HAL
open science

The electric vehicle routing problem with capacitated charging stations

Aurélien Froger, Jorge E. Mendoza, Ola Jabali, Gilbert Laporte

To cite this version:

Aurélien Froger, Jorge E. Mendoza, Ola Jabali, Gilbert Laporte. The electric vehicle routing problem with capacitated charging stations. 2019. hal-02386167v1

HAL Id: hal-02386167

https://hal.science/hal-02386167v1

Preprint submitted on 29 Nov 2019 (v1), last revised 31 Dec 2021 (v3)

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

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

The electric vehicle routing problem with capacitated charging stations

Aurélien Froger[1] Jorge E. Mendoza[2] Ola Jabali[3] Gilbert Laporte[2]

[1] Université de Bordeaux, UMR CNRS 5251 – Inria Bordeaux Sud-Ouest

[2] HEC Montréal, Montréal, Canada H3T 2A7

[3] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy

Submitted for publication in November 2019

Abstract

Much of the existing research on electric vehicle routing problems (E-VRPs) assumes that the charging stations (CSs) can simultaneously charge an unlimited number of electric vehicles, but this is not the case. In this research, we investigate how to model and solve E-VRPs taking into account these capacity restrictions. In particular, we study an E-VRP with non-linear charging functions, multiple charging technologies, en route charging, and variable charging quantities, while explicitly accounting for the capacity of CSs expressed in the number of chargers. We refer to this problem as the E-VRP with non-linear charging functions and capacitated stations (E-VRP-NL-C). This problem advances the E-VRP literature by considering the scheduling of charging operations at each CS. We first introduce two mixed integer linear programming formulations showing how CS capacity constraints can be incorporated into E-VRP models. We then introduce an algorithmic framework to the E-VRP-NL-C, that iterates between two main components: *a route generator* and *a solution assembler*. The route generator uses an iterated local search algorithm to build a pool of high-quality routes. The solution assembler applies a branch-and-cut algorithm to select a subset of routes from the pool. We report on computational experiments comparing four different assembly strategies on a large and diverse set of instances. Our results show that our algorithm deals with the CS capacity constraints effectively. Furthermore, considering the well-known uncapacitated version of the E-VRP-NL-C, our solution method identifies new best-known solutions for 80 out of 120 instances.

Keywords— Electric vehicle routing; non-linear charging function; synchronization constraints; mixed integer linear programming; matheuristic; iterated local search; branch-and-cut

1 Introduction

In recent years, competitive prices and technological advances have made electric vehicles (EVs) an attractive alternative to internal combustion engine-powered vehicles for logistics operations (Juan et al., 2016; Pelletier et al., 2016). Consequently, the operations research community has started paying attention to the study of electric vehicle routing problems (E-VRPs), which consist of designing routes to serve a set of customers using a fleet of EVs. Due to their relatively short driving range, EVs may need to detour to charging stations (CSs) to replenish their battery, especially in the context of mid-haul or long-haul routing (Schiffer et al., 2018; Villegas et al., 2018). Therefore, key decisions in E-VRPs concern not only the sequence in which the customers are served, but also where and by how much to charge the vehicles.

One of the main modeling elements in E-VRPs is the charging process of batteries. Some studies assume that EVs are fully recharged whenever they detour to a CS. In the green vehicle routing problem (G-VRP) tackled by Erdoğan and Miller-Hooks (2012), Koç and Karaoglan (2016), Montoya et al. (2016), and Bartolini and Andelmin (2017), the charging time is assumed to be constant, while in the works of Schneider et al. (2014), Hiermann et al. (2016), Keskin

and Çatay (2016), and Desaulniers et al. (2016) the charging time linearly depends on the state of charge (SoC) of the EV upon arrival at the CS. A full charging policy may be too restrictive, especially when considering en route charging. As Montoya et al. (2017) pointed out, this policy may lead to unnecessary out-of-the-depot charging, which translates into expensive driver idling time and overpriced energy purchases. To overcome this drawback, several researchers have studied problem variants in which the charging time is a decision variable (Desaulniers et al. (2016), Froger et al. (2019), Felipe et al. (2014), and Montoya et al. (2017)). The latter three papers examine the case in which the CSs may have different charging speeds. In reality, however, the battery charging process follows a non-linear function with respect to time (Pelletier et al. (2017)). To account for this, Montoya et al. (2017), Koç et al. (2018), and Froger et al. (2019) have modeled the charging process through concave piecewise linear functions. In the resulting problem, known as the electric vehicle routing problem with non-linear charging function (E-VRP-NL), the charging times are estimated more precisely than in the variants that assume a linear charging process.

All papers just reviewed assume that charging stations are always available, meaning that EVs do not need to wait to start charging. This implicitly means that the E-VRP research has mainly focused on problem variants where the charging infrastructure is privately owned by the route planner. In some contexts, this is a plausible assumption, since large companies may decide to invest in their own infrastructure to avoid dealing with the uncertainty in the availability of public CSs (Villegas et al., 2018). To the best of our knowledge, there exist only on a handful of references dealing with public charging stations for E-VRPs (see Sweda et al. (2017) and Kullman et al. (2018) for further details). Keskin et al. (2019) considered deterministic time-dependent queueing times at the stations.

Another common assumption in most, if not all, E-VRP publications is that CSs are always readily available to charge a vehicle when it arrives. Thus, implicitly assuming that CSs are uncapacitated and can simultaneously charge an unlimited number of EVs. In practice, however, each CS has a fixed and often small number of chargers. The intuition behind neglecting the CS capacity constraints is that accounting for the detour and charging times (or costs) while planning the routes is enough to capture the impact of the charging decisions on the cost and feasibility of solutions. Nonetheless, the long charging times (from tens of minutes to several hours) and the small number of chargers typically available at private CSs may generate congestion. To illustrate this point, we ran a feasibility test on the 120 best-known solutions (BKSs) for the E-VRP-NL reported in Montoya et al. (2017) limiting the number of chargers per CS to one, two, three, and four. According to our results, 55 of the BKS become infeasible if there is only one charger per CS. This figure drops to 23 and three for the cases with two and three chargers. The only scenario where all BKSs are feasible is when CSs have four chargers (see Appendix A for details). Intuitively, one may think that by merely shifting the starting time of the charging operations marginally, the feasibility problem will be solved. However, our experiments show that this is not only unnecessarily expensive, but it may also be infeasible. Another option to mitigate the effects of congestion is to increase the number of chargers installed at each CS, but this may not be a viable solution in practice. Indeed, Gnann et al. (2018) predict that in 2020, the purchase and installation costs of a fast charging point (i.e., power rates above 22 kW) will be around 40,000€ and its annual operation cost will reach 4,000€. Thus, if a company decides to invest in out-of-the-depot charging infrastructure, there is little chance that it will decide to install more than a couple of chargers at each CS. In conclusion, we argue that in most practical situations, the congestion at the CSs should be taken into account when planning the routes by explicitly modeling the CS capacity constraints.

We are aware of only one study taking into account CS capacity constraints when optimizing routing decisions. Bruglieri et al. (2019) explicitly limited the number of vehicles simultaneously refueling at each alternative fuelling station according to its number of fueling pumps. Some researchers have studied related but different problems. For instance, Sassi and Oulamara (2014) and Pelletier et al. (2018) considered the problem of scheduling charging operations at the depot, assuming that the routes are given as an input. Both papers considered not only constraints on the number of available chargers, but also electricity grid constraints limiting the amount of power that can be drawn from the electric grid at any given time. The latter is typically a binding constraint for central depots with a large number of available chargers, but is usually not a concern for CSs since they are designed to operate at full capacity.

In this paper we introduce the E-VRP-NL with capacitated CSs (E-VRP-NL-C). The problem extends classical E-VRPs to account for the fixed number of chargers available at each CS. The E-VRP-NL-C is a complex combined routing-scheduling problem belonging to the family of VRPs with synchronization constraints. More specifically, according to the taxonomy introduced by Drexl (2012), the E-VRP-NL-C belongs to the class of VRPs with *resource synchronization*, where vehicles compete to access scarce resources (i.e., the chargers at every CS). The E-VRP-NL-C shares similarities

with problems where vehicles need to wait while the loading equipment is busy. Examples of problems in this class include the log truck scheduling problem (El Hachemi et al., 2013; Rix et al., 2015) and routing and scheduling problems arising in public works (Grimault et al., 2017). The E-VRP-NL-C also relates to VRPs with inter-tour resource constraints. In these problems, the scarce resources are located only at the terminal node of the routes (i.e., the depot). An example of a problem in this category is the VRP introduced by Hempsch and Irnich (2008), where there is a limited number of ramps at the depot, and therefore only a fixed number of vehicles can be served simultaneously. The problem that is most closely related to the E-VRP-NL-C is the VRP with location congestion introduced by Lam and Van Hentenryck (2016). In this problem, each customer has a given number of requests and a limited number of resources (e.g., conveyors). When a vehicle arrives at a customer location, it must wait until at least one resource becomes available to handle the shipping. However, there exist two fundamental differences between their problem and our E-VRP-NL-C. The first is that in our problem, visiting nodes with limited resources (the CSs) is needed for feasibility reasons. As a consequence, not only the number, but also the location and duration of the charging operations depend on the configuration of the routes. In other words, the *tasks* that must be synchronized are not known a priori. The second difference is that in contrast to their problem, in our E-VRP-NL-C, the task synchronization has a direct impact on the solution cost. Dealing with these two features adds a thick layer of complexity to the models and solution methods.

The contribution of this paper is twofold. First, we propose two mixed integer linear programming (MILP) formulations for the E-VRP-NL-C, showing how capacity constraints can be integrated into standard E-VRP models. Second, we introduce a framework to solve E-VRPs with CS capacity constraints in general, and the E-VRP-NL-C in particular. It is made up of two interacting components: a *route generator* and a *solution assembler*. The first component builds a set of high-quality solutions, while relaxing the CS capacity constraints. The routes making up these solutions are stored in a pool which is sent to the assembler every few iterations. The latter combines routes from the pool in an attempt to build a solution meeting the CS capacity constraints. If such a solution does not exist or cannot be found within a given computing time, the assembler sends a signal to the generator to modify the search space in order to favor feasibility. For the particular case of the E-VRP-NL-C, we have developed a route generator based on a new, powerful, and efficient iterated local search (ILS) metaheuristic for the E-VRP-NL, and a solution assembler based on branch-and-cut. We present four assembly strategies allowing for different tradeoffs between efficiency and effectiveness. We have carried out extensive computational experiments on a large set of instances with different characteristics adapted from available benchmarks. The results demonstrate that our approach can handle the CS capacity constraints effectively. In addition, we report new BKS for 80 out of the 120 instances of a well-established benchmark set for the closely related E-VRP-NL.

The remainder of this paper is organized as follows. Section 2 formally introduces the E-VRP-NL-C. Section 3 describes MILP formulations of the problem. Section 4 presents the proposed solution method. Section 5 shows the computational results. Finally, Section 6 concludes and outlines research perspectives.

2 Problem description

We define the E-VRP-NL-C as follows. Let I be the set of customers that need to be served and let F be the set of charging stations (CSs) at which recharging can take place. Each customer $i \in I$ has a service time g_i. The customers are served using a homogeneous fleet of EVs. Each EV has a battery of capacity Q (expressed in kWh). At the beginning of the planning horizon, the EVs are located in a single depot, from which they leave fully charged. The depot is continuously open for T_{max} hours. Traveling from one location i (the depot, a customer, or a CS) to another location j incurs a driving time $t_{ij} \geqslant 0$ and an energy consumption $e_{ij} \geqslant 0$. Driving times and energy consumption both satisfy the triangular inequality. Due to their limited battery capacity, EVs may need to stop en route at CSs. Charging operations can occur at any CS, they are non-preemptive, and EVs can be partially recharged. The CS $j \in F$ has a concave piecewise linear charging function $\phi_j(\Delta)$ that maps for an empty battery the time Δ spent charging at j and the SoC of the vehicle upon departing from j. If q is the SoC of the EV upon arrival at j and Δ is the charging time, then the SoC of the EV upon departure from j is given by $\phi_j(\Delta + \phi_j^{-1}(q))$ (Figure 1). We denote by $B_j = \{0, \ldots, b_j\}$ the ordered set of breakpoints of the charging function at j, sorted in non-decreasing order. We also introduce a_{jk} and q_{jk} to denote the charging time and the SoC for breakpoint $k \in B_j$ of CS j.

Each CS $j \in F$ also has a *capacity*, given by the number of available chargers C_j. Due to the limited capacity of CSs, vehicles may incur waiting times while they queue for a charger. We therefore note that optimal solutions are not

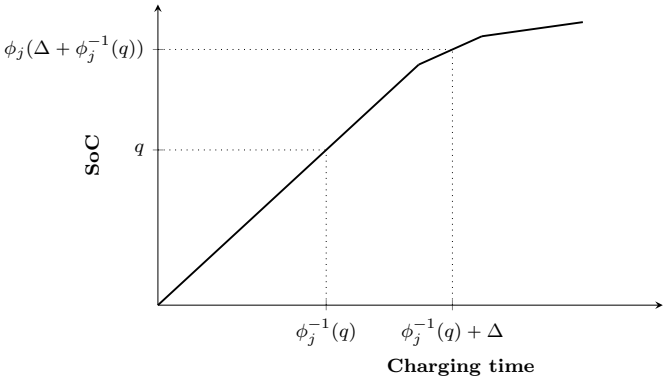


Figure 1: Piecewise linear charging function of a CS.

necessarily *left-shifted schedules*, as is the case for the E-VRP-NL and other E-VRP variants assuming uncapacitated CSs.

Feasible solutions to the E-VRP-NL-C must satisfy the following conditions: 1) each customer is visited exactly once by a single vehicle, 2) each route starts at the depot not before time 0 and ends at the depot not later than time T_{max}, 3) each route is energy-feasible, i.e., the SoC of an EV when it enters and leaves from any location lies between 0 and Q, and 4) no more than C_j EVs simultaneously charge at each CS $j \in F$. The objective of the E-VRP-NL-C is to minimize the total time needed to serve all customers, including driving, service, charging, and waiting times. Note that to avoid congestion at CSs, the starting times of the routes can be delayed at no cost. Moreover, an E-VRP-NL-C instance may have no feasible solution.

3 Two mixed-integer linear programming formulations

In this section we extend the formulations of Froger et al. (2019) for the closely related E-VRP-NL in order to deal with CS capacity constraints. These formulations belong to two families: CS replication-based formulations and path-based formulations. The former are akin to the MILP formulations which are typically used in the E-VRP literature. The latter correspond to a more intricate modeling strategy similar to the one proposed by Bartolini and Andelmin (2017) for the G-VRP. According to the results obtained by Froger et al. (2019) on small-size instances, path-based formulations outperform CS replication-based formulations. Nonetheless, we decided to explore both types of formulations to provide insights into how CS capacity constraints can be modeled under both families of formulations.

3.1 CS replication-based formulation

The E-VRP-NL-C can be defined on a digraph $G = (V, A)$, where $V = \{0\} \cup I \cup F'$ is the set of nodes and A is the set of arcs connecting the nodes of V. Node 0 represents the depot. The set F' is equal to $\cup_{i \in F} F'_i$ where for each CS $i \in F$ F'_i denotes the set containing β_i copies of i (i.e., $|F'_i| = \beta_i$). The value of β_i corresponds to an upper bound on the number of visits to CS i. Note that we must have $\beta_i \gg C_i$; otherwise, the capacity constraints are conservatively respected (the maximum number of charging operations at a CS being smaller than its capacity). We denote by $F'_i \subseteq F'$ the set containing the β_i copies of CS i (i.e., $|F'_i| = \beta_i$ and $F' = \bigcup_{i \in F} F'_i$). We assume that F'_i is an ordered set whose elements are numbered from 1 to β_i. In the remainder of this paper, depending on the context, we refer to an element of F' or F'_i as a CS copy or as a potential charging operation. We use the preprocessing procedure presented in Froger et al. (2019) to reduce the number of arcs in A. This technique primarily removes arcs that cannot feasibly be used due to the capacity of the battery.

According to the experiments carried out by Froger et al. (2019), the best CS replication-based formulation for the E-VRP-NL uses an arc-based tracking of the time and of the SoC. We therefore decided to use arc-based tracking in our first formulation. The binary variable x_{ij} is equal to 1 if and only if an EV travels on arc $(i, j) \in A$. The continuous variables τ_{ij} and y_{ij} represent the time and SoC of an EV when it departs from vertex $i \in V$ to travel on arc (i, j). If no vehicle travels on this arc, then both variables are equal to 0. The continuous variables $\underline{q}_j$ and $\overline{q}_j$ specify the SoC of an

EV when it enters and leaves the CS copy $j \in F'$. The variables $\underline{a}_j$ and $\overline{a}_j$ are the scaled arrival and departure times, according to the charging function of CS copy j. The continuous variable Δ_j represents the duration of the charging operation performed at j. For $k \in B_j \backslash \{0\}$, the binary variables $\underline{w}_{jk}$ and $\overline{w}_{jk}$ are equal to 1 if and only if the SoC lies between $q_{j,k-1}$ and q_{jk} when the EV enters and leaves CS copy j, respectively. Finally, the continuous variables $\underline{\lambda}_{jk}$ and $\overline{\lambda}_{jk}$ are the coefficients associated with the breakpoints (a_{jk}, q_{jk}) of the piecewise linear charging function ϕ_j when the EV enters and leaves CS copy j, respectively. Without loss of generality, we restrict waiting times to occur only before charging operations. We introduce a continuous variable ∇_j representing the waiting time of an EV before the start of its charging operation at $j \in F'$.

To model the CS capacity constraints, we rely on concepts borrowed from the Resource Constrained Scheduling Problem (RCPSP) literature. More precisely, we propose a *flow-based* formulation inspired from Artigues et al. (2003). There are, however, differences between our CS scheduling problem and the RCPSP. In our problem: *i)* the duration of each task (i.e., the charging operation), and *ii)* the number of tasks executed by each resource (i.e., a CS) are decision variables, whereas these are given parameters in the RCPSP. To the best of our knowledge this particular case has not yet been considered. In our flow-based formulation (hereafter referred to as FB), we consider C_i resources (represented by the chargers) for each CS $i \in F$. Each resource can execute at most one operation at any given time. Let $i \in F$ be a CS and 0_i and $\beta_i + 1$ be two dummy operations acting as the source and the sink of the flow. We denote by $\widetilde{F'}_i$ the ordered set $\{0_i\} \cup F'_i \cup \{\beta_i + 1\}$ of potential charging operations (CS copies are considered here as potential charging operations). A visit to a CS copy requires the scheduling of the corresponding charging operation on one of the chargers of its associated CS. To break symmetries created by the introduction of copies of a CS i, without loss of generality, if $\gamma_i \leqslant \beta_i$ visits to a CS are needed, we enforce the visit of the CS copies numbered from 1 to γ_i. Moreover, among the CS copies of i that are visited, we force them to be visited in the reverse order in which they appear in F_i (i.e, a charging operation $j \in F'_i$ must start after a charging operation $l \in F'_i$ if $l > j$). The reverse order is used since the departure time and charging duration variables are equal to zero when a CS copy is not visited. The sequential binary variable u_{jl} is equal to 1 only if (potential) operation l starts later than the completion of (potential) operation j with $j, l \in F'_i$ and $j > l$. Let now $j, l \in \widetilde{F'}_i$ be two operations (potentially dummy) such that $j > l$. The continuous flow variable f_{jl} denotes the number of chargers that are transferred from operation j to operation l. This number is naturally equal to one if j and l are not both dummy. Specifically, if both j and l are not dummy, f_{jl} is equal to one if and only if the operations j and l are scheduled on the same charger, l is scheduled after j, and no other operation uses the charger between the completion of j and the beginning of l. For notational convenience, we define $\widetilde{C}_j := 1$ for all $j \in F'_i$ and $\widetilde{C}_{0_i} := \widetilde{C}_{\beta_i+1} := C_i$.

To illustrate our flow-based formulation, consider an example in which a CS i has two chargers (i.e., $C_i = 2$) and set F'_i contains five copies of this CS ($\beta_i = 5$). Figure 2 depicts the structure of the flow network. Assume that the model decides to schedule four charging operations at i. Figure 3 illustrates the structure of the flow network derived from the schedule (by eliminating the flow variables associated to sequential variables equal to zero) and describes a feasible flow.

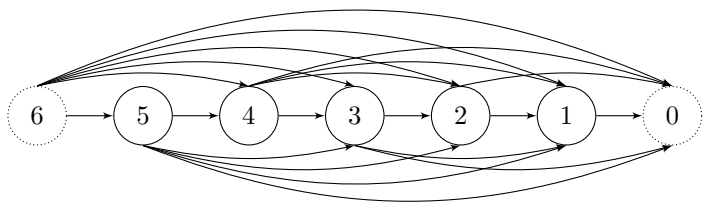
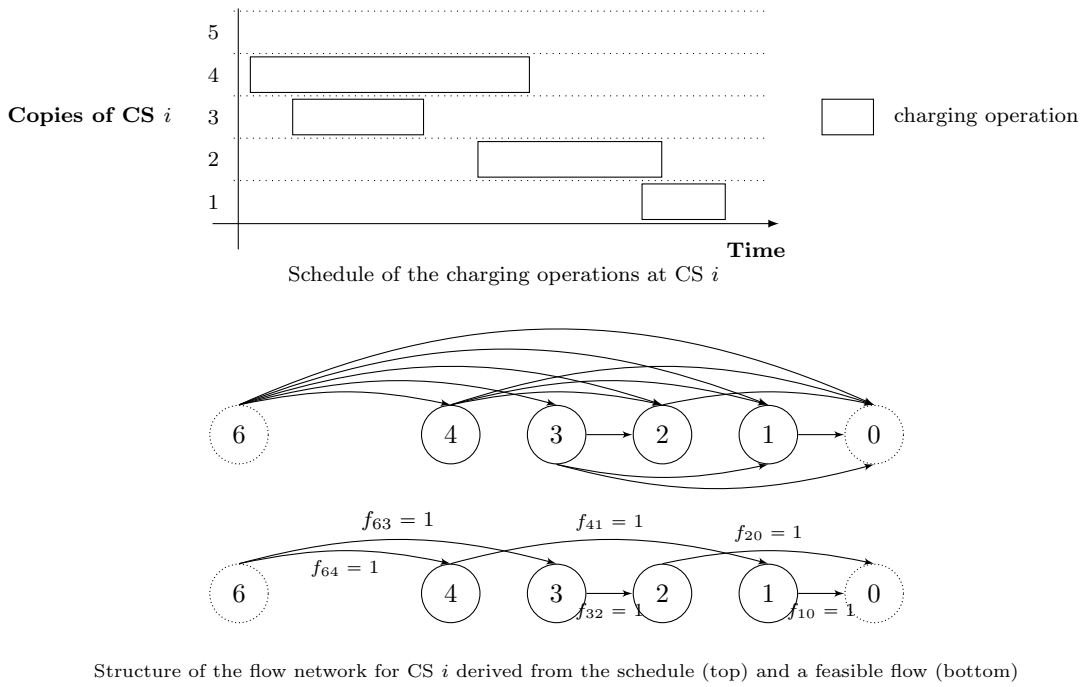


Figure 2: Structure of the flow network for CS i.



Structure of the flow network for CS i derived from the schedule (top) and a feasible flow (bottom)

Figure 3: Illustration of the flow-based formulation modeling approach for a given schedule at CS i

Building on this concept, the CS replication-based formulation for the E-VRP-NL, denoted as $[F^{CS_{rep}}]$, is as follows:

$$[F^{CS_{rep}}] \quad \text{minimize} \sum_{(i,j)\in A} t_{ij}x_{ij} + \sum_{i\in F'} (\Delta_i + \nabla_i) \tag{1}$$

$$\text{subject to} \sum_{(i,l)\in A} x_{il} = 1 \quad i \in I \tag{2}$$

$$\sum_{(j,l)\in A} x_{jl} \leqslant 1 \quad j \in F' \tag{3}$$

$$\sum_{(l,i)\in A} x_{li} - \sum_{(i,l)\in A} x_{il} = 0 \quad i \in V \tag{4}$$

$$y_{0l} = Qx_{0l} \quad (0,l) \in A \tag{5}$$

$$\sum_{(l,i)\in A} y_{li} - \sum_{(l,i)\in A} e_{lj}x_{li} = \sum_{(i,l)\in A} y_{il} \quad i \in I \tag{6}$$

$$\sum_{(l,j)\in A} y_{lj} - \sum_{(l,j)\in A} e_{lj}x_{lj} = \underline{q}_j \quad j \in F' \tag{7}$$

$$\sum_{(j,l)\in A} y_{jl} = \overline{q}_j \quad j \in F' \tag{8}$$

$$y_{ij} \leqslant \left(Q - \min_{l\in F\cup\{0\}} \{e_{li}\}\right) x_{ij} \quad (i,j) \in A \tag{9}$$

$$y_{ij} \geqslant \left(e_{ij} + \min_{l\in F\cup\{0\}} \{e_{jl}\}\right) x_{ij} \quad (i,j) \in A \tag{10}$$

$$\underline{q}_j \leqslant \overline{q}_j \quad j \in F' \tag{11}$$

$$\underline{q}_j = \sum_{k\in B_j} \underline{\lambda}_{jk} q_{jk} \quad j \in F' \tag{12}$$

$$\underline{a}_j = \sum_{k\in B_j} \underline{\lambda}_{jk} a_{jk} \quad j \in F' \tag{13}$$

$$\sum_{k\in B_j} \underline{\lambda}_{jk} = \sum_{k\in B_j\setminus\{0\}} \underline{w}_{jk} \quad j \in F' \tag{14}$$

$$\sum_{k\in B_j\setminus\{0\}} \underline{w}_{jk} = \sum_{(j,l)\in A} x_{jl} \quad j \in F' \tag{15}$$

$$\underline{\lambda}_{j0} \leqslant \underline{w}_{j1} \quad j \in F' \tag{16}$$

$$\underline{\lambda}_{jk} \leqslant \underline{w}_{jk} + \underline{w}_{j,k+1} \quad j \in F', k \in B_j \backslash \{0, b_j\} \tag{17}$$

$$\underline{\lambda}_{jb_j} \leqslant \underline{w}_{jb_j} \quad j \in F' \tag{18}$$

$$\overline{q}_j = \sum_{k \in B_j} \overline{\lambda}_{jk} q_{jk} \quad j \in F' \tag{19}$$

$$\overline{a}_j = \sum_{k \in B_j} \overline{\lambda}_{jk} a_{jk} \quad j \in F' \tag{20}$$

$$\sum_{k \in B_j} \overline{\lambda}_{jk} = \sum_{k \in B_j \backslash \{0\}} \overline{w}_{jk} \quad j \in F' \tag{21}$$

$$\sum_{k \in B_j \backslash \{0\}} \overline{w}_{jk} = \sum_{(j,l) \in A} x_{jl} \quad j \in F' \tag{22}$$

$$\overline{\lambda}_{j0} \leqslant \overline{w}_{j1} \quad j \in F' \tag{23}$$

$$\overline{\lambda}_{jk} \leqslant \overline{w}_{jk} + \overline{w}_{j,k+1} \quad j \in F', k \in B_j \backslash \{0, b_j\} \tag{24}$$

$$\overline{\lambda}_{jb_j} \leqslant \overline{w}_{jb_j} \quad j \in F' \tag{25}$$

$$\Delta_j = \overline{a}_j - \underline{a}_j \quad j \in F' \tag{26}$$

$$\sum_{(l,i) \in A} (\tau_{li} + (t_{li} + g_i)\, x_{li}) = \sum_{(i,l) \in A} \tau_{il} \quad i \in I \tag{27}$$

$$\sum_{(l,j) \in A} (\tau_{lj} + t_{lj} x_{lj}) + \Delta_j + \nabla_j = \sum_{(j,l) \in A} \tau_{jl} \quad j \in F' \tag{28}$$

$$\tau_{li} \leqslant (T_{max} - t_{li} - g_i - t_{i0})\, x_{li} \quad (l,i) \in A, i \in I \tag{29}$$

$$\tau_{lj} \leqslant \left(T_{max} - t_{lj} - \Delta_j^{min} - t_{j0}\right) x_{lj} \quad (l,j) \in A, j \in F' \tag{30}$$

$$\sum_{(h,j) \in A} x_{hj} \leqslant \sum_{(h,l) \in A} x_{hl}, \quad i \in F, j, l \in F'_i, j < l \tag{31}$$

$$\sum_{(j,h) \in A} \tau_{jh} - \Delta_j \geqslant \sum_{(l,h) \in A} \tau_{lh} - \Delta_l \quad i \in F, j, l \in F'_i : j < l \tag{32}$$

$$\sum_{l \in \widetilde{F'}_i, l > j} f_{lj} = \sum_{(l,j) \in A} x_{lj} \quad i \in F, j \in F'_i \tag{33}$$

$$\sum_{l \in \widetilde{F'}_i, l > j} f_{lj} - \sum_{l \in \widetilde{F'}_i, l < j} f_{jl} = 0 \quad i \in F, j \in F'_i \tag{34}$$

$$C_i - \sum_{l \in \widetilde{F'}_i, l > 0_i} f_{l,0_i} = 0 \quad i \in F \tag{35}$$

$$\sum_{l \in \widetilde{F'}_i, l < \beta_i + 1} f_{\beta_i + 1, l} - C_i = 0 \quad i \in F \tag{36}$$

$$u_{jh} \geqslant u_{jl} + u_{lh} - 1 \quad i \in F, j, l, h \in F'_i : j > l > h \tag{37}$$

$$\sum_{(l,h) \in A} \tau_{lh} - \Delta_l - \sum_{(j,h) \in A} \tau_{jh} \geqslant (T_{max} - t_{i0})\,(u_{jl} - 1) \quad i \in F, j, l \in F'_i, j > l \tag{38}$$

$$f_{jl} \leqslant \min(\widetilde{C}_j, \widetilde{C}_l) u_{jl} \quad i \in F, (j,l) \in F'_i, j > l \tag{39}$$

$$x_{ij} \in \{0,1\}, \tau_{ij} \geqslant 0, y_{ij} \geqslant 0 \quad (i,j) \in A \tag{40}$$

$$\underline{w}_{jk} \in \{0,1\}, \overline{w}_{jk} \in \{0,1\} \quad j \in F', k \in B_j \backslash \{0\} \tag{41}$$

$$\underline{\lambda}_{jk} \geqslant 0, \overline{\lambda}_{jk} \geqslant 0 \quad j \in F', k \in B_j \tag{42}$$

$$\underline{a}_j \geqslant 0, \overline{a}_j \geqslant 0, \underline{q}_j \geqslant 0, \overline{q}_j \geqslant 0, \Delta_j \geqslant 0, \nabla_j \geqslant 0 \quad j \in F' \tag{43}$$

$$u_{jl} \in \{0,1\} \quad i \in F, j, l \in F'_i, j > l \tag{44}$$

$$f_{jl} \geqslant 0 \quad i \in F, j, l \in \widetilde{F'}_i, j > l. \tag{45}$$

The objective function (1) includes the driving, charging, and waiting times. Constraints (2) ensure that each customer is visited exactly once, while constraints (3) ensure that each CS copy is visited at most once. Constraints (4) impose the flow conservation. Constraints (5) state that the EVs leave the depot fully charged. Constraints (6) track the SoC at each customer. Constraints (7) track the SoC of the EV when it arrives at a CS copy. Constraints (8) track the SoC of the EV when it leaves a CS copy. Constraints (9) couple the y_{ij} and x_{ij} variables. Constraints (10) state that if an

EV traverses the arc (i, j) its SoC when leaving i must be enough to traverse the arc and then to reach the closest CS or the depot. Constraints (11) couple the SoC of an EV upon arrival at a CS with its SoC at departure. Constraints (12)–(18) define the SoC and the corresponding charging time with respect to ϕ_j of an EV upon arrival at CS copy j. Specifically, we express $(\underline{a}_j, \underline{q}_j)$ as a convex combination of the breakpoints (a_{jk}, q_{jk}) using the $\underline{\lambda}_{jk}$ variables, where $k \in B_j$. Similarly, constraints (19)–(25) define the SoC and the corresponding charging time with respect to ϕ_j of an EV upon departure from CS copy j. Specifically, we express $(\overline{a}_j, \overline{q}_j)$ as a convex combination of the breakpoints (a_{jk}, q_{jk}) using the $\overline{\lambda}_{jk}$ variables, where $k \in B_j$. Constraints (26) define the time spent at a CS copy. Constraints (27) track the departure time at each customer. Constraints (28) track the departure time at CS copies. Constraints (29) and (30) couple the τ_{ij} and x_{ij} variables. Specifically, if an EV traverses an arc (i, j), then its departure time must guarantee that the EV returns to the depot before the closing time. The parameter Δ_j^{min} is a lower bound on the time spent charging to recover the energy consumed to make the detour to $j \in F'$. It is equal to the time needed to charge the energy quantity $\min_{l,l' \in V \setminus \{j\}: (l,j) \in A \wedge (j,l') \in A} \{e_{lj} + e_{jl'} - e_{ll'}\}$ if the EV has an empty battery. Constraints (31) and (32) break the symmetries created by the introduction of the CS copies. Constraints (33) state that a resource has to be allocated to a charging operation in F_i' if an EV reaches the corresponding CS. Constraints (34)–(36) ensure the flow conservation for the chargers. Constraints (37) express the transitivity of the precedence relations. Constraints (38) are the disjunctive constraints coupling the start time of charging operations j and l to u_{jl}. For a pair (j, l), the constraint is active when $u_{jl} = 1$ and, in this case, it enforces the precedence relation between the charging operations j and l (i.e., l cannot start before the completion of j). Constraints (39) couple the flow variables for the chargers to the charging operation sequence variables. Finally, constraints (40)–(45) define the domains of the decision variables.

3.2 Path-based formulation

One drawback of the previous formulation is the need to replicate the CSs. To ensure that no optimal solution is cut off, the number of needed copies is very large (Froger et al. (2019) provide an example where this value is equal to $4|I|$), which yields intractable MILPs. To overcome this difficulty, Froger et al. (2019) proposed an alternative model of for the E-VRP-NL based on the concept of CS paths between each couple of nodes (either customers or the depot). Given two nodes (customer or depot) i and j, we call a CS path (CSP) a simple path starting from i visiting a sequence of CSs or none and ending at j.

The concept of CSPs leads to a redefinition of the problem on a directed multigraph $\widetilde{G} = (\widetilde{V}, \widetilde{A})$, where $\widetilde{V} = \{0\} \cup I$, and $\widetilde{A}$ is the set of arcs connecting the nodes of $\widetilde{V}$. More specifically, an arc in $\widetilde{A}$ represents a CSP p, starting in $\texttt{org}(p) \in \widetilde{V}$ and ending in $\texttt{dest}(p) \in \widetilde{V}$. We denote n_p the number of CSs in p and $L_p = \{0, \cdots, n_p - 1\}$ as the ordered set of CS positions in p. We note that $L_p = \varnothing$ indicates that p does not visit any CS between $\texttt{org}(p)$ and $\texttt{dest}(p)$. When $L_p \neq \varnothing$, we denote $\mu_p(l)$ the CS at position $l \in L_p$. We define e^p and t^p as the energy consumption and the driving time associated with CSP $p \in P$. Given two nodes $i, j \in \widetilde{V}$, we define P_{ij} as the set of CSPs connecting i to j, and we define $P = \bigcup_{i,j \in \widetilde{V}, i \neq j} P_{ij}$ as the set of all CSPs.

Our path-based formulation of the E-VRP-NL-C involves the following decisions variables. The binary variable x_p is 1 if and only if an EV travels CSP $p \in P$. The continuous variables τ_p and y_p track the time and SoC of an EV when it departs from node $\texttt{org}(p)$ to $\texttt{dest}(p)$ using CSP p. We associate with each CS in a CSP of set P a unique potential charging operation (a CSP can at most be traveled by a single EV). The continuous variables $\underline{q}_{pl}$ and $\overline{q}_{pl}$ specify the SoC of an EV when it enters and leaves $\mu_p(l)$ (i.e., the CS at position $l \in L_p$). The continuous variable Δ_{pl} represents the duration of the charging operation performed at $\mu_p(l)$. The variables $\underline{a}_{pl}$ and $\overline{a}_{pl}$ are the scaled arrival and departure times, according to the charging function of CS $\mu_p(l)$. For $k \in B_i \setminus \{0\}$, the binary variables $\underline{w}_{plk}$ and $\overline{w}_{plk}$ are equal to 1 if and only if the SoC is between $q_{\mu_p(l),k-1}$ and $q_{\mu_p(l),k}$ when the EV enters and leaves CS $\mu_p(l)$, respectively. The continuous variables $\underline{\lambda}_{plk}$ and $\overline{\lambda}_{plk}$ represent the coefficients associated with the breakpoints $(a_{\mu_p(l),k}, q_{\mu_p(l),k})$ of the piecewise linear charging function, when the EV enters and leaves CS $\mu_p(l)$, respectively. Let e^p and t^p be the energy consumption and the driving time associated with CSP $p \in P$. The continuous variable ∇_{pl} is the waiting time at the CS at position l in CSP p before charging. We also need to track the starting and completion time of every charging operation. The continuous variables $\underline{s}_{pl}$ and $\overline{s}_{pl}$ represent the starting and completion time of the charging operation performed at $\mu_p(l)$.

The modeling of the capacity constraints follows the same idea as the one described in §3.1. For convenience, we introduce the set O_i of potential charging operations at CS $i \in F$. Every operation $o \in O_i$ represents the visit of a

specific CS of a CSP of P. We denote by $p(o) \in P$ the corresponding CSP and by $l(o) \in L_{p(o)}$ the position of the visited CS. Specifically, an operation o is associated with CS $\mu_{p(o)}(l(o))$. Binary variable $u_{oo'}$ is equal to 1 if and only if operation o' starts after the completion of operation $o \neq o'$. For every CS $i \in F$, we introduce two dummy operations ε_i^+ and ε_i^-, which act as the source and the sink of the flow. Let $(o, o') \in \big(O_i \cup \{\varepsilon_i^+\}\big) \times \big(O_i \cup \{\varepsilon_i^-\}\big)$ be a couple of charging operations; then the continuous flow variable $f_{oo'}$ denotes the number of chargers that are transferred from operation o to operation o'. When both these operations are not dummy, $f_{oo'}$ is equal to one if and only if these operations are scheduled on the same charger, o' is scheduled after o, and no other operation is scheduled on the charger between the completion of o and the beginning of o'. For notational convenience, we define $\widetilde{C}_o := 1$ for all $o \in O_i$ and $\widetilde{C}_{\varepsilon_i^+} := \widetilde{C}_{\varepsilon_i^-} := C_i$. The second model for the E-VRP-NL-C, denoted as $[P^{path}]$, is as follows:

$$[F^{path}] \quad \text{minimize} \sum_{p \in P} \left(t^p x_p + \sum_{l \in L_p} (\Delta_{pl} + \nabla_{pl}) \right) \tag{46}$$

$$\text{subject to} \quad \sum_{j \in \widetilde{V}, i \neq j} \sum_{p \in P_{ij}} x_p = 1 \quad i \in I \tag{47}$$

$$\sum_{j \in \widetilde{V}, i \neq j} \sum_{p \in P_{ji}} x_p - \sum_{j \in \widetilde{V}, i \neq j} \sum_{p \in P_{ij}} x_p = 0 \quad i \in \widetilde{V} \tag{48}$$

$$\sum_{l \in \widetilde{V}, l \neq j} \sum_{p \in P_{lj}} \left(y_p - e^p x_p + \sum_{l \in L_p} (\overline{q}_{pl} - \underline{q}_{pl}) \right) = \sum_{l \in \widetilde{V}, l \neq j} \sum_{p \in P_{jl}} y_p \quad j \in I \tag{49}$$

$$y_p - e_{\mathtt{org}(p), \mu_p(0)} x_p = \underline{q}_{p0} \quad p \in P \tag{50}$$

$$\overline{q}_{p,l-1} - e_{\mu_p(l-1), \mu_p(l)} x_p = \underline{q}_{pl} \quad p \in P, l \in L_p \backslash \{0\} \tag{51}$$

$$y_p - e^p x_p - \sum_{l \in L_p} (o_{pl} - q_{pl}) \geqslant 0 \quad i \in I, p \in P_{i0} \tag{52}$$

$$y_p = Q x_p \quad i \in \widetilde{V} \backslash \{0\}, p \in P_{0i} \tag{53}$$

$$y_p \leqslant Q x_p \quad p \in P \tag{54}$$

$$\underline{q}_{pl} = \sum_{k \in B_{\mu_p(l)}} \underline{\lambda}_{plk} q_{\mu_p(l)k} \quad p \in P, l \in L_p \tag{55}$$

$$\underline{a}_{pl} = \sum_{k \in B_{\mu_p(l)}} \underline{\lambda}_{plk} a_{\mu_p(l)k} \quad p \in P, l \in L_p \tag{56}$$

$$\sum_{k \in B_{\mu_p(l)}} \underline{\lambda}_{plk} = \sum_{k \in B_{\mu_p(l)} \backslash \{0\}} \underline{w}_{plk} \quad p \in P, l \in L_p \tag{57}$$

$$\sum_{k \in B_{\mu_p(l)} \backslash \{0\}} \underline{w}_{plk} = x_p \quad p \in P, l \in L_p \tag{58}$$

$$\underline{\lambda}_{pl0} \leqslant \underline{w}_{pl1} \quad p \in P, l \in L_p \tag{59}$$

$$\underline{\lambda}_{plk} \leqslant \underline{w}_{plk} + \underline{w}_{pl,k+1} \quad p \in P, l \in L_p, k \in B_{\mu_p(l)} \backslash \{0, b_{\mu_p(l)}\} \tag{60}$$

$$\underline{\lambda}_{plb_{\mu_p(l)}} \leqslant \underline{w}_{plb_{\mu_p(l)}} \quad p \in P, l \in L_p \tag{61}$$

$$\overline{q}_{pl} = \sum_{k \in B_{\mu_p(l)}} \overline{\lambda}_{plk} q_{\mu_p(l)k} \quad p \in P, l \in L_p \tag{62}$$

$$\overline{a}_{pl} = \sum_{k \in B_{\mu_p(l)}} \overline{\lambda}_{plk} a_{\mu_p(l)k} \quad p \in P, l \in L_p \tag{63}$$

$$\sum_{k \in B_{\mu_p(l)}} \overline{\lambda}_{plk} = \sum_{k \in B_{\mu_p(l)} \backslash \{0\}} \overline{w}_{plk} \quad p \in P, l \in L_p \tag{64}$$

$$\sum_{k \in B_{\mu_p(l)} \backslash \{0\}} \overline{w}_{plk} = x_p \quad p \in P, l \in L_p \tag{65}$$

$$\overline{\lambda}_{i0} \leqslant \overline{w}_{pl1} \quad p \in P, l \in L_p \tag{66}$$

$$\overline{\lambda}_{plk} \leqslant \overline{w}_{plk} + \overline{w}_{pl,k+1} \quad p \in P, l \in L_p, k \in B_{\mu_p(l)} \backslash \{0, b_{\mu_p(l)}\} \tag{67}$$

$$\overline{\lambda}_{plb_{\mu_p(l)}} \leqslant \overline{w}_{plb_{\mu_p(l)}} \quad p \in P, l \in L_p \tag{68}$$

$$\Delta_{pl} = \overline{a}_{pl} - \underline{a}_{pl} \quad p \in P, l \in L_p \tag{69}$$

$$\sum_{i\in V, i\neq j}\sum_{p\in P_{ij}}\left(\tau_p + t^p x_p + \sum_{l\in L_p}(\Delta_{pl}+\nabla_{pl})\right) + g_j = \sum_{l\in \widetilde{V}, l\neq j}\sum_{p\in P_{jl}} \tau_p \quad j\in I \tag{70}$$

$$\tau_p + \sum_{l\in L_p}(\Delta_{pl}+\nabla_{pl}) \leqslant \left(T_{max} - t^p - p_{\mathtt{dest}(p)} - t_{\mathtt{dest}(p),0}\right) x_p \quad p\in P \tag{71}$$

$$\tau_p + t_{\mathtt{org}(p),\mu_p(0)} x_p + \nabla_{p0} = \underline{s}_{p0} \quad p\in P \tag{72}$$

$$\overline{s}_{p,l-1} + t_{\mu_p(l-1),\mu_p(l)} x_p + \nabla_{pl} = \underline{s}_{pl} \quad p\in P, l\in L_p, l\neq 0 \tag{73}$$

$$\Delta_{pl} = \overline{s}_{pl} - \underline{s}_{pl} \quad p\in P, l\in L_p \tag{74}$$

$$\sum_{o'\in O_i\cup\{\varepsilon_i^+\}} f_{o'o} = x_{p(o)} \quad i\in F, o\in O_i \tag{75}$$

$$\sum_{o'\in O_i\cup\{\varepsilon_i^+\}} f_{o'o} - \sum_{o'\in O_i\cup\{\varepsilon_i^-\}} f_{oo'} = 0 \quad i\in F, o\in O_i \tag{76}$$

$$C_i - \sum_{o\in O_i\cup\{\varepsilon_i^-\}} f_{\varepsilon_i^+,o} = 0 \quad i\in F \tag{77}$$

$$\sum_{o\in O_i\cup\{\varepsilon_i^+\}} f_{o,\varepsilon_i^-} - C_i = 0 \quad i\in F \tag{78}$$

$$\underline{s}_{p(o),l(o)} - \overline{s}_{p(o'),l(o')} \geqslant T_{max}\left(u_{o'o} - 1\right) \quad i\in F, o, o'\in O_i \tag{79}$$

$$f_{oo'} \leqslant \min(\widetilde{C}_o, \widetilde{C}_{o'}) u_{oo'} \quad i\in F, (o,o')\in \left(O_i\cup\{\varepsilon_i^+\}\right)\times\left(O_i\cup\{\varepsilon_i^-\}\right) \tag{80}$$

$$x_p\in\{0,1\} \quad p\in P \tag{81}$$

$$\tau_p \geqslant 0, y_p \geqslant 0 \quad p\in P \tag{82}$$

$$\underline{q}_{pl}, \overline{q}_{pl}, \underline{a}_{pl}, \overline{a}_{pl}, \underline{s}_{pl}, \overline{s}_{pl}, \Delta_{pl}, \nabla_{pl} \geqslant 0 \quad p\in P, l\in L_p \tag{83}$$

$$\underline{\lambda}_{plk} \geqslant 0, \overline{\lambda}_{plk} \geqslant 0 \quad p\in P, l\in L_p, k\in B_{\mu_p(l)}\backslash\{0\} \tag{84}$$

$$\underline{w}_{plk}\in\{0,1\}, \overline{w}_{plk}\in\{0,1\} \quad p\in P, l\in L_p, k\in B_{i(p,l)}\backslash\{0\} \tag{85}$$

$$u_{oo'}\in\{0,1\} \quad i\in F, o, o'\in O_i \tag{86}$$

$$f_{oo'} \geqslant 0 \quad i\in F, (o,o')\in\left(O_i\cup\{\varepsilon_i^+\}\right)\times\left(O_i\cup\{\varepsilon_i^-\}\right). \tag{87}$$

The objective function (46) minimizes the total driving, charging, and waiting time. Constraints (47) ensure that each customer is visited exactly once. Constraints (48) impose flow conservation. Constraints (49) track the SoC of EVs at each customer location. Constraints (50) track the SoC at the arrival at the first CS of each CSP. Constraints (51) couple the SoC of an EV that leaves a CS to go to another CS. Constraints (52) ensure that if the EV travels between a vertex and the depot, it has sufficient energy to reach its destination. Constraints (53) state that every EV leaves the depot with a fully charged battery. Constraints (54) couple the SoC tracking variable with the arc travel variables. Constraints (55)–(69) model the relationship between the SoC of an EV upon arrival at a CS, the charging time, and its SoC upon departing from this CS. Constraints (70) track the departure time at each vertex. Constraints (71) couple the time tracking variable with the arc travel variables, and impose the route duration limit. Constraints (72) and (73) define the starting and completion times of every charging operation, as well as the potential waiting time before the start of the operation. Constraints (74) define the duration of the charging operations based on their starting and completion times. Constraints (75) state that a charger has to be allocated to every charging operation of each selected CSP. Constraints (76) ensure flow conservation. Constraints (77) and (78) compute the flow value at the beginning and at the end of the time horizon. Constraints (79) couple the sequencing variables with the starting time of the corresponding charging operations. Constraints (80) couple the flow variables with the sequence variables. Specifically, a charger can be sent from a charging operation o' to another charging operation o if o starts after the completion of o'. Finally, constraints (81)–(87) define the domains of the decision variables.

Without preprocessing, the number of CSPs explodes with the number of CSs and the number of customers. However, a large number of these arcs cannot be part of an optimal solution. Froger et al. (2019) presented a filtering procedure to reduce the number of CSPs in the path-based formulation based on the definition of a dominance rule between two CSPs having the same origin and destination. Due to the potential waiting times that can occur at CSs, we cannot apply the dominance rule described in the work of Froger et al. (2019) between CSPs with the same origin and destination if they both contain CSs. However, in our computational experiments we apply the dominance rule in the special case

between the unique CSP with no CS and every other CSP of P_{ij}, with i and j in $\widetilde{V}$.

4 Solution method

The results obtained by Froger et al. (2019) suggest that the models introduced in §3.1 are only applicable to small-size instances (around 20 customers). In this section, we introduce a matheuristic to solve the E-VRP-NL-C. The method relies on two interacting components: a route generator and a solution assembler. The first component builds a pool of high-quality solutions by relaxing the CS capacity constraints, while the latter component recombines routes from the pool trying to build a solution satisfying the CS capacity constraints. Algorithm 1 outlines the general structure of the method. The algorithms starts by generating an initial solution without taking into account the capacity constraints (line 1). In our implementation, this step is carried out using a modified version of the classical Clarke and Wright heuristic. Next, the algorithm enters the main loop (lines 3–28). During n_{max} iterations the algorithm alternates between the *route generation* and *solution assembly* phases. In particular, the algorithm uses procedure `generateRoutes(·)` to retrieve a triplet $(\Phi_O, \Phi_R, \Omega')$, where Φ_O, Φ_R are sets of high-quality solutions for the *original* and *relaxed* problems (i.e., with and without CS capacity constraints), and Ω' is a set of independent, feasible, and high-quality routes for the original problem. The latter will be referred to as the *short-term* pool. Next (lines 4–12) the algorithm adds the routes making up the solutions in Φ_O and Φ_R to a *long-term* pool Ω while keeping track of the best solutions for both the original (s_O^*) and the relaxed problem (s_R^*). The intuition behind adding routes coming from potentially infeasible solutions to the original problem (i.e., solutions to the relaxed problem) to the long-term pool is to foster diversity. Indeed, these routes tend to be of high quality (in terms of the objective function) and might be recombined efficiently with others later.

The algorithm then enters the assembly phase (lines 13–14). In the first step of this phase, the algorithm merges the short- and long-term pools. Thus, seeking to combine the past and recent history of the search into a single set of routes that has a size that is manageable for the solution assembler. The algorithm then calls procedure `assemble(·)` to retrieve a tuple (s', S'), where s' is the best solution and S' is the set of all improving solutions (with respect to s_O^*) for the original problem found during the call. If the assembler retrieves a solution, the algorithm adds its routes in S' to the long-term pool Ω and updates the incumbent s_O^* (lines 15–18). If after the call to the assembler, the algorithm still has not found a feasible solution (i.e., s_O^* is still equal to `null`), it implements two actions. First, it slightly modifies the search space to favor feasibility. In our implementation, we modify the solution space by artificially reducing the opening hours of the depot (line 21). The underlying idea, is that the reduction of the depot's operating hours would lead to shorter routes with more slack to accommodate waiting times at charging stations during the assembly phase. Second, it tries to *repair* the best-known solution for the relaxed problem (i.e., s_R^*) by making it comply with the new solution space. In our implementation, if a route visiting n customers has a duration strictly greater than the new value T of the depot hours, the algorithm creates two new routes by adding a return to the depot after the $\lfloor n/2 \rfloor^{th}$ customer and reoptimizes the charging decisions within these routes. To avoid feasibility issues, the new route maximum duration limit is not considered when a route visits only one customer. The same procedure is performed on the newly created routes as long as the routes contain more than one customer and their duration exceeds T. On the other hand, if after the call to the assembler a feasible solution is available (i.e., s_O^* is different than `null`), the algorithm then reestablishes (if needed) the original solution space (i.e., resets the value of the depot opening hours to T_{max}) and moves to a new iteration.

Both the route generation and solution assembly phases consist in solving complex combinatorial optimisation problems. In our framework, we develop an iterative local search algorithm as the route generator and a branch-and-cut procedure as the solution assembler. The remainder of this section describes these two components.

4.1 Route generator: an iterated local search algorithm for the E-VRP-NL

Local search (LS) techniques have been commonly used to solve E-VRPs. In this respect, researchers have devoted substantial effort in finding move evaluation strategies that offer the best trade-off between accuracy and efficiency. A first approach relies on LS operators that focus on customers or CSs, more or less indistinctively, while other approaches focus on one type of node (i.e., either customers or CSs) while not modifying the positioning of the others ones. Typically, given the resulting sequence of customers and CSs, the quantity to be charged at each CS is determined. The feasibility and the evaluation of a move is therefore a result of this process. While this has the advantage of

Algorithm 1: Solution method - general structure

```
/* f(s) denotes the value of the objective function for a solution s and we assume that f(NULL) = +∞*/
s_R^0 ←generateInitialSolution()
n ← 0, T ← T_max, Ω ← ∅, s_R^* ← s_R^0, s_R ← s_R^0, s_FO^* ← NULL, s_O ← NULL
while n < n_max do
    (Φ_O, Φ_R, Ω') ←generateRoutes(s_R,s_O,δ,T)    (see Algorithm 2)
    for each s'_R ∈ Φ_R do
        addRoutesToPool(Ω,s'_R)
        if f(s'_R) < f(s_R^*) then  s_R^* ← s'_R
    end
    for each s'_O ∈ Φ_O do
        addRoutesToPool(Ω,s'_O)
        if f(s'_O) < f(s_O^*) then  s_O^* ← s'_O
    end
    Ω' ← merge(Ω, Ω')
    (s', S') ← assemble(Ω', s_O^*)    (see Algorithm 4) /* s' = NULL if no improving solution is found */
    if s' ≠ NULL then
        s_O^* ← s'
        if f(s') < f(s_R^*) then  s_R^* ← s'
        for each s ∈ S' do  addRoutesToPool(Ω,s)
    end
    if s_O^* = NULL then
        T ← max{T_min, α · T} /* T_min : minimum possible value for T, α < 1 a tuning parameter */
        s_R ← repair(s_R^*), s_O ← NULL
    else
        if T < T_max then  T ← T_max
        s_O ← s_O^*, s_R ← s_R^*
    end
    n ← n + 1
end
return s_O^*
```

requiring a low computational effort, it can have undesired effects. For example, a move may be deemed infeasible or non-improving, while the resulting sequence(s) of customer visits can lead to an excellent solution if the charging decisions are reoptimized, in terms of which CSs to visit, when to visit them and how much to charge the EV during these visits.

Most of the LS-based algorithms designed for E-VRPs use the aforementioned approaches (e.g., Felipe et al. (2014); Schneider et al. (2014); Goeke and Schneider (2015); Koç and Karaoglan (2016)). In some cases, more advanced strategies are also called periodically to improve the charging decisions inside the routes, by means of heuristic or exact procedures, or a combination of both (Hiermann et al. (2016); Montoya et al. (2017)). Another approach couples the charging decisions (in terms of which CSs to visit, when to visit them and how much to charge the EV during these visits) with the evaluation of the customer sequencing moves. Specifically, it relies on LS operators that solely function on customers, while the charging decisions are only made to check the feasibility of a move and to evaluate it. This coupling approach comes with a computational burden. To avoid the need to explicitly model CSs, Andelmin and Bartolini (2019) designed for the Green VRP a multi-start local search heuristic that works on a multigraph (where each arc represents a CSP). Nonetheless, their solution method is only halfway between the previously described approaches for two reasons. First, vehicles are fully charged when visiting a CS, and thus no optimisation of the recharging quantities is performed. Second, they do not necessarily guarantee that the charging decisions inside the routes are optimal. To our knowledge, only Hiermann et al. (2019) designed an LS-based algorithm based on the coupling approach. Specifically, they used a dynamic programming procedure to determine for each single route the CS visits and greedy policies to decide the propulsion mode for a plug-in hybrid electric vehicle. They call this procedure every time they evaluate a move.

The previous discussion shows that charging decisions should be handled with care in LS-based algorithms for E-VRPs. In the E-VRP-NL-C, given a set of routes without CSs, making optimal charging decisions consists in defining

for every route where, when and how much to charge in order to minimize the total time to visit the customers while meeting the CS capacity constraints. Therefore, the charging decisions of the routes need to be coupled, thus adding another complicating decisional layer. As a consequence, solving this charging problem whenever an LS-based algorithm evaluates a move is computationally demanding for the E-VRP-NL-C. However, for the vast majority of E-VRPs this computational burden can be reduced since charging decisions can be independently optimized for each route (defined here as a sequence of customer visits). This optimisation problem was explicitly handled by Montoya et al. (2017) and Froger et al. (2019) in the context of the E-VRP-NL. The authors referred to this problem as the fixed route vehicle charging problem (FRVCP).

We have developed a novel route generator, which is based on an ILS algorithm that solves the E-VRP-NL, while populating a pool of routes to be used in constructing solutions to the E-VRP-NL-C. In particular, we considered only customer nodes in the local search, and we used the labeling algorithm of Froger et al. (2019) to tackle the FRVCP, as their results suggest that it can be embedded within heuristics.

Introduced by Lourenço et al. (2003), ILS is a metaheuristic that iteratively applies an LS phase to produce local optima, and perturbation mechanisms to escape from them. It is initialized with a solution generally provided by a constructive heuristic. In our implementation, we combine ILS with a variable neighborhood descent (VND) search strategy for the LS phase. Algorithm 2 outlines the general structure of our method. First, the current best solution s_R^* is set equal to the initial solution s_R^0 provided as input. The algorithm then enters an iterative process. Except during the first iteration, it perturbs the current best solution s_R^* to escape from the current local optimum and potentially explore a new region of the search space (see §4.1.2). This produces a new start point s'_R for the VND which computes a new local optimum s_R to the E-VRP-NL (see §4.1.1) and also returns the best solution s_O to the E-VRP-NL-C it has encountered. Note that we only check the capacity constraints after accepting a move (i.e., when building a new solution to the E-VRP-NL). If appropriate, the algorithm updates the best-known solutions s_R^* and s_O^*, as well as the sets Φ_R and Φ_O of improving solutions. It also populates a pool of routes Ω. Note that we only add a route to Ω if it does not already contains a route visiting the same sequence of nodes. This procedure is reiterated until the targeted number of iterations has been reached. The optimization returns the sets of improving solutions to the E-VRP-NL and E-VRP-NL-C and the generated pool of routes Ω. To speed up the ILS algorithm, its implementation is based on the static move descriptor (SMD) concept introduced by Zachariadis and Kiranoudis (2010) which prevents unnecessary reevaluations of moves and provides an efficient way of exploring neighborhoods (see Appendix B).

Algorithm 2: The ILS algorithm

Input : a solution s_R^0 to the E-VRP-NL, a solution s_O^0 to the E-VRP-NL-C (possibly equal to `NULL`), a maximum number of iterations δ^{max}, and a maximum route duration limit T

Output: a set Φ_R and a set Φ_O of improving solutions to the E-VRP-NL and to the E-VRP-NL-C, a pool of routes Ω

Procedure `generateRoutes`(s_R^0,s_O^0,δ^{max},T):
  `/* We denote` $f(s)$ `the value of the objective function for a solution` s `and we assume` $f(\texttt{NULL}) = +\infty$`*/`
  $\delta \leftarrow 0$, $\Phi_R \leftarrow \emptyset$, $\Phi_O \leftarrow \emptyset$, $\Omega \leftarrow \emptyset$
  $s_R^* \leftarrow s_R^0$, $s_O^* \leftarrow s_O^0$
  while $\delta < \delta^{max}$ **do**
    if $\delta = 0$ **then** $s'_R \leftarrow s_R^0$
    else $s'_R \leftarrow$`perturb`(s_R^*,T) (see §4.1.2)
    $(s_R, s_O) \leftarrow$ `VND`(s'_R,T) (see Algorithm 5)
    `addRoutesToPool`(Ω,s_R)
    if $f(s_R) < f(s_R^*)$ **then** $\Phi_R \leftarrow \Phi_R \cup \{s_R\}$, $s_R^* \leftarrow s_R$
    if $f(s_O) < f(s_O^*)$ **then** $\Phi_O \leftarrow \Phi_O \cup \{s_O\}$, $s_O^* \leftarrow s_O$
    $\delta \leftarrow \delta + 1$
  end
  return (Φ_R, Φ_O, Ω)

4.1.1 The VND search phase

The VND relies on an ordered list of LS operators. A single operator is considered at a time. If an improving move is found, the search restarts with the first operator of the list. Otherwise, it moves to the next operator. The search

reaches a local optimum when the last operator fails to improve the current solution.

Our VND employs several classical VRP operators focusing on sequencing decisions. These operators are defined for solutions represented as sequences of customer visits without CSs. We use five vertex exchanges operators that work by relocating or exchanging customer visits : 1-0, 2-0, 1-1, 2-1, and 2-2 vertex exchanges. We also use the inter-route and intra-route versions of 2-opt. We also define a specific operator for the E-VRP-NL, referred to as *separate*. This operator creates two routes from a single route by inserting a return to the depot after a customer visit. It may improve the cost of a solution if at least one CS is part of the split route. Indeed, creating two routes rather than one may decrease the total time when an expensive detour to a CS is avoided. We stop applying an operator as soon as it has found an improving move. We refer to Algorithm 5 in Appendix C for a description of the general scheme of the VND search phase.

We only consider CSs when evaluating LS moves. In order to make charging decisions in such a way that every route involved in a move has the lowest possible duration, we solve one (inter-route moves) or more (intra-route moves) FRVCPs. To this end, we use the labeling algorithm described by Froger et al. (2019). After each call, the duration of the route is stored in a cache memory to avoid recomputing it. Moreover, since only improving solutions are accepted in the VND framework, we only solve FRVCPs for potentially improving moves. Therefore, we rely on a procedure that filters unpromising or infeasible moves. Such moves are determined by establishing a lower bound on the duration of the routes resulting from a move. The value of this lower bound is computed as the sum of two terms: the minimum increase in time to detour to a CS between two successive nodes in the route, and a lower bound on the charging time. The latter term is computed by dividing the sum of the energy consumption of the route and the minimum increase in energy consumption to detour to a CS between two successive nodes in the route, by the steepest slope for a segment of the piecewise linear charging functions. We note that this lower bound corresponds to the true duration of a route in the case where its energy consumption does not exceed Q. If the lower bound on the route duration exceeds the limit, then the route is infeasible. Otherwise, the procedure checks whether the route duration is stored in a cache memory and modifies the lower bound value accordingly. We use the lower bound on the duration of the routes to determine whether the move is strictly non-improving. We refer to Algorithm 6 in Appendix C for a detailed explanation on this procedure. If a move is not discarded by the lower bound on the basis that it is infeasible or non-improving, we evaluate it exactly when necessary (i.e., before accepting it). To this end, we solve FRVCPs for all the routes that have not been evaluated exactly, as long as the move remains feasible and potentially improving.

4.1.2 The perturbation phase

Whenever we reach a local optimum, we perturb the current solution by removing geographically close customers and by reinserting them at different positions. First, we randomly select a customer $i \in I$. We then remove the κ closest customers to i from their respective routes, with κ randomly selected in the interval $[\min\{|I|, 5\}, \max\{\min\{|I|, 5\}, \lceil\sqrt{|I|}\rceil\}]$. We set the distance between two customers i_1 and i_2 as equal to $0.5\left(t_{i_1,i_2} + t_{i_2,i_1} + (e_{i_1,i_2} + e_{i_2,i_1})/\rho^*\right)$. The value of ρ^* corresponds to the steepest slope for a segment of the piecewise linear charging functions. Finally, customers are reinserted in the solution one at a time and in a random order by applying the following rules. A removed customer cannot be reinserted in the same route from which it was removed. We evaluate the increase in time of every feasible insertion of the customer. This is done by reoptimizing the charging decisions. The probability of selecting a given feasible insertion is set inversely proportional to the time increase due to the insertion. If there exists no feasible insertion, we simply create a new route with the customer.

4.2 Solution assembler: a branch-and-cut method

The objective of the second component of the matheuristic is to construct the best possible solution to the E-VRP-NL-C from a pool of routes Ω, obtained from the route generator component. We recall that the charging decisions made for a route are such that its total duration is minimized. The main challenge of the second component of the heuristic is to combine the routes in a solution while satisfying the capacity constraints.

Deriving the best possible solution by solving a set partitioning (SP) model over a pool of routes is a strategy that has been successfully applied to several hard VRPs (Alvarenga et al., 2007; Subramanian et al., 2013; Villegas et al., 2013; Montoya et al., 2017; Andelmin and Bartolini, 2019). It is used either as a post-optimization phase or as an

intensification phase within a methaheuritic. An example of the latter is the matheuristic proposed by Subramanian et al. (2013) to solve a class of VRPs. This approach has been mostly applied to problems without route coupling constraints (i.e., the feasibility of one route is independent of the feasibility of other routes). Due to the CS capacity constraints, the route coupling constraints need to be accounted for in the E-VRP-NL-C. To the best of our knowledge, only two studies have dealt with route coupling constraints in the assembly phase of a solution from a pool of routes: Morais et al. (2014) and Grangier et al. (2017), both in the context of cross-dock VRPs.

We propose assembling the solutions using a decomposition of the problem into a route selection master problem and a CS capacity management subproblem. The master problem consists in selecting a set of routes such that every customer is covered by exactly one route. The charging decisions within the selected routes (as imported from the first component) may lead to a violation of the capacity constraints. In such cases, the subproblem checks whether the CS capacity constraints can be met by revising some of the charging decisions.

We propose four versions of the subproblem, which primarily depend on the degree of allowed modifications on the selected routes by the master problem. In the first version (denoted by N), we do not revise the charging decisions in the selected routes and we only check whether the CS capacity constraints are satisfied. In the second version (denoted by D), we allow delaying the starting time of each route (i.e., we postpone their starting time) to satisfy the CS capacity constraints. We note that in versions N and D no increase in the total time of the solution is incurred. In the third version (denoted by DW), we allow delaying the starting time of each route, and we also allow vehicles to wait for a charger if a CS is overcrowded by EVs. In DW, waiting times can only occur when multiple CSs are visited within a route, since, when only on CS is visited within a route, delaying the starting time of the route is preferable, as this does not penalize the objective function. In the fourth version (denoted by DWR), we also revise the amount of energy charged at the CSs. If a route contains at least two CSs, we may decide to charge more at the first CS in order to avoid waiting before charging the EV at the second CS. We note that this strategy may sometimes avoid detouring to one of the CSs visited in the original route.

We designed a branch-and-cut method to efficiently solve the problem while exploiting the above discussed decomposition. While solving the route selection problem, we dynamically solve the CS capacity management subproblem. More specifically, at each node, we solve the subproblem for the current selection of routes. Depending on the version of the subproblem, we generate different cuts to discard infeasible route selections. For versions DW and DWR, we also add cuts to account for the underestimation of the increase in the total time to visit the customers.

In the following subsections, we provide a detailed description of our four versions of the CS capacity management subproblem. We use the following notation. The binary parameter a_{ri} is equal to one if and only if route $r \in \Omega$ serves customer $i \in I$. We define parameter t_r as the duration of a route $r \in \Omega$, obtained from the route generation component. We denote by $O(\Omega)$ the set containing all the charging operations occurring in the routes belonging to Ω, and by $O_j(\Omega) \subseteq O(\Omega)$ all the charging operations occurring at CS $j \in F$ in these routes. We denote by $O(\{r\})$ the list of charging operations occurring in route r. Let $r(o)$ and $j(o)$ be the route and the CS associated with a charging operation o. Let $\bar{S}(o)$ and $\bar{\Delta}(o)$ be the starting time and duration of charging operation o in the route $r(o) \in \Omega$. For each route $r \in \Omega$, we assume that the operations in $O(\{r\})$ are sorted in non-decreasing order of their starting times. We denote by $\pi(o)$ the charging operation following o in route $r(o)$. If there does not exist any charging operation after o, we set $\pi(o) = -1$. We also denote by $\Pi^+(o)$ the charging operations occurring after o in $r(o)$. For every route $r \in \Omega$ and for each charging operation $o \in O(\{r\})$ we denote by $t^+(o)$ the total time spent by the EV in r between the CS associated with o and the CS associated with $\pi^+(o)$ or the depot. For each operation o, we also introduce two parameters $ES(o)$ and $LS(o)$ representing its earliest and latest possible starting times. The values of these parameters depend on the different versions of the subproblem and they are specified in the following subsections.

4.2.1 Version N and version D of the subproblem

To model the route selection problem, we introduce a binary variable x_r equal to 1 if and only if route $r \in \Omega$ is selected. A MILP formulation of this problem is then the following classical SP model:

$$[HC1] \quad \text{minimize} \sum_{r \in \Omega} t_r x_r \tag{88}$$

$$\text{subject to} \sum_{r \in \Omega} a_{ri} x_r = 1 \quad i \in I \tag{89}$$

$$x_r \in \{0,1\} \quad r \in \Omega. \tag{90}$$

The objective (88) is to select a subset of routes from Ω that minimizes the total duration. Constraints (89) ensure that each customer is visited exactly once. Constraints (90) set the domains of the decision variables.

Let $\Omega_{1^+}(\bar{x})$ denote the set of routes with at least one charging operation resulting from a feasible solution $\bar{x}$ to the above problem, i.e., $\Omega_{1^+}(\bar{x}) = \{r \in \Omega : \bar{x}_r = 1 \wedge O(\{r\}) \neq \varnothing\}$. Note that the routes without mid-route charging need not be considered when checking the capacity constraints. We also define $\Omega_{1^+}(\bar{x}, j) \subseteq \Omega_{1^+}(\bar{x})$ as the set of routes with a charging operation at $j \in F$.

Version N In this version of the subproblem, the CS capacity management subproblem does not revise the charging operations scheduled in the selected routes. It only checks the feasibility of the charging operations with respect to the capacity constraints. We therefore set $ES(o) = LS(o) = \bar{S}(o)$. Let $\text{CMP}_1(\bar{x})$ be this subproblem defined for the routes that belong to $\Omega_{1^+}(\bar{x})$. It can be decomposed into $|F|$ independent problems, one for every CS. To solve $\text{CMP}_1(\bar{x})$, for every CS $j \in F$ we apply a polynomial algorithm to check the existence of subsets of operations overloading the CS. Specifically, we call procedure `CheckCapacityCut`$(O_j(\Omega_{1^+}(\bar{x})), C_j)$ to check whether the capacity constraints are satisfied according to the scheduled charging operations in $O_j(\Omega_{1^+}(\bar{x}))$. We refer the reader to Algorithm 7 in Appendix D for the details of this procedure.

Version D The main drawback of version N is that it may discard promising routes. Indeed, in some cases simply delaying the starting time of the routes renders them feasible. Note that shifting the start time of a route does not increase the objective function. In version D, we seek to resolve capacity conflicts by shifting starting times of the routes. Let $\text{CMP}_2(\bar{x})$ be this subproblem. In contrast to $\text{CMP}_1(\bar{x})$, $\text{CMP}_2(\bar{x})$ does not decompose into an independent problem for each CS. Let o be a charging operation. Its earliest starting time $ES(o)$ is equal to $\bar{S}(o)$ since by construction the operations are left shifted in each route of the pool. The parameter $LS(o)$ is computed by subtracting from T_{max} the time needed to complete the route (considering the duration of the next charging operations, the driving times, and no waiting times). Specifically, $LS(o) = T_{max} - \sum_{o' \in \{o\} \cup \Pi^+(o)} (t^+(o') + \bar{\Delta}(o'))$.

We now define a continuous-time MILP formulation of $\text{CMP}_2(\bar{x})$. Let the variable S_o define the starting time of operation o. We model the capacity constraints using a flow-based formulation. For each CS $j \in F$, we consider two dummy operations ε_j^+ and ε_j^-, and we define $\widetilde{C}_o := 1$ for all $o \in O_j(\Omega_{1^+}(\bar{x}))$ and $\widetilde{C}_{\varepsilon_j^+} := \widetilde{C}_{\varepsilon_j^-} := C_j$. We then introduce the continuous variable $f_{oo'}$ representing the quantity of resource (i.e., chargers) that is transferred from charging operation o to charging operation o'. We also define the sequential binary variable $u_{oo'}$ taking the value of 1 if operation o is processed before operation o'. The MILP formulation of $\text{CMP}_2(\bar{x})$ is as follows:

$$[\text{CMP}_2(\bar{x})] \quad \text{minimize } 0 \tag{91}$$

$$\text{subject to } u_{oo'} + u_{o'o} \leqslant 1 \quad j \in F, o, o' \in O_j(\Omega_{1^+}(\bar{x})) : o < o' \tag{92}$$

$$u_{oo''} \geqslant u_{oo'} + u_{o'o''} - 1 \quad j \in F, o, o', o'' \in O_j(\Omega_{1^+}(\bar{x})) \tag{93}$$

$$S_{o'} \geqslant S_o + \bar{\Delta}(o) u_{oo'} + (LS(o) - ES_{o'})(u_{oo'} - 1) \quad j \in F, o, o' \in O_j(\Omega_{1^+}(\bar{x})) \tag{94}$$

$$S_{\pi(o)} = S_o + \bar{\Delta}(o) + t^+(o) \quad o \in O(\Omega_{1^+}(\bar{x})) : \pi(o) \neq -1 \tag{95}$$

$$\sum_{o' \in O_j(\Omega_{1^+}(\bar{x})) \cup \{\varepsilon_j^+\}} f_{o'o} = 1 \quad j \in F, o \in O_j(\Omega_{1^+}(\bar{x})) \tag{96}$$

$$\sum_{o' \in O_j(\Omega_{1^+}(\bar{x})) \cup \{\varepsilon_j^+\}} f_{o'o} - \sum_{o' \in O_j(\Omega_{1^+}(\bar{x})) \cup \{\varepsilon_j^-\}} f_{oo'} = 0 \quad j \in F, o \in O_j(\Omega_{1^+}(\bar{x})) \tag{97}$$

$$\sum_{o \in O_j(\Omega_{1^+}(\bar{x})) \cup \{\varepsilon_j^-\}} f_{\varepsilon_j^+, o} = C_j \quad j \in F \tag{98}$$

$$\sum_{o \in O_j(\Omega_{1^+}(\bar{x})) \cup \{\varepsilon_j^+\}} f_{o, \varepsilon_j^-} = C_j \quad j \in F \tag{99}$$

$$f_{oo'} \leqslant \max\left(\widetilde{C_o}, \widetilde{C_{o'}}\right) u_{oo'} \quad j \in F, o, o' \in O_j(\Omega_{1^+}(\bar{x})) \tag{100}$$

$$ES(o) \leqslant S_o \leqslant LS(o) \quad o \in O(\Omega_{1^+}(\bar{x})) \tag{101}$$

$$f_{oo'} \geqslant 0 \quad j \in F,$$

$$(o, o') \in \big(O_j(\Omega_{1^+}(\bar{x})) \cup \{\varepsilon_j^+\}\big) \cup \big(O_j(\Omega_{1^+}(\bar{x})) \cup \{\varepsilon_j^-\}\big) \tag{102}$$

$$u_{oo'} \in \{0, 1\} \quad j \in F, o, o' \in O_j(\Omega_{1^+}(\bar{x})). \tag{103}$$

[$\text{CMP}_2(\bar{x})$] is a feasibility problem. Constraints (92) state that for two distinct operations o and o', either o precedes o', o' precedes o, or o and o' are processed in parallel (if there is more than one charger at the CS). Constraints (93) express the transitivity of the precedence relations. Constraints (94) are the disjunctive constraints on the operations related to the same CS. Each such constraint is active when $u_{oo'} = 1$ and, in which case, it enforces the precedence relation between charging operations o and o'. Note that no waiting times can occur before a charging operation. Constraints (95) enforce the precedence relation and the time lag between the charging operations occurring in the same route. Constraints (96) state that a charger has to be allocated to each charging operation. Constraints (97) ensure flow conservation. Constraints (98) and (99) define the value of the flow leaving the source and the flow entering the sink. Constraints (100) couple the flow variables with the sequence variables. Constraints (101) and (103) define the domains of the decision variables.

4.2.2 Version DW and version DWR of the subproblem

In these versions of the subproblem, we allow a possible increase in the total duration of the routes. Indeed, introducing waiting times at CSs or revising the amount of charged energy may help resolve capacity violations, but such modifications may extend routes due to the non-linearity of the charging functions and the consideration of multiple charging technologies. Let θ be a non-negative variable estimating the added delay when solving the CS capacity management subproblem. A MILP formulation of the route selection problem (derived directly from [$HC1$]) follows:

$$[HC2] \quad \text{minimize} \sum_{r \in \Omega} t_r x_r + \theta \tag{104}$$

$$\text{subject to} \quad (89), (90)$$

$$\theta \geqslant 0. \tag{105}$$

Thereafter, we assume that we have a fixed selection $\Omega_{1^+}(\bar{x})$ of routes given by fixing the variables $\{x_r\}_{r \in \Omega}$ to values respecting the current constraints of the route selection problem.

Version DW In this version of the subproblem, we assume that EVs can wait at CSs if delaying the starting times of the routes is not sufficient to avoid capacity violations. Let $\text{CMP}_3(\bar{x})$ be the scheduling subproblem of the routes $\Omega_{1^+}(\bar{x})$, which has the objective of minimising the addition of waiting times. The MILP formulation of $\text{CMP}_3(\bar{x})$ uses the decision variables S_o, $f_{oo'}$, $u_{oo'}$ defined in $\text{CMP}_2(\bar{x})$. We also introduce variable ∇_o that represents the waiting time incurred before the start of charging operation $o \in O(\Omega_{1^+}(\bar{x}))$. For every charging operation o, its earliest starting time $ES(o)$ is equal to $\bar{S}(o)$ and its latest starting time $LS(o)$ is equal to $T_{max} - \sum_{o' \in \{o\} \cup \Pi^+(o)} (t^+(o') + \bar{\Delta}(o'))$. The MILP formulation of $\text{CMP}_3(\bar{x})$ is as follows:

$$[\text{CMP}_3(\bar{x})] \quad \text{minimize} \sum_{o \in O(\Omega_{1^+}(\bar{x}))} \nabla_o \tag{106}$$

$$\text{subject to} \quad (92) - (94), (96) - (103)$$

$$S_{\pi(o)} = S_o + \bar{\Delta}(o) + t^+(o) + \nabla_{\pi(o)} \quad o \in O(\Omega_{1^+}(\bar{x})) : \pi(o) \neq -1 \tag{107}$$

$$\nabla_o \geqslant 0, \quad o \in O(\Omega_{1^+}(\bar{x})). \tag{108}$$

The objective (106) is to minimize the waiting time inserted in each route. Constraints (107) define the minimum time lag between the charging operations occurring in the same route. Constraints (108) define the domains of the waiting decision variables.

Version DWR In this version of the subproblem, in addition to the introduction of waiting times, resolving conflicts at CSs can also be achieved by revising the amounts of energy charged at each CS in every route. We denote by $\text{CMP}_4(\bar{x})$ the subproblem in which we want to minimize the increase in the duration of the selected routes. Indeed, revising the

charging operations leads to an increase in time when the CSs in a route have different charging technologies or when the charging functions are non-linear. Note that if we substantially increase the charging amounts at a CS we may not need to visit the subsequent CS in the route. We therefore need to account for the potential removal of visits to CSs.

We denote by $e^-(o)$ the energy consumption of the EV between its departure from i_o and its arrival at $j_{\pi^+(o)}$ if $\pi(o) \neq -1$. This takes into account the energy consumed to visit all the customers scheduled in the route between charging operations o and $\pi^+(o)$ or the depot. Similarly, we denote by $e^+(o)$ the energy consumption of the EV from its departure from $j_{\pi^-(o)}$, if o is not the first charging operation of the route or from the depot to its arrival at $j(o)$. Since a charging operation can be skipped by charging more energy during the previous or next charging operations of the same route, we define $\widetilde{t}(o)$ and $\widetilde{e}(o)$ as the time and energy saved if the EV does not detour to perform the charging operation o. We define $\Omega_{2+}(\bar{x})$ as the subset of $\Omega_{1+}(\bar{x})$ that contains only the routes including at least two charging operations (i.e., $\Omega_{2+}(\bar{x}) = \{r \in \Omega_{1+}(\bar{x}) : |O(\{r\})| \geqslant 2\}$).

Our formulation of $\text{CMP}_4(\bar{x})$ draws upon formulation $[\text{CMP}_3(\bar{x})]$. Aside from using decision variables defined in the latter, $[\text{CMP}_4(\bar{x})]$ also uses the following decision variables for the operations in $O(\Omega_{2+}(\bar{x}))$. The variables $\underline{t}_j$ and $\overline{t}_j$ are the scaled arrival and departure times, according to the charging function of CS $j(o)$. The binary variables $\underline{w}_{ok}$ and $\overline{w}_{ok}$ are equal to 1 if and only if the SoC lies between $q_{j(o),k-1}$ and $q_{j(o),k}$, with $k \in B_{j(o)} \backslash \{0\}$, upon starting and finishing operation o, respectively. The continuous variables $\underline{\lambda}_{ok}$ and $\overline{\lambda}_{jk}$ are the coefficients associated with the breakpoints $(a_{j(o),k}, q_{j(o),k})$ of $\phi_{j(o)}$ upon starting and finishing operation o, respectively. The continuous variables $\underline{y}_o$ and $\overline{y}_o$ represent the SoC of the EV upon starting and finishing charging operation o. The continuous variable Δ_o represents the duration of charging operation o. For each route, we check whether it might be possible for a charging operation o to be skipped by considering that the EV leaves the previous CS (or depot) with a fully replenished battery. If this allows the EV to reach the next CS or to return to the depot without performing o, then we allow the EV not to detour to the corresponding CS. To this end, we introduce the binary variable z_o equal to 1 if and only if the charging operation o is executed. We also compute for every charging operation the time windows during which it must be scheduled. The earliest starting time $ES(o)$ of a charging operation o is computed assuming that the EV skips (if the previous computation has shown it is possible) the previous charging operations (if any), and charges the maximum between the energy needed to recover the detour to the CS and the energy required to reach the next CS. To compute the latter, we consider that the SoC of the EV upon arriving at the CS is maximal (a full charge occurs at the previous CS). Then, we estimate the charging times assuming that the EV arrives with an empty battery. The latest starting time $LS(o)$ of operation o is computed assuming that the EV returns to the depot at time T_{max} and assuming that the EV skips the next charging operations (if possible). The MILP formulation of $\text{CMP}_4(\bar{x})$ is as follows:

$$[\text{CMP}_4(\bar{x})] \quad \text{minimize} \sum_{o \in O(\Omega_{1+}(\bar{x}))} \nabla_o + \sum_{o \in O(\Omega_{2+}(\bar{x}))} \big(\Delta_o - \bar{\Delta}(o) - (1 - z_o)\widetilde{t}(o)\big) \tag{109}$$

$$\text{subject to} \quad (92) - (93), (97) - (103)$$

$$\underline{y}_o = \sum_{k \in B_{j(o)}} \underline{\lambda}_{jk} q_{j(o)k} \quad o \in O(\Omega_{2+}(\bar{x})) \tag{110}$$

$$\underline{t}_o = \sum_{k \in B_{j(o)}} \underline{\lambda}_{jk} a_{j(o)k} \quad o \in O(\Omega_{2+}(\bar{x})) \tag{111}$$

$$\sum_{k \in B_{j(o)}} \underline{\lambda}_{jk} = \sum_{k \in B_{j(o)} \backslash \{0\}} \underline{w}_{ok} \quad o \in O(\Omega_{2+}(\bar{x})) \tag{112}$$

$$\sum_{k \in B_{j(o)} \backslash \{0\}} \underline{w}_{ok} = z_o \quad o \in O(\Omega_{2+}(\bar{x})) \tag{113}$$

$$\underline{\lambda}_{j0} \leqslant \underline{w}_{o1} \quad o \in O(\Omega_{2+}(\bar{x})) \tag{114}$$

$$\underline{\lambda}_{jk} \leqslant \underline{w}_{ok} + \underline{w}_{o,k+1} \quad o \in O(\Omega_{2+}(\bar{x})), k \in B_{j(o)} \backslash \{0, b_{j(o)}\} \tag{115}$$

$$\underline{\lambda}_{jb_j} \leqslant \underline{w}_{ob_{j(o)}} \quad o \in O(\Omega_{2+}(\bar{x})) \tag{116}$$

$$\overline{y}_o = \sum_{k \in B_{j(o)}} \overline{\lambda}_o q_{j(o)k} \quad o \in O(\Omega_{2+}(\bar{x})) \tag{117}$$

$$\overline{t}_o = \sum_{k \in B_{j(o)}} \overline{\lambda}_o a_{j(o)k} \quad o \in O(\Omega_{2+}(\bar{x})) \tag{118}$$

$$\sum_{k \in B_{j(o)}} \lambda_{ok} = \sum_{k \in B_{j(o)} \setminus \{0\}} \overline{w}_{ok} \quad o \in O(\Omega_{2^+}(\bar{x})) \tag{119}$$

$$\sum_{k \in B_{j(o)} \setminus \{0\}} \overline{w}_{ok} = 1 \quad o \in O(\Omega_{2^+}(\bar{x})) \tag{120}$$

$$\overline{\lambda}_{o0} \leqslant \overline{w}_{o1} \quad o \in O(\Omega_{2^+}(\bar{x})) \tag{121}$$

$$\overline{\lambda}_{ok} \leqslant \overline{w}_{ok} + \overline{w}_{o,k+1} \quad o \in O(\Omega_{2^+}(\bar{x})), k \in B_{j(o)} \setminus \{0, b_{j(o)}\} \tag{122}$$

$$\overline{\lambda}_{ob_{j(o)}} \leqslant \overline{w}_{ob_{j_0}} \quad o \in O(\Omega_{2^+}(\bar{x})) \tag{123}$$

$$\Delta_o = \overline{t}_o - \underline{t}_o \quad o \in O(\Omega_{2^+}(\bar{x})) \tag{124}$$

$$\Delta_o \leqslant a_{j(o),b_{j(o)}} z_o \quad o \in O(\Omega_{2^+}(\bar{x})) \tag{125}$$

$$\underline{y}_{o(r,1)} = \bar{q}^{\text{first}}(r) \quad r \in \Omega_{2^+}(\bar{x}) \tag{126}$$

$$\underline{y}_{\pi(o)} = \overline{y}_o - e^+(o) + \widetilde{e}(o)(1 - z_o) \quad o \in O(\Omega_{2^+}(\bar{x})) : \pi(o) \neq -1 \tag{127}$$

$$\overline{y}_o - e^+(o) + \widetilde{e}(o)(1 - z_o) \geqslant 0 \quad o \in O(\Omega_{2^+}(\bar{x})) : \pi(o) = -1 \tag{128}$$

$$S_{o'} \geqslant S_o + \bar{\Delta}(o) u_{oo'} + (LS(o) - ES_{o'})(u_{oo'} - 1) \quad j \in F, o, o' \in O_j(\Omega_{1^+}(\bar{x}) \setminus \Omega_{2^+}(\bar{x})) \tag{129}$$

$$S_{o'} \geqslant S_o + \Delta_o + (LS(o) - ES_{o'})(u_{oo'} - 1) \quad j \in F, o, o' \in O_j(\Omega_{2^+}(\bar{x})) \tag{130}$$

$$S_{\pi(o)} = S_o + \bar{\Delta}(o) + t^+(o) + \nabla_{\pi(o)} \quad o \in O(\Omega_{1^+}(\bar{x}) \setminus \Omega_{2^+}(\bar{x})) : \pi(o) \neq -1 \tag{131}$$

$$S_{\pi(o)} = S_o + \Delta_o + t^+(o) - \widetilde{t}(o)(1 - z_o) + \nabla_{\pi(o)} \quad o \in O(\Omega_{2^+}(\bar{x})) : \pi(o) \neq -1 \tag{132}$$

$$S_o + \Delta_o + t^+(o) - \widetilde{t}(o)(1 - z_o) \leqslant T_{max} \quad o \in O(\Omega_{2^+}(\bar{x})) : \pi(o) = -1 \tag{133}$$

$$\sum_{o' \in O_{j(o)}(\Omega_{1^+}(\bar{x})) \cup \{\varepsilon^+_{j(o)}\}} f_{o'o} = 1 \quad o \in O(\Omega_{1^+}(\bar{x}) \setminus \Omega_{2^+}(\bar{x})) \tag{134}$$

$$\sum_{o' \in O_{j(o)}(\Omega_{1^+}(\bar{x})) \cup \{\varepsilon^+_{j(o)}\}} f_{o'o} = z_o \quad o \in O(\Omega_{2^+}(\bar{x})) \tag{135}$$

$$ES(o) \leqslant S_o \leqslant LS(o) \quad o \in O(\Omega_{1^+}(\bar{x})) \tag{136}$$

$$z_o \in \{0, 1\}, \Delta_o \geqslant 0, 0 \leqslant \underline{y}_o \leqslant Q, 0 \leqslant \overline{y}_o \leqslant Q \quad o \in O(\Omega_{2^+}(\bar{x})) \tag{137}$$

$$\underline{w}_{ok}, \overline{w}_{ok} \in \{0, 1\}, \quad o \in O(\Omega_{2^+}(\bar{x})), k \in B_{j(o)} \setminus \{0\} \tag{138}$$

$$\underline{\lambda}_{ok}, \overline{\lambda}_{ok} \in \{0, 1\} \quad o \in O(\Omega_{2^+}(\bar{x})), k \in B_{j(o)}. \tag{139}$$

The objective (109) is to minimize the total additional time inserted in each route. Constraints (110)–(124) model the piecewise linear charging functions. Constraints (125) impose a duration equal to 0 for each charging operation that is not executed anymore. For each route $r \in \Omega_{2^+}(\bar{x})$, constraints (126) define the SoC $\bar{q}^{\text{first}}(r)$ of the EV upon starting its first charging operation. Constraints (127) couple the SoC of the EV after finishing a charging operation with its SoC when starting the next charging operation occurring in the route. Note that if z_o is equal to 0, then the SoC $\underline{y}_o = \overline{y}_o$ still takes into account the energy consumed to detour to CS $j(o)$. The energy saved by not visiting this CS is subtracted when computing the SoC at the beginning of the next operation of the route or at the arrival at the depot (see (128)). For each route, constraints (128) force the corresponding EV to have enough SoC at the end of the last charging operation to reach the depot. Constraints (129) and (130) are the disjunctive constraints on the operations related to the same CS. Constraints (131) and (132) define a minimum time lag between the charging operations occurring in the same route. Note that if z_o is equal to 0, then the starting time S_o still takes into account the detour to CS $j(o)$. The time saved by not visiting this CS is subtracted during the computation of the departure time for the next operation of the route. Constraints (133) limit the route duration. Constraints (134) and (135) assign a charger to each operation that is executed. Constraints (136)–(139) define the domains of the new decision variables.

4.2.3 Cut generation procedure

To couple the route selection problem with the CS capacity management subproblem, we generate constraints to cut off infeasible selections of routes in $\Omega_{1^+}(\bar{x})$ and to bound the variable θ (for versions DW and DWR). The efficiency of our branch-and-cut method is based on the strength of these cuts. Rather than only cutting the current solution, we investigate a strategy that generates cuts for a large portion of the solution space of the route selection problem. Moreover, we try to add several cuts at a time to speed up the convergence of the algorithm.

We first focus on the cuts that are applicable only to version N. Given a CS j, let $\mathcal{O}_j$ be a set containing subsets of operations with a cardinality strictly larger than C_j, which overlap in time. To discard the current solution $\bar{x}$ in the route selection problem, we add to $[HC1]$ the following cuts:

$$\sum_{r \in \Omega_{1+}(\bar{x}): O(\{r\}) \cap O \neq \varnothing} x_r \leqslant C_j \quad j \in F, O \in \mathcal{O}_j. \tag{140}$$

For the other versions of the subproblem, Algorithm 3 summarizes the procedure used to solve the capacity management subproblem and potentially generating cuts. We first try to detect, before solving $\text{CMP}_{\cdot}(\bar{x})$, the potential infeasibility of the subproblem. To this end, we apply two algorithms (frequently used in scheduling problems) to detect capacity violations that cannot be solved by the CS capacity management subproblem. These algorithms focus on a single CS and are based on a reasoning rooted in the time windows (derived from the opening hours of the depot) in which every charging operation can be scheduled. The first algorithm focuses on the fixed part of the charging operations (line 3). We call a fixed part of a charging operation o the time interval (if it exists) between the latest starting time $LS(o)$ and earliest completion time $ES(o) + \bar{\Delta}(o)$. Based on the fixed part of the charging operations, we detect subsets of operations that will necessarily lead to a violation of the capacity constraints and we add cuts to forbid the underlying subset of routes (line 6). Hereafter, we refer to this algorithm as the *fixed part based algorithm*. The details of this procedure are provided in Algorithm 8 (Appendix D). The second algorithm is based on an energy reasoning (line 8). The required energy consumption of a charging operation o during a time interval $[t_1, t_2)$ is equal to the minimum duration (possibly equal to 0) for which o is surely executed within the interval. For a CS j, the total required energy consumption by the charging operation o scheduled at j over time interval $[t_1, t_2)$ cannot exceed $C_j(t_2 - t_1)$. The difference between the last term and the total required energy consumption is called the slack over $[t_1, t_2)$. The slack over any time interval must always be non-negative. The number of intervals that needs consideration is bounded (see Baptiste et al. (2001) for more details). When there exists an interval for which a slack is strictly negative, the subset of operations having a non-zero energy consumption on this interval cannot be performed without leading to a violation of the capacity constraints. We add a no-good cut to discard it (line 10). Hereafter, we refer to this algorithm as the *energy reasoning based algorithm*. The details of this procedure are provided in Algorithm 9 (Appendix D).

To limit the computation time, if the two previous algorithms prove the infeasibility of the subproblem, we add to the route selection problem the generated cuts and we do not solve the subproblem. Otherwise, since the absence of capacity violation detection from these latter algorithms does not ensure the feasibility of the subproblem, we solve the latter exactly. We observe that the subproblem ($\text{CMP}_2(\bar{x})$, $\text{CMP}_3(\bar{x})$, or $\text{CMP}_4(\bar{x})$) may often be decomposed into several independents smaller subproblems. This allows us to potentially formulate cuts for each of these small subproblems. We introduce a graph $G(\bar{x})$ in which each node represents a CS and there exists an edge between two CSs if there exists a route in $\Omega_{1+}(\bar{x})$ with charging operations at these two CSs. We can then decompose the subproblem into as many independent subproblems as the number of connected components of $G(\bar{x})$. Let $\mathcal{C}(G(\bar{x}))$ be the connected components of $G(\bar{x})$. For every connected component $C \in \mathcal{C}(G(\bar{x}))$, only the charging operations scheduled at the CSs in C need consideration. We denote $\text{CMP}(\bar{x}, C)$ the subproblem restricted to the CSs in connected component C. When $\text{CMP}(\bar{x}, C)$ is infeasible, we generate an integer Benders cut, also called combinatorial Benders cut (Codato and Fischetti, 2006), to invalidate the current solution to the route selection problem (line 18). When it is feasible, we compare the current value $\bar{\theta}$ of θ with the subproblem objective function value. This only holds for versions DW and DWR of the subproblem. To build strong cuts, we consider the set $\mathcal{C}^{\text{feas}}$ of connected components for which the corresponding subproblem is feasible. For each subset C of $\mathcal{C}^{\text{feas}}$, we generate an integer Benders optimality cut if the route selection problem underestimates the increase in time needed to resolve conflict(s) at CS(s) (line 25).

4.2.4 The branch-and-cut algorithm

We now outline the general structure of the solution assembler component in Algorithm 4. When provided as an input, we give to the solver the objective function value of a solution to the E-VRP-NL-C as a cutoff value (i.e., an upper bound on the value of the objective function of the master problem). Therefore, the solver only considers the solutions with an objective function value strictly less than this cutoff value. The solver returns the best solution assembled from a given pool of routes by the branch-and-cut method. The component also returns a set L of improving solutions (compared to the initial solution s) computed throughout the algorithm.

Algorithm 3: Solving the CS capacity management subproblem and generating cuts for version D, DW, or DWR

Input: a solution $\bar{x}$ to the current master problem

Procedure `SolveSubProblem(`$\bar{x}$`)`:

  Compute $ES(o)$ and $LS(o)$ for each charging operation $o \in O(\Omega_{1^+}(\bar{x}))$

  for $j \in F$ **do**

    $\mathcal{U} \leftarrow$ `FixedPartsAlgorithm`$(O_j(\bar{x}), C_j)$ (see Algorithm 8)

    if $\mathcal{U} \neq \emptyset$ **then**

      for $U \in \mathcal{U}$ **do** Generate the following cut and add it to $[HC.]$: $\sum_{r \in \Omega_{1^+}(\bar{x},j): O(\{r\}) \cap U \neq \emptyset} x_r \leqslant C_j$

    else

      $\mathcal{U} \leftarrow$ `EnergeticAlgorithm`$(O_j(\bar{x}), C_j)$ (see Algorithm 9)

      if $\mathcal{U} \neq \emptyset$ **then**

        for $U \in \mathcal{U}$ **do** Generate the following cut and add it to $[HC.]$:
        $$\sum_{r \in \Omega_{1^+}(\bar{x},j): O(\{r\}) \cap U \neq \emptyset} x_r \leqslant |\{r \in \Omega_{1^+}(\bar{x}, j) : O(\{r\}) \cap U \neq \emptyset\}| - 1$$

      end

    end

  end

  if no cuts have been generated and $\bar{x}$ is integer **then**

    $\mathcal{C}^{\text{feas}} \leftarrow \emptyset$

    for each $C \in \mathcal{C}(G(\bar{x}))$ **do**

      `/* Let CMP(`$\bar{x}, C$`) be the subproblem restricted to the CSs in connected component` C `and` z`(CMP(`$\bar{x}, C$`)) denotes the value of the objective function (when a solution is found) */`

      Solve the MILP formulation of $\text{CMP}(\bar{x}, C)$ that corresponds to the selected version of the subproblem

      if no solution is found **then**

        Generate the following cut and add it to $[HC.]$: $\sum_{j \in C} \sum_{r \in \Omega_{1^+}(\bar{x},j)} x_r \leqslant \sum_{j \in C} |\Omega_{1^+}(\bar{x}, j)| - 1$

      else

        `/* Only for versions DW and DWR`

        $\mathcal{C}^{\text{feas}} \leftarrow \mathcal{C}^{\text{feas}} \cup \{C\}$

      end

    end

    `/* Only for versions DW and DWR`

    for every subset $\mathcal{C}$ of connected components of $\mathcal{C}^{\text{feas}}$ **do**

      if $\sum_{C \in \mathcal{C}} z(\text{CMP}(\bar{x}, C)) > \bar{\theta}$ **then** Generate the following cut and add it to $[HC.]$:
      $$\left(\sum_{C \in \mathcal{C}} z(\text{CMP}(\bar{x}, C))\right)\left(1 + \sum_{C \in \mathcal{C}} \sum_{j \in C} \sum_{r \in \Omega_{1^+}(\bar{x},j)} x_r - \sum_{C \in \mathcal{C}, j \in C} |\Omega_{1^+}(\bar{x}, j)|\right) \leqslant \theta$$

    end

  end

  if no cuts have been generated **then**

    Save the solution result of calling the subproblem on the routes of $\Omega_{1^+}(\bar{x})$

  end

Algorithm 4: The branch-and-cut algorithm

Input : a pool Ω of routes / a solution s (possibly equal to `NULL`) to the E-VRP-NL-C / a CPU time limit τ for the algorithm / a CPU time limit for each call to the subproblem τ^{SP}

Output: the best solution s^* computed by the branch-and-cut method / a set L of improving solutions to the E-VRP-NL-C (in comparison with s)

Procedure `assemble`(Ω,s,τ,τ^{SP})**:**

  Build the route selection problem [HC.] from pool Ω

  Define the callback function (function called at every node of the branch-and-bound tree) as procedure `SolveSubProblem`($\bar{x}$) (see Algorithm 3) and associate to it a CPU time limit of τ^{SP} per call

  Define the cutoff value $f(s)$ (objective function value of s) and the CPU time limit τ of the solver

  Give the model [HC.] and the callback function to the solver and launch it

  Retrieve the best solution s^* computed within the CPU time limit and the set Φ of improving solutions (compared to s)

  return (s^*, Φ)

5 Computational results

We used Gurobi 7.5.0 (through its Java API) to solve the MILP models. Each instance was executed on a single thread with 12 GB and on a cluster of 27 computers, each having 12 cores and two Intel(R) Xeon® X5675 3.07 GHz processors. We implemented all the algorithms using Java 8. In all the tables, the CPU time is given in seconds and rounded to the nearest integer.

We considered the 120-instance testbed of Montoya et al. (2017)[1]. In this testbed, there are six sets of 20 instances, each with 10, 20, 40, 80, 160, or 320 customers. The EVs are Peugeot iOns which have a consumption rate of 0.125 kWh/km and a battery of 16 kWh. When dealing with the E-VRP-NL-C, we adapted each instance by fixing a number of chargers for each CS. We decided to consider instances in which all the CSs have one or two chargers.

The first aim of our computational experiments is to assess and compare the performance of the CS replication-based and the path-based models (Section 3) for solving small-size instances of the E-VRP-NL-C. These results are presented in §5.1. The second aim of our computational experiments is to assess the quality of the ILS presented in §4.1 as a solution method for the E-VRP-NL. A comparison with results from the literature is presented in §5.2. The third aim of our experiments is to assess the effectiveness of our algorithmic framework to build high-quality solutions to the E-VRP-NL-C. We compare the results given by our matheuristic according to the version of the subproblem selected during the assembly phase. These results are presented in §5.3.

5.1 Results for E-VRP-NL-C formulations

For the E-VRP-NL-C we considered the twenty 10-customer and the twenty 20-customer instances. We ran the MILP models with a three-hour CPU time limit. In the CS replication-based models [$F^{CS\text{-}rep}$], the number of copies of each CS was set to $\beta \geqslant 1$. Since there does not yet exist a procedure for fixing β while guaranteeing optimality, we tested $\beta = 4$ and $\beta = 5$.

Table 1 reports for the CS replication-based and path-based formulations the number of instances with a solution proven to be optimal (#Opt), the average CPU time in seconds (Time) for these latter instances, and the average gap (Gap) for the remaining instances. We compute the gap as $(z - z^{LB})/z$, where z is the objective function value of the best integer solution returned by the solver, and z^{LB} is the best lower bound retrieved by the solver. The detailed results for all the tested instances are reported in Appendix E.1.

Our results show that the path-based model outperforms the CS replication-based models on the 10-customer instances. Indeed, the MILP solver can optimally solve all the 10-customer instances within an average computation time of roughly 10 minutes for the former model, whereas four instances cannot be solved to optimality within the CPU time limit for the latter model. We also observe that the 20-customer instances are already challenging for the models since very few instances were solved. The introduction of the capacity constraints weakens the dominance rules between CSPs in the path-based model. This is due to the waiting times which are difficult to estimate and embed within the filtering technique. The number of paths therefore grows very quickly. Regarding the CS replication-based model,

[1] available from `http://www.vrp-rep.org/` (Retrieved October 25, 2019)

increasing the value of β increases computation times. As expected, since the linear relaxation of each of these models is weak, we conclude that solving the MILP formulations with a commercial solver does not constitute an efficient solution method for the E-VRP-NL-C. Still, 22 out of the 40 tested instances were optimally solved when considering one or two chargers per CS.

Table 1: Computational results for the CS replication-based and path-based formulations on the 10-customer and 20-customer instances.

Capacity	$\|I\|$	$[F^{CSrep}]$ ($\beta = 4$)			$[F^{CSrep}]$ ($\beta = 5$)			$[F^{path}]$		
		#Opt	Time	Gap	#Opt	Time	Gap	#Opt	Time	Gap
1	10	16	372	14.4%	16	503	14.7%	20	572	-
	20	5	1464	12.7%	5	708	12.8%	2	543	20.5%
2	10	16	431	13.6%	16	581	16.3%	20	621	
	20	5	597	*13.2%	5	1387	*14.2%	2	656	21.0%

*: There are 4 instances for which the solver reaches the CPU time limit without finding any solution. They are not considered when computing the gap.

5.2 Results for the E-VRP-NL

Since the route generator of our matheuristic essentially solves the E-VRP-NL, we wanted to assess its quality on this problem. We considered the 120 instances of the original Montoya et al. (2017) testbed. We performed 10 test runs for each instance using Algorithm 2 and compared the results with those obtained in the E-VRP-NL literature through two solutions methods: the metaheuristic of Montoya et al. (2017) which combines an ILS with a heuristic concentration (HC), and the large neighborhood search (LNS) of Koç et al. (2018). Table 2 shows the results of this comparison. Given a number of customers in the instances, it reports for each solution method the number of BKS to the E-VRP-NL (#BKS), the average gap to the BKS (Gap), and the average number of routes in the best computed solution. We compute the gap as $(z - z^*)/z$, where z is the objective function value of the best solution returned by the solution method, and z^* is the objective function value of the BKS. Table 2 also reports the average gap to the best average objective function value (Gap BA) as $(z_{\text{avg}} - z^*_{\text{avg}})/z_{\text{avg}}$, where z_{avg} is the average objective function value of the solutions obtained in 10 runs by the solution method, and z^*_{avg} is the best average (BA) objective function value obtained in 10 runs by one of the existing solution method for the E-VRP-NL. The detailed results for all the tested instances are reported in Appendix E.2.

Table 2: Comparison of the results obtained by ILS with the results of Montoya et al. (2017) and Koç et al. (2018) for the E-VRP-NL.

	ILS + HC 2.33 GHz processor - 16GB of RAM					LNS 3.6 GHz processor - 32 GB of RAM					ILS 3.07 GHz processor - 12 GB of RAM				
$\|I\|$	#BKS	Gap BKS	#R	Gap BA	Time	#BKS	Gap BKS	#R	Gap BA	Time	#BKS	Gap BKS	#R	Gap BA	Time
10	20	0.0%	2.7	0.3%	6	20	0.0%	2.7	0.1%	8	20	0.0%	2.7	0.0%	5
20	11	0.3%	3.7	0.7%	11	12	0.2%	3.6	0.4%	14	20	0.0%	3.6	0.0%	8
40	3	1.0%	6.5	2.6%	35	6	0.9%	6.4	1.1%	45	20	0.0%	6.2	0.0%	20
80	0	3.8%	9.2	5.4%	80	0	3.8%	8.8	3.8%	99	20	0.0%	8.6	0.0%	64
160	0	7.3%	16.7	8.1%	568	0	7.7%	16.6	7.8%	632	20	0.0%	15.3	0.0%	295
320	0	11.2%	32.6	12.6%	4398	0	11.7%	32.6	12.4%	4555	20	0.0%	29.0	0.0%	1118
All	34	3.9%	11.9	4.9%	850	38	4.1%	11.8	4.3%	892	120	0.0%	10.9	0.0%	252

ILS + HC: (Montoya et al., 2017), LNS: (Koç et al., 2018).

The results presented in Table 2 clearly show that our ILS outperforms all existing methods for the E-VRP-NL. We have identified 80 new BKS and matched the existing BKS for the remaining instances. We have improved the previous solutions by about 4.0%. The improvement increases with the number of customers. The algorithm is also stable as the average results on 10 test runs for each instance are better than those reported for previous methods. While it is

difficult to draw definitive conclusions on the computation times since these tests were run on different machines, it seems that the ILS is at least as fast as the other methods, if not the fastest one. The major difference between our method and the existing ones comes from the reoptimization of the charging decisions when evaluating a move.

5.3 Results for the E-VRP-NL-C

We have performed several tests to assess the effectiveness of our matheuristic in obtaining high-quality solutions for the E-VRP-NL-C. After some preliminary experiments, we set the values of the parameters of the matheuristic as presented in Table 3. Setting the stopping criterion to 12 iterations and the number of ILS iterations to 200 proved to be a good compromise between solution quality and computation time. In some rare cases we were not able to optimally solve the assembly phase using version DWR of the subproblem. Indeed, when it is difficult to find a solution satisfying the capacity constraints or when such a solution does not exist, the MILP formulation becomes computationally expensive. Nonetheless, after testing higher CPU time limits for the second component, we observed that the impact on the results was negligible.

Table 3: Value of the matheuristic parameters.

n_{max}	δ	τ	τ^{SP}	α	T_{min}
12	200	180 s	5 s	0.9	$0.67 \cdot T_{max}$

We first compare the results obtained by our matheuristic for the four versions of the subproblem used by the solution assembler. For one or two chargers at each CS and each version of the subproblem, we performed 10 test runs for each instance. Table 4 reports the best results obtained during our tests. Specifically, given a number of chargers per CS and a selected version of the subproblem used in the branch-and-cut method, this table reports the number of instances with a solution in each of the 10 tests (#F), the number of instances with the best solution (BS) to the E-VRP-NL-C computed all over our tests (#BS), including those of the path-based formulation, and the average gap to the BS (Gap BS). Table 5 reports the average CPU time in seconds over 10 runs for the whole algorithm (T), and for the first (T_1) and second (T_2) components. It also reports the average gap with respect to the best average objective function value obtained using a given version of the subproblem (Gap BA). The gaps are computed as in §5.2. Detailed results for all the tested instances are reported in Appendix E.3.

Table 4: Comparison of the best results obtained by the matheuristic for the E-VRP-NL-C according to the version of the subproblem it uses and the number of chargers at each CS.

Cap.	$\|I\|$	Version N			Version D			Version DW			Version DWR		
		#F	#BS	Gap BS	#F	#BS	Gap BS	#F	#BS	Gap BS	#F	#BS	Gap BS
1	10	20	17	0.06%	20	20	0.00%	20	20	0.00%	20	20	0.00%
	20	20	15	0.09%	20	20	0.00%	20	20	0.00%	20	20	0.00%
	40	19	15	0.17%	19	18	0.01%	19	18	0.01%	19	19	0.00%
	80	20	15	0.02%	20	20	0.00%	20	20	0.00%	20	20	0.00%
	160	19	5	0.27%	20	13	0.06%	20	13	0.06%	20	15	0.09%
	320	10	5	0.32%	20	1	0.28%	20	2	0.27%	20	13	0.10%
	All	**105**	**72**	**0.14%**	**119**	**92**	**0.06%**	**119**	**93**	**0.06%**	**119**	**107**	**0.03%**
2	10	20	20	0.00%	20	20	0.00%	20	20	0.00%	20	20	0.00%
	20	20	20	0.00%	20	20	0.00%	20	20	0.00%	20	20	0.00%
	40	19	19	0.09%	20	20	0.00%	20	20	0.00%	20	20	0.00%
	80	20	18	0.01%	20	20	0.00%	20	20	0.00%	20	19	0.01%
	160	20	13	0.05%	20	10	0.18%	20	10	0.18%	20	14	0.09%
	320	19	4	0.33%	20	9	0.12%	20	9	0.12%	20	8	0.17%
	All	**118**	**94**	**0.08%**	**120**	**99**	**0.05%**	**120**	**99**	**0.05%**	**120**	**101**	**0.05%**

Table 5: Comparison of the average results obtained by the matheuristic for the E-VRP-NL-C according to the version of the subproblem it uses and the number of chargers at each CS.

Cap.	$\|I\|$	Version N				Version D				Version DW				Version DWR			
		T	T_1	T_2	Gap BA	T	T_1	T_2	Gap BA	T	T_1	T_2	Gap BA	T	T_1	T_2	Gap BA
1	10	6	5	1	0.06%	6	5	1	0.00%	6	5	1	0.00%	6	5	1	0.00%
	20	11	10	1	0.09%	11	10	1	0.00%	11	10	1	0.00%	11	10	1	0.00%
	40	24	23	1	0.31%	23	22	1	0.03%	23	22	1	0.03%	87	22	65	0.00%
	80	74	73	1	0.10%	74	73	1	0.00%	73	72	1	0.00%	74	73	1	0.04%
	160	340	335	5	0.45%	339	335	4	0.10%	340	336	4	0.10%	397	333	64	0.08%
	320	1345	1274	71	0.34%	1480	1253	227	0.17%	1487	1259	228	0.15%	1933	1244	689	0.05%
	All	**300**	**287**	**13**	**0.22%**	**322**	**283**	**39**	**0.05%**	**323**	**284**	**39**	**0.05%**	**418**	**281**	**137**	**0.03%**
2	10	5	5	0	0.00%	5	5	0	0.00%	5	5	0	0.00%	5	5	0	0.00%
	20	10	10	0	0.00%	10	10	0	0.00%	10	10	0	0.01%	10	10	0	0.00%
	40	23	23	0	0.05%	23	23	0	0.11%	23	23	1	0.09%	45	23	22	0.01%
	80	73	72	1	0.18%	73	73	1	0.19%	73	73	1	0.25%	73	73	1	0.14%
	160	336	334	2	0.24%	336	334	2	0.25%	337	335	2	0.24%	338	336	2	0.28%
	320	1340	1279	61	0.21%	1283	1276	7	0.12%	1276	1269	7	0.14%	1327	1269	58	0.08%
	All	**298**	**287**	**11**	**0.11%**	**288**	**287**	**2**	**0.11%**	**287**	**286**	**2**	**0.12%**	**300**	**286**	**14**	**0.09%**

First, it should be noted that our matheuristic returns an optimal solution for the 22 small-size instances for which the MILP path formulation yields an optimal solution in about 10% of the computation time. Considering that a single charger exists in each CS, we observe that using version N of the subproblem prevents the algorithm from finding solutions to the E-VRP-NL-C for 15 instances. In this case, the algorithm finds the best solution (computed all over our tests) for only 72 out of 120 instances. Delaying the starting time of the routes yields better solutions for 20 more instances. In contrast, allowing waiting times in version DW leads to very marginal improvements, compared with version D. Indeed, making the EVs wait for an available charger can eliminate capacity violations only if at least two CSs are visited in a route; otherwise it is sufficient to make the EVs leave the depot after time 0, as is done in version D. Although the same requirement (more than one CS) holds when allowing the revision of the amount of energy charged at the CSs in version DWR, the latter strategy yields almost all the best solutions to the E-VRP-NL-C.

When increasing the number of chargers to two per CS, the capacity constraints become less binding and all our assembly strategies yield very similar results. This was somewhat expected but it should be noted that delaying the starting time of the routes increases the probability of ending up with a feasible solution.

The first general conclusion is that allowing delays when solving the CS capacity management subproblem is a suitable and efficient way of to assembling high-quality solutions to the E-VRP-NL-C. Allowing the revision of the amounts of energy charged at the CSs on top of it significantly improves the results, but it is slightly more computationally expensive (Table 5). We also note that in general, most of the computation time is spent generating routes as assembling a solution is usually very fast, especially when a cutoff value is provided to the branch-and-cut method. Not surprisingly the computation time increases with the number of customers but it remains reasonable since it takes around 30 minutes to tackle the largest instances.

To assess the relevance of our cuts, we show in Table 6 the distribution of the different type of cuts generated in the branch-and-cut tree. When the capacity constraints are very binding, the energetic reasoning based algorithm detects infeasible route selections much more frequently than the fixed part reasoning based algorithm. We observe that due to its high degree of flexibility, compared with the other versions, DWR requires solving the MILP formulation more often. Indeed, the two previously mentioned algorithms are weaker in that case since the mandatory part of each charging operation may have a very short duration due to the potential revision of the amount of energy charged at the CSs. When we increase the number of chargers at each CS, the reasoning algorithms lead to the generation of fewer cuts. This is not surprising since these are based on relaxations of the subproblem. However, they are useful since the generated cuts are stronger, which avoids computationally expensive calls to the MILP solver. We also see that integer optimality cuts are seldom generated. Since increases in the total time of the routes are penalized, the algorithm tends to avoid adding waiting times. This indicates that revising the charging operations does not always lead to an increase in time,

and a trade-off between delay and a revision at no cost is often found. To support this conclusion, we have analyzed the solutions returned by the matheuristic. Given the number of chargers at each CS, Table 7 reports the percentage of solutions for which there exists at least one route with a delayed starting time, a waiting time before a charging operation, and a revision of the amount of energy charged in the EV. We see that for most of the solutions returned by the algorithm, satisfying the capacity constraints can be achieved without increasing the cost.

Table 6: Characteristics of the cuts generating in the solution assembly phase.

Cap.	Version D			Version DW				Version DWR			
	FP	NRG	BF	FP	NRG	BF	BO	FP	NRG	BF	BO
1	2.3%	94.5%	3.2%	2.3%	95.0%	2.5%	0.2%	0.6%	16.9%	80.5%	2.0%
2	29.0%	28.2%	42.8%	28.9%	27.1%	42.5%	1.5%	1.3%	0.7%	96.2%	1.8%

FP (feasibility cuts generated after calling the fixed part reasoning based algorithm), NRG (feasibility cuts generated after calling the energetic based reasoning algorithm), BF (feasibility cuts of type generated after solving the MILP formulation), BO (optimality cuts of type generated after solving the MILP formulation)

Table 7: Analysis of the solutions retrieved by the matheuristic.

Capacity	Version D	Version DW		Version DWR				
	= D	= D	> W	= D	= NRG	> W	> NRG	> R
1	42.1%	42.2%	1.7%	42.0%	17.4%	1.3%	6.3%	2.5%
2	11.7%	11.7%	0.0%	11.4%	8.5%	0.0%	0.4%	0.0%

D (delay), W (waiting time), NRG (revision of the amount of energy charged), R (removal), = (no increase of the total time), > (increase of the total time)

Since we have imposed a CPU time limit for the branch-and-cut execution and since the maximum route duration limit may decrease according to the state of the solution method, no version of the subproblem dominates the other versions. To perform a fair comparison between the different strategies, and to quantify the benefit of allowing the addition of waiting times and the revision of the amount of energy charged at the CSs, we took the 2,400 long-term pools of routes obtained at the end of the matheuristic for version DWR of the subproblem. For every pool of routes, we ran the branch-and-cut algorithm to generate the best E-VRP-NL-C solution. Table 8 reports for each version of the subproblem the number of pool of routes for which the algorithm obtains a feasible solution (#F), as well as the best solution (#Best) among those obtained with the other versions on the same pool. It also shows the average gap between the solution returned and the best solution obtained during these tests under all the assembly strategies. The conclusions are very similar to those drawn from the results of the matheuristic. Compared to version N, version DWR makes nearly 400 extra routes feasible when assuming one charger. Not surprisingly, when increasing the number of chargers, the results tend to be more independent of the selected version of the subproblem.

Table 8: Comparison of the results of the branch-and-cut method according to the different versions of the subproblem.

Cap.	Version N			Version D			Version DW			Version DWR		
	#F	#Best	Gap	#F	#Best	Gap	#F	#Best	Gap	#F	#Best	Gap
1	1916	1361	0.315%	2262	2095	0.096%	2270	2112	0.091%	2307	2307	0.000%
2	2321	2125	0.073%	2390	2371	0.013%	2390	2371	0.013%	2390	2390	0.000%

6 Conclusion and perspectives

We have modeled and solved an electric vehicle routing problem that embeds several features such as piecewise linear charging functions, multiple charging technologies, and multiple visits to CSs. Our methodology explicitly considers the fact that the number of EVs simultaneously charging at every CS is limited by the number of chargers. We have proposed two continuous-time formulations of the E-VRP-NL-C: a CS replication-based and a path-based formulation. Our results show that optimally solving even small-size instances is challenging. To handle larger instances, we have developed an algorithmic framework which iteratively calls a route generator and a solution assembler. The first component focuses on generating a pool of high-quality and diversified routes from solutions to the E-VRP-NL obtained by means of an ILS metaheuristic. Computational experiments have shown that this first component produces high quality E-VRP-NL solutions. Indeed, we have improved 80 out of 120 best-known E-VRP-NL solutions. The second component assembles a solution to the E-VRP-NL-C by selecting a subset of routes from the generated pool. We decomposed this assembly problem into a route selection problem and a CS capacity management subproblem. We have designed a branch-and-cut method based on this decomposition scheme. We have developed and compared four versions of the CS capacity management subproblem, including a simple check of the capacity constraints, the introduction of waiting times, and the revision of the charging amounts. Our results show that there exists a serious risk of ending up with no solution if the method disregards the capacity constraints or if the routes cannot be modified by the subproblem. Delaying the starting time and revising the amount of energy charged at the visited CSs lessens this risk. Using more complex strategies to solve capacity violation issues at CSs does not significantly increase the computation time and yields better solutions. The matheuristic is also able to find all optimal solutions of small-size instances.

Acknowledgments— This research was partly funded by the French Agence Nationale de la Recherche through project e-VRO (ANR-15-CE22-0005-01) and by the Canadian Natural Sciences and Engineering Research Council under grants 436014-2013 and 2015-06189. This support is gratefully acknowledged.

A Experiments on the feasibility of solutions from the literature when considering capacitated CSs

We verified the feasibility of the 120 best-known solutions (BKSs) for the E-VRP-NL reported in Montoya et al. (2017), by limiting the number of chargers per CS to one, two, three, and four. Table 9 presents the results of our experiments. We observe that if there exists only one charger at each CS, then almost half of the solutions are infeasible (i.e., they violate the capacity constraints). This proportion drops to 11% when allowing two chargers per station, and four chargers at each CS are needed to ensure the feasibility of all solutions. In practice, however, there are usually only one or two chargers available at each CS.

Table 9: Results of the feasibility tests performed on the best solutions obtained in (Montoya et al., 2017) (All CSs have the same number of chargers).

Experiment	# Infeasible solutions	Average duration with capacity violation	Average #EVs during capacity violation
$\mathcal{C}_1$	55/120	32 min	2.1
$\mathcal{C}_2$	23/120	23 min	3.2
$\mathcal{C}_3$	3/120	33 min	4.0
$\mathcal{C}_4$	0/120	0 min	0.0

B Implementation details for the ILS

Our implementation of the ILS is based on the static move descriptor (SMD) concept. An SMD is static information that describes a move independently of the current solution. A cost tag is associated with every SMD to store its evaluation and it is updated throughout the LS. By storing SMDs in special data structures, one can access and update the moves in an efficient way in order to reduce computation time. Specifically, SMDs are stored in a priority queue (PQ) and organized according to their dynamic cost tag. The exploration of a neighborhood consists in looping over the PQ until a feasible SMD can be applied. Then, to avoid unnecessary reevaluations of moves, only the cost tag of the SMDs impacted by the previously applied SMD is updated. One major difference with the previous use of the SMD framework lies in the complexity of the evaluation of the moves. To our knowledge, until now there has been an emphasis on feasibility issues when using the SMD framework. Indeed, since checking the feasibility of each move can be time consuming, this is only done during the exploration phase of the neighborhood. In our case, not only the feasibility of a move can be computationally expensive to check, but also its evaluation.

Our SMD implementation is inspired from the one proposed by Beek et al. (2018) who implemented the PQ with a binary heap[2]. For every operator, we use a matrix data structure to store all the SMDs. We build these matrices during the initialization phase of the algorithm and we store them during the entire runtime of the algorithm. To avoid a computationally expensive initialization phase of the ILS, the cost tag of each SMD is set equal to the value computed after the first step of the evaluation (see §4.1.1). Since we are only interested in the moves that can potentially improve the current solution, only those SMDs with a cost tag strictly below 0 need consideration (we refer to them as improving SMDs). Throughout the algorithm, we use two binary heaps to store potentially improving SMDs: an "exact" binary heap (EBH) for the SMDs that are exactly evaluated and a "approximate" binary heap (ABH) for the remaining SMDs. In the initialization of the algorithm, we add to EBH all the improving SMDs that are exactly evaluated using the first step, and we add to ABH the other improving SMDs. When we start exploring a neighborhood, we apply the first move in EBH if it is not empty. Otherwise, we iterate over ABH. We apply the second step of the evaluation to compute the exact cost tag of each SMD in ABH until we find an improving move or we reach the end of the heap. After selecting a move m, we update the values of the SMDs if they contain at least one node in the routes impacted by

[2]A binary heap is a complete binary tree satisfying the heap ordering property: the value of the key stored in each node is less than or equal to the value of the keys in the node's children.

m. Although in some cases the cost tag of a certain number of these SMDs may remain identical, we want to keep the SMD framework as simple as possible. To avoid a computationally expensive update phase, each cost tag is set equal to the value computed during the first step evaluation. It can therefore be a lower bound on the exact evaluation of the move. In contrast with Beek et al. (2018), we keep cross-operator effects, meaning that we update the cost tag of SMDs associated with other operators. However, we perform this update for the other operators only when the search moves to them. To this end, we associate with each route the iteration number for which has been lastly updated, and to each operator the iteration number for which it has last been called (we increment the iteration number each time the search accepts a move or changes operator).

C Algorithmic details for the ILS

Algorithm 5 describes the general scheme of the VND search phase. Algorithm 6 describes the move evaluation procedure.

Algorithm 5: The VND algorithm

Input : a solution s^0 to the E-VRP-NL and a maximum route duration limit T
Output: a solution s_R to the E-VRP-NL and a solution s_O (possibly equal to `NULL`) to the E-VRP-NL-C

Procedure `VND`(s^0,T)**:**
  `/* We denote` $f(s)$ `the value of the objective function for a solution` s `and we assume` $f(\texttt{NULL}) = +\infty$`*/`
  $s_R \leftarrow s^0$, $s_O \leftarrow$ `NULL`, $k \leftarrow 0$
  $N \leftarrow$ [(1-0,2-0 vertex exchanges), (1-1,2-1,2-2 vertex exchanges), (2-opt intra and inter-routes), separate]
  while $k < |N|$ **do**
    `/* The procedure search(`$N[k], s, T$`) searches an improving solution to` s `in the neighborhood` $N[k]$ `respecting the maximum route duration limit` T`. It uses Algorithm 6 for move evaluation */`
    $s' \leftarrow$ `search`$(N[k], s_R, T)$ `/*` $s' =$ `NULL if no improving solution is found */`
    if s' is a solution to the E-VRP-NL-C and $f(s') < f(s_O)$ **then** $s_O \leftarrow s'$
    if $f(s') < f(s_R)$ **then** $s_R \leftarrow s'$, $k \leftarrow 0$
    else $k \leftarrow k + 1$
  end
  return (s_R, s_O)

D Algorithmic details for the branch-and-cut method

Algorithm 7 describes the procedure to check the capacity constraints for version N of the subproblem. Algorithm 8 and Algorithm 9 provide a detailed description of the fixed part and energy based reasoning algorithms.

Algorithm 6: Evaluation of a move

Input : a move m and a maximum route duration limit T

Output: a couple (c,b) where c is a lower estimation of the move evaluation and b is a boolean equal to `true` if the move is exactly evaluated or if it is a non-improving move and it is equal to `false` otherwise.

```
Procedure EvaluateMove(m,T):
    /* The evaluation of an infeasible move is set to ∞ */
    b ← true, c ← −CurrentDuration(R_m) /* R_m denotes the routes impacted by move m */
    /* Let R'_m denotes the newly created or modified routes obtained after applying move m */
    for each route r ∈ R'_m do
        t ← DurationWithoutDetourToCS(r)
        if t > T then return (∞,true) /* The move m is infeasible */ ;
        e^LB ← EnergyWithoutDetourToCS(r)
        if e^LB ⩽ Q then
            (c_r,b_r) ← (t,true)
        else
            t ← t + MinDurationToDetourToCS(r) /* The computation of the lower bound
            if t > T then return (∞,true) /* The move m is infeasible */ ;
            if the duration of r is in the cache memory then (c_r,b_r) ← (DurationFromMemory(r),true) else (c_r,b_r) ←
              (t,false)
        end
        c ← c + c_r
        if b_r = false then b ← false
    end
    if c ⩾ 0 then b ← true /* The move m is non-improving */
    return (c,b)
```

Algorithm 7: Checking the violation of capacity constraints for version N of the subproblem

Input : a list of charging operations L numbered from 1 to n ($L(i)$ denotes the operation at position i in the list L) / an integer number $C \geqslant 1$ representing the maximum number of operations that can be scheduled simultaneously

Output: a set containing all the maximal subsets of charging operations leading to a violation of the CS capacity constraint

```
Procedure CheckCapacityCut(O,C):
    Sort the operations in L in non-decreasing order of their starting time /* if two charging operations have the same
      starting time, then the charging operation with the minimum duration comes first */
    U ← ∅ /* subsets of charging operations leading to a violation of the CS capacity constraint*/
    U ← {L(1)} /* subset of charging operations currently executed */
    k ← 2, excess ← false
    while k ⩽ n do
        for every operation o ∈ U do
            if S̄(L(k)) ⩾ S̄(o) + Δ̄(o) then
                if excess then 𝒰 ← 𝒰 ∪ {U}, excess ← false
                else U ← U\{o}
            end
        end
        U ← U ∪ {L(k)}
        if |U| > C then excess ← true
    end
    if excess then 𝒰 ← 𝒰 ∪ {U}
    return 𝒰
```

Algorithm 8: The fixed part based reasoning algorithm

Input : a list of charging operations O numbered from 1 to n ($O(k)$ denotes the operation at position k in the list O) / an integer number $C \geqslant 1$ representing the maximum number of operations that can be scheduled simultaneously

Output: a set containing all the maximal subsets of charging operations leading to a violation of the CS capacity constraint

Procedure `FixedPartsAlgorithm`(O,C):
- Remove from O all the charging operations such that $LS(o) \geqslant ES(o) + \bar{\Delta}(o)$
- Sort the operations in O in non-decreasing order of starting time `/* if two charging operations have the same starting time, then the charging operation with the minimum duration comes first */`
- $\mathcal{U} \leftarrow \varnothing$ `/* subsets of charging operations leading to a violation of the CS capacity constraint */`
- $U \leftarrow \{O(1)\}$ `/* subset of charging operations currently executed */`
- $k \leftarrow 2$, excess $\leftarrow false$
- **while** $k \leqslant n$ **do**
  - **for** every operation $o \in U$ **do**
    - **if** $LS_{O(k)} \geqslant ES(o) + \bar{\Delta}(o)$ **then**
      - **if** *excess* **then** $\mathcal{U} \leftarrow \mathcal{U} \cup \{U\}$, excess $\leftarrow$ `false`
      - **else** $U \leftarrow U \backslash \{o\}$
    - **end**
  - **end**
  - $U \leftarrow U \cup \{O(k)\}$
  - **if** $|U| > C$ **then** excess $\leftarrow$ `true`
- **end**
- **if** excess **then** $\mathcal{U} \leftarrow \mathcal{U} \cup \{U\}$
- **return** $\mathcal{U}$

Algorithm 9: The energy reasoning based algorithm

Input : a set of charging operations O / an integer number $C \geqslant 1$ representing the maximum number of operations that can be scheduled simultaneously

Output: a set containing all subsets of charging operations leading to a violation of the CS capacity constraint

Procedure `EnergeticAlgorithm`(O,C):
- $T_1 := \bigcup_{o \in O} \left(\{ES(o)\} \cup \{ES(o) + \bar{\Delta}(o)\} \cup \{LS(o)\}\right)$
- $T_2 := \bigcup_{o \in O} \left(\{LS(o) + \bar{\Delta}(o)\} \cup \{ES(o) + \bar{\Delta}(o)\} \cup \{LS(o)\}\right)$
- $T(t) := \bigcup_{o \in O} \{ES(o) + LS(o) + \bar{\Delta}(o) - t\}$
- $T = \{(t_1, t_2) \in T_1 \times T_2\} \cup \{(t_1, t_2) \in T_1 \times T(t_1) : t_1 \in O_1\} \cup \{(t_1, t_2) \in T(t_2) \times T_2 : t_2 \in O_2\}$
- $p_o^+(t_1) = \max\{0, \bar{\Delta}(o) - \max\{0, t_1 - ES(o)\}\}$ `/* duration during which` o `is executed after time` t_1 `if it is scheduled as soon as possible */`
- $p_o^-(t_2) = \max\{0, \bar{\Delta}(o) - \max\{0, LS(o) + \bar{\Delta}(o) - t_2\}\}$ `/* duration during which` o `is executed before time` t_2 `if it is scheduled as late as possible */`
- $W(o, t_1, t_2) = \min\{0, t_2 - t_1, p_o^+(t_1), p_o^-(t_2)\}$
- **for** $(t_1, t_2) \in T_1 \times T_2$ **do**
  - **if** $t_1 < t_2 \wedge \sum_{o \in O} W(o, t_1, t_2) > C(t_2 - t_1)$ **then** $\mathcal{U} \leftarrow \mathcal{U} \bigcup \{\{o \in O : W(o, t_1, t_2) > 0\}\}$
- **end**
- **return** $\mathcal{U}$

E Detailed computational results

Based on notation introduced by Montoya et al. (2017), we write each instance using the symbol $tc\gamma_1 c\gamma_2 s\gamma_3 c\gamma_4\#$ where γ_1 is the method used to place the customers (i.e., 0: randomization, 1: mixture of randomization and clustering, 2: clustering), γ_2 is the number of customers, γ_3 is the number of the CSs, γ_4 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).

E.1 Detailed results for the E-VRP-NL-C formulations

Tables 10 and 11 report the detailed results obtained by the CS replication-based and path-based formulations on the E-VRP-NL-C. The symbol "-" means that no feasible solution has been found by the solver.

E.2 Detailed results for the E-VRP-NL

Tables 12 and 13 report the detailed results obtained by our ILS and two methods from the literature on the E-VRP-NL (Montoya et al., 2017; Koç et al., 2018). The best values (objective function value of the BKS and best average value of the objective function obtained over 10 runs) are indicated in boldface.

E.3 Detailed results for the E-VRP-NL-C

Tables 14, 15, 16, and 17 report the detailed results obtained by our matheuristic on the E-VRP-NL-C instances. The best values (objective function value of the BKS and best average value of the objective function obtained over 10 runs) are indicated in boldface.

Table 10: Detailed results for the CS replication-based and path-based MILP formulations on the 10-customer and 20-customer instances if the number of chargers at each CS is equal to 1.

Instance	$[F^{CSrep}]$ ($\beta = 4$)		$[F^{CSrep}]$ ($\beta = 5$)		$[F^{path}]$	
	Obj	Time	Obj	Time	Obj	Time
tc0c10s2cf1	19.75	112	19.75	123	19.75	9
tc0c10s2ct1	12.30	76	12.30	115	12.30	21
tc0c10s3cf1	19.75	10800	19.75	10800	19.76	832
tc0c10s3ct1	10.80	11	10.80	37	10.80	25
tc1c10s2cf2	9.03	7	9.03	10	9.03	8
tc1c10s2cf3	16.37	49	16.37	77	16.37	46
tc1c10s2cf4	16.10	30	16.10	60	16.10	12
tc1c10s2ct2	10.75	127	10.75	159	10.75	446
tc1c10s2ct3	13.17	41	13.17	54	13.17	6
tc1c10s2ct4	13.83	32	13.83	23	13.83	5
tc1c10s3cf2	9.03	6	9.03	11	9.03	14
tc1c10s3cf3	16.37	1558	16.37	1433	16.37	299
tc1c10s3cf4	14.90	65	14.90	252	14.90	36
tc1c10s3ct2	9.20	243	9.20	956	9.20	272
tc1c10s3ct3	13.02	1105	13.02	515	13.02	156
tc1c10s3ct4	13.21	90	13.21	466	13.21	36
tc2c10s2cf0	21.77	10800	21.83	10800	21.77	988
tc2c10s2ct0	12.45	2391	12.45	3750	12.45	482
tc2c10s3cf0	21.83	10800	21.77	10800	21.77	3444
tc2c10s3ct0	11.51	10800	11.51	10800	11.51	4309
tc0c20s3cf2	-	10800	27.47	10800	27.49	10800
tc0c20s3ct2	17.08	10800	17.45	10800	17.08	10800
tc0c20s4cf2	-	10800	-	10800	27.51	10800
tc0c20s4ct2	17.22	10800	17.35	10800	17.24	10800
tc1c20s3cf1	17.49	10800	17.49	10800	17.49	10800
tc1c20s3cf3	16.80	10800	16.81	10800	16.83	10800
tc1c20s3cf4	17.00	179	17.00	197	17.00	573
tc1c20s3ct1	18.94	10800	18.94	10800	19.39	10800
tc1c20s3ct3	12.60	4315	12.60	1181	12.60	10800
tc1c20s3ct4	16.21	277	16.21	341	16.21	513
tc1c20s4cf1	16.38	10800	16.38	10800	16.38	10800
tc1c20s4cf3	16.80	10800	16.80	10800	16.86	10800
tc1c20s4cf4	17.00	1309	17.00	640	-	10800
tc1c20s4ct1	17.81	10800	17.81	10800	19.12	10800
tc1c20s4ct3	14.43	10800	14.43	10800	14.43	10800
tc1c20s4ct4	17.00	1241	17.00	1184	-	10800
tc2c20s3cf0	25.05	10800	-	10800	24.85	10800
tc2c20s3ct0	25.79	10800	25.80	10800	25.82	10800
tc2c20s4cf0	-	10800	-	10800	24.73	10800
tc2c20s4ct0	-	10800	-	10800	26.36	10800

Table 11: Detailed results for the CS replication-based and path-based MILP formulations on the 10-customer and 20-customer instances if the number of chargers at each CS is equal to 2.

Instance	$[F^{CSrep}]$ ($\beta = 4$)		$[F^{CSrep}]$ ($\beta = 5$)		$[F^{path}]$	
	Obj	Time	Obj	Time	Obj	Time
tc0c10s2cf1	19.75	46	19.75	195	19.75	9
tc0c10s2ct1	12.30	105	12.30	172	12.30	22
tc0c10s3cf1	19.75	10800	19.75	10800	19.75	829
tc0c10s3ct1	10.80	14	10.80	23	10.80	28
tc1c10s2cf2	9.03	6	9.03	2	9.03	9
tc1c10s2cf3	16.37	58	16.37	60	16.37	53
tc1c10s2cf4	16.10	22	16.10	51	16.10	13
tc1c10s2ct2	10.75	139	10.75	166	10.75	535
tc1c10s2ct3	13.17	22	13.17	61	13.17	6
tc1c10s2ct4	13.83	24	13.83	64	13.83	5
tc1c10s3cf2	9.03	2	9.03	25	9.03	16
tc1c10s3cf3	16.37	1328	16.37	2994	16.37	306
tc1c10s3cf4	14.90	59	14.90	235	14.90	43
tc1c10s3ct2	9.20	120	9.20	535	9.20	302
tc1c10s3ct3	13.02	1567	13.02	789	13.02	188
tc1c10s3ct4	13.21	81	13.21	83	13.21	40
tc2c10s2cf0	21.77	10800	21.77	10800	21.77	1224
tc2c10s2ct0	12.45	3305	12.45	3847	12.45	577
tc2c10s3cf0	21.77	10800	21.77	10800	21.77	4002
tc2c10s3ct0	11.51	10800	11.51	10800	11.51	4222
tc0c20s3cf2	-	10800	-	10800	27.49	10800
tc0c20s3ct2	17.08	10800	17.08	10800	17.08	10800
tc0c20s4cf2	-	10800	-	10800	27.51	10800
tc0c20s4ct2	16.99	10800	17.22	10800	17.43	10800
tc1c20s3cf1	17.49	10800	17.49	10800	17.49	10800
tc1c20s3cf3	16.45	10800	16.80	10800	16.83	10800
tc1c20s3cf4	17.00	108	17.00	177	17.00	712
tc1c20s3ct1	18.94	10800	18.94	10800	19.39	10800
tc1c20s3ct3	12.60	529	12.60	4341	12.60	10800
tc1c20s3ct4	16.21	309	16.21	329	16.21	599
tc1c20s4cf1	16.38	10800	16.38	10800	16.38	10800
tc1c20s4cf3	16.80	10800	16.80	10800	17.04	10800
tc1c20s4cf4	17.00	545	17.00	1530	20.06	10800
tc1c20s4ct1	18.02	10800	18.25	10800	19.12	10800
tc1c20s4ct3	14.43	10800	14.43	10800	14.43	10800
tc1c20s4ct4	17.00	1495	17.00	559	-	10800
tc2c20s3cf0	-	10800	24.87	10800	24.85	10800
tc2c20s3ct0	25.79	10800	25.79	10800	25.83	10800
tc2c20s4cf0	24.86	10800	-	10800	24.73	10800
tc2c20s4ct0	-	10800	-	10800	26.36	10800

Table 12: Detailed comparison of the results obtained by ILS with the results of Montoya et al. (2017) and Koç et al. (2018) for the E-VRP-NL (instances with 10, 20, or 40 customers)

		ILS + HC		LNS		ILS			ILS + HC		LNS		ILS	
Instance	BKS	Best	#R	Best	#R	Best	#R	BA	Avg	Time	Avg	Time	Avg	Time
tc0c10s2cf1	19.75	**19.75**	3	**19.75**	3	**19.75**	3	19.75	20.12	4	19.77	8	**19.75**	5
tc0c10s2ct1	12.30	**12.30**	2	**12.30**	2	**12.30**	2	12.30	12.34	4	12.31	8	**12.30**	3
tc0c10s3cf1	19.75	**19.75**	3	**19.75**	3	**19.75**	3	19.75	20.12	4	19.76	7	**19.75**	5
tc0c10s3ct1	10.80	**10.80**	2	**10.80**	2	**10.80**	2	10.80	**10.80**	5	10.81	8	**10.80**	5
tc1c10s2cf2	9.03	**9.03**	3	**9.03**	3	**9.03**	3	9.03	9.07	2	9.04	9	**9.03**	4
tc1c10s2cf3	16.37	**16.37**	3	**16.37**	3	**16.37**	3	16.37	**16.37**	6	16.38	9	**16.37**	5
tc1c10s2cf4	16.10	**16.10**	3	**16.10**	3	**16.10**	3	16.10	**16.10**	5	16.11	7	**16.10**	5
tc1c10s2ct2	10.75	**10.75**	3	**10.75**	3	**10.75**	3	10.75	**10.75**	4	10.76	8	**10.75**	3
tc1c10s2ct3	13.17	**13.17**	2	**13.17**	2	**13.17**	2	13.17	13.18	8	13.18	9	**13.17**	6
tc1c10s2ct4	13.83	**13.83**	2	**13.83**	2	**13.83**	2	13.83	**13.83**	5	13.84	9	**13.83**	5
tc1c10s3cf2	9.03	**9.03**	3	**9.03**	3	**9.03**	3	9.03	9.06	2	9.04	10	**9.03**	5
tc1c10s3cf3	16.37	**16.37**	3	**16.37**	3	**16.37**	3	16.37	**16.37**	6	16.39	8	**16.37**	3
tc1c10s3cf4	14.90	**14.90**	3	**14.90**	3	**14.90**	3	14.90	**14.90**	7	14.91	8	**14.90**	2
tc1c10s3ct2	9.20	**9.20**	3	**9.20**	3	**9.20**	3	9.20	9.34	5	9.21	9	**9.20**	6
tc1c10s3ct3	13.02	**13.02**	2	**13.02**	2	**13.02**	2	13.02	**13.02**	10	13.03	7	**13.02**	5
tc1c10s3ct4	13.21	**13.21**	2	**13.21**	2	**13.21**	2	13.21	**13.21**	6	13.22	9	**13.21**	5
tc2c10s2cf0	21.77	**21.77**	3	**21.77**	3	**21.77**	3	21.77	**21.77**	9	21.78	8	**21.77**	6
tc2c10s2ct0	12.45	**12.45**	3	**12.45**	3	**12.45**	3	12.45	**12.45**	5	12.46	8	**12.45**	6
tc2c10s3cf0	21.77	**21.77**	3	**21.77**	3	**21.77**	3	21.77	**21.77**	9	21.79	7	**21.77**	3
tc2c10s3ct0	11.51	**11.51**	3	**11.51**	3	**11.51**	3	11.51	11.54	7	11.52	9	**11.51**	5
tc0c20s3cf2	27.47	27.60	4	**27.47**	4	**27.47**	4	27.47	27.66	12	27.52	12	**27.47**	10
tc0c20s3ct2	17.08	**17.08**	3	**17.08**	3	**17.08**	3	17.08	17.13	8	17.11	18	**17.08**	6
tc0c20s4cf2	27.47	27.48	4	27.60	4	**27.47**	4	27.47	27.61	13	27.65	14	**27.47**	9
tc0c20s4ct2	16.99	**16.99**	3	**16.99**	3	**16.99**	3	16.99	17.10	9	17.02	16	**16.99**	9
tc1c20s3cf1	17.49	17.50	3	17.50	3	**17.49**	3	17.49	17.53	12	17.53	13	**17.49**	10
tc1c20s3cf3	16.44	16.63	4	16.48	3	**16.44**	3	16.44	16.78	8	16.50	17	**16.44**	7
tc1c20s3cf4	17.00	**17.00**	4	**17.00**	4	**17.00**	4	17.00	**17.00**	4	17.03	15	**17.00**	5
tc1c20s3ct1	18.94	18.95	4	18.95	4	**18.94**	4	18.94	19.38	15	18.97	14	**18.94**	9
tc1c20s3ct3	12.60	12.65	3	**12.60**	3	**12.60**	3	12.60	12.72	9	12.62	17	**12.60**	9
tc1c20s3ct4	16.21	**16.21**	4	**16.21**	4	**16.21**	4	16.21	16.25	5	16.24	11	**16.21**	8
tc1c20s4cf1	16.38	16.39	4	16.47	3	**16.38**	4	16.38	16.40	13	16.49	18	**16.38**	6
tc1c20s4cf3	16.44	16.56	3	16.48	3	**16.44**	3	16.44	16.80	9	16.51	11	**16.44**	11
tc1c20s4cf4	17.00	**17.00**	4	**17.00**	4	**17.00**	4	17.00	**17.00**	4	17.03	15	**17.00**	8
tc1c20s4ct1	17.80	18.25	4	18.25	4	**17.80**	4	17.80	18.32	16	18.28	18	**17.80**	11
tc1c20s4ct3	14.43	**14.43**	3	**14.43**	3	**14.43**	3	14.43	14.50	8	14.46	12	**14.43**	7
tc1c20s4ct4	17.00	**17.00**	4	**17.00**	4	**17.00**	4	17.00	**17.00**	6	17.03	11	**17.00**	6
tc2c20s3cf0	24.68	**24.68**	4	**24.68**	4	**24.68**	4	24.68	**24.68**	14	24.70	11	**24.68**	7
tc2c20s3ct0	25.79	**25.79**	4	**25.79**	4	**25.79**	4	25.79	**25.79**	15	25.83	15	**25.79**	10
tc2c20s4cf0	24.67	**24.67**	4	**24.67**	4	**24.67**	4	24.67	24.69	15	24.71	13	**24.67**	11
tc2c20s4ct0	26.02	**26.02**	4	**26.03**	4	**26.02**	4	26.02	**26.02**	15	26.07	16	**26.02**	9
tc0c40s5cf0	32.20	32.67	8	32.67	8	**32.20**	7	32.30	33.25	24	32.75	52	**32.30**	16
tc0c40s5cf4	30.25	30.77	6	30.60	6	**30.25**	6	30.25	31.49	33	30.69	49	**30.25**	22
tc0c40s5ct0	27.91	28.72	7	28.70	7	**27.91**	6	27.91	29.35	25	28.78	46	**27.91**	17
tc0c40s5ct4	28.63	**28.63**	6	**29.17**	5	**28.63**	6	28.63	28.72	33	29.25	59	**28.63**	18
tc0c40s8cf0	30.40	31.28	7	31.23	7	**30.40**	6	30.40	32.02	34	31.31	63	**30.40**	18
tc0c40s8cf4	28.11	29.32	6	28.25	5	**28.11**	5	28.23	29.86	43	28.30	52	**28.23**	25
tc0c40s8ct0	26.22	26.35	6	**26.22**	6	**26.22**	6	26.22	26.89	29	26.27	58	**26.22**	17
tc0c40s8ct4	29.07	29.20	6	29.22	6	**29.07**	5	29.07	29.27	47	29.28	48	**29.07**	22
tc1c40s5cf1	64.51	65.16	10	65.52	10	**64.51**	10	64.51	66.03	44	65.67	33	**64.51**	25
tc1c40s5ct1	52.33	52.68	9	52.60	9	**52.33**	8	52.33	53.36	59	52.72	40	**52.33**	23
tc1c40s8cf1	40.64	40.75	7	41.63	7	**40.64**	7	40.64	42.33	70	41.71	34	**40.64**	21
tc1c40s8ct1	40.18	40.56	7	40.56	7	**40.18**	7	40.18	41.19	71	40.67	49	**40.18**	24
tc2c40s5cf2	27.54	**27.54**	6	**27.54**	6	**27.54**	6	27.54	27.67	32	27.62	42	**27.54**	17
tc2c40s5cf3	19.65	19.74	5	**19.65**	5	**19.65**	5	19.65	20.18	17	19.70	50	**19.65**	21
tc2c40s5ct2	26.91	**26.91**	6	**26.91**	6	**26.91**	6	26.91	27.02	23	26.99	42	**26.91**	14
tc2c40s5ct3	23.39	23.54	6	23.71	6	**23.39**	6	23.39	23.77	26	23.75	51	**23.39**	22
tc2c40s8cf2	27.13	27.15	6	27.14	6	**27.13**	6	27.13	27.31	29	27.20	35	**27.13**	16
tc2c40s8cf3	19.65	19.66	5	**19.65**	5	**19.65**	5	19.65	20.24	19	19.69	36	**19.65**	22
tc2c40s8ct2	26.28	26.33	6	26.29	6	**26.28**	6	26.28	26.71	26	26.34	40	**26.28**	16
tc2c40s8ct3	22.45	22.71	5	**22.45**	5	**22.45**	5	22.45	23.23	25	22.52	30	**22.45**	24

Table 13: Detailed comparison of the results obtained by ILS with the results of Montoya et al. (2017) and Koç et al. (2018) for the E-VRP-NL (instances with 80, 160, or 320 customers)

		ILS + HC		LNS		ILS			ILS + HC		LNS		ILS	
Instance	BKS	Best	#R	Best	#R	Best	#R	BA	Avg	Time	Avg	Time	Avg	Time
tc0c80s12cf0	34.16	34.64	9	35.24	8	**34.16**	8	34.16	35.59	57	35.40	105	**34.16**	66
tc0c80s12cf1	40.91	42.90	10	42.30	9	**40.91**	9	40.94	44.07	75	42.47	85	**40.94**	68
tc0c80s12ct0	37.51	39.31	9	39.27	9	**37.51**	8	38.08	39.83	66	39.41	86	**38.08**	65
tc0c80s12ct1	39.91	41.94	10	41.64	9	**39.91**	9	40.06	43.03	73	41.83	103	**40.06**	59
tc0c80s8cf0	39.08	39.43	9	40.64	10	**39.08**	9	39.16	39.86	56	40.77	88	**39.16**	48
tc0c80s8cf1	43.38	45.23	10	46.65	9	**43.38**	9	43.95	45.73	121	46.80	98	**43.95**	73
tc0c80s8ct0	40.52	41.90	10	41.44	9	**40.52**	9	41.44	42.76	54	41.59	87	**41.44**	61
tc0c80s8ct1	43.85	45.27	10	45.25	10	**43.85**	9	44.07	45.85	130	45.37	100	**44.07**	73
tc1c80s12cf2	28.65	29.54	8	29.54	8	**28.65**	7	28.77	30.73	61	29.66	113	**28.77**	52
tc1c80s12ct2	28.73	29.52	8	29.38	8	**28.73**	8	29.18	30.66	59	29.47	114	**29.18**	54
tc1c80s8cf2	29.15	30.81	8	31.38	8	**29.15**	8	29.15	31.83	51	31.47	94	**29.15**	51
tc1c80s8ct2	30.45	31.74	8	31.72	8	**30.45**	8	30.52	32.36	60	31.82	98	**30.52**	57
tc2c80s12cf3	30.60	31.97	9	31.28	8	**30.60**	8	30.60	32.70	76	31.37	105	**30.60**	57
tc2c80s12cf4	42.10	43.89	9	43.69	9	**42.10**	9	42.14	44.97	131	43.81	86	**42.14**	83
tc2c80s12ct3	29.90	30.83	9	30.31	8	**29.90**	8	29.90	31.59	58	30.39	114	**29.90**	54
tc2c80s12ct4	40.27	42.40	9	42.56	9	**40.27**	9	40.27	42.82	134	44.68	103	**40.27**	74
tc2c80s8cf3	31.70	32.44	9	31.94	8	**31.70**	8	31.93	32.60	64	32.06	87	**31.93**	55
tc2c80s8cf4	46.03	49.29	10	49.67	10	**46.03**	9	46.78	49.69	100	49.84	128	**46.78**	93
tc2c80s8ct3	31.38	32.31	9	32.71	9	**31.38**	8	31.43	32.55	65	32.82	89	**31.43**	65
tc2c80s8ct4	43.83	44.83	10	44.16	10	**43.83**	9	44.00	46.61	111	44.31	103	**44.00**	75
tc0c160s16cf2	57.91	61.20	16	62.09	15	**57.91**	15	58.00	62.99	365	62.55	442	**58.00**	242
tc0c160s16cf4	76.90	82.92	18	82.77	18	**76.90**	16	77.55	83.84	1213	83.41	709	**77.55**	367
tc0c160s16ct2	57.64	59.90	15	59.75	15	**57.64**	15	57.73	62.80	342	60.29	811	**57.73**	247
tc0c160s16ct4	76.14	82.37	18	82.90	18	**76.14**	16	76.90	83.08	945	83.85	983	**76.90**	353
tc0c160s24cf2	56.32	59.27	15	59.26	15	**56.32**	14	56.76	60.92	403	59.79	732	**56.76**	253
tc0c160s24cf4	75.53	81.44	18	81.43	18	**75.53**	16	76.30	82.13	1209	82.33	595	**76.30**	370
tc0c160s24ct2	55.42	59.25	16	59.67	16	**55.42**	14	56.47	60.19	410	60.21	915	**56.47**	253
tc0c160s24ct4	75.05	80.96	18	81.38	18	**75.05**	16	75.87	82.11	957	82.21	436	**75.87**	372
tc1c160s16cf0	74.54	79.80	18	79.76	18	**74.54**	16	75.32	80.75	766	80.52	420	**75.32**	327
tc1c160s16cf3	66.45	71.76	17	71.98	17	**66.45**	15	67.20	72.75	462	72.77	729	**67.20**	307
tc1c160s16ct0	74.20	79.04	17	80.21	17	**74.20**	16	75.31	79.90	643	80.99	472	**75.31**	326
tc1c160s16ct3	65.31	73.29	17	73.24	17	**65.31**	15	66.20	75.11	279	73.82	750	**66.20**	289
tc1c160s24cf0	73.62	78.60	17	79.48	17	**73.62**	16	74.05	79.30	741	80.32	460	**74.05**	331
tc1c160s24cf3	62.90	68.56	17	68.73	17	**62.90**	15	63.64	69.57	483	69.28	522	**63.64**	282
tc1c160s24ct0	73.34	78.21	17	78.32	17	**73.34**	16	74.00	79.35	578	79.05	553	**74.00**	319
tc1c160s24ct3	63.19	68.72	17	69.17	17	**63.19**	15	63.66	69.98	358	69.76	889	**63.66**	280
tc2c160s16cf1	56.65	60.34	16	60.25	15	**56.65**	14	57.39	61.26	274	60.70	716	**57.39**	252
tc2c160s16ct1	55.37	60.27	15	59.86	15	**55.37**	14	55.52	60.62	288	60.40	408	**55.52**	232
tc2c160s24cf1	56.70	59.82	16	60.01	16	**56.70**	14	57.27	61.14	305	60.63	564	**57.27**	260
tc2c160s24ct1	55.03	59.13	16	59.97	15	**55.03**	14	55.15	59.72	340	60.53	531	**55.15**	238
tc1c320s24cf2	133.32	152.13	36	153.12	36	**133.32**	31	133.99	153.99	7106	154.65	4155	**133.99**	1287
tc1c320s24cf3	106.43	117.48	30	117.39	30	**106.43**	28	107.00	118.36	3066	118.43	3258	**107.00**	1060
tc1c320s24ct2	131.63	148.77	36	148.57	36	**131.63**	30	132.49	154.13	6853	149.89	4727	**132.49**	1231
tc1c320s24ct3	105.93	116.64	31	117.50	31	**105.93**	27	106.67	119.17	3274	118.53	5105	**106.67**	1045
tc1c320s38cf2	129.19	141.63	33	142.25	33	**129.19**	30	129.76	147.08	7236	144.17	4249	**129.76**	1178
tc1c320s38cf3	106.01	116.22	30	117.31	30	**106.01**	28	106.36	117.74	3114	118.78	5978	**106.36**	1129
tc1c320s38ct2	128.82	140.96	32	142.75	32	**128.82**	30	129.51	145.09	6974	144.50	6078	**129.51**	1167
tc1c320s38ct3	105.73	116.07	30	117.91	30	**105.73**	27	106.74	117.71	3063	119.40	3157	**106.74**	1186
tc2c320s24cf0	158.80	182.45	38	182.90	38	**158.80**	33	160.55	186.94	6566	185.27	4014	**160.55**	1343
tc2c320s24cf1	87.46	95.51	29	95.71	29	**87.46**	26	87.64	96.42	1456	96.81	5150	**87.64**	890
tc2c320s24cf4	111.16	122.74	32	122.83	32	**111.16**	28	111.62	124.68	3681	124.51	3923	**111.62**	989
tc2c320s24ct0	159.70	181.45	37	182.29	37	**159.70**	33	160.49	186.23	7204	183.80	6191	**160.49**	1309
tc2c320s24ct1	87.25	94.73	27	94.97	27	**87.25**	26	87.83	96.49	1259	95.96	3530	**87.83**	863
tc2c320s24ct4	111.09	121.94	32	122.09	32	**111.09**	28	111.62	123.85	4274	123.45	5196	**111.62**	1041
tc2c320s38cf0	158.70	176.92	37	178.17	37	**158.70**	33	159.53	182.31	6734	179.81	3350	**159.53**	1356
tc2c320s38cf1	86.92	94.29	28	95.73	28	**86.92**	26	87.25	95.07	1602	96.79	5343	**87.25**	890
tc2c320s38cf4	109.80	122.32	32	122.26	32	**109.80**	28	110.66	123.47	2661	123.46	3724	**110.66**	1087
tc2c320s38ct0	158.71	190.97	41	192.23	41	**158.71**	33	159.35	192.15	7637	194.66	4448	**159.35**	1374
tc2c320s38ct1	86.59	94.53	28	94.66	28	**86.59**	26	86.97	95.29	1409	95.87	3973	**86.97**	894
tc2c320s38ct4	110.05	121.66	32	121.64	32	**110.05**	28	110.55	123.15	2785	123.06	5554	**110.55**	1034

Table 14: Detailed comparison of the best results obtained by the matheuristic for the E-VRP-NL-C according to the version of the subproblem it uses if the number of chargers at every CS is equal to 1 (instances with 10, 20, or 40 customers).

Instance	Version N				Version D				Version DW				Version DWR				BS	BA
	Time	#F	Best	Avg	Time	#F	Best	Avg	Time	#F	Best	Avg	Time	#F	Best	Avg		
tc0c10s2cf1	5	10	**19.75**	**19.75**	5	10	**19.75**	**19.75**	6	10	**19.75**	**19.75**	5	10	**19.75**	**19.75**	19.75	19.75
tc0c10s2ct1	4	10	**12.30**	**12.30**	4	10	**12.30**	**12.30**	4	10	**12.30**	**12.30**	4	10	**12.30**	**12.30**	12.30	12.30
tc0c10s3cf1	5	10	**19.75**	**19.75**	6	10	**19.75**	**19.75**	6	10	**19.75**	**19.75**	6	10	**19.75**	**19.75**	19.75	19.75
tc0c10s3ct1	5	10	**10.80**	**10.80**	5	10	**10.80**	**10.80**	5	10	**10.80**	**10.80**	5	10	**10.80**	**10.80**	10.80	10.80
tc1c10s2cf2	5	10	**9.03**	**9.03**	5	10	**9.03**	**9.03**	6	10	**9.03**	**9.03**	5	10	**9.03**	**9.03**	9.03	9.03
tc1c10s2cf3	6	10	**16.37**	**16.37**	6	10	**16.37**	**16.37**	6	10	**16.37**	**16.37**	6	10	**16.37**	**16.37**	16.37	16.37
tc1c10s2cf4	5	10	**16.10**	**16.10**	6	10	**16.10**	**16.10**	5	10	**16.10**	**16.10**	6	10	**16.10**	**16.10**	16.10	16.10
tc1c10s2ct2	4	10	**10.75**	**10.75**	4	10	**10.75**	**10.75**	4	10	**10.75**	**10.75**	4	10	**10.75**	**10.75**	10.75	10.75
tc1c10s2ct3	5	10	13.25	13.25	5	10	**13.17**	**13.17**	5	10	**13.17**	**13.17**	5	10	**13.17**	**13.17**	13.17	13.17
tc1c10s2ct4	5	10	**13.83**	**13.83**	5	10	**13.83**	**13.83**	5	10	**13.83**	**13.83**	5	10	**13.83**	**13.83**	13.83	13.83
tc1c10s3cf2	4	10	**9.03**	**9.03**	5	10	**9.03**	**9.03**	5	10	**9.03**	**9.03**	5	10	**9.03**	**9.03**	9.03	9.03
tc1c10s3cf3	4	10	**16.37**	**16.37**	4	10	**16.37**	**16.37**	4	10	**16.37**	**16.37**	4	10	**16.37**	**16.37**	16.37	16.37
tc1c10s3cf4	3	10	**14.90**	**14.90**	4	10	**14.90**	**14.90**	4	10	**14.90**	**14.90**	4	10	**14.90**	**14.90**	14.90	14.90
tc1c10s3ct2	7	10	**9.20**	**9.20**	7	10	**9.20**	**9.20**	7	10	**9.20**	**9.20**	7	10	**9.20**	**9.20**	9.20	9.20
tc1c10s3ct3	5	10	**13.02**	**13.02**	6	10	**13.02**	**13.02**	6	10	**13.02**	**13.02**	6	10	**13.02**	**13.02**	13.02	13.02
tc1c10s3ct4	5	10	**13.21**	**13.21**	6	10	**13.21**	**13.21**	6	10	**13.21**	**13.21**	5	10	**13.21**	**13.21**	13.21	13.21
tc2c10s2cf0	5	10	21.83	21.83	5	10	**21.77**	**21.77**	6	10	**21.77**	**21.77**	6	10	**21.77**	**21.77**	21.77	21.77
tc2c10s2ct0	5	10	**12.45**	**12.45**	5	10	**12.45**	**12.45**	4	10	**12.45**	**12.45**	5	10	**12.45**	**12.45**	12.45	12.45
tc2c10s3cf0	4	10	21.83	21.83	4	10	**21.77**	**21.77**	4	10	**21.77**	**21.77**	4	10	**21.77**	**21.77**	21.77	21.77
tc2c10s3ct0	6	10	**11.51**	**11.51**	6	10	**11.51**	**11.51**	6	10	**11.51**	**11.51**	6	10	**11.51**	**11.51**	11.51	11.51
tc0c20s3cf2	12	10	27.62	27.62	12	10	**27.47**	**27.47**	11	10	**27.47**	**27.47**	12	10	**27.47**	**27.47**	27.47	27.47
tc0c20s3ct2	8	10	**17.08**	**17.08**	8	10	**17.08**	**17.08**	7	10	**17.08**	**17.08**	8	10	**17.08**	**17.08**	17.08	17.08
tc0c20s4cf2	10	10	27.62	27.62	10	10	**27.47**	**27.47**	10	10	**27.47**	**27.47**	9	10	**27.47**	**27.47**	27.47	27.47
tc0c20s4ct2	10	10	**16.99**	**16.99**	10	10	**16.99**	**16.99**	10	10	**16.99**	**16.99**	11	10	**16.99**	**16.99**	16.99	16.99
tc1c20s3cf1	12	10	**17.49**	**17.49**	12	10	**17.49**	**17.49**	13	10	**17.49**	**17.49**	12	10	**17.49**	**17.49**	17.49	17.49
tc1c20s3cf3	9	10	**16.44**	**16.44**	9	10	**16.44**	**16.44**	9	10	**16.44**	**16.44**	10	10	**16.44**	**16.44**	16.44	16.44
tc1c20s3cf4	6	10	**17.00**	**17.00**	7	10	**17.00**	**17.00**	6	10	**17.00**	**17.00**	6	10	**17.00**	**17.00**	17.00	17.00
tc1c20s3ct1	11	10	**18.94**	**18.94**	10	10	**18.94**	**18.94**	11	10	**18.94**	**18.94**	11	10	**18.94**	**18.94**	18.94	18.94
tc1c20s3ct3	11	10	**12.60**	**12.60**	11	10	**12.60**	**12.60**	11	10	**12.60**	**12.60**	11	10	**12.60**	**12.60**	12.60	12.60
tc1c20s3ct4	9	10	**16.21**	**16.21**	10	10	**16.21**	**16.21**	9	10	**16.21**	**16.21**	9	10	**16.21**	**16.21**	16.21	16.21
tc1c20s4cf1	8	10	**16.38**	**16.38**	8	10	**16.38**	**16.38**	8	10	**16.38**	**16.38**	8	10	**16.38**	**16.38**	16.38	16.38
tc1c20s4cf3	13	10	**16.44**	**16.44**	12	10	**16.44**	**16.44**	12	10	**16.44**	**16.44**	13	10	**16.44**	**16.44**	16.44	16.44
tc1c20s4cf4	10	10	**17.00**	**17.00**	10	10	**17.00**	**17.00**	9	10	**17.00**	**17.00**	10	10	**17.00**	**17.00**	17.00	17.00
tc1c20s4ct1	12	10	**17.80**	**17.80**	13	10	**17.80**	**17.80**	12	10	**17.80**	**17.80**	12	10	**17.80**	**17.80**	17.80	17.80
tc1c20s4ct3	7	10	**14.43**	**14.43**	7	10	**14.43**	**14.43**	7	10	**14.43**	**14.43**	8	10	**14.43**	**14.43**	14.43	14.43
tc1c20s4ct4	8	10	**17.00**	**17.00**	8	10	**17.00**	**17.00**	8	10	**17.00**	**17.00**	8	10	**17.00**	**17.00**	17.00	17.00
tc2c20s3cf0	9	10	**24.68**	**24.68**	9	10	**24.68**	**24.68**	8	10	**24.68**	**24.68**	9	10	**24.68**	**24.68**	24.68	24.68
tc2c20s3ct0	12	10	25.80	25.80	12	10	**25.79**	**25.79**	12	10	**25.79**	**25.79**	12	10	**25.79**	**25.79**	25.79	25.79
tc2c20s4cf0	12	10	24.68	24.68	12	10	**24.67**	**24.67**	12	10	**24.67**	**24.67**	12	10	**24.67**	**24.67**	24.67	24.67
tc2c20s4ct0	11	10	26.17	26.18	11	10	**26.02**	**26.02**	12	10	**26.02**	**26.02**	12	10	**26.02**	**26.02**	26.02	26.02
tc0c40s5cf0	21	10	**32.20**	**32.23**	20	10	**32.20**	**32.23**	20	10	**32.20**	**32.23**	20	10	**32.20**	**32.23**	32.20	32.23
tc0c40s5cf4	26	10	**30.25**	**30.25**	25	10	**30.25**	**30.25**	25	10	**30.25**	**30.25**	26	10	**30.25**	**30.25**	30.25	30.25
tc0c40s5ct0	20	10	**27.91**	**27.91**	20	10	**27.91**	**27.91**	21	10	**27.91**	**27.91**	21	10	**27.91**	**27.91**	27.91	27.91
tc0c40s5ct4	23	10	**28.63**	**28.63**	23	10	**28.63**	**28.63**	22	10	**28.63**	**28.63**	22	10	**28.63**	**28.63**	28.63	28.63
tc0c40s8cf0	22	10	30.46	30.51	21	10	**30.40**	**30.40**	21	10	**30.40**	**30.40**	21	10	**30.40**	**30.40**	30.40	30.40
tc0c40s8cf4	28	10	**28.24**	**28.25**	29	10	**28.24**	**28.25**	29	10	**28.24**	**28.25**	28	10	**28.24**	**28.25**	28.24	28.25
tc0c40s8ct0	20	10	**26.22**	**26.22**	20	10	**26.22**	**26.22**	20	10	**26.22**	**26.22**	20	10	**26.22**	**26.22**	26.22	26.22
tc0c40s8ct4	27	10	**29.07**	**29.07**	26	10	**29.07**	**29.07**	26	10	**29.07**	**29.07**	27	10	**29.07**	**29.07**	29.07	29.07
tc1c40s5cf1	27	0	-	-	14	0	-	-	14	0	-	-	1299	0	-	-	-	-
tc1c40s5ct1	26	10	52.42	52.70	28	10	52.42	52.60	26	10	52.42	52.60	27	10	**52.33**	**52.33**	52.33	52.33
tc1c40s8cf1	25	10	41.67	42.21	24	10	**40.64**	**40.64**	23	10	**40.64**	**40.64**	24	10	**40.64**	**40.64**	40.64	40.64
tc1c40s8ct1	29	10	40.45	40.73	28	10	**40.35**	40.36	28	10	**40.35**	40.36	29	10	**40.35**	**40.35**	40.35	40.35
tc2c40s5cf2	20	10	**27.54**	**27.54**	19	10	**27.54**	**27.54**	20	10	**27.54**	**27.54**	21	10	**27.54**	**27.54**	27.54	27.54
tc2c40s5cf3	26	10	**19.65**	**19.65**	25	10	**19.65**	**19.65**	25	10	**19.65**	**19.65**	26	10	**19.65**	**19.65**	19.65	19.65
tc2c40s5ct2	20	10	**26.91**	**26.91**	19	10	**26.91**	**26.91**	19	10	**26.91**	**26.91**	19	10	**26.91**	**26.91**	26.91	26.91
tc2c40s5ct3	23	10	**23.39**	**23.39**	22	10	**23.39**	**23.39**	22	10	**23.39**	**23.39**	21	10	**23.39**	**23.39**	23.39	23.39
tc2c40s8cf2	20	10	**27.13**	**27.13**	20	10	**27.13**	**27.13**	20	10	**27.13**	**27.13**	20	10	**27.13**	**27.13**	27.13	27.13
tc2c40s8cf3	26	10	**19.65**	**19.65**	25	10	**19.65**	**19.65**	26	10	**19.65**	**19.65**	24	10	**19.65**	**19.65**	19.65	19.65
tc2c40s8ct2	19	10	**26.28**	**26.28**	19	10	**26.28**	**26.28**	18	10	**26.28**	**26.28**	18	10	**26.28**	**26.28**	26.28	26.28
tc2c40s8ct3	27	10	**22.45**	**22.45**	28	10	**22.45**	**22.45**	28	10	**22.45**	**22.45**	28	10	**22.45**	**22.45**	22.45	22.45

Table 15: Detailed comparison of the best results obtained by the matheuristic for the E-VRP-NL-C according to the version of the subproblem it uses if the number of charger at every CS is equal to 1(instances with 80, 160, or 320 customers).

Instance	Version N				Version D				Version DW				Version DWR				BS	BA
	Time	#F	Best	Avg	Time	#F	Best	Avg	Time	#F	Best	Avg	Time	#F	Best	Avg		
tc0c80s12cf0	74	10	**34.06**	**34.20**	75	10	**34.06**	**34.20**	78	10	**34.06**	**34.20**	77	10	**34.06**	**34.20**	34.06	34.20
tc0c80s12cf1	78	10	**40.83**	40.95	77	10	**40.83**	**40.94**	77	10	**40.83**	**40.94**	77	10	**40.83**	**40.94**	40.83	40.94
tc0c80s12ct0	76	10	**37.51**	37.95	77	10	**37.51**	**37.89**	77	10	**37.51**	**37.89**	77	10	**37.51**	37.98	37.51	37.89
tc0c80s12ct1	72	10	**39.72**	40.01	73	10	**39.72**	**40.00**	71	10	**39.72**	**40.00**	72	10	**39.72**	**40.00**	39.72	40.00
tc0c80s8cf0	61	10	**38.59**	**38.79**	61	10	**38.59**	**38.79**	62	10	**38.59**	**38.79**	60	10	**38.59**	**38.79**	38.59	38.79
tc0c80s8cf1	83	10	43.41	44.28	80	10	**43.38**	**44.05**	81	10	**43.38**	**44.05**	85	10	**43.38**	44.13	43.38	44.05
tc0c80s8ct0	71	10	**40.53**	**41.47**	71	10	**40.53**	**41.47**	69	10	**40.53**	**41.47**	69	10	**40.53**	**41.47**	40.53	41.47
tc0c80s8ct1	82	10	43.86	43.99	82	10	**43.85**	**43.98**	80	10	**43.85**	**43.98**	82	10	**43.85**	**43.98**	43.85	43.98
tc1c80s12cf2	58	10	**28.65**	**28.77**	59	10	**28.65**	**28.77**	58	10	**28.65**	**28.77**	59	10	**28.65**	**28.77**	28.65	28.77
tc1c80s12ct2	60	10	**28.73**	29.17	60	10	**28.73**	**29.09**	59	10	**28.73**	**29.09**	62	10	**28.73**	29.14	28.73	29.09
tc1c80s8cf2	63	10	**29.15**	**29.15**	62	10	**29.15**	**29.15**	60	10	**29.15**	**29.15**	62	10	**29.15**	**29.15**	29.15	29.15
tc1c80s8ct2	61	10	**29.88**	**30.42**	63	10	**29.88**	**30.42**	62	10	**29.88**	**30.42**	63	10	**29.88**	**30.42**	29.88	30.42
tc2c80s12cf3	63	10	30.61	30.61	62	10	**30.60**	**30.60**	63	10	**30.60**	**30.60**	62	10	**30.60**	**30.60**	30.60	30.60
tc2c80s12cf4	89	10	42.20	42.62	88	10	**42.12**	**42.25**	89	10	**42.12**	**42.25**	105	10	**42.12**	42.26	42.12	42.25
tc2c80s12ct3	62	10	**29.90**	**29.90**	61	10	**29.90**	**29.90**	62	10	**29.90**	**29.90**	61	10	**29.90**	**29.90**	29.90	29.90
tc2c80s12ct4	87	10	40.28	40.28	86	10	**40.27**	**40.27**	85	10	**40.27**	**40.27**	86	10	**40.27**	**40.27**	40.27	40.27
tc2c80s8cf3	61	10	**31.70**	**31.93**	60	10	**31.70**	**31.93**	61	10	**31.70**	**31.93**	60	10	**31.70**	**31.93**	31.70	31.93
tc2c80s8cf4	100	10	**46.19**	46.85	102	10	**46.19**	**46.84**	104	10	**46.19**	**46.84**	102	10	**46.19**	46.94	46.19	46.84
tc2c80s8ct3	75	10	**31.38**	**31.38**	76	10	**31.38**	**31.38**	76	10	**31.38**	**31.38**	74	10	**31.38**	**31.38**	31.38	31.38
tc2c80s8ct4	85	10	**43.98**	44.03	86	10	**43.98**	**44.02**	84	10	**43.98**	**44.02**	88	10	**43.98**	**44.02**	43.98	44.02
tc0c160s16cf2	285	10	57.93	58.04	272	10	**57.91**	58.02	273	10	**57.91**	58.02	268	10	**57.91**	**58.01**	57.91	58.01
tc0c160s16cf4	428	10	77.56	78.47	429	10	**77.30**	**77.63**	425	10	**77.30**	**77.63**	862	10	**77.30**	77.67	77.30	77.63
tc0c160s16ct2	286	10	**57.16**	**57.63**	288	10	**57.16**	57.64	289	10	**57.16**	57.64	279	10	**57.16**	57.64	57.16	57.63
tc0c160s16ct4	417	10	**76.18**	77.25	404	10	76.30	**76.97**	414	10	76.30	76.98	798	10	**76.18**	**76.97**	76.18	76.97
tc0c160s24cf2	291	10	**56.86**	**56.93**	284	10	**56.86**	56.94	285	10	**56.86**	56.94	286	10	**56.86**	**56.93**	56.86	56.93
tc0c160s24cf4	430	10	76.21	77.04	429	10	**75.63**	76.60	432	10	**75.63**	**76.55**	538	10	75.83	76.67	75.63	76.55
tc0c160s24ct2	298	10	55.53	56.30	303	10	**55.42**	56.23	301	10	**55.42**	56.23	294	10	**55.42**	**56.17**	55.42	56.17
tc0c160s24ct4	420	10	75.13	76.34	432	10	**75.09**	**76.26**	430	10	**75.09**	**76.26**	577	10	75.36	76.29	75.09	76.26
tc1c160s16cf0	382	8	75.81	77.16	372	10	74.69	76.53	379	10	74.69	76.53	413	10	**74.64**	**75.83**	74.64	75.83
tc1c160s16cf3	348	10	66.52	67.80	347	10	66.52	**67.17**	352	10	66.52	67.18	352	10	**66.49**	67.26	66.49	67.17
tc1c160s16ct0	369	10	74.30	74.94	367	10	74.24	74.91	360	10	74.24	74.91	382	10	**74.20**	**74.62**	74.20	74.62
tc1c160s16ct3	323	10	65.45	66.40	316	10	65.24	**66.02**	318	10	65.24	**66.02**	313	10	**65.23**	66.08	65.23	66.02
tc1c160s24cf0	381	10	73.93	74.60	372	10	**73.62**	73.96	374	10	**73.62**	**73.90**	445	10	**73.62**	74.03	73.62	73.90
tc1c160s24cf3	328	10	63.22	63.93	323	10	63.06	**63.60**	331	10	63.06	**63.60**	331	10	**63.05**	63.78	63.05	63.60
tc1c160s24ct0	384	10	73.95	74.61	380	10	**73.32**	73.95	371	10	**73.32**	73.95	375	10	**73.32**	**73.77**	73.32	73.77
tc1c160s24ct3	335	10	62.76	**63.22**	334	10	**62.70**	63.30	335	10	**62.70**	63.30	328	10	**62.70**	63.26	62.70	63.22
tc2c160s16cf1	284	10	56.71	57.44	286	10	**56.59**	**57.33**	285	10	**56.59**	**57.33**	284	10	56.66	57.38	56.59	57.33
tc2c160s16ct1	257	10	**55.37**	55.45	257	10	**55.37**	**55.43**	255	10	**55.37**	**55.43**	261	10	**55.37**	55.44	55.37	55.43
tc2c160s24cf1	289	10	**56.00**	57.24	298	10	56.47	**57.20**	299	10	56.47	**04:48**	297	10	56.47	57.36	56.00	57.20
tc2c160s24ct1	276	10	55.03	**55.07**	286	10	**54.96**	55.10	281	10	**54.96**	55.10	270	10	55.03	55.15	54.96	55.07
tc1c320s24cf2	1600	0	-	-	1582	10	133.89	**135.14**	1570	10	133.91	135.16	3219	10	**133.66**	135.37	133.66	135.14
tc1c320s24cf3	1250	10	105.88	106.73	1252	10	105.70	**106.70**	1268	10	105.70	**106.70**	1262	10	**105.47**	106.77	105.47	106.70
tc1c320s24ct2	1480	0	-	-	2337	10	133.46	135.57	2366	10	133.46	135.23	3461	9	**132.50**	**134.69**	132.50	134.69
tc1c320s24ct3	1166	10	**105.40**	106.31	1162	10	105.68	**106.29**	1156	10	105.68	**106.29**	1186	10	105.89	106.47	105.40	106.29
tc1c320s38cf2	1417	0	-	-	1370	10	129.09	129.61	1375	10	129.09	129.68	1561	10	**128.96**	**129.40**	128.96	129.40
tc1c320s38cf3	1286	10	105.60	**106.21**	1276	10	105.71	106.47	1265	10	105.71	106.47	1232	10	**105.41**	106.23	105.41	106.21
tc1c320s38ct2	1422	0	-	-	1397	10	128.86	129.80	1399	10	128.86	129.79	2603	10	**128.69**	**129.65**	128.69	129.65
tc1c320s38ct3	1342	10	106.42	106.82	1356	10	105.93	**106.60**	1377	10	105.93	**106.60**	1395	10	**105.44**	106.66	105.44	106.60
tc2c320s24cf0	1840	0	-	-	2534	10	163.39	165.89	2587	10	163.39	165.94	3472	10	**162.15**	**165.15**	162.15	165.15
tc2c320s24cf1	1014	10	**87.19**	**87.59**	1004	10	87.27	87.65	1021	10	87.27	87.65	1028	10	87.26	87.68	87.19	87.59
tc2c320s24cf4	1113	4	112.46	114.09	1095	10	111.74	112.45	1083	10	111.74	112.39	1433	10	**111.59**	**112.19**	111.59	112.19
tc2c320s24ct0	1683	0	-	-	2482	10	164.81	166.51	2498	10	164.81	**166.46**	3413	10	**164.18**	166.59	164.18	166.46
tc2c320s24ct1	1016	10	**86.86**	87.57	1026	10	87.26	87.64	1043	10	87.26	87.64	1017	10	87.01	**87.51**	86.86	87.51
tc2c320s24ct4	1214	5	111.66	113.18	1191	10	111.00	111.90	1182	10	111.00	**111.76**	1231	10	**110.95**	111.82	110.95	111.76
tc2c320s38cf0	1697	0	-	-	1875	10	160.69	162.51	1886	10	160.69	162.51	3388	10	**160.59**	**161.69**	160.59	161.69
tc2c320s38cf1	1016	10	**86.65**	**87.17**	1013	10	86.97	87.25	1029	10	86.97	87.25	999	10	86.69	87.21	86.65	87.17
tc2c320s38cf4	1274	10	**109.48**	110.62	1226	10	110.19	110.63	1224	10	110.19	110.63	1253	10	109.91	**110.46**	109.48	110.46
tc2c320s38ct0	1825	0	-	-	2169	10	160.54	162.97	2166	10	**160.36**	162.94	3252	10	161.25	**162.34**	160.36	162.34
tc2c320s38ct1	1076	10	86.59	86.97	1075	10	**86.23**	**86.72**	1068	10	**86.23**	**86.72**	1061	10	86.53	86.88	86.23	86.72
tc2c320s38ct4	1182	10	109.97	110.73	1201	10	109.60	110.29	1189	10	109.60	110.29	1210	10	**109.44**	**110.27**	109.44	110.27

Table 16: Detailed comparison of the best results obtained by the matheuristic for the E-VRP-NL-C according to the version of the subproblem it uses if the number of chargers at every CS is equal to 2 (instances with 10, 20, or 40 customers).

Instance	Version N				Version D				Version DW				Version DWR				BS	BA
	Time	#F	Best	Avg	Time	#F	Best	Avg	Time	#F	Best	Avg	Time	#F	Best	Avg		
tc0c10s2cf1	6	10	**19.75**	**19.75**	6	10	**19.75**	**19.75**	5	10	**19.75**	**19.75**	5	10	**19.75**	**19.75**	19.75	19.75
tc0c10s2ct1	4	10	**12.30**	**12.30**	4	10	**12.30**	**12.30**	4	10	**12.30**	**12.30**	4	10	**12.30**	**12.30**	12.30	12.30
tc0c10s3cf1	6	10	**19.75**	**19.75**	5	10	**19.75**	**19.75**	6	10	**19.75**	**19.75**	6	10	**19.75**	**19.75**	19.75	19.75
tc0c10s3ct1	5	10	**10.80**	**10.80**	5	10	**10.80**	**10.80**	5	10	**10.80**	**10.80**	5	10	**10.80**	**10.80**	10.80	10.80
tc1c10s2cf2	6	10	**9.03**	**9.03**	6	10	**9.03**	**9.03**	5	10	**9.03**	**9.03**	6	10	**9.03**	**9.03**	9.03	9.03
tc1c10s2cf3	6	10	**16.37**	**16.37**	6	10	**16.37**	**16.37**	6	10	**16.37**	**16.37**	6	10	**16.37**	**16.37**	16.37	16.37
tc1c10s2cf4	6	10	**16.10**	**16.10**	5	10	**16.10**	**16.10**	5	10	**16.10**	**16.10**	6	10	**16.10**	**16.10**	16.10	16.10
tc1c10s2ct2	4	10	**10.75**	**10.75**	4	10	**10.75**	**10.75**	4	10	**10.75**	**10.75**	4	10	**10.75**	**10.75**	10.75	10.75
tc1c10s2ct3	5	10	**13.17**	**13.17**	5	10	**13.17**	**13.17**	5	10	**13.17**	**13.17**	5	10	**13.17**	**13.17**	13.17	13.17
tc1c10s2ct4	5	10	**13.83**	**13.83**	5	10	**13.83**	**13.83**	5	10	**13.83**	**13.83**	5	10	**13.83**	**13.83**	13.83	13.83
tc1c10s3cf2	5	10	**9.03**	**9.03**	5	10	**9.03**	**9.03**	5	10	**9.03**	**9.03**	5	10	**9.03**	**9.03**	9.03	9.03
tc1c10s3cf3	4	10	**16.37**	**16.37**	4	10	**16.37**	**16.37**	4	10	**16.37**	**16.37**	4	10	**16.37**	**16.37**	16.37	16.37
tc1c10s3cf4	3	10	**14.90**	**14.90**	4	10	**14.90**	**14.90**	3	10	**14.90**	**14.90**	3	10	**14.90**	**14.90**	14.90	14.90
tc1c10s3ct2	6	10	**9.20**	**9.20**	7	10	**9.20**	**9.20**	7	10	**9.20**	**9.20**	6	10	**9.20**	**9.20**	9.20	9.20
tc1c10s3ct3	6	10	**13.02**	**13.02**	5	10	**13.02**	**13.02**	5	10	**13.02**	**13.02**	5	10	**13.02**	**13.02**	13.02	13.02
tc1c10s3ct4	5	10	**13.21**	**13.21**	6	10	**13.21**	**13.21**	5	10	**13.21**	**13.21**	5	10	**13.21**	**13.21**	13.21	13.21
tc2c10s2cf0	6	10	**21.77**	**21.77**	6	10	**21.77**	**21.77**	6	10	**21.77**	**21.77**	6	10	**21.77**	**21.77**	21.77	21.77
tc2c10s2ct0	5	10	**12.45**	**12.45**	5	10	**12.45**	**12.45**	5	10	**12.45**	**12.45**	5	10	**12.45**	**12.45**	12.45	12.45
tc2c10s3cf0	4	10	**21.77**	**21.77**	4	10	**21.77**	**21.77**	4	10	**21.77**	**21.77**	4	10	**21.77**	**21.77**	21.77	21.77
tc2c10s3ct0	6	10	**11.51**	**11.51**	6	10	**11.51**	**11.51**	7	10	**11.51**	**11.51**	6	10	**11.51**	**11.51**	11.51	11.51
tc0c20s3cf2	11	10	**27.47**	**27.47**	11	10	**27.47**	**27.47**	12	10	**27.47**	**27.47**	12	10	**27.47**	**27.47**	27.47	27.47
tc0c20s3ct2	8	10	**17.08**	**17.08**	8	10	**17.08**	**17.08**	8	10	**17.08**	**17.08**	7	10	**17.08**	**17.08**	17.08	17.08
tc0c20s4cf2	9	10	**27.47**	**27.47**	10	10	**27.47**	**27.47**	10	10	**27.47**	**27.47**	9	10	**27.47**	**27.47**	27.47	27.47
tc0c20s4ct2	11	10	**16.99**	**16.99**	11	10	**16.99**	**16.99**	11	10	**16.99**	**16.99**	10	10	**16.99**	**16.99**	16.99	16.99
tc1c20s3cf1	12	10	**17.49**	**17.49**	12	10	**17.49**	**17.49**	12	10	**17.49**	**17.49**	12	10	**17.49**	**17.49**	17.49	17.49
tc1c20s3cf3	10	10	**16.44**	**16.44**	9	10	**16.44**	**16.44**	9	10	**16.44**	**16.44**	9	10	**16.44**	**16.44**	16.44	16.44
tc1c20s3cf4	7	10	**17.00**	**17.00**	7	10	**17.00**	**17.00**	6	10	**17.00**	**17.00**	7	10	**17.00**	**17.00**	17.00	17.00
tc1c20s3ct1	11	10	**18.94**	**18.94**	11	10	**18.94**	**18.94**	11	10	**18.94**	**18.94**	10	10	**18.94**	**18.94**	18.94	18.94
tc1c20s3ct3	11	10	**12.60**	**12.60**	11	10	**12.60**	**12.60**	10	10	**12.60**	**12.60**	10	10	**12.60**	**12.60**	12.60	12.60
tc1c20s3ct4	9	10	**16.21**	**16.21**	9	10	**16.21**	**16.21**	10	10	**16.21**	**16.21**	10	10	**16.21**	**16.21**	16.21	16.21
tc1c20s4cf1	9	10	**16.38**	**16.38**	8	10	**16.38**	**16.38**	8	10	**16.38**	**16.38**	8	10	**16.38**	**16.38**	16.38	16.38
tc1c20s4cf3	13	10	**16.44**	**16.44**	13	10	**16.44**	**16.44**	13	10	**16.44**	**16.44**	13	10	**16.44**	**16.44**	16.44	16.44
tc1c20s4cf4	10	10	**17.00**	**17.00**	9	10	**17.00**	**17.00**	10	10	**17.00**	**17.00**	9	10	**17.00**	**17.00**	17.00	17.00
tc1c20s4ct1	12	10	**17.80**	**17.80**	13	10	**17.80**	**17.80**	13	10	**17.80**	**17.80**	13	10	**17.80**	**17.80**	17.80	17.80
tc1c20s4ct3	7	10	**14.43**	**14.43**	8	10	**14.43**	**14.43**	7	10	**14.43**	**14.43**	8	10	**14.43**	**14.43**	14.43	14.43
tc1c20s4ct4	8	10	**17.00**	**17.00**	8	10	**17.00**	**17.00**	8	10	**17.00**	**17.00**	8	10	**17.00**	**17.00**	17.00	17.00
tc2c20s3cf0	9	10	**24.68**	**24.68**	9	10	**24.68**	**24.68**	8	10	**24.68**	**24.68**	8	10	**24.68**	**24.68**	24.68	24.68
tc2c20s3ct0	12	10	**25.79**	**25.79**	12	10	**25.79**	**25.79**	12	10	**25.79**	**25.79**	11	10	**25.79**	**25.79**	25.79	25.79
tc2c20s4cf0	12	10	**24.67**	**24.67**	12	10	**24.67**	**24.67**	11	10	**24.67**	**24.67**	11	10	**24.67**	**24.67**	24.67	24.67
tc2c20s4ct0	12	10	**26.02**	**26.02**	12	10	**26.02**	**26.02**	11	10	**26.02**	**26.02**	11	10	**26.02**	**26.02**	26.02	26.02
tc0c40s5cf0	20	10	**32.20**	**32.23**	20	10	**32.20**	**32.23**	21	10	**32.20**	**32.23**	20	10	**32.20**	**32.23**	32.20	32.23
tc0c40s5cf4	26	10	**30.25**	**30.25**	26	10	**30.25**	**30.25**	25	10	**30.25**	**30.25**	26	10	**30.25**	**30.25**	30.25	30.25
tc0c40s5ct0	20	10	**27.91**	**27.91**	20	10	**27.91**	**27.91**	19	10	**27.91**	**27.91**	19	10	**27.91**	**27.91**	27.91	27.91
tc0c40s5ct4	23	10	**28.63**	**28.63**	23	10	**28.63**	**28.63**	23	10	**28.63**	**28.63**	22	10	**28.63**	**28.63**	28.63	28.63
tc0c40s8cf0	20	10	**30.40**	**30.40**	21	10	**30.40**	**30.40**	21	10	**30.40**	**30.40**	20	10	**30.40**	**30.40**	30.40	30.40
tc0c40s8cf4	28	10	**28.24**	**28.25**	28	10	**28.24**	**28.25**	29	10	**28.24**	**28.25**	29	10	**28.24**	**28.25**	28.24	28.25
tc0c40s8ct0	21	10	**26.22**	**26.22**	20	10	**26.22**	**26.22**	20	10	**26.22**	**26.22**	21	10	**26.22**	**26.22**	26.22	26.22
tc0c40s8ct4	26	10	**29.07**	**29.07**	26	10	**29.07**	**29.07**	26	10	**29.07**	**29.07**	26	10	**29.07**	**29.07**	29.07	29.07
tc1c40s5cf1	28	2	66.55	66.55	28	10	**65.38**	67.50	28	10	**65.38**	67.50	454	10	**65.38**	**66.51**	65.38	66.51
tc1c40s5ct1	27	10	**52.33**	**52.33**	27	10	**52.33**	**52.33**	26	10	**52.33**	**52.33**	26	10	**52.33**	**52.33**	52.33	52.33
tc1c40s8cf1	23	10	**40.64**	**40.64**	24	10	**40.64**	**40.64**	24	10	**40.64**	**40.64**	23	10	**40.64**	**40.64**	40.64	40.64
tc1c40s8ct1	28	10	**40.18**	**40.18**	27	10	**40.18**	**40.18**	27	10	**40.18**	**40.18**	28	10	**40.18**	**40.18**	40.18	40.18
tc2c40s5cf2	19	10	**27.54**	**27.54**	19	10	**27.54**	**27.54**	20	10	**27.54**	**27.54**	20	10	**27.54**	**27.54**	27.54	27.54
tc2c40s5cf3	25	10	**19.65**	**19.65**	25	10	**19.65**	**19.65**	25	10	**19.65**	**19.65**	26	10	**19.65**	**19.65**	19.65	19.65
tc2c40s5ct2	18	10	**26.91**	**26.91**	19	10	**26.91**	**26.91**	19	10	**26.91**	**26.91**	19	10	**26.91**	**26.91**	26.91	26.91
tc2c40s5ct3	22	10	**23.39**	**23.39**	22	10	**23.39**	**23.39**	23	10	**23.39**	**23.39**	22	10	**23.39**	**23.39**	23.39	23.39
tc2c40s8cf2	20	10	**27.13**	**27.13**	20	10	**27.13**	**27.13**	20	10	**27.13**	**27.13**	19	10	**27.13**	**27.13**	27.13	27.13
tc2c40s8cf3	25	10	**19.65**	**19.65**	26	10	**19.65**	**19.65**	25	10	**19.65**	**19.65**	25	10	**19.65**	**19.65**	19.65	19.65
tc2c40s8ct2	19	10	**26.28**	**26.28**	19	10	**26.28**	**26.28**	18	10	**26.28**	**26.28**	19	10	**26.28**	**26.28**	26.28	26.28
tc2c40s8ct3	29	10	**22.45**	**22.45**	28	10	**22.45**	**22.45**	29	10	**22.45**	**22.45**	29	10	**22.45**	**22.45**	22.45	22.45

Table 17: Detailed comparison of the best results obtained by the matheuristic for the E-VRP-NL-C according to the version of the subproblem it uses if the number of chargers at every CS is equal to 2 (instances with 80, 160, or 320 customers).

Instance	Version N				Version D				Version DW				Version DWR				BS	BA
	Time	#F	Best	Avg	Time	#F	Best	Avg	Time	#F	Best	Avg	Time	#F	Best	Avg		
tc0c80s12cf0	76	10	**34.06**	**34.20**	74	10	**34.06**	**34.20**	75	10	**34.06**	**34.20**	77	10	**34.06**	**34.20**	34.06	34.20
tc0c80s12cf1	77	10	**40.83**	**40.94**	77	10	**40.83**	**40.94**	77	10	**40.83**	**40.94**	76	10	**40.83**	**40.94**	40.83	40.94
tc0c80s12ct0	75	10	**37.51**	37.95	76	10	**37.51**	**37.89**	74	10	**37.51**	**37.89**	75	10	**37.51**	37.98	37.51	37.89
tc0c80s12ct1	71	10	**39.72**	**40.00**	70	10	**39.72**	**40.00**	72	10	**39.72**	**40.00**	71	10	**39.72**	**40.00**	39.72	40.00
tc0c80s8cf0	61	10	**38.59**	**38.84**	62	10	**38.59**	**38.84**	62	10	**38.59**	**38.84**	61	10	**38.59**	**38.84**	38.59	38.84
tc0c80s8cf1	83	10	**43.38**	**44.09**	82	10	**43.38**	44.12	85	10	**43.38**	44.12	87	10	**43.38**	44.10	43.38	44.09
tc0c80s8ct0	67	10	**40.53**	**41.47**	69	10	**40.53**	**41.47**	70	10	**40.53**	**41.47**	71	10	**40.53**	**41.47**	40.53	41.47
tc0c80s8ct1	78	10	**43.85**	**43.98**	80	10	**43.85**	**43.98**	79	10	**43.85**	**43.98**	81	10	**43.85**	**43.98**	43.85	43.98
tc1c80s12cf2	59	10	**28.65**	**28.77**	57	10	**28.65**	**28.77**	60	10	**28.65**	**28.77**	59	10	**28.65**	**28.77**	28.65	28.77
tc1c80s12ct2	60	10	**28.73**	29.17	62	10	**28.73**	**29.09**	63	10	**28.73**	**29.09**	62	10	**28.73**	29.14	28.73	29.09
tc1c80s8cf2	62	10	**29.15**	**29.15**	63	10	**29.15**	**29.15**	62	10	**29.15**	**29.15**	62	10	**29.15**	**29.15**	29.15	29.15
tc1c80s8ct2	64	10	**29.88**	**30.42**	64	10	**29.88**	**30.42**	64	10	**29.88**	**30.42**	62	10	**29.88**	**30.42**	29.88	30.42
tc2c80s12cf3	63	10	**30.60**	**30.60**	62	10	**30.60**	**30.60**	61	10	**30.60**	**30.60**	62	10	**30.60**	**30.60**	30.60	30.60
tc2c80s12cf4	87	10	42.12	42.20	89	10	**42.11**	**42.18**	88	10	**42.11**	**42.18**	88	10	**42.11**	**42.18**	42.11	42.18
tc2c80s12ct3	63	10	**29.90**	**29.90**	64	10	**29.90**	**29.90**	64	10	**29.90**	**29.90**	62	10	**29.90**	**29.90**	29.90	29.90
tc2c80s12ct4	88	10	**40.27**	**40.27**	86	10	**40.27**	**40.27**	88	10	**40.27**	**40.27**	85	10	**40.27**	**40.27**	40.27	40.27
tc2c80s8cf3	59	10	**31.70**	**31.93**	60	10	**31.70**	**31.93**	59	10	**31.70**	**31.93**	59	10	**31.70**	**31.93**	31.70	31.93
tc2c80s8cf4	101	10	46.18	46.65	101	10	**46.06**	**46.64**	103	10	**46.06**	**46.64**	102	10	46.18	46.65	46.06	46.64
tc2c80s8ct3	75	10	**31.38**	**31.38**	78	10	**31.38**	**31.38**	76	10	**31.38**	**31.38**	76	10	**31.38**	**31.38**	31.38	31.38
tc2c80s8ct4	87	10	**43.72**	**43.97**	85	10	**43.72**	**43.97**	84	10	**43.72**	**43.97**	87	10	**43.72**	**43.97**	43.72	43.97
tc0c160s16cf2	277	10	**57.91**	**58.01**	279	10	**57.91**	**58.01**	276	10	**57.91**	**58.01**	279	10	**57.91**	**58.01**	57.91	58.01
tc0c160s16cf4	416	10	77.04	**77.59**	418	10	77.04	77.62	424	10	77.04	77.62	427	10	**77.03**	77.60	77.03	77.59
tc0c160s16ct2	286	10	**57.16**	**57.63**	280	10	**57.16**	**57.63**	285	10	**57.16**	**57.63**	286	10	**57.16**	57.64	57.16	57.63
tc0c160s16ct4	403	10	76.66	76.93	395	10	76.37	**76.86**	408	10	76.37	**76.86**	399	10	**76.14**	76.87	76.14	76.86
tc0c160s24cf2	284	10	**56.86**	56.93	283	10	**56.86**	**56.92**	283	10	**56.86**	**56.92**	285	10	**56.86**	**56.92**	56.86	56.92
tc0c160s24cf4	432	10	**75.50**	**76.46**	415	10	76.00	76.61	410	10	76.00	76.61	430	10	75.59	76.48	75.50	76.46
tc0c160s24ct2	289	10	55.43	56.18	291	10	**55.42**	**56.16**	302	10	**55.42**	**56.16**	296	10	**55.42**	56.17	55.42	56.16
tc0c160s24ct4	418	10	**74.99**	76.16	428	10	75.06	**76.01**	429	10	75.06	**76.01**	436	10	**74.99**	76.19	74.99	76.01
tc1c160s16cf0	365	10	**74.55**	75.63	370	10	**74.55**	**75.51**	368	10	**74.55**	**75.51**	374	10	**74.55**	**75.51**	74.55	75.51
tc1c160s16cf3	348	10	**66.14**	**66.92**	352	10	66.32	67.02	351	10	66.32	67.02	344	10	66.32	66.98	66.14	66.92
tc1c160s16ct0	372	10	**74.19**	74.70	372	10	74.20	75.16	368	10	74.20	75.16	379	10	**74.19**	**74.66**	74.19	74.66
tc1c160s16ct3	307	10	**65.22**	**66.08**	309	10	66.03	66.42	311	10	66.03	66.42	319	10	65.26	66.27	65.22	66.08
tc1c160s24cf0	377	10	73.67	74.05	367	10	**73.62**	**73.85**	371	10	**73.62**	**73.85**	372	10	**73.62**	74.06	73.62	73.85
tc1c160s24cf3	323	10	**63.17**	63.86	334	10	63.22	63.83	325	10	63.22	63.83	333	10	**63.17**	**63.74**	63.17	63.74
tc1c160s24ct0	371	10	73.34	**73.81**	367	10	**73.32**	**73.81**	361	10	**73.32**	**73.81**	366	10	73.34	74.12	73.32	73.81
tc1c160s24ct3	330	10	**62.70**	63.28	326	10	**62.70**	**63.27**	337	10	**62.70**	**63.27**	336	10	**62.70**	63.32	62.70	63.27
tc2c160s16cf1	285	10	57.04	57.65	279	10	**57.01**	**57.62**	284	10	**57.01**	**57.62**	280	10	57.04	57.65	57.01	57.62
tc2c160s16ct1	260	10	**55.37**	**55.44**	258	10	**55.37**	**55.44**	262	10	**55.37**	**55.44**	258	10	**55.37**	55.46	55.37	55.44
tc2c160s24cf1	298	10	**56.00**	57.27	300	10	56.47	**57.18**	294	10	56.47	**57.18**	288	10	56.67	57.38	56.00	57.18
tc2c160s24ct1	281	10	55.03	55.16	291	10	55.03	55.14	283	10	55.03	55.14	274	10	**54.96**	**55.10**	54.96	55.10
tc1c320s24cf2	1561	10	133.66	134.28	1464	10	**132.45**	**133.57**	1458	10	**132.45**	**133.57**	1461	10	132.82	133.75	132.45	133.57
tc1c320s24cf3	1237	10	**105.29**	**106.53**	1233	10	105.47	106.60	1224	10	105.47	106.60	1264	10	106.21	106.98	105.29	106.53
tc1c320s24ct2	1601	10	132.16	133.09	1380	10	131.73	132.30	1379	10	131.73	132.30	1386	10	**131.69**	**132.28**	131.69	132.28
tc1c320s24ct3	1199	10	105.92	106.40	1142	10	**105.25**	**106.06**	1150	10	**105.25**	**106.06**	1156	10	105.37	106.27	105.25	106.06
tc1c320s38cf2	1359	10	129.01	129.40	1360	10	**128.87**	**129.21**	1375	10	**128.87**	**129.21**	1371	10	**128.87**	129.28	128.87	129.21
tc1c320s38cf3	1231	10	105.55	106.24	1243	10	**104.90**	**105.88**	1227	10	**104.90**	**105.88**	1253	10	105.72	106.24	104.90	105.88
tc1c320s38ct2	1395	10	128.84	129.44	1366	10	**128.71**	129.43	1349	10	**128.71**	129.43	1418	10	128.79	**129.34**	128.71	129.34
tc1c320s38ct3	1350	10	105.83	106.50	1385	10	105.39	106.64	1379	10	105.39	106.64	1339	10	**105.23**	**106.42**	105.23	106.42
tc2c320s24cf0	1880	10	160.36	161.73	1539	10	159.66	**160.55**	1555	10	159.66	**160.55**	1703	10	**158.96**	160.60	158.96	160.55
tc2c320s24cf1	1012	10	**87.26**	87.73	1032	10	87.35	87.71	1047	10	87.35	87.71	1039	10	87.31	**87.66**	87.26	87.66
tc2c320s24cf4	1065	10	110.98	111.52	1088	10	**110.58**	**111.12**	1074	10	**110.58**	**111.12**	1085	10	110.73	111.37	110.58	111.12
tc2c320s24ct0	1826	10	160.19	161.36	1505	10	**159.58**	160.22	1485	10	**159.58**	160.30	2184	10	159.62	**160.01**	159.58	160.01
tc2c320s24ct1	996	10	87.26	87.68	1030	10	**87.02**	**87.52**	1024	10	**87.02**	**87.52**	1014	10	87.28	87.70	87.02	87.52
tc2c320s24ct4	1175	10	111.07	111.63	1177	10	111.04	**111.54**	1155	10	111.04	**111.54**	1171	10	**110.99**	111.63	110.99	111.54
tc2c320s38cf0	1571	10	159.35	159.93	1588	10	**158.29**	**159.33**	1580	10	**158.29**	**159.33**	1589	10	158.44	159.48	158.29	159.33
tc2c320s38cf1	1008	10	86.88	87.22	1024	10	86.78	**87.19**	1015	10	86.78	**87.19**	1021	10	**86.77**	87.30	86.77	87.19
tc2c320s38cf4	1264	10	109.93	110.56	1264	10	109.94	**110.39**	1255	10	109.94	**110.39**	1246	10	**109.80**	110.57	109.80	110.39
tc2c320s38ct0	1782	9	158.59	159.53	1617	10	158.30	159.06	1580	10	158.30	159.06	1586	10	**157.81**	**158.88**	157.81	158.88
tc2c320s38ct1	1093	10	**85.90**	**86.81**	1038	10	86.52	86.95	1049	10	86.52	86.95	1074	10	86.26	86.85	85.90	86.81
tc2c320s38ct4	1204	10	**109.50**	**110.12**	1191	10	109.75	110.13	1164	10	109.75	110.13	1187	10	109.85	110.18	109.50	110.12

References

Alvarenga, G. B., Mateus, G., and De Tomi, G. (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34(6):1561–1584.

Andelmin, J. and Bartolini, E. (2019). A multi-start local search heuristic for the green vehicle routing problem based on a multigraph reformulation. *Computers & Operations Research*, 109:43–63.

Artigues, C., Michelon, P., and Reusser, S. (2003). Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, 149(2):249–267.

Baptiste, P., Le Pape, C., and Nuijten, W. (2001). *Constraint-based scheduling: applying constraint programming to scheduling problems*, volume 39. Springer, Boston, MA.

Bartolini, E. and Andelmin, J. (2017). An exact algorithm for the green vehicle routing problem. *Transportation Science*, 51(4):1288–1303.

Beek, O., Raa, B., Dullaert, W., and Vigo, D. (2018). An efficient implementation of a static move descriptor-based local search heuristic. *Computers & Operations Research*, 94:1–10.

Bruglieri, M., Mancini, S., and Pisacane, O. (2019). The green vehicle routing problem with capacitated alternative fuel stations. *Computers & Operations Research*, 112:104759.

Codato, G. and Fischetti, M. (2006). Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766.

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

Drexl, M. (2012). Synchronization in vehicle routing – a survey of vrps with multiple synchronization constraints. *Transportation Science*, 46(3):297–316.

El Hachemi, N., Gendreau, M., and Rousseau, L.-M. (2013). A heuristic to solve the synchronized log-truck scheduling problem. *Computers & Operations Research*, 40(3):666–673.

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

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

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

Gnann, T., Funke, S., Jakobsson, N., Plötz, P., Sprei, F., and Bennehag, A. (2018). Fast charging infrastructure for electric vehicles: Today's situation and future needs. *Transportation Research Part D: Transport and Environment*, 62:314–329.

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

Grangier, P., Gendreau, M., Lehuédé, F., and Rousseau, L.-M. (2017). A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Computers & Operations Research*, 84:116–126.

Grimault, A., Bostel, N., and Lehuédé, F. (2017). An adaptive large neighborhood search for the full truckload pickup and delivery problem with resource synchronization. *Computers & Operations Research*, 88:1–14.

Hempsch, C. and Irnich, S. (2008). Vehicle routing problems with inter-tour resource constraints. In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 421–444. Springer, Boston, MA.

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

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

Juan, A. A., Mendez, C. A., Faulin, J., de Armas, J., and Grasman, S. E. (2016). Electric vehicles in logistics and transportation: A survey on emerging environmental, strategic, and operational challenges. *Energies*, 9(2):86.

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

Keskin, M., Laporte, G., and Çatay, B. (2019). Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Computers & Operations Research*, 107:77–94.

Koç, Ç., Jabali, O., Mendoza, J. E., and Laporte, G. (2018). The electric vehicle routing problem with shared charging stations. *International Transactions in Operational Research*, 26(4):1211–1243.

Koç, C. and Karaoglan, I. (2016). The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing*, 39:154–164.

Kullman, N., Goodson, J., and Mendoza, J. E. (2018). Dynamic Electric Vehicle Routing: Heuristics and Dual Bounds. Working paper available at `https://hal.archives-ouvertes.fr/hal-01928730`.

Lam, E. and Van Hentenryck, P. (2016). A branch-and-price-and-check model for the vehicle routing problem with location congestion. *Constraints*, 21(3):1–19.

Lourenço, H. R., Martin, O. C., and Stützle, T. (2003). Iterated local search. In Glover, F. and Kochenberger, G. A., editors, *Handbook of metaheuristics*, volume 57, pages 320–353. Springer, Boston, MA.

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

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

Morais, V. W. C., Mateus, G. R., and Noronha, T. F. (2014). Iterated local search heuristics for the vehicle routing problem with cross-docking. *Expert Systems with Applications*, 41(16):7495–7506.

Pelletier, S., Jabali, O., and Laporte, G. (2016). 50th anniversary invited article—goods distribution with electric vehicles: Review and research perspectives. *Transportation Science*, 50(1):3–22.

Pelletier, S., Jabali, O., and Laporte, G. (2018). Charge scheduling for electric freight vehicles. *Transportation Research Part B: Methodological*, 115:246–269.

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

Rix, G., Rousseau, L.-M., and Pesant, G. (2015). A column generation algorithm for tactical timber transportation planning. *Journal of the Operational Research Society*, 66(2):278–287.

Sassi, O. and Oulamara, A. (2014). Joint scheduling and optimal charging of electric vehicles problem. In Murgante, B., Misra, S., Rocha, A. M. A. C., Torre, C., Rocha, J. G., Falcão, M. I., Taniar, D., Apduhan, B. O., and Gervasi, O., editors, *Computational Science and Its Applications – ICCSA 2014*, volume 8580, pages 76–91, Cham. Springer.

Schiffer, M., Stütz, S., and Walther, G. (2018). Electric commercial vehicles in mid-haul logistics networks. In Pistoia, G. and Liaw, B., editors, *Behaviour of Lithium-Ion Batteries in Electric Vehicles*, pages 153–173. Springer, Cham.

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

Subramanian, A., Uchoa, E., and Ochi, L. S. (2013). A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, 40(10):2519–2531.

Sweda, T. M., Dolinskaya, I. S., and Klabjan, D. (2017). Adaptive routing and recharging policies for electric vehicles. *Transportation Science*, 51(4):1326–1348.

Villegas, J. G., Guéret, C., Mendoza, J. E., and Montoya, A. (2018). The technician routing and scheduling problem with conventional and electric vehicle. Working paper available at `https://hal.archives-ouvertes.fr/hal-01813887`.

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., and Velasco, N. (2013). A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, 230(2):231–244.

Zachariadis, E. E. and Kiranoudis, C. T. (2010). A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. *Computers & Operations Research*, 37(12):2089–2105.