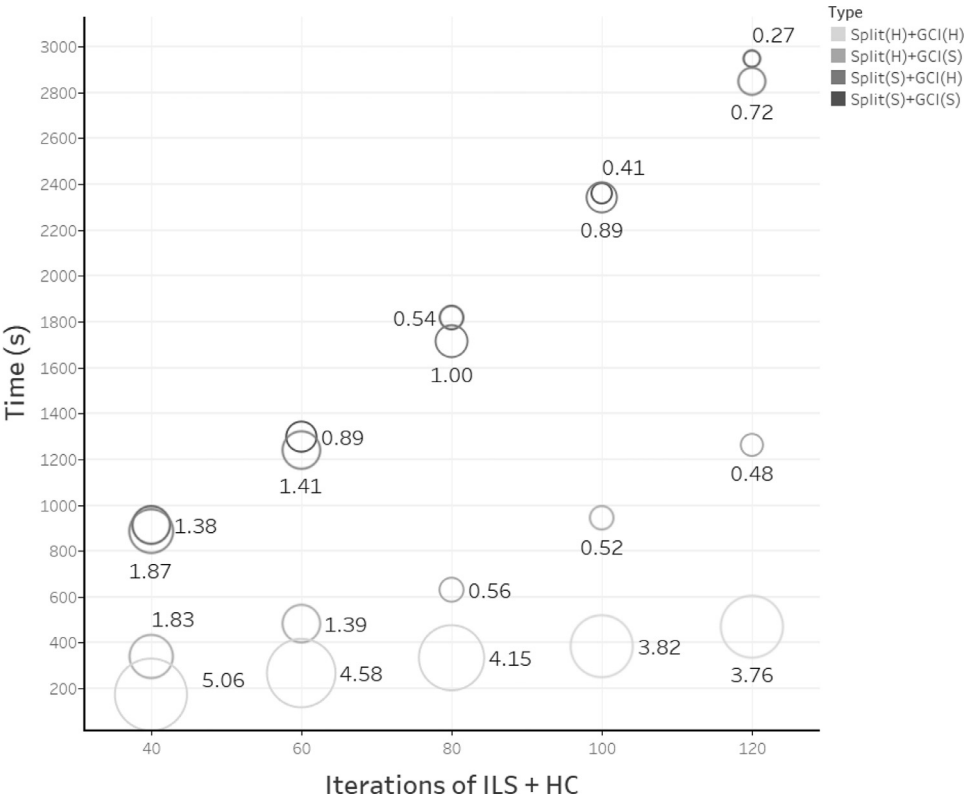


Fig. 9. Results of the parameter tuning: testing 20 possible configurations.

As mentioned before, L1 assumes that batteries charge faster than they do in reality (Fig. 2b). As a consequence, routes based on L1 may in practice need more time to reach the planned charge levels. The extra time may make a route infeasible if there is little slack in the duration constraint. Indeed, the post-hoc evaluation shows that for 14 instances, the L1 solutions are infeasible in practice. On the other hand, L2 assumes that batteries charge slower than in reality (Fig. 2c). Overestimating the charging times does not lead to feasibility issues, but the resulting routes may be overly conservative. For instance, in our experiments L2 leads to solutions that are (on average) 1.45% more expensive, and it increases the number of routes in two instances.

4.3. Results of our ILS+HC on E-VRP-NL instances

4.3.1. Experimental environment

We implemented our ILS in Java (jre V.1.8.0) and used Gurobi Optimizer (version 5.6.0) to solve the FRVCP and the set partitioning problem in the HC component. We set a time limit of 800 s in Gurobi to control the running time of the HC phase. All the experiments were run on a computing cluster with 2.33 GHz Inter Xeon E5410 processors with 16GB of RAM running under Linux Rocks 6.1.1. The results of our ILS+HC are computed over 10 runs. Each replication of the experiments was run on a single processor.

Table 3
Summary results of our ILS+HC on instances with proven optima.

Metric	ILS + HC
Number of optimal solutions	27/27
Avg. Gap (%)	0.29
Max. Gap (%)	1.87
Avg. Best Gap (%)	0.00
Avg. Time (s)	6.31

Table 4
Summary results of our ILS+HC on instances without proven optima.

Metric	ILS + HC
Number of BKSs	18/18
Avg. Gap (%)	1.09
Max. Gap (%)	2.95
Avg. Best Gap (%)	0.00
Avg. Time (s)	17.17

4.3.2. Parameter settings

Three main parameters define the configuration of our ILS+HC: the number of iterations (K) and the methods used to solve the FRVCP in (i) the split procedure (Split(·)) and (ii) the GCI neighborhood (GCI(·)). We conducted a computational study to fine-tune these parameters. Table 2 summarizes the values tested for each parameter.

To avoid overfitting, we conducted the parameter tuning on a *training* instance set. The latter is made up of 24 additional instances of 6 different sizes (in terms of number of customers): 10, 20, 40, 80, 160, 320. For each size we generated 4 instances.

Fig. 9 presents the results of our parameter tuning. The *X* coordinate represents *K*, the *Y* coordinate represents the CPU time, the circle diameter represents the average gap[4] with respect to the BKS, and the color represents the combination of methods used to solve the FRVCP.

As expected, with higher values of *K* the algorithm achieves better results: between the best configuration with $K = 40$ and that with $K = 120$ there is a difference of more than 1.1% in the average gap. To simplify the discussion, we will focus on the results obtained with $K = 120$, but the conclusions are valid for any *K*. Not surprisingly, with {Split(S), GCI(S)} our ILS delivers the best results but also consumes the most CPU time. The opposite is true with {Split(H), GCI(H)}: the algorithm is fast, but it delivers poor solutions. This result highlights the importance of making optimal charging decisions when solving the E-VRP-NL (and E-VRPs in general). With {Split(H), GCI(S)} the method obtains better results than with {Split(S), GCI(H)}. This result was expected and is consistent with the notion that an aggressive local search is more important than an excellent initial solution generator. After analyzing our results, we decided that {$K = 80$, Split(H), GCI(S)} is the configuration that offers the best trade-off between solution accuracy and computational performance. The experiments reported in the remainder of the paper used this configuration.

4.3.3. Performance of the hybrid metaheuristic

Since the E-VRP-NL is a new problem, there are no results or algorithms to benchmark against. To get an idea of the quality of the solutions delivered by our ILS+HC, we compared its results with the results obtained by Gurobi using the MILP formulation introduced in Section 2. Gurobi reported integer solutions for 45 of the 120 instances; 27 of those solutions are proven optima. The instances with proven optima are the twenty 10-customer instances and seven 20-customer instances. Table 3 summarizes the results of our ILS+HC on the instances with proven optima using five metrics: the number of optimal solutions found, the average and maximum gap with respect to the optimal solution, the average best gap,[5] and the average computing time. The results suggest that our ILS+HC is able to deliver high-quality solutions for the E-VRP-NL. It matched the 27 optimal solutions, and the average and maximum gaps show that our method is stable.

Table 4 summarizes the results of our ILS+HC on the 18 instances in which Gurobi found an integer solution but did not prove optimality. We use the metrics introduced above, replacing the number of optimal solutions by the number of BKSs found and computing the average and maximum gaps with respect to the BKSs. On these instances, our ILS+HC found all the BKSs (18/18) and reported an average gap of 1.09.

Since we do not have benchmark solutions for the remaining 93 instances, we do not discuss in this section the results of our ILS+HC on those instances. To allow future comparisons with our method, we report in Appendix B detailed results for each of the 120 instances. These results are also available at http://www.vrp-rep.org.

[4] $G(\%) = (of - of_{BKS})/of_{BKS} \times 100$.

[5] The best gap is the gap between the best solution found over 10 runs and the optimal solution.

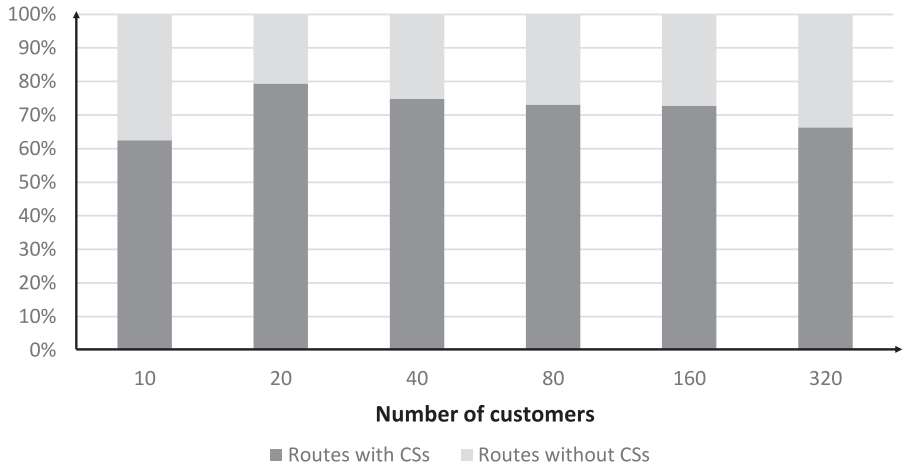
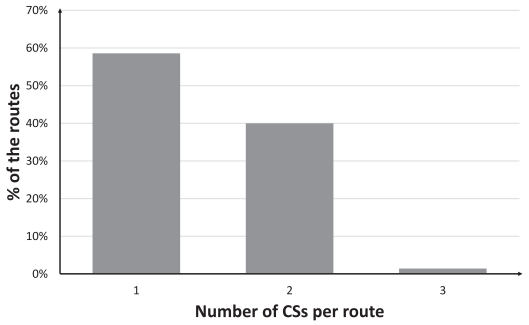
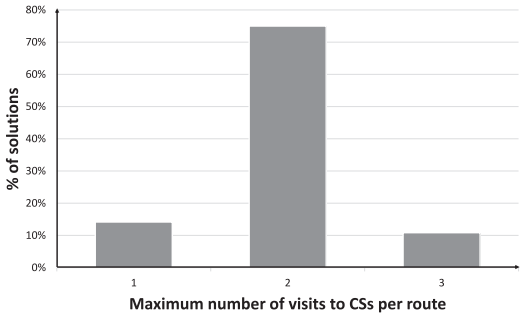


Fig. 10. Percentage of the routes with/without visits to CSs by instance size.



(a) Histogram of number of visits to CS per route.



(b) Histogram of the maximum number of visited CSs in the routes of each solution.

Fig. 11. Analysis of the number of visits to CSs.

4.3.4. Characteristics of good E-VRP-NL solutions

We analyze in this section the characteristics of the BKSs found in our experiments. We aim to provide the reader with insight that may be useful for the design of new solution methods for the E-VRP-NL.

In total, our BKSs are made up of 1426 routes. Our first analysis concerns the fraction of those routes that exploit mid-route charging. Fig. 10 presents the percentage of routes with and without visits to CSs, grouped by instance size. The data shows that on average 71.47% of the routes in the BKSs visit at least one CS. This percentage is roughly the same for each instance size. This figure provides two insights. First, mid-route charging is a key element of good E-VRP-NL solutions (probably because it gives algorithms flexibility in assigning the customers to the routes). Second, since charging decisions affect most of the routes making up a good solution, they play a critical role in its quality.

The second analysis concerns the number of mid-route charges per route. Fig. 11a and b presents histograms of the number of mid-route charges per route and the maximum number of mid-route charges per route in a solution. Fig. 11a shows that for those routes performing mid-route charging, 58.58% charge once, 40.00% twice, and 1.43% three times. Although a large proportion of the routes perform a single mid-route charge, 85.83% of the solutions contain at least one route with more than one mid-route charge (Fig. 11b). These figures suggest that models and methods for the E-VRP-NL can benefit from relaxing the upper bound of one visit to a CS per route that is sometimes used in the literature.

The third analysis concerns the energy recovered through mid-route charges. Fig. 12 presents a histogram of the average battery level (in % of the total battery capacity) after a mid-route charge. The numbers show that over 90% of the mid-route charges are partial charges (i.e., they do not fully charge the battery). This figure indicates the importance of embedding components capable of making partial charging decisions into E-VRP-NL solution methods.[6] A second interesting observation from the data displayed in Fig. 12 is the percentage (around 12%) of mid-route charges that restore the battery to above 80%

[6] Note that until 2016 this was the exception rather than the rule in the E-VRP literature.

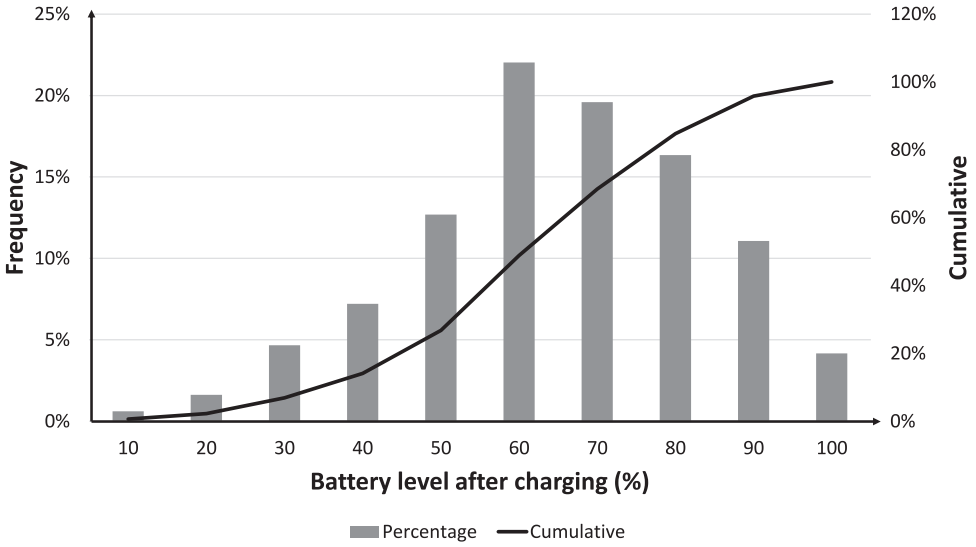


Fig. 12. Histogram of the average battery level (in % of total battery capacity) after a mid-route charge.

of its capacity. As mentioned in Section 2, a common assumption in the E-VRP literature is that the battery can be charged only in the linear segment of the charging curve (which ends at roughly 80% of the capacity). Our data suggests that good E-VRP-NL solutions often include routes with mid-route charges that take the battery level into the nonlinear part of the charging function.

5. Conclusion and future work

In this paper, we have introduced a new E-VRP that captures the nonlinear charging behavior of the charging process using a piecewise linear approximation: the electric vehicle routing problem with nonlinear charging function (E-VRP-NL). To solve the problem we proposed an ILS enhanced with HC. At the heart of our method is a neighborhood scheme that solves a new variant of the FRVCP. This problem consists in optimizing the charging decisions (where and how much to charge) of a route serving a fixed sequence of customers. We conducted an experiment comparing our charging function approximation to those in the literature. Our results show that neglecting the nonlinear nature of the charging function may lead to infeasible or overly expensive solutions. To assess the performance of our ILS+HC, we ran it on a new set of instances. We compared its solutions with the solutions obtained by Gurobi using the MILP formulation of the E-VRP-NL. The results suggest that our ILS+HC is able to deliver high-quality solutions for the E-VRP-NL. Finally, we analyzed the solutions delivered by our method to provide fellow researchers with insight into the characteristics of good E-VRP-NL solutions. Our analysis concluded that good solutions tend to use multiple mid-route charges, exploit partial recharges, and employ the nonlinear segment of the battery charging function.

Interesting research directions include designing alternative E-VRP-NL methods (both exact and heuristic) to give a better basis for comparison with our results. Another interesting possibility would be to develop approaches for the FRVCP that offer a different trade-off between accuracy and efficiency than those of the two approaches proposed in this paper. Last but not least, it would be interesting to extend the E-VRP-NL to consider capacitated CSs. To adapt our ILS+HC to this new variant two components would need to be re-designed: the GCI neighborhood and the HC. The former would need to solve the FRVCP simultaneously for all routes ensuring that CS capacity constraints are satisfied. The latter would need to guarantee that CS capacities are satisfied in the set-partitioning formulation. In ongoing research, we are working on these extensions.

Acknowledgement

The authors would like to thank the Universidad EAFIT scientific computing center (APOLO) for its support for the computational experiments. This research was partly funded in Colombia by Universidad EAFIT, Programa de Movilidad Doctoral hacia Francia (Colfuturo - Emb. de Francia - ASCUN - Colciencias - Min. de Educación), Programa Enlaza Mundos (Alcaldía de Medellín), and Universidad de Antoquia through project UdeA-2016-10705; and in France by Agence Nationale de la Recherche through project e-VRO (ANR-15-CE22-0005-01).

Appendix A. Solution approaches for the fixed-route vehicle-charging problem

We present in this appendix two approaches for the FRVCP. The first solves an MILP formulation using a commercial solver, and the second solves the problem using a greedy heuristic adapted from the literature. Note that the FRVCP can also be modeled as an (electric vehicle) shortest path problem which can be solved using the Zündorf (2014) algorithm.

A.1. MILP formulation running on commercial solver

The model introduced in Section 2.4 may seem too complex to be efficiently solved using out-of-the-box software. However, in practice the number of customers per route and the number of available CSs tend to be low, so the resulting MILP formulations are within the scope of commercial solvers. For instance, on a problem with 8 CSs, the MILP formulation for a fixed route serving 10 customers has 539 continuous variables, 352 integer variables, and 1263 constraints.[7] We optimally solved this problem using Gurobi Optimizer (version 5.6.0) in 0.06 s. We therefore decided to embed into our ILS+HC a component that uses a commercial solver for the FRVCP.

To further reduce the size of the MILP formulation and improve the solver's performance, we propose four preprocessing strategies that eliminate infeasible CS insertions. Our strategies rely on the following two premises: (i) the energy consumption and the travel time between vertices satisfy the triangular inequality; and (ii) since the piecewise-linear charging function is concave (i.e., $\rho_{j,k-1} \geq \rho_{jk}, \forall j \in F, k \in B$), the first segment has the fastest charging rate.

We propose four strategies. The first three filter CS insertions that are infeasible independently of how the customers are sequenced in the routes. These strategies are applied only once before running the ILS+HC. The fourth strategy filters CS insertions that are infeasible for a specific fixed route.

Strategy 1: This strategy estimates the minimum time τ needed to visit CS $j \in F$ between two vertices i and $h \in I \cup \{0\}$. This time is defined as the sum of a lower bound on the travel time (u) and a lower bound on the charging time (v) of any route serving customers i and h. Note that in any route serving customer i, t_{0i} is a lower bound on the route's duration from the start to customer i. Similarly, note that in any route serving customer h, t_{h0} is a lower bound on the time to complete a route from vertex h. From these two observations we can compute the minimum duration of a route visiting CS j between vertices i and h as $u = t_{0i} + t_{ij} + t_{jh} + t_{h0} + p_i + p_h$. To compute the minimum charging time v, we need to compute the minimum amount of energy (ec) that an EV coming from vertex i and traveling to vertex h must charge at CS j. This amount is the charge needed to recover the energy consumed to make the detour to j, i.e., $ec = e_{ij} + e_{jh} - e_{ih}$. Because the battery level when the EV arrives at i in any E-VRP-NL solution is unknown a priori, we consider that the battery is charged at j using the fastest charging rate (ρ_{0j}). Then $v = \frac{ec}{\rho_{0j}}$. It is clear that if $\tau = u + v > T_{\max}$ then any route visiting CS j between customers i and h is infeasible. We therefore forbid this insertion in our MILP.

Strategy 2: This strategy computes a lower bound on the remaining energy that an EV must have on arrival at customer i to be able to visit CS j next. Note that in terms of energy remaining, the best way to reach vertex i is to visit it immediately after fully charging at CS $c(i) = \arg\min_{l \in F \cup \{0\}} e_{li}$. If $Q - e_{c(i)i} < e_{ij}$ any route visiting j after i is energy-infeasible and we can safely forbid this insertion in our MILP.

Strategy 3: Note that $o = Q - e_{ji}$ is a lower bound on the energy remaining when an EV arrives at customer i immediately after charging at CS j. Note also that o must be enough to at least close the route (reach the depot) or reach the closest CS in terms of energy consumption. If $o < e_{ic(i)}$, where $c(i) = \arg\min_{l \in F \cup \{0\}} e_{il}$, any route visiting j before i is energy-infeasible and we can safely forbid this insertion.

Strategy 4: This strategy estimates a lower bound on the new duration of a given fixed route if CS $j \in F$ is inserted between vertices $\pi(i)$ and $\pi(i+1) \in \Pi$, $i \neq n_r$. This bound, $\overline{t'}$, is defined as the sum of the new travel time (u) and a lower bound on the charging time (v). It is easy to see that $u = \bar{t} + t_{\pi(i)j} + t_{j\pi(i+1)} - t_{\pi(i)\pi(i+1)}$. Similarly to Strategy 1, to estimate v, we consider that the battery is charged at j using the fastest charging rate. Therefore, $v = \frac{ec}{\rho_{0j}}$, where $ec = e_{\pi(i)j} + e_{j\pi(i+1)} - e_{\pi(i)\pi(i+1)}$ is the charge needed to recover the energy consumed in the detour to j. If $\overline{t'} = (u + v) > T_{\max}$, inserting j between vertices $\pi(i)$ and $\pi(i+1)$ leads to an infeasible route, so we can safely forbid this insertion.

Applying our preprocessing strategies to the MILP formulation for the 10-customer route of the example above, we reduce the model to 71 continuous variables, 40 integer variables, and 165 constraints. We solved the model in Gurobi in 0.02 s.

A.2. Greedy heuristic

Existing metaheuristics for E-VRPs use various approaches to make charging decisions. One popular approach is the *recharge relocation operator* proposed by Felipe et al. (2014) for the green vehicle routing problem with multiple technologies and partial recharges (GVRP-MTPR). This approach considers the insertion of only one CS per route. Starting from an energy-feasible fixed-route the procedure first deletes the current CS. Then, it tries to improve the charging decisions by

[7] We randomly picked the route from the best solution found for a randomly selected instance.

inserting each CS into each arc of the fixed-route. To decide how much energy to charge at the inserted CS, the algorithm applies a simple rule: charge the minimum amount of energy needed to reach the depot (i.e., to complete the route). We propose here a heuristic to solve the FRVCP based on this approach.

Our heuristic has two phases: *location of CSs* and *setting of the charge*. In the first phase, it iteratively inserts CSs into the arcs of the fixed-route Π to ensure feasibility in terms of energy. In the second phase, it improves the charging decisions by adjusting the energy charged at each visited CS. Algorithm 1 describes our heuristic. The approach uses four important procedures: `trackBattery(·)`, `sumNegative(·)`, `totalTime(·)`, and `copyAndInsert(·)`. Procedure `trackBattery(·)` computes the battery level Y_i at each vertex $i \in \Pi$, assuming that the EV fully charges its battery at each visited CS. Note that Y_i may take negative values. Procedure `sumNegative(·)` computes the sum of the battery levels with negative values (i.e., $s = \sum_{i \in \Pi} \min\{0, Y_i\}$). Procedure `totalTime(·)` computes the total time t of the route (assuming a full charging policy). Finally, procedure `copyAndInsert(·)` takes as input a fixed route, a CS, and a position in the route; it returns a copy of the fixed route with the CS inserted at the given position.

The heuristic starts with the location phase (lines 2–27). After computing s for the current fixed-route Π, it enters the outer loop (lines 7–27). In each pass through the inner loop (lines 8–25), the heuristic (i) evaluates the insertion of a CS into each arc of Π assuming that the EV fully charges its battery, and (ii) selects the insertion that maximizes s (lines 14–18). If $s = 0$ (i.e., the route is energy-feasible), it selects the insertion that minimizes t (lines 19–23). Then, it performs the selected insertion (line 26). If the route Π is still energy-infeasible (i.e., $s < 0$), the heuristic starts again at line 8 and tries to insert additional CSs until feasibility in terms of energy is reached.

In the charge-setting phase (line 28), the heuristic invokes procedure `ruleMinEnergy(`Π`)` to set the energy charged at each CS according to the Felipe et al. (2014) rule. Finally, it evaluates if the route satisfies the maximum-duration constraint (lines 29–33). It returns a Boolean variable indicating whether or not the fixed-route is feasible (f) and the route Π with the newly inserted CSs.

Algorithm 1 Greedy heuristic.

```
1: function GreedyHeuristic(Π^0, F)
2:     Π ⟵ Π^0
3:     Y ⟵ trackBattery(Π)
4:     s ⟵ sumNegative(Y)
5:     t ⟵ ∞
6:     f ⟵ false
7:     while s < 0 do
8:         for j = 1 to |F| do
9:             for i = 0 to n_r − 1 do
10:                Π′ ⟵ copyAndInsert(Π, F_j, i)
11:                Y′ ⟵ trackBattery(Π′)
12:                s′ ⟵ sumNegative(Y′)
13:                t′ ⟵ totalTime(Π′)
14:                if s′ > s then
15:                    s ⟵ s′
16:                    u ⟵ j
17:                    v ⟵ i
18:                end if
19:                if s′ = 0 and t′ < t then
20:                    t ⟵ t′
21:                    u ⟵ j
22:                    v ⟵ i
23:                end if
24:            end for
25:        end for
26:        Π ⟵ copyAndInsert(Π, F_u, v)
27:    end while
28:    ⟨t, Π⟩ ⟵ RuleMinEnergy(Π)
29:    if t ≤ T_max then
30:        f ⟵ true
31:    else
32:        Π ⟵ Π^0
33:    end if
34:    return f, Π
35: end function
```

Table B.5
Results of ILS+HC on the 120 instances .

		Gurobi		ILS+HC				
Instance	BKS	Best	G (%)	Best	G (%)	Avg.	G (%)	t (s)
tc2c10s2cf0	21.77	**21.77**	0.00	**21.77**	0.00	21.77	0.00	8.53
tc0c10s2cf1	19.75	**19.75**	0.00	**19.75**	0.00	20.12	1.87	3.86
tc1c10s2cf2	9.03	**9.03**	0.00	**9.03**	0.00	9.07	0.44	2.43
tc1c10s2cf3	16.37	**16.37**	0.00	**16.37**	0.00	16.37	0.00	5.63
tc1c10s2cf4	16.10	**16.10**	0.00	**16.10**	0.00	16.10	0.00	4.79
tc2c10s2ct0	12.45	**12.45**	0.00	**12.45**	0.00	12.45	0.00	5.38
tc0c10s2ct1	12.30	**12.30**	0.00	**12.30**	0.00	12.34	0.33	3.99
tc1c10s2ct2	10.75	**10.75**	0.00	**10.75**	0.00	10.75	0.00	4.21
tc1c10s2ct3	13.17	**13.17**	0.00	**13.17**	0.00	13.18	0.08	7.56
tc1c10s3ct4	13.21	**13.21**	0.00	**13.21**	0.00	13.21	0.00	6.01
tc2c10s3cf0	21.77	**21.77**	0.00	**21.77**	0.00	21.77	0.00	8.90
tc0c10s3cf1	19.75	**19.75**	0.00	**19.75**	0.00	20.12	1.87	4.41
tc1c10s3cf2	9.03	**9.03**	0.00	**9.03**	0.00	9.06	0.33	2.36
tc1c10s3cf3	16.37	**16.37**	0.00	**16.37**	0.00	16.37	0.00	6.06
tc1c10s3cf4	14.90	**14.90**	0.00	**14.90**	0.00	14.90	0.00	6.72
tc2c10s3ct0	11.51	**11.51**	0.00	**11.51**	0.00	11.54	0.26	6.81
tc0c10s3ct1	10.80	**10.80**	0.00	**10.80**	0.00	10.80	0.00	4.83
tc1c10s3ct2	9.20	**9.20**	0.00	**9.20**	0.00	9.34	1.52	5.33
tc1c10s3ct3	13.02	**13.02**	0.00	**13.02**	0.00	13.02	0.00	9.77
tc1c10s2ct4	13.83	**13.83**	0.00	**13.83**	0.00	13.83	0.00	4.84
tc2c20s3cf0	24.68	24.73	0.20	**24.68**	0.00	24.68	0.00	13.86
tc1c20s3cf1	17.50	17.55	0.29	**17.50**	0.00	17.53	0.17	12.32
tc0c20s3cf2	27.60	28.54	3.41	**27.60**	0.00	27.66	0.22	11.77
tc1c20s3cf3	16.63	16.81	1.08	**16.63**	0.00	16.78	0.90	8.41
tc1c20s3cf4	17.00	**17.00**	0.00	**17.00**	0.00	17.00	0.00	3.77
tc2c20s3ct0	25.79	**25.79**	0.00	**25.79**	0.00	25.79	0.00	14.66
tc1c20s3ct1	18.95	19.38	2.27	**18.95**	0.00	19.38	2.27	15.25
tc0c20s3ct2	17.08	17.11	0.18	**17.08**	0.00	17.13	0.29	8.49
tc1c20s3ct3	12.65	12.68	0.24	**12.65**	0.00	12.72	0.55	8.86
tc1c20s3ct4	16.21	**16.21**	0.00	**16.21**	0.00	16.25	0.25	5.16
tc2c20s4cf0	24.67	25.36	2.80	**24.67**	0.00	24.69	0.08	14.63
tc1c20s4cf1	16.39	16.40	0.06	**16.39**	0.00	16.40	0.06	13.47
tc0c20s4cf2	27.48	–	–	**27.48**	0.00	27.61	0.47	12.81
tc1c20s4cf3	16.56	16.80	1.45	**16.56**	0.00	16.80	1.45	8.69
tc1c20s4cf4	17.00	**17.00**	0.00	**17.00**	0.00	17.00	0.00	4.17
tc2c20s4ct0	26.02	–	–	**26.02**	0.00	26.02	0.00	15.25
tc1c20s4ct1	18.25	18.25	0.00	**18.25**	0.00	18.32	0.38	16.14
tc0c20s4ct2	16.99	17.21	1.29	**16.99**	0.00	17.10	0.65	9.33
tc1c20s4ct3	14.43	**14.43**	0.00	**14.43**	0.00	14.50	0.49	7.99
tc1c20s4ct4	17.00	**17.00**	0.00	**17.00**	0.00	17.00	0.00	6.08
tc0c40s5cf0	32.67	–	–	**32.67**	0.00	33.25	1.78	23.85
tc1c40s5cf1	65.16	–	–	**65.16**	0.00	66.03	1.34	44.01
tc2c40s5cf2	27.54	38.93	41.36	**27.54**	0.00	27.67	0.47	31.64
tc2c40s5cf3	19.74	21.04	6.59	**19.74**	0.00	20.18	2.23	16.85

(*continued on next page*)

Appendix B. Detailed results of the hybrid metaheuristic

Table B.5 gives the results reported by our ILS+HC on the E-VRP-PNL instances. We compare our results with the results obtained by Gurobi running the MILP introduced in Section 2. For each instance, we give the problem name[8] and the BKS reported by Gurobi or ILS+HC.

For the results obtained with Gurobi, we report the best solution (Best) and the gap with respect to the BKS (G). For the results obtained with the ILS+HC, we report the best solution, the average solution (Avg.), and the average computing time (CPU) in seconds over ten runs. For the ILS+HC, we provide the gap of the average solution and best solution with reference to the BKS. The last rows of the table summarize the average and maximum gaps with respect to the BKSs, the number of times each method found the BKS, and the average and maximum running times. Values in bold indicate that a method found the BKS.

[8] tcαcβsμcϵ#, where $\boldsymbol{\alpha}$ is the method used to place the customers (i.e., 0: randomization, 1: clustering, 2: mixture of both), $\boldsymbol{\beta}$ is the number of customers, $\boldsymbol{\mu}$ is the number of the CSs, $\boldsymbol{\epsilon}$ is 't' if we use a p-median heuristic to locate the CSs and 'f' otherwise, and **#** is the number of the instance for each combination of parameters (i.e., # = 0, 1, 2, 3, 4).

Table B.5 (*continued*)

		Gurobi		ILS+HC				
Instance	BKS	Best	G (%)	Best	G (%)	Avg.	G (%)	t (s)
tc0c40s5cf4	30.77	36.47	18.52	**30.77**	0.00	31.49	2.34	33.33
tc0c40s5ct0	28.72	–	–	**28.72**	0.00	29.35	2.19	24.50
tc1c40s5ct1	52.68	–	–	**52.68**	0.00	53.36	1.29	58.52
tc2c40s5ct2	26.91	–	–	**26.91**	0.00	27.02	0.41	22.85
tc2c40s5ct3	23.54	–	–	**23.54**	0.00	23.77	0.98	26.48
tc0c40s5ct4	28.63	–	–	**28.63**	0.00	28.72	0.31	32.55
tc0c40s8cf0	31.28	–	–	**31.28**	0.00	32.02	2.37	33.59
tc1c40s8cf1	40.75	–	–	**40.75**	0.00	42.33	3.88	69.99
tc2c40s8cf2	27.15	29.19	7.51	**27.15**	0.00	27.31	0.59	28.92
tc2c40s8cf3	19.66	22.01	11.95	**19.66**	0.00	20.24	2.95	19.46
tc0c40s8cf4	29.32	–	–	**29.32**	0.00	29.86	1.84	43.05
tc0c40s8ct0	26.35	30.29	14.95	**26.35**	0.00	26.89	2.05	28.54
tc1c40s8ct1	40.56	–	–	**40.56**	0.00	41.19	1.55	70.50
tc2c40s8ct2	26.33	–	–	**26.33**	0.00	26.71	1.44	25.64
tc2c40s8ct3	22.71	23.51	3.52	**22.71**	0.00	23.23	2.29	25.25
tc0c40s8ct4	29.20	–	–	**29.20**	0.00	29.27	0.24	47.46
tc0c80s8cf0	39.43	–	–	**39.43**	0.00	39.86	1.09	56.41
tc0c80s8cf1	45.23	–	–	**45.23**	0.00	45.73	1.11	121.27
tc1c80s8cf2	30.81	–	–	**30.81**	0.00	31.83	3.31	50.99
tc2c80s8cf3	32.44	–	–	**32.44**	0.00	32.60	0.49	64.05
tc2c80s8cf4	49.29	–	–	**49.29**	0.00	49.69	0.81	99.84
tc0c80s8ct0	41.90	–	–	**41.90**	0.00	42.76	2.05	54.35
tc0c80s8ct1	45.27	–	–	**45.27**	0.00	45.85	1.28	129.66
tc1c80s8ct2	31.74	–	–	**31.74**	0.00	32.36	1.95	59.73
tc2c80s8ct3	32.31	–	–	**32.31**	0.00	32.55	0.74	65.15
tc2c80s8ct4	44.83	–	–	**44.83**	0.00	46.61	3.97	111.24
tc0c80s12cf0	34.64	–	–	**34.64**	0.00	35.59	2.74	57.24
tc0c80s12cf1	42.90	–	–	**42.90**	0.00	44.07	2.73	74.58
tc1c80s12cf2	29.54	–	–	**29.54**	0.00	30.73	4.03	61.34
tc2c80s12cf3	31.97	–	–	**31.97**	0.00	32.70	2.28	75.64
tc2c80s12cf4	43.89	–	–	**43.89**	0.00	44.97	2.46	131.13
tc0c80s12ct0	39.31	–	–	**39.31**	0.00	39.83	1.32	65.54
tc0c80s12ct1	41.94	–	–	**41.94**	0.00	43.03	2.60	73.32
tc1c80s12ct2	29.52	–	–	**29.52**	0.00	30.66	3.86	58.85
tc2c80s12ct3	30.83	–	–	**30.83**	0.00	31.59	2.47	57.57
tc2c80s12ct4	42.40	–	–	**42.40**	0.00	42.82	0.99	134.33
tc1c160s16cf0	79.80	–	–	**79.80**	0.00	80.75	1.19	765.69
tc2c160s16cf1	60.34	–	–	**60.34**	0.00	61.26	1.52	273.86
tc0c160s16cf2	61.20	–	–	**61.20**	0.00	62.99	2.92	365.10
tc1c160s16cf3	71.76	–	–	**71.76**	0.00	72.75	1.38	461.58
tc0c160s16cf4	82.92	–	–	**82.92**	0.00	83.84	1.11	1213.20
tc1c160s16ct0	79.04	–	–	**79.04**	0.00	79.90	1.09	643.27
tc2c160s16ct1	60.27	–	–	**60.27**	0.00	60.62	0.58	287.64
tc0c160s16ct2	59.90	–	–	**59.90**	0.00	62.80	4.84	341.86
tc1c160s16ct3	73.29	–	–	**73.29**	0.00	75.11	2.48	278.67
tc0c160s16ct4	82.37	–	–	**82.37**	0.00	83.08	0.86	944.60
tc1c160s24cf0	78.60	–	–	**78.60**	0.00	79.30	0.89	741.12
tc2c160s24cf1	59.82	–	–	**59.82**	0.00	61.14	2.21	304.66
tc0c160s24ct2	59.25	–	–	**59.25**	0.00	60.19	1.59	409.80
tc1c160s24ct3	68.72	–	–	**68.72**	0.00	69.98	1.83	358.35
tc0c160s24cf4	81.44	–	–	**81.44**	0.00	82.13	0.85	1209.32
tc1c160s24ct0	78.21	–	–	**78.21**	0.00	79.35	1.46	577.83
tc2c160s24ct1	59.13	–	–	**59.13**	0.00	59.72	1.00	340.40
tc0c160s24cf2	59.27	–	–	**59.27**	0.00	60.92	2.78	403.33
tc1c160s24cf3	68.56	–	–	**68.56**	0.00	69.57	1.47	483.10
tc0c160s24ct4	80.96	–	–	**80.96**	0.00	82.11	1.42	956.94
tc2c320s24cf0	182.45	–	–	**182.45**	0.00	186.94	2.46	6566.41
tc2c320s24cf1	95.51	–	–	**95.51**	0.00	96.42	0.95	1456.16
tc1c320s24cf2	152.13	–	–	**152.13**	0.00	153.99	1.22	7105.63
tc1c320s24cf3	117.48	–	–	**117.48**	0.00	118.36	0.75	3065.82
tc2c320s24cf4	122.74	–	–	**122.74**	0.00	124.68	1.58	3681.14
tc2c320s24ct0	181.45	–	–	**181.45**	0.00	186.23	2.63	7204.02
tc2c320s24ct1	94.73	–	–	**94.73**	0.00	96.49	1.86	1259.26
tc1c320s24ct2	148.77	–	–	**148.77**	0.00	154.13	3.60	6853.35
tc1c320s24ct3	116.64	–	–	**116.64**	0.00	119.17	2.17	3273.79
tc2c320s24ct4	121.94	–	–	**121.94**	0.00	123.85	1.57	4273.94

(*continued on next page*)

Table B.5 (*continued*)

		Gurobi		ILS+HC				
Instance	BKS	Best	G (%)	Best	G (%)	Avg.	G (%)	t (s)
tc2c320s38cf0	176.92	–	–	**176.92**	0.00	182.31	3.05	6733.82
tc2c320s38cf1	94.29	–	–	**94.29**	0.00	95.07	0.83	1601.78
tc1c320s38cf2	141.63	–	–	**141.63**	0.00	147.08	3.85	7235.62
tc1c320s38cf3	116.22	–	–	**116.22**	0.00	117.74	1.31	3113.71
tc2c320s38cf4	122.32	–	–	**122.32**	0.00	123.47	0.94	2660.68
tc2c320s38ct0	190.97	–	–	**190.97**	0.00	192.15	0.62	7636.50
tc2c320s38ct1	94.53	–	–	**94.53**	0.00	95.29	0.80	1408.88
tc1c320s38ct2	140.96	–	–	**140.96**	0.00	145.09	2.93	6974.34
tc1c320s38ct3	116.07	–	–	**116.07**	0.00	117.71	1.41	3062.95
tc2c320s38ct4	121.66	–	–	**121.66**	0.00	123.15	1.22	2784.91
Avg. Gap			2.61		0.00		1.32	
Max. Gap			41.36		0.00		4.44	
Found solution		45		120				
Best		27		120				
Avg. Time								849.55
Max. Time								7636.50

References

Adler, J.D., Mirchandani, P.B., 2014. Online routing and battery reservations for electric vehicles with swappable batteries. Transp. Res. Part B 70, 285–302.

Afroditi, A., Boile, M., Theofanis, S., Sdoukopoulos, E., Margaritis, D., 2014. Electric vehicle routing problem with industry constraints: trends and insights for future research. Transp. Res. Procedia 3, 452–459.

Bansal, P., 2015. Charging of electric vehicles: technology and policy implications. J. Sci. Policy Governance 6 (1).

Becker, T.A., Sidhu, I., Tenderich, B., 2009. Electric Vehicles in the United States: A New Model with Forecasts to 2030. Technical Report. Center for Entrepreneurship & Technology (CET) , University of California, 2009.1.v.2.0.

Bruglieri, M., Colorni, A., Lué, A., 2014. The vehicle relocation problem for the one-way electric vehicle sharing: an application to the Milan case. Procedia - Social Behav. Sci. 111, 18–27.

Bruglieri, M., Pezzella, F., Pisacane, O., Suraci, S., 2015. A variable neighborhood search branching for the electric vehicle routing problem with time windows. Electron. Notes Discrete Math. 47, 221–228.

Conrad, R.G., Figliozzi, M.A., 2011. The recharging vehicle routing problem. In: Doolen, T., Aken, E.V. (Eds.), Proceedings of the 2011 Industrial Engineering Research Conference. Reno, NV, USA.

De Cauwer, C., Van Mierlo, J., Coosemans, T., 2015. Energy consumption prediction for electric vehicles based on real-world data. Energies 8 (8), 8573–8593.

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

Erdoğan, S., Miller-Hooks, E., 2012. A green vehicle routing problem. Transp. Res. Part E 48 (1), 100–114.

Felipe, A., Ortuño, M.T., Righini, G., Tirado, G., 2014. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transp. Res. Part E 71, 111–128.

Goeke, D., Schneider, M., 2015. Routing a mixed fleet of electric and conventional vehicles. Eur. J. Oper. Res. 245 (1), 81–99.

Hiermann, G., Puchinger, J., Ropke, S., Hartl, R.F., 2016. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. Eur. J. Oper. Res. 252 (3), 995–1018.

Hof, J., Schneider, M., 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. Transp. Res. Part B 97, 102–112.

Hõimoja, H., Rufer, A., Dziechciaruk, G., Vezzini, A., 2012. An ultrafast ev charging station demonstrator. In: Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2012 International Symposium on. IEEE, pp. 1390–1395.

Keskin, M., Çatay, B., 2016. Partial recharge strategies for the electric vehicle routing problem with time windows. Transp. Res. Part C 65, 111–127.

Kleindorfer, P.R., Neboian, A., Roset, A., Spinler, S., 2012. Fleet renewal with electric vehicles at La Poste. Interfaces 42 (5), 465–477.

Kullman, N., Goodson, J., Mendoza, J.E., 2016. Electric vehicle routing with mid-route recharging and uncertain charging station availability. INFORMS Annual Meeting 2016.

Lenstra, J.K., Rinnooy Kan, A., 1981. Complexity of vehicle routing and scheduling problems. Networks 11 (2), 221–227.

Liao, C.-S., Lu, S.-H., Shen, Z.-J.M., 2016. The electric vehicle touring problem. Transp. Res. Part B 86, 163–180.

Lin, J., Zhou, W., Wolfson, O., 2016. Electric vehicle routing problem. Transp. Res. Procedia 12, 508–521.

Lourenço, H.R., Martin, O.C., Stützle, T., 2010. Iterated local search: framework and applications. In: Handbook of Metaheuristics. Springer, Boston, MA, pp. 363–397.

Marcacci, S., 2013. GM triples size of its Michigan electric vehicle battery laboratory (CT exclusive). https://cleantechnica.com/2013/09/17/gm-opens-50000-square-foot-electric-vehicle-battery-laboratory/, Last Accessed= 26/11/2016.

Mendoza, J.E., Guéret, C., Hoskins, M., Lobit, H., Pillac, V., Vidal, T., Vigo, D., 2014. VRP-REP: a vehicle routing community repository. Third Meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog). Oslo (Norway).

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Comput. Oper. Res. 24 (11), 1097–1100.

Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2015. A multi-space sampling heuristic for the green vehicle routing problem. Transp. Res. Part C 70, 113–128.

Pelletier, S., Jabali, O., Laporte, G., 2014. Battery Electric Vehicles for Goods Distribution: A Survey of Vehicle Technology, Market Penetration, Incentives and Practices. Technical Report. CIRRELT-2014-43.

Pelletier, S., Jabali, O., Laporte, G., 2016. Goods distribution with electric vehicles: review and research perspectives. Transp. Sci. 50 (1), 3–22.

Pelletier, S., Jabali, O., Laporte, G., Veneroni, M., 2017. Battery degradation and behaviour for electric vehicles: review and numerical analyses of several models. Transp. Res. Part B. In press.

Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Comput. Oper. Res. 31 (12), 1985–2002.

Restrepo, J., Rosero, J., Tellez, S., 2014. Performance testing of electric vehicles on operating conditions in Bogotá DC, Colombia. In: Transmission & Distribution Conference and Exposition-Latin America (PES T&D-LA), 2014 IEEE PES. IEEE, pp. 1–8.

Rosenkrantz, D.J., Stearns, R.E., Lewis, P.M., 1977. An analysis of several heuristics for the traveling salesman problem. SIAM J. Comput. 6, 563–581.
Rosing, K.E., ReVelle, C.S., 1997. Heuristic concentration: two stage solution construction. Eur. J. Oper. Res. 97 (1), 75–86.
Sassi, O., Cherif, W.R., Oulamara, A., 2014. Vehicle Routing Problem with Mixed fleet of Conventional and Heterogenous Electric Vehicles and Time Dependent Charging Costs. Technical Report. hal-01083966.
Sassi, O., Cherif-Khettaf, W., Oulamara, A., 2015. Iterated tabu search for the mix fleet vehicle routing problem with heterogenous electric vehicles. In: Le Thi, H.A., Pham Dinh, T., Nguyen, N.T. (Eds.), Modelling, Computation and Optimization in Information Systems and Management Sciences. In: Advances in Intelligent Systems and Computing, 359. Springer International Publishing, Cham, pp. 57–68.
Schiffer, M., Walther, G., 2017. The electric location routing problem with time windows and partial recharging. Eur. J. Oper. Res. doi:10.1016/j.ejor.2017.01.011. In press
Schneider, M., Stenger, A., Goeke, D., 2014. The electric vehicle-routing problem with time windows and recharging stations. Transp. Sci. 48 (4), 500–520.
Schneider, M., Stenger, A., Hof, J., 2015. An adaptive VNS algorithm for vehicle routing problems with intermediate stops. OR Spectr. 37 (2), 353–387.
Suzuki, Y., 2014. A variable-reduction technique for the fixed-route vehicle-refueling problem. Comput. Ind. Eng. 67, 204–215.
Sweda, T.M., Dolinskaya, I.S., Klabjan, D., 2014. Optimal recharging policies for electric vehicles. Transp. Sci. doi:10.1287/trsc.2015.0638. In press
Sweda, T.M., Dolinskaya, I.S., Klabjan, D., 2015. Adaptive routing and recharging policies for electric vehicles. Transp. Sci.. In press
Szeto, W.Y., Cheng, Y., 2016. Artificial bee colony approach to solving the electric vehicle routing problem. Annual Meeting of the Transportation Research Board, TRB 2016.
Uhrig, M., Weiß, L., Suriyah, M., Leibfried, T., 2015. E-mobility in car parks–guidelines for charging infrastructure expansion planning and operation based on stochastic simulations. the 28th International Electric Vehicle Symposium and Exhibition, KINTEX, Korea. Goyang, Korea.
Wang, H., Liu, Y., Fu, H., Li, G., 2013. Estimation of state of charge of batteries for electric vehicles. Int. J. Control Autom. 6 (2), 185–194.
Zündorf, T., 2014. Electric Vehicle Routing with Realistic Recharging Models. Karlsruhe Institute of Technology, Karlsruhe, Germany. Master's thesis.