New formulations for the electric vehicle routing problem with nonlinear charging functions

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

CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

New Formulations for the Electric Vehicle Routing Problem with Nonlinear Charging Functions

Aurélien Froger
Jorge E. Mendoza
Ola Jabali
Gilbert Laporte

June 2017

CIRRELT-2017-30

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

New Formulations for the Electric Vehicle Routing Problem with Nonlinear Charging Functions

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

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Management Sciences, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[2] Université François-Rabelais de Tours, LI EA 6300, ROOT ERL CNRS 6305, Tours, France

[3] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano 20133, Italy

Abstract. Electric vehicle routing problems (E-VRPs) are receiving growing attention from the operations research community. Electric vehicles (EVs) differ substantially from internal combustion engine vehicles. In the routing context, the main difference is due to the limited EV autonomy, which can be recuperated at charging stations. These are rather scarce, thus EVs typically have to perform detours to reach them. Various assumptions on the charging policy and objective function{, among others}, have led to the definition of several variants E-VRPs. Modeling the charging functions is a focal point of the majority of these problems. The majority of the research has focused on constant or linear charging function. To account for the nonlinear relation between the times spent charging and the amount of energy charged, the electric vehicle routing problem with nonlinear charging function has been recently introduced. In this research we propose new formulations for this problem. We present an arc-based tracking of the time and the state of charge which, according to our experiments, outperform the classical node-based tracking of these values. We also propose alternative formulations of the piecewise linear approximation of the nonlinear charging function. To prevent the use of charging stations nodes replication, we propose a recharging path-based model. We present a labeling algorithm to generate a tractable number of these paths. This latter model overcomes the limits of the classical models presented in the literature and also outperforms them in our experiments.

Keywords: Vehicle routing problem, electric vehicle routing problem with nonlinear charging function, mixed integer linear programming, labeling algorithm.

Acknowledgements. This research was partly funded in France by Agence Nationale de la Recherche through project e-VRO (ANR-15-CE22-0005-01).

---

* Corresponding author: Aurelien.Froger@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2017

1. Introduction

Electric vehicles (EVs) are assumed to be an environmental viable alternative to internal combustion engine vehicles (ICEVs). While this assumption is conditioned to EVs being supplied by environmentally friendly energy resources, the diffusion of EV is being encouraged through numerous governmental and private policies. Transporters are thus in the process of adapting their operational conventions in a manner that fits the adoption of EVs. As a result, electric vehicle routing problems (E-VRPs) are receiving much attention in the operational research literature. Introducing EVs modifies the definition of the "classic" vehicle routing problems (VRPs), as the EV technological constraints need to be accounted for. Indeed, the EVs have a driving autonomy limited by their battery capacity, which can be recovered at charging stations (CSs). These are much more scarce compared to refueling stations for ICEVs, therefore EVs typically perform on-route detours to reach CS. Moreover, the charging times of EVs could be much longer compared to refueling the activity of ICEVs. Therefore, considering EVs introduces additional decisions in VRPs related to planning the charging operations. More specifically, decisions in E-VRPs not only relate to the assignment of customers to vehicles and establishing the sequence of customer visits per vehicle, but also determine where and how long to charge.

Research on the E-VRPs primarily started with the introduction of the Green Vehicle Routing Problem (G-VRP) by Erdoğan and Miller-Hooks (2012). In the G-VRP, a fleet of homogeneous alternative fuel vehicles must visit a set of customers from a single depot while traveling a minimum total distance. The vehicles have a limited fuel tank capacity but can detour to CSs to extend their driving range. The routes have a duration limit constraint. The tank is fully recharged at each visit to a CS in constant time. This assumption fits the vehicles operating on alternative fuels, e.g., biofuels and liquid nitrogen. In terms of EV such an assumption is plausible when working with battery swapping stations, where the battery of an EV is replaced by a fully charged one. However, to our knowledge, battery swapping stations do not seem to exist nowadays: the company Better Place declared bankruptcy in 2013 and Tesla's only battery swapping station shutdown at the end of 2016. To tackle the G-VRP, Erdoğan and Miller-Hooks (2012) proposed two heuristics. Koč and Karaoglan (2016) developed a branch-and-cut algorithm that can optimally solve instances with up to 20 customers. These authors also presented a simulated annealing-based heuristic algorithm to tackle larger instances. Montoya et al. (2016) developed a multi-space sampling two-stage heuristic. The first stage of this heuristic builds a pool of high-quality routes, by means of randomized route-first cluster-second heuristics, and the second stage assembles a solution from the pool by solving a set partitioning (SP) formulation. Bartolini and Andelmin (2017) proposed an exact method based on a SP formulation of the problem, which is strengthened by the addition of a number of cuts. One distinct feature of their formulation lies in defining the problem on a multigraph using the concept of refuel-paths. They optimally solved instances with up to 110 customers.

Several variants of the G-VRP have been proposed in the literature. Schneider et al. (2014) defined the electric routing problem with time windows and cargo capacity constraints (E-VRPTW). In the E-VRPTW, the batteries are fully recharged at CSs. However, the

charging time is not longer presumed constant, as in the G-VRP, it linearly depends on the state of charge (SoC) of the EV upon its arrival at the CS. Finally, no route duration limit is considered. To solve the E-VRPTW, Schneider et al. (2014) proposed a hybrid metaheuristic that couples variable neighborhood and tabu search. Schneider et al. (2015) proposed an adaptive variable neighborhood search for a more general class of problems, which improved the G-VRP results of Schneider et al. (2014). Keskin and Čatay (2016) designed an adaptive large neighborhood search (ALNS) algorithm with customized operators for the E-VRPTW. Desaulniers et al. (2016) studied a version of the E-VRPTW, in which the fleet size is fixed. These authors proposed a branch-and-cut-and-price approach that is able to solve instances with up to 100 customers to optimality. This algorithm also applies when adopting a partial charging policy for the EVs, i.e., a charging activity does not entail fully charging the battery.

In the vast majority of E-VRPs, the energy consumption rate of EVs depends only on the traveled distance. Goeke and Schneider (2015) considered a mixed fleet of EVs and ICEVs, the energy consumption of the EVs depends on road gradients, cargo load, and vehicle speed. Such parameters are used to accurately model energy consumption. They developed an ALNS algorithm which combines local search. Furthermore, they adapted their algorithm to tackle the related E-VRPTW.

Hiermann et al. (2016) introduced an extension of the E-VRPTW called the Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations (E-FSMFTW). They considered a heterogeneous fleet of vehicles, each type of vehicle having different capacity, energy consumption and acquisition costs. They developed a branch-and-price algorithm, and optimally solved instances with up to 15 customers. They also developed an ALNS algorithm based metaheuristic for solving realistic-sized instances. Both these algorithms produce competitive results on the E-VRPTW of Schneider et al. (2014) instances. Locating CSs is another critical point when routing EVs. The joint problem of locating CSs and routing EVs has been recently studied. Yang and Sun (2015) considered battery swapping stations and proposed heuristic algorithms for the resulting problem. Schiffer and Walther (2017) introduced the electric location routing problem with time windows and partial recharging and proposed a mathematical model.

With the exception of the work of Desaulniers et al. (2016) and Schiffer and Walther (2017), the above literature assumed that whenever a charging activity is performed it is concluded only when the battery is fully charged. This assumption yields more tractable models, which is partly the reason behind its adoption by earlier versions of E-VRPs. Nonetheless, assuming that the battery is fully charged each time a charging operation takes place is rather conservative. Partial recharges were first been introduced by Felipe et al. (2014). Indeed, compared to a full charging policy, partial charging allows savings in both energy consumption and time, since only the energy required can be charged. It is worth noting that the time to reach a fully-charged battery is usually prohibitive when charging operations occur in routes. The authors also considered CSs with multiple technologies. More specifically, the charging rate may be different from one CS to another. This reflects the variety of chargers one can encounter in practice (e.g., slow, medium, and fast chargers). The authors also considered capacitated vehicles and route duration constraints. They developed a simulated annealing metaheuristic for the resulting problem. Bruglieri et al. (2015) consid-

ered the E-VRPTW using a partial charging policy with an alternative objective function, which consists in minimizing a weighted sum of two different components: the number of EVs serving the customers and the total time consumed in traveling, waiting and charging. This objective better captures the impact of charging decisions. Indeed, the objectives solely based on cost or distance associated with each arc neglect the significant time that can be spent at CSs. The same analysis is done by Montoya et al. (2017) who therefore set the minimization of the total time as the objective of their problem. They also used nonlinear charging functions to model the charging process, which better depict the real charging process. The resulting problem is called the electric vehicle routing problem with nonlinear charging function (E-VRP-NL). Indeed, from a 80%-SoC the battery level generally increases concavely with time. The authors therefore proposed a piecewise linear approximation of the charging process. Moreover, Montoya et al. (2017) showed that accounting for nonlinear charging functions is pivotal for the E-VRPs. Indeed, they showed that previously adopted linear charging functions yield infeasible routes with respect to tour duration limits, when projected on nonlinear functions. Moreover, even if routes produced under linear charging functions are feasible, when subjected to nonlinear charging functions they tend to be of poor quality. To solve the E-VRP-NL, they developed a hybrid metaheuristic that combines an iterated local search and a SP problem.

The research outlined above implicitly assumes that the charging infrastructure is owned by the EV operator. This is plausible for large transportation companies. In contrast, public infrastructure involves uncertainty with respect to the availability of the CS. Using public infrastructure to charge EVs in a routing context has received far less attention in the literature. We refer the reader to (Kullman et al., 2016) and, especially, to (Sweda et al., 2017) for specific details.

Research on the E-VRP is moving towards more accurately representing the characteristics of EVs. This tendency is yielding more intricate models posing various algorithmic challenges. In this paper, we study the E-VRP-NL introduced by Montoya et al. (2017), which as previously mentioned, is pivotal in that it realistically models the charging operation via nonlinear functions, while accounting for various charging technologies.

First, we revisit the mixed integer linear programming (MILP) formulation introduced by Montoya et al. (2017) and propose alternative strategies to model the time and SoC tracking as well as the piecewise linear charging functions. Second, we define the problem on a multigraph, which allows us to derive a formulation of the E-VRP-NL that does not use replication of CS nodes, as it is typical in E-VRPs. We discuss how this formulation better fits the problem from a modeling perspective, we also show the superiority of this formulation on a number of experiments.

The remainder of this document is organized as follows. We formally describe the E-VRP-NL in Section 2. In Section 3, we present MILP formulations of the problem based on its "classical" definition on a simple digraph. We then present a MILP formulation of the E-VRP-NL using the concept of recharging paths. We present the computational results in Section 5. Finally, we present our conclusions and outline research perspectives in Section 6.

2. Problem description

The electric vehicle routing problem with nonlinear charging function introduced by (Montoya et al., 2017) is defined 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 the vehicles can stop to recharge their battery. Each customer $i \in I$ has a service time g_i. The customers are served using an unlimited and homogeneous fleet of EVs. All the EVs have a battery of capacity Q (expressed in kWh) and a maximum tour duration of T_{max}. At the beginning of the planning horizon, the EVs are located in a single depot, from which they leave fully charged. Traveling from one location i (the depot, a customer, or a CS) to another location j incurs a diriving time $t_{ij} \geqslant 0$ and an energy consumption $e_{ij} \geqslant 0$. The triangular inequality holds for both the driving times and energy consumptions. Due to their limited battery capacity, EVs may require to stop en route at CSs. Charging operations can occur at any CS and EVs can be partially recharged. Each CS $i \in F$ has a charging mode (e.g., slow, moderate, fast) associated with a piecewise linear charging function $\Phi_i(\Delta)$ that maps, for an empty battery, the time Δ spent charging at i to the SoC of the vehicle when it leaves i. If q is the SoC of the EV when it arrives at i and Δ the time spent charging, the SoC of the EV when it departs from i is given by $\Phi_i(\Delta + \Phi_i^{-1}(q))$. We denote as $B_i = \{0, \ldots, b_i\}$ the set of breakpoints of the piecewise linear approximation of the charging curve at i (sorted in ascending order). Let c_{ik} and a_{ik} be the charging time and the SoC of breakpoint $k \in B_i$, of the CS i. Figure 1 shows the piecewise linear approximation for a CS i yielding a power of 22 kW charging a vehicle equipped with a 16 kWh battery. Let ρ_{ik} be the slope of the segment between $c_{i,k-1}$ and c_{ik} (i.e. $\rho_{ik} = (a_{ik} - a_{i,k-1})/(c_{ik} - c_{i,k-1})$) and η_{ik} the y-intercept of the segment between $c_{i,k-1}$ and c_{ik} (i.e. $\eta_{ik} = a_{ik} - c_{ik}\rho_{ik}$). We say that a charging function g dominates another charging function g' if $\forall t \in [0, S_{max}]$, we have $g(t) \geqslant g'(t)$. The parameter $S_{max} = max_{i \in F}\{c_{ib}\}$ corresponds to the maximum charging time, i.e., the time to fully charge the battery when the initial SoC is zero. An important characteristic of charging functions is that given two such functions, corresponding to different technologies, one will dominate the other.

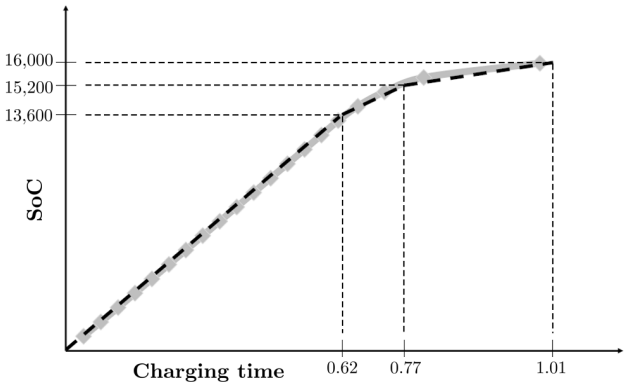


Figure 1: Real data (solid line) vs. piecewise linear approximation (dashed line) for a CS yielding a power of 22 kW charging a 16 kWh battery. Taken from (Montoya et al., 2017).

Feasible solutions to the E-VRP-NL satisfy the following conditions:

1. Each customer is visited exactly once by a single vehicle;
2. Each route starts and ends at the depot;
3. Each route satisfies the maximum-duration limit T_{max};
4. Each route is energy-feasible, i.e., the SoC of an EV when it arrives at and departs from any location is between 0 and Q

The objective of the E-VRP-NL is to minimize the total time. This takes into account driving, service, and charging times. The motivation for having a temporal-based objective, stems from the fact that charging times should be explicitly accounted for in this problem.

3. CS replication-based formulation

Following the established conventions in the E-VRP literature, the E-VRP-NL is defined on a complete digraph $G = (V, A)$, where $V = \{0\} \cup I \cup F'$ is the set of nodes and $A = \{(i, j) : i, j \in V, i \neq j\}$ is the set of arcs connecting nodes of V. The symbol 0 represents the single depot. The set F' contains β_i copies of each CS $i \in F$ (i.e., $|F'| = \sum_{i \in F} \beta_i$). The value of β_i corresponds to an upper bound on the number of times that each CS i can be visited. These copies are introduced for modeling convenience, each visit of an EV to a CS is modeled as a visit to a distinct copy of that particular CS, i.e, each node in F' can be restricted to be visited at most once. Therefore, the use of CS copies facilitates tracking of the SoC and the driving time of each route. In the remainder of this manuscript, depending on the context, we refer to an element of F' or F'_i as a CS copy. We denote as $F'_i \subseteq F'$ 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 and that its elements are numbered from 1 to β_i.

It is noteworthy that the graph G is not necessary complete. Indeed, some arcs from the arc set A can be safely removed without cutting off the optimal solution. Specifically, we introduce the following preprocessing steps to reduce the size of the arc set:

1. Let $\overrightarrow{e}_i^{max} = Q - \min_{l \in F \cup \{0\}} e_{li}$ be the maximum possible SoC at departure of node $i \in V$. Similarly, let $\overrightarrow{e}_i^{min} = \min_{l \in F \cup \{0\}} e_{il}$ be the minimum SoC at departure of node $i \in V$ to build an energy-feasible route. Considering $i, j \in V$, in the case where $\overrightarrow{e}_i^{max} + e_{ij} + \overrightarrow{e}_j^{min} > Q$ we infer that even with the maximum SoC at i, after traversing the arc (i, j) the EV does not have enough SoC to reach the nearest CS or the depot. Therefore, the corresponding arcs (i, j) to this case are removed from A.
2. Let $t^{\star}_{ij}$ be the duration of the shortest path in G from i to j with respect to arc duration t_{ij}, of the resulting graph from step 1. Let ec_{ij}^{min} the minimum charging amount necessary to build a route traversing arc (i, j). We have $ec_{ij}^{min} = \max(0, (e_{0i} + e_{ij} + e_{j0}) - Q)$. We also define t_{ij}^{charge} the minimum charging time necessary to perform a route traversing the arc (i, j). Namely, t_{ij}^{charge} is equal to 0 if $ec_{ij}^{min} = 0$. Otherwise, t_{ij}^{charge} is equal to:

$$t_{ij}^{charge} = \min_{l \in F} \left(\min \left(\left(ec_{ij}^{min} + e_{0l} + e_{li} - e_{0i} \right) / \rho_{l1}, \left(ec_{ij}^{min} + e_{il} + e_{lj} - e_{ij} \right) / \rho_{l1}, \left(ec_{ij}^{min} + e_{jl} + e_{l0} - e_{j0} \right) / \rho_{l1} \right) \right)$$

More specifically, t_{ij}^{charge} is the minimum time spent for the detour to a CS and for charging, as it accounts for fastest possible CS charging for three mutually exclusive detour options. If $t_{0i}^{\star} + t_{ij}^{\star} + t_{j0}^{\star} + t_{ij}^{charge} > T_{max}$, we remove the arc (i, j) from set A, since traversing the arc (i, j) leads to violation of the tour duration limit.

3. We also remove all the arcs between CS copies of the same CS. More specifically, for every $i \in F$ and $j, l \in F_i'$, we remove the arc (j, l) from set A.

In the following, for the sake of completeness and clarity, we present the model introduced in (Montoya et al., 2017) in Section 3.1. In Section 3.2 we introduce alternative formulations, which relate to modeling the SoC as a function of time tracking, as well as the piecewise linearization of the charging functions.

3.1. The E-VRP-NL formulation

The formulation of Montoya et al. (2017) involves the following decisions variables. Binary variable x_{ij} is 1 if and only if an EV travels arc $(i, j) \in A$. Continuous variables τ_j and y_j track the time and SoC of the EV when it departs from node $j \in V$. Continuous variables q_i and o_i specify (according to the piecewise linear approximation of the charging function of i) the SoC of an EV when it arrives at and departs from CS copy $i \in F'$, and s_i and d_i are the scaled arrival time and departure time, according to the charging function of CS copy i. Continuous variable $\Delta_i = d_i - s_i$ represents the duration of the charging operation performed at CS copy i. Binary variables z_{ik} and w_{ik} are 1 if the SoC is between $a_{i,k-1}$ and a_{ik}, with $k \in B_i \backslash \{0\}$, when the EV arrives at and departs from CS copy i, respectively. Finally, continuous variables α_{ik} and λ_{ik} are the coefficients associated with the breakpoint (c_{ik}, a_{ik}) in the piecewise linear approximation, when the EV arrives at and departs from CS copy $i \in F'$ respectively. Specifically, the variables $\{\alpha_{ik}\}_{k \in B_i}$ enable the expression of (s_i, q_i) as a convex combination of the breakpoints $\{(c_{ik}, a_{ik})\}_{k \in B_i}$. Similarly, the variables $\{\lambda_{ik}\}_{k \in B_i}$ enable the expression of (d_i, o_i) as a convex combination of the breakpoints $\{(c_{ik}, a_{ik})\}_{k \in B_i}$. The formulation of the E-VRP-NL, denoted as $[F^M]$, is as follows:

$$[F^M] \quad \min \sum_{i,j \in V, i \neq j} t_{ij} x_{ij} + \sum_{i \in F'} \Delta_i + \sum_{i \in I} g_i \tag{1}$$

subject to

$$\sum_{(i,j) \in A} x_{ij} = 1, \qquad \forall i \in I \tag{2}$$

$$\sum_{(i,j) \in A} x_{ij} \leqslant 1, \qquad \forall i \in F' \tag{3}$$

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

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

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

$$y_i \geqslant e_{i0} x_{i0}, \qquad \forall (i,0) \in A : i \in V \backslash \{0\} \tag{7}$$

$$y_i = o_i, \qquad \forall i \in F' \tag{8}$$
$$y_0 = Q \tag{9}$$
$$q_i \leqslant o_i, \qquad \forall i \in F' \tag{10}$$
$$q_i = \sum_{k \in B_i} \alpha_{ik} a_{ik}, \qquad \forall i \in F' \tag{11}$$
$$s_i = \sum_{k \in B_i} \alpha_{ik} c_{ik}, \qquad \forall i \in F' \tag{12}$$
$$\sum_{k \in B_i} \alpha_{ik} = \sum_{k \in B_i \backslash \{0\}} z_{ik}, \qquad \forall i \in F' \tag{13}$$
$$\sum_{k \in B_i \backslash \{0\}} z_{ik} = \sum_{(i,j) \in A} x_{ij}, \qquad \forall i \in F' \tag{14}$$
$$\alpha_{i0} \leqslant z_{i1}, \qquad \forall i \in F' \tag{15}$$
$$\alpha_{ik} \leqslant z_{ik} + z_{i,k+1}, \qquad \forall i \in F', \forall k \in B_i \backslash \{0, b_i\} \tag{16}$$
$$\alpha_{ib_i} \leqslant z_{ib_i}, \qquad \forall i \in F' \tag{17}$$
$$o_i = \sum_{k \in B_i} \lambda_{ik} a_{ik}, \qquad \forall i \in F' \tag{18}$$
$$d_i = \sum_{k \in B_i} \lambda_{ik} c_{ik}, \qquad \forall i \in F' \tag{19}$$
$$\sum_{k \in B_i} \lambda_{ik} = \sum_{k \in B_i \backslash \{0\}} w_{ik}, \qquad \forall i \in F' \tag{20}$$
$$\sum_{k \in B_i \backslash \{0\}} w_{ik} = \sum_{(i,j) \in A} x_{ij}, \qquad \forall i \in F' \tag{21}$$
$$\lambda_{i0} \leqslant w_{i1}, \qquad \forall i \in F' \tag{22}$$
$$\lambda_{ik} \leqslant w_{ik} + w_{i,k+1}, \qquad \forall i \in F,' \forall k \in B_i \backslash \{0, b_i\} \tag{23}$$
$$\lambda_{ib_i} \leqslant w_{ib_i}, \qquad \forall i \in F' \tag{24}$$
$$\Delta_i = d_i - s_i, \qquad \forall i \in F' \tag{25}$$
$$\tau_i + (t_{ij} + p_j) x_{ij} - T_{max}(1 - x_{ij}) \leqslant \tau_j, \qquad \forall (i,j) \in A : i \in V, j \in I \tag{26}$$
$$\tau_i + \Delta_j + t_{ij} x_{ij} - (S_{max} + T_{max})(1 - x_{ij}) \leqslant \tau_j, \qquad \forall (i,j) \in A : i \in V, j \in F', \tag{27}$$
$$\tau_j + t_{j0} \leqslant T_{max}, \qquad \forall j \in V \tag{28}$$
$$\tau_0 = 0 \tag{29}$$
$$\tau_l - \Delta_l \geqslant \tau_j - \Delta_j, \qquad \forall i \in F, \forall j, l \in F_i', j < l \tag{30}$$
$$\sum_{(h,j) \in A} x_{hj} \geqslant \sum_{(h,l) \in A} x_{hl}, \qquad \forall i \in F, \forall j, l \in F_i', j < l \tag{31}$$
$$x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A \tag{32}$$
$$\tau_i \geqslant 0, y_i \geqslant 0 \qquad \forall i \in V \tag{33}$$
$$z_{ik} \in \{0,1\}, w_{ik} \in \{0,1\}, \qquad \forall i \in F', \forall k \in B_i \backslash \{0\} \tag{34}$$
$$\alpha_{ik} \geqslant 0, \lambda_{ik} \geqslant 0, \qquad \forall i \in F', \forall k \in B_i \tag{35}$$
$$q_i \geqslant 0, o_i \geqslant 0, \Delta_i \geqslant 0, \qquad \forall i \in F' \tag{36}$$
$$s_i \geqslant 0, d_i \geqslant 0 \qquad \forall i \in F' \tag{37}$$

Equation (1) gives the objective of the problem: minimizing the total time (driving times, service times, and charging times). Constraints (2) ensure that each customer is visited once.

Constraints (3) ensure that each CS copy is visited at most once. Constraints (4) impose the flow conservation. Constraints (5) and (6) track the SoC of an EV at each node. Constraints (7) ensure that if an EV travels between a node (customer or CS copy) and the depot, it has sufficient energy to reach its destination. Constraints (8) reset the battery tracking to o_i upon departure from CS copy $i \in F'$. Constraint (9) states that the SoC of an EV leaving the depot is Q. Constraints (10) couple the SoC of an EV when it arrives at and departs from any CS copy. Constraints (11)–(17) define the SoC (and its corresponding charging time) of an EV when it arrives at CS copy i (based on the piecewise linear approximation of the charging function). Similarly, constraints (18)–(24) define the SoC (and its corresponding charging time) of an EV when it departs from CS copy i. Constraints (25) define the time spent at any CS copy. Constraints (26) and (27) track the departure time at each node. Constraints (28) and (29) ensure that the EVs return to the depot no later than T_{max}. While not necessary, constraints (30) and (31) avoid potential symmetry between copies of the CSs. These constraints ensure that the copies of CS i are visited in the order they appear in F_i (i.e, a charging operation at $j \in F_i'$ must start after a charging operation at $l \in F_i'$ if $j > l$). Finally, constraints (32)–(37) define the domain of the decision variables.

To strengthen the previous formulation, we propose adding the following valid inequalities:

$$y_i \geqslant \min_{l \in F \cup \{0\}} e_{il} \qquad \forall i \in V \backslash \{0\} \tag{38}$$

These inequalities state that the SoC of an EV at the departure from a node must be enough to reach the depot or the nearest CS (according to energy consumption).

3.2. Alternative formulations

There usually exists multiple ways to model a problem. Keeping in mind that the efficiency of a MILP solver running a model is very sensitive to different formulations, we now investigate several modeling alternatives to the E-VRP-NL.

3.2.1. Modeling the time and SoC tracking

Formulation $[F^M]$ tracks the time and the SoC of the EVs with variables indexed on the nodes. We therefore refer to constraints (5)-(9), (26)-(29), (31), (33) and (38) as *node-based tracking constraints*. We now propose an alternative way to model the time and SoC tracking by introducing arc-based variables. More precisely, we replace variables τ_j and y_j by the continuous variables τ_{ij} and y_{ij} tracking (respectively) the time and SoC of an EV when it departs from node $i \in V$ to travel arc $(i, j) \in A$. If no vehicle travels the arc (i, j), both variables are 0. We model the time and SoC tracking with the following constraints (hereafter referred to as *arc-based tracking constraints*), that replace the node-based tracking constraints in formulation $[F^M]$:

$$\sum_{(i,j) \in A} y_{ij} - \sum_{(i,j) \in A} e_{ij} x_{ij} = \sum_{(j,l) \in A} y_{jl}, \qquad \forall j \in I \tag{39}$$

$$\sum_{(i,j) \in A} y_{ij} - \sum_{(i,j) \in A} e_{ij} x_{ij} = q_j, \qquad \forall j \in F' \tag{40}$$

$$\sum_{(j,l)\in A} y_{jl} = o_j, \qquad \forall j \in F' \tag{41}$$

$$y_{ij} \leqslant \left(Q - \min_{l\in F\cup\{0\}} e_{li}\right) x_{ij}, \qquad \forall (i,j) \in A \tag{42}$$

$$y_{ij} \geqslant \left(e_{ij} + \min_{l\in F\cup 0} e_{jl}\right) x_{ij}, \qquad \forall (i,j) \in A \tag{43}$$

$$\sum_{(i,j)\in A: i\neq 0} (\tau_{ij} + (t_{ij} + p_j)\, x_{ij}) = \sum_{(j,l)\in A} \tau_{jl}, \qquad \forall j \in I \tag{44}$$

$$\sum_{(i,j)\in A: i\neq 0} (\tau_{ij} + t_{ij} x_{ij}) + \Delta_j = \sum_{(j,l)\in A} \tau_{jl}, \qquad \forall j \in F' \tag{45}$$

$$\tau_{ij} \leqslant (T_{max} - t_{ij} - p_j - t_{j0})\, x_{ij}, \qquad \forall (i,j) \in A : i \neq 0, j \in I \tag{46}$$

$$\tau_{ij} \leqslant \left(T_{max} - t_{ij} - \Delta_j^{min} - t_{j0}\right) x_{ij}, \qquad \forall (i,j) \in A : i \neq 0, j \in F' \tag{47}$$

$$\sum_{(j,h)\in A: j\neq 0} \tau_{jh} - \Delta_j \geqslant \sum_{(l,h)\in A: l\neq 0} \tau_{lh} - \Delta_l, \qquad \forall i \in F, \forall j,h \in F_i' : j < l \tag{48}$$

$$\tau_{ij} \geqslant 0, \qquad \forall (i,j) \in A : i \neq 0 \tag{49}$$

$$y_{ij} \geqslant 0, \qquad \forall (i,j) \in A \tag{50}$$

Constraints (39) track the SoC of an EV at each customer. Constraints (40) track the SoC of an EV when it arrives at a CS copy. Constraints (41) track the SoC of the EV when it leaves a CS copy. Constraints (42) couple the variables y_{ij} and x_{ij}. Constraints (43) 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 depot. They are an adaptation of the valid inequalities (38) associated with the node-based tracking constraints. Constraints (44) track the departure time at each customer. Constraints (45) track the departure time at CS copies. Constraints (46) and (47) 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 without exceeding the tour duration limit. The parameter Δ_j^{min} represents the minimum duration of a charging operation at $j \in F'$. We have $\Delta_j^{min} = \min_{l,l'\in V\setminus\{j\}|(l,j)\in A\wedge(j,l')\in A} (e_{lj} + e_{jl'} - e_{ll'})/\rho_{i1}$. This is a lower bound on the time spent charging to recover the energy consumed to make the detour to j considering the fastest charging rate. Constraints (48) break symmetries created by the introduction of CS copies. Constraints (49) and (50) define the domain of the newly introduced decision variables.

According to the experiments conducted by Ascheuer et al. (2001) on the asymmetric traveling salesman problem with time windows, MILP solvers tend to perform better on arc-based tracking constraints, similar to (39)-(50) than on those based on node-based tracking constraints. Therefore, there is ground to believe that re-formulating the time and SoC tracking in $[F^M]$ using this new model may yield better performance.

3.2.2. Modeling the piecewise linear charging function: two alternative formulations

In formulation $[F^M]$, the piecewise linear charging functions are modeled using the widely used convex combination (CB) model (also referred to as the Lambda-method (Vielma et al., 2010)). Alternatively, we propose two E-VRP-NL models based on new formulations for the

piecewise linear charging functions. Namely, a multiple choice model (MC) and a reduced model (R).

- First, we present the MC model. According to (Croxton et al., 2003) the MC and CB models have the same LP relaxation and lead to the same bounds. However, Vielma et al. (2010) conducted experiments showing that MILP solvers tend to perform better on MC models. Therefore, there is room to believe that re-formulating the charging constraints in the E-VRP-NL using an MC model may yield a MILP that is easier to solve in practice.

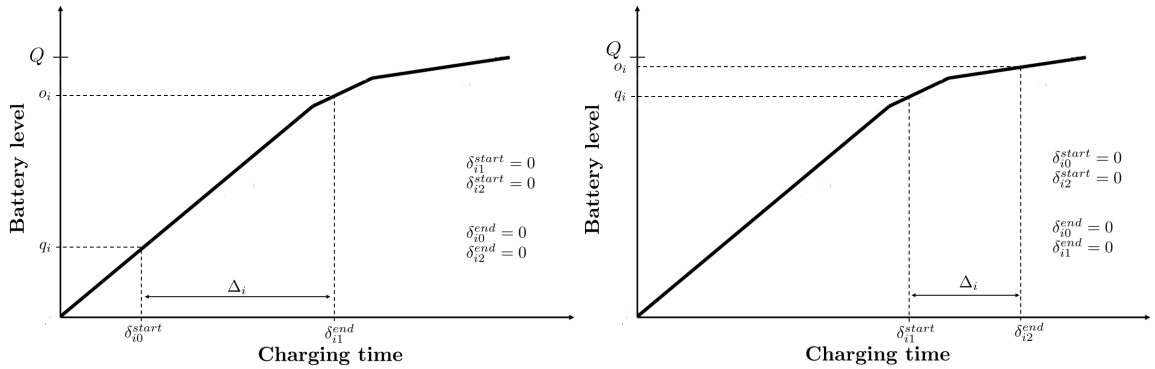


Figure 2: Illustration of the definition of variables δ_{ik}^{start} and δ_{ik}^{end}

To model the piecewise linear charging constraints, we introduce for each $i \in F'$ and $k \in B_i$ the continuous variables δ_{ik}^{start} and δ_{ik}^{end}. If the SoC of the EV lays between $a_{i,k-1}$ and a_{ik} when the EV arrives at (resp. leaves from) CS copy i, δ_{ik}^{start} (resp. δ_{ik}^{end}) takes the value of the abscissa in the charging function corresponding to the SoC. In any other case, δ_{ik}^{start} (resp. δ_{ik}^{end}) takes the value 0. Figure 2 illustrates the definition of these variables. Reusing variables z_{ik} and w_{ik} as defined in §3.1, we can model the piecewise linear charging constraints as:

$$q_i = \sum_{k \in B_i \setminus \{0\}} \rho_{ik} \delta_{ik}^{start} + \eta_{ik} z_{ik}, \qquad \forall i \in F' \tag{51}$$

$$c_{ik-1} z_{ik} \leqslant \delta_{ik}^{start} \leqslant c_{ik} z_{ik}, \qquad \forall i \in F', \forall k \in B_i \setminus \{0\} \tag{52}$$

$$o_i = \sum_{k \in B_i \setminus \{0\}} \rho_{ik} \delta_{ik}^{end} + \eta_{ik} w_{ik}, \qquad \forall i \in F' \tag{53}$$

$$c_{ik-1} w_{ik} \leqslant \delta_{ik}^{end} \leqslant c_{ik} w_{ik}, \qquad \forall i \in F', \forall k \in B_i \setminus \{0\} \tag{54}$$

$$s_i = \sum_{k \in B_i \setminus \{0\}} \delta_{ik}^{start}, \qquad \forall i \in F' \tag{55}$$

$$d_i = \sum_{k \in B_i \setminus \{0\}} \delta_{ik}^{end}, \qquad \forall i \in F' \tag{56}$$

$$\delta_{ik}^{start} \geqslant 0, \delta_{ik}^{end} \geqslant 0, \qquad \forall i \in F', \forall k \in B_i \setminus \{0\} \tag{57}$$

Constraints (51) and (52) couple the SoC of an EV when it arrives at CS copy i to the correct segment and abscissa of the charging function. Similarly, constraints (53) and (54) couple the battery level when an EV departs from CS copy i to the correct segment and abscissa of the charging function. Constraints (55) and (56) help define the time spent at any CS copy (see constraints (25)). Constraints (32)–(37) define the domain of the newly introduced decision variables.

- Second, we present the R model. It takes advantage of the concavity of the charging function to reduce the number of variables and constraints used to model its piecewise linear approximation. We omit variables z_{ik}, w_{ik}, α_{ik} and λ_{ik}. Let each CS copy $i \in F'$ and $k \in B_i \backslash \{0\}$. We introduce the continuous variable ϕ_{ik} that represents the amount of energy charged at CS copy i on the segment that lies between the points $(c_{i,k-1}, a_{i,k-1})$ and (c_{ik}, a_{ik}). We also introduce the binary variable ω_{ik}, which is equal to 1 if and only if an EV charges at CS copy i on the segment between the points $(c_{i,k-1}, a_{i,k-1})$ and (c_{ik}, a_{ik}). Figure 3 illustrates the definition of these variables.

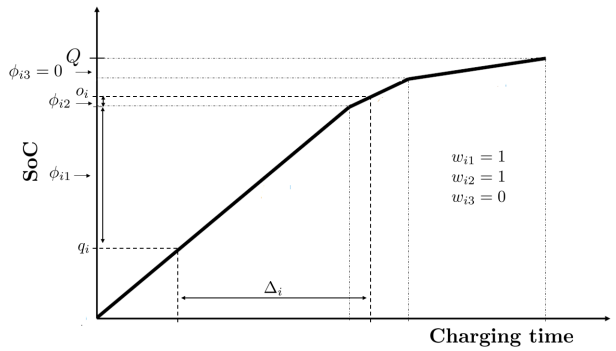


Figure 3: Illustration of the definition of variables used in the R model

A shorter formulation of the piecewise linear charging constraints is then as follows:

$$q_i + \phi_{ik} \leqslant a_{ik}\omega_{ik} + Q(1 - \omega_{ik}), \qquad \forall i \in F', \forall k \in B_i \backslash \{0\} \tag{58}$$

$$\phi_{ik} \leqslant (a_{ik} - a_{i,k-1})\omega_{ik}, \qquad \forall i \in F', \forall k \in B_i \backslash \{0\} \tag{59}$$

$$\sum_{k \in B_i \backslash \{0\}} \omega_{ik} \geqslant \sum_{(i,j) \in A} x_{ij}, \qquad \forall i \in F' \tag{60}$$

$$\omega_{ik} \leqslant \sum_{(i,j) \in A} x_{ij}, \qquad \forall i \in F', \forall k \in B_i \backslash \{0\} \tag{61}$$

$$o_i = q_i + \sum_{k \in B_i \backslash \{0\}} \phi_{ik}, \qquad \forall i \in F' \tag{62}$$

$$\Delta_i = \sum_{k \in B_i \backslash \{0\}} \phi_{ik}/\rho_{ik}, \qquad \forall i \in F' \tag{63}$$

$$\phi_{ik} \geqslant 0 \qquad \forall i \in F', \forall k \in B_i \backslash \{0\} \tag{64}$$

$$\omega_{ik} \in \{0, 1\} \qquad \forall i \in F', \forall k \in B_i \backslash \{0\} \tag{65}$$

Constraints (58) restrict the segments on which EVs can charge according to the SoC they have at their arrival at CS copies. Constraints (59) restrict the charging amount that can be charged on each segment. Constraints (60) impose the activation of one segment whenever an EV visits a CS copy. Constraints (61) are valid inequalities. Constraints (62) define the SoC after a charging operation, by summing up the amounts charged on the relevant segments. Constraints (63) define the time spent charging at each CS copy. Constraints (64)–(65) define the domain of the newly introduced decision variables.

Hereafter, we refer to:

- constraints (11)-(13), (15)-(20), (22)- (24), and (35) defined in $[F^M]$ as *CB piecewise linear constraints*,
- constraints (51)-(57) as *MC piecewise linear constraints*
- to constraints (58)-(65) as *R piecewise linear constraints*

3.2.3. Valid inequalities

To strengthen all the previous formulations, we introduce a new set of valid inequalities. Let ec_l^{min} be the minimum amount of energy that an EV must charge at a CS copy $l \in F'$. This amount is the charge needed to recover the minimum energy consumed to make the detour from the node visited before node l and the node visited after l. Specifically, we have $ec_l^{min} = \min_{i,j \in V \setminus \{l\} | (i,l) \in A \wedge (l,j) \in A} (e_{il} + e_{lj} - e_{ij})$. We can therefore build the following valid constraints:

$$o_i - q_i \geqslant ec_i^{min} \left(\sum_{(i,j) \in A} x_{ij} \right), \qquad \forall i \in F' \tag{66}$$

4. Recharging path-based formulation

A major drawback of the previous formulations defined on a simple graph is the need to replicate CSs nodes. To ensure that no optimal solutions are cut off, the number of copies should be very large, we computed an upper bound of $4|I|$ for this value (see AppendixA). Indeed, the number of CS copies impacts the efficiency of MILP solvers to compute optimal solutions in a reasonable amount of time. Therefore working with an order of $|I|$ copies for each CS is impracticable.

Montoya et al. (2017) proposed a procedure to set the number of CS copies that need to be used in the model. They use the same number β of CS copies associated with every CS. Starting with $\beta = 0$, they solve the MILP formulation to optimality with a time limit of 100h, β is then incremented and the resulting MILP is solved. If $\beta \geqslant 1$ and either the time limit is reached or the optimal solution obtained with $\beta = l$ has the same value as the

one obtained with $\beta = l - 1$, they stop the procedure and fix $\beta = l - 1$. This procedure is not optimal when the time limit is reached. Moreover, even when all problems are solved to optimality, the procedure might cut off the optimal solution in many different cases (see AppendixB for examples).

The main objective of this section is to propose a formulation for the E-VRP-NL that does not explicitly have copies of CSs. To address this issue, we propose an alternative model for the E-VRP-NL based on the concept of recharging paths between each couple of nodes (either customers or the depot). A recharging path is simply a route connecting two nodes (customers or depot) and possibly visiting one or multiple CSs in between. This concept has been introduced (under the name of refuel paths) by Bartolini and Andelmin (2017) for the design of an exact algorithm for the G-VRP. The concept has also been mentioned by Roberti and Wen (2016) for the electric traveling salesman problem. The difference between this concept and the models of Koč and Karaoglan (2016) and Leggieri and Haourari (2017) is that the latter authors assumed that only one CS can be visited between each pair of customers. This assumption may, however, exclude the optimal solution to the G-VRP (see (Bartolini and Andelmin, 2017) for an example).

The concept of recharging paths (hereafter sometimes referred to as paths) leads to the definition of the E-VRP-NL on a directed multigraph $\widetilde{G} = (\widetilde{V}, \widetilde{A})$, where $\widetilde{V} = \{0\} \cup I$ and $\widetilde{A}$ is the set of arcs associated with paths connecting nodes of $\widetilde{V}$.

Let $i, j \in \widetilde{V}$ be two nodes such that $i \neq j$. We define P_{ij} as the set of recharging paths connecting node i to node j by visiting none or a number of CSs. Let P be the set of all recharging paths connecting any couple of nodes in the graph. Specifically, we have $P = \bigcup_{i,j \in \widetilde{V}, i \neq j} P_{ij}$. We denote $o(p)$ and $d(p)$ as the origin and destination of a path $p \in P$. For each path p, we define an arc in $\widetilde{A}$ from $o(p)$ to $d(p)$. Let n_p denote the number of CSs in path p and let $L_p = \{0, 1..., n_p - 1\} \subset \mathbb{N}$ be the set of CS positions in the path p. Let $i(p, l)$, we create a special path $p^0_{ij} \in P_{ij}$ from i to j without having a CS in between.

4.1. Mixed integer program

A path-based formulation of the E-VRP-NL involves the following decisions variables. Binary variable x_p is 1 if and only if an EV travels recharging path $p \in P$. Continuous variables τ_p and y_p track the time and SoC of an EV when it departs from node $o(p)$ to $d(p)$ using path p. For a path p, continuous variables q_{pl} and o_{pl} specify (according to the piecewise linear approximation of the charging function) the SoC of an EV when it arrives at and departs from $i(p, l)$ (i.e. the CS at position $l \in L_p$). Continuous variable Δ_{pl} represent the duration of the charging operation performed at $i(p, l)$. Continuous variable ϕ_{plk} represents the amount of energy charged on the segment that lies between the points $(c_{i(p,l),k-1}, a_{i(p,l),k-1})$ and $(c_{i(p,l),k}, a_{i(p,l),k})$ at the CS $i(p, l)$. Binary variables ω_{plk} equal to one if and only if an EV charges at the CS at position l in path p on the segment between the points $(c_{i(p,l),k-1}, a_{i(p,l),k-1})$ and $(c_{i(p,l),k}, a_{i(p,l),k})$. Let e^p and t^p be the energy consumption and the driving time associated with path $p \in P$.

A path-based formulation of the E-VRP-NL, denoted as $[F^{path}]$, is as follows:

$$[F^{path}] \quad \min \sum_{p\in P}\left(t^p x_p + \sum_{l\in L_p}\Delta_{pl}\right) + \sum_{i\in I} g_i \tag{67}$$

subject to

$$\sum_{j\in\tilde{V},i\neq j}\sum_{p\in P_{ij}} x_p = 1, \quad \forall i \in I \tag{68}$$

$$\sum_{j\in\tilde{V},i\neq j}\sum_{p\in P_{ji}} x_p - \sum_{j\in\tilde{V},i\neq j}\sum_{p\in P_{ij}} x_p = 0, \quad \forall i \in \tilde{V} \tag{69}$$

$$\sum_{l\in\tilde{V},l\neq j}\sum_{p\in P_{lj}}\left(y_p - e^p x_p + \sum_{l\in L_p}(o_{pl}-q_{pl})\right) = \sum_{l\in\tilde{V},l\neq j}\sum_{p\in P_{jl}} y_p, \quad \forall j \in I \tag{70}$$

$$y_p - e_{o(p),i(p,0)} x_p = q_{p0}, \quad \forall p \in P : |L_p| \neq 0 \tag{71}$$

$$o_{p,l-1} - e_{i(p,l-1),i(p,l)} x_p = q_{pl}, \quad \forall p \in P, \forall l \in L_p \backslash \{0\} \tag{72}$$

$$\sum_{i\in\tilde{V},i\neq 0}\sum_{p\in P_{i0}}\left(y_p - e^p x_p - \sum_{l\in L_p}(o_{pl}-q_{pl})\right) \geqslant 0, \quad \forall i \in I \tag{73}$$

$$y_p \leqslant Q x_p, \quad \forall p \in P \tag{74}$$

$$\sum_{i\in\tilde{V}\backslash\{0\},i\neq j}\sum_{p\in P_{ij}}\tau_p + \sum_{i\in V,i\neq j}\sum_{p\in P_{ij}}\left(t^p x_p + \sum_{l\in L_p}\Delta_{pl}\right) + p_j = \sum_{l\in\tilde{V},l\neq j}\sum_{p\in P_{jl}}\tau_p, \quad \forall j \in I \tag{75}$$

$$\tau_p + \sum_{l\in L_p}\Delta_{pl} \leqslant \left(T_{max} - t^p - p_{d(p)} - t_{d(p),0}\right) x_p, \quad \forall p \in P \tag{76}$$

$$q_{pl} + \phi_{plk} \leqslant a_{i(p,l),k}\omega_{plk} + Q(1-\omega_{plk}), \quad \forall p \in P, \forall l \in L_p, \forall k \in B_{i(p,l)}\backslash\{0\} \tag{77}$$

$$\phi_{plk} \leqslant (a_{i(p,l),k} - a_{i(p,l),k-1})\omega_{plk}, \quad \forall p \in P, \forall l \in L_p, \forall k \in B_{i(p,l)}\backslash\{0\} \tag{78}$$

$$\sum_{k\in B_{i(p,l)}\backslash\{0\}}\omega_{plk} \geqslant x_p, \quad \forall p \in P, \forall l \in L_p \tag{79}$$

$$\omega_{plk} \leqslant x_p, \quad \forall p \in P, \forall l \in L_p, \forall k \in B_{i(p,l)}\backslash\{0\} \tag{80}$$

$$o_{pl} = q_{pl} + \sum_{k\in B_{i(p,l)}\backslash\{0\}}\phi_{plk}, \quad \forall p \in P, \forall l \in L_p \tag{81}$$

$$\Delta_{pl} = \sum_{k\in B_{i(p,l)}\backslash\{0\}}\phi_{plk}/\rho_{i(p,l),k}, \quad \forall p \in P, \forall l \in L_p \tag{82}$$

$$x_p \in \{0,1\}, \quad \forall p \in P \tag{83}$$

$$\tau_p \geqslant 0, y_p \geqslant 0 \quad \forall p \in P \tag{84}$$

$$q_{pl}, o_{pl}, \Delta_{pl} \geqslant 0, \quad \forall p \in P, \forall l \in L_p \tag{85}$$

$$\phi_{plk} \geqslant 0 \quad \forall p \in P, \forall l \in L_p, \forall k \in B_{i(p,l)}\backslash\{0\} \tag{86}$$

$$\omega_{plk} \in \{0,1\} \quad \forall p \in P, \forall l \in L_p, \forall k \in B_{i(p,l)}\backslash\{0\} \tag{87}$$

Equation (67) gives the objective of the problem: minimizing the total time (driving times, service times, and charging times). Constraints (68) ensure that each customer is visited once. Constraints (69) impose the flow conservation. Constraints (70) track the SoC

of EVs at each customer. Constraints (71) track the SoC at the arrival at the first CS of each refuel path. Constraints (72) couple the SoC of an EV that leaves a CS to go to another CS. Constraints (73) ensure that if the EV travels between a node (customer or CS copy) and the depot, it has sufficient energy to reach its destination. Constraints (74) couple the SoC tracking variable to the arc travel variables. Constraints (75) track the departure time at each node. Constraints (76) couple the time tracking variable to the arc travel variables, and impose a tour duration limit. Constraints (77) restrict the segments on which EVs can charge according to the SoC they have at their arrival at CSs. Constraints (78) restrict the charging amount that can be charged on each segment. Constraints (79) impose the activation of at least one segment whenever an EV visit a CS. Constraints (80) are coupling constraints. Constraints (81) define the SoC after a charging operation. Constraints (82) define the time spent charging at each CS. Finally, constraints (83)–(87) define the domains of the decision variables.

Notice that this model widely differs from the one introduced by Andelmin (2014) for the G-VRP. First, this latter problem implies several simplifying hypothesis: a full charging policy and a linear approximation of the charging function. Second, the authors used variables indexed on the nodes to track the SoC and the time. It is also noteworthy that the alternative ways of modeling introduced in Section 3 are still valid when working with a multigraph. More specifically, time and SoC tracking can be performed on the node instead of on the arcs. Moreover, CB and MC piecewise linear constraints can be used as alternatives to the R piecewise linear constraints (77)-(82),(86), and (87) of $[F^{path}]$. An adaptation of the $[F^{path}]$ to deal with CB piecewise linear constraints is described in AppendixC. We do not present the other models as preliminary experiments show that the MILP solver running them performs at most as well as when running model $[F^{path}]$.

4.2. Paths and dominance

Without the use of preprocessing techniques, the arc set $\widetilde{A}$ contains a very large number of arcs. One can compute an upper bound on the number of these arcs. For each pair of nodes, we have to consider all the combinations of CSs. For each combination, we need to consider all the possible permutations allowing each CS to be possibly visited twice if we consider CSs with different technologies. An upper bound Γ on the number of arcs is therefore defined as follows:

$$\Gamma = |I|(|I|-1) \cdot \left(\sum_{k=1}^{|F|} \left(\begin{array}{c} |F| \\ k \end{array} \right) \cdot \left(\sum_{l=1}^{k} \frac{(k+l)!}{2^l} \right) \right) \tag{88}$$

If all the CSs have the same technologies, the upper bound Γ can be reduced to:

$$\Gamma = |I|(|I|-1) \cdot \sum_{k=1}^{|F|} \frac{|F|!}{k!} \tag{89}$$

The previous bounds (although quite loose) clearly demonstrate how the number of paths 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. To find these paths, we exploit the concept of state of charge (SoC) functions (hereafter referred to as SoC-functions) associated with paths. This concept has been introduced in (Zündorf, 2014) to solve an electric-shortest path problem[1]. Let $p \in P$ be a path and $0 \leqslant q \leqslant Q$ an initial SoC for an EV when leaving the origin $o(p)$. A SoC-function SoC_p^q for a path p with an initial SoC q maps a duration $t \in \mathbb{R}$ to a final SoC at the destination $d(p)$. Since we assume that the charging functions are piecewise linear, a SoC-function can be defined by a set of supporting points (t_1, q_1),...,(t_k, q_k) sorted by ascending order. t_1 represents the minimal duration for an EV to travel the path according to the battery constraints (no under-overcharging is allowed and the SoC is sufficient to travel between each arc) and the initial SoC. A SoC-function represented by the set of supporting points (t_1, q_1),...,(t_k, q_k) is defined as follows:

$$SoC_p^q(t) = \begin{cases} -\infty & \text{if } 0 \leqslant t < t_1 \\ \dfrac{(t - t_1)(q_2 - q_1)}{t_2} + q_1 & \text{if } t_1 \leqslant t < t_2 \\ \ldots & \ldots \\ \dfrac{(t - t_{k-1})(q_2 - q_1)}{t_{k+1}} + q_{k-1} & \text{if } t_{k-1} \leqslant t < t_k \\ q_k & \text{else} \end{cases} \tag{90}$$

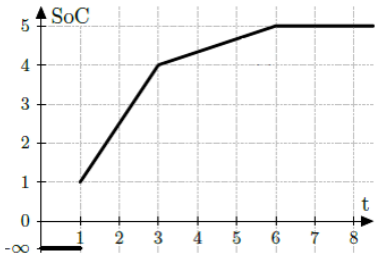


Figure 4: Example of a SoC-function associated with the supporting points (1,1),(3,4) and (6,5) taken from (Zündorf, 2014).

Figure 4 illustrates the concept of SoC-function. In this example, the first supporting point of the curve is (1,1). This means that the minimum duration to travel the path is equal to 1, and in that case, the arrival SoC at the destination of the EV is equal to 1. If at the destination of the path (i.e. at $d(p)$) the SoC needed for the EV is 4, the duration of the corresponding path is equal to 3 (the time is increased due to a larger amount of energy charged at a CS part of the path).

For a known initial SoC, we now introduce the concept of dominance between paths having same origin and destination.

[1]This problem consists in finding the route with the minimum traveling time respecting battery constraints and taking into account that the EV can stop in route at CSs.

Definition 4.1. *Let $p, p' \in P$ be two paths with same origin $o = o(p) = o(p')$ and same destination $d = d(p) = d(p')$ and $0 \leqslant q \leqslant Q$ an initial SoC of the EV when leaving the origin node o. Path p is said to dominate path p' with respect to q if $SoC_p^q(t) \geqslant SoC_{p'}^q(t)$ for every $t \geqslant 0$. We denote $p \succ_q p'$.*

The previous definition states that a path p dominates another path p' only if for every possible travel time the EV has a higher SoC at the destination. Zündorf (2014) proved that this dominance rule does not remove any path that could be part of an optimal solution. Let us now introduce the concept of dominance between paths with the same origin and destination, independently of the initial SoC of the EV when leaving the origin.

Definition 4.2. *Let $p, p' \in P$ be two paths with same origin $o = o(p) = o(p')$ and same destination $d = d(p) = d(p')$. Path p is said to dominate path p' if $p \succ_q p'$ for every $0 \leqslant q \leqslant Q$. We denote $p \succ p'$.*

Let SoC_i^{MIN} and SoC_i^{MAX} be the minimal and maximal SoC of an EV at node $i \in \widetilde{V}$. Let $SoC_i^{MIN} = \min_{j \in F \cup 0} e_{ij}$ (the SoC must be enough to reach the depot or the nearest CS according to energy consumption). Furthermore, let $SoC_i^{MAX} = Q - \min_{l \in F \cup 0} e_{li}$. Let $F(i)$ be the set of CSs reachable from i by an EV having a SoC between SoC_i^{MIN} and SoC_i^{MAX}. More specifically, we have $F(i) = \{l \in F | SoC_i^{MIN} \leqslant e_{il} \leqslant SoC_i^{MAX}\}$.

Proposition 4.1. *Let $i, j \in \widetilde{V}, i \neq j$ be two nodes of the multigraph and $p, p' \in P_{ij}$ be two paths. Path p dominates path p' (i.e., $p \succ p'$) if $p \succ_q p'$ for every $q \in \{e_{ij} | j \in F(i)\}$.*

Proof. First, observe that the path going directly from i to j without visiting a CS, if feasible (i.e. $e_{ij} \leqslant SoC_i^{MAX}$), can never be dominated since the driving times verify the triangular inequality. Let $SoC_i^{MIN} \leqslant q \leqslant SoC_i^{MAX}$ be the initial SoC of the EV when it departs from node i and p be a feasible non-dominated path (containing at least one CS) with respect to q. We need to prove that there exists $q' \in \{e_{ij} | j \in F(i)\}$ such that p is non-dominated with respect to q'. The cases $q < SoC_i^{MIN}$ and $q > SoC_i^{MAX}$ do not need to be considered since it cannot be part of any feasible solution to the E-VRP-NL. Also, since the path p contains at least one CS, there exists $l \in F(i)$ such that $e_{il} \leqslant q$ (to satisfy the battery constraints). Let $l_1(p)$ be the first CS visited in path p, $Y(l_1(p))$ the SoC of the EV when it arrives at $l_1(p)$, and $\Phi(l_1(p))$ the amount of energy charged at $l_1(p)$. Let us prove that p is non-dominated with respect to $q' = e_{i,l_1(p)}$. First, the CS $l_1(p)$ is obviously reachable without visiting any CS if the initial SoC of the EV is q'. Second, the EV arrives at $l_1(p)$ with a larger SoC (i.e. $Y(l_1(p)) \geqslant 0$) if its initial SoC is q rather than q'. Given that charging first at CS $l_1(p)$ in path p makes this latter non-dominated with respect to q, charging the amount of energy $\Phi(l_1(p)) + q - q'$ at CS $l_1(p)$ makes p a non-dominated path with respect to q' (because of the dominance property between charging functions). □

To compute the set P_{ij} for each couple of nodes $i, j \in \widetilde{V}, i \neq j$, we apply the procedure described in Algorithm 1. Using the result of Proposition 4.1, we first fix the SoC q at the departure of i before using a label-correcting algorithm to compute all the non-dominated paths between i and j if the initial SoC is q. This algorithm is inspired by to the one

presented by Zündorf (2014). The underlying directed graph simply consists of a graph containing nodes i and j and every CS node. For each CS node l, we create the arcs (l, l') where l' is another CS different from l and the arcs (i, l) and (l, j). We also create the arc (i, j). When visiting a CS, we must determine the amount of energy to charge. However, without knowing the further course of the path, this cannot be done. The idea of the algorithm is thus to delay the decision about how much energy should be charge at CSs as long as possible. For this purpose, it uses SoC-functions as labels. When the EV traverses an arc, the missing energy (if there is some) is charged retroactively at the previous CS (if possible). Otherwise traversing the edge is impossible when extending the label. As soon as we want to set a CS node, we need to set the energy we charged at the previous CS (if one exists in the current path). Zündorf (2014) proved that changing CSs is only meaningful at a supporting point of the current SoC-function. Using this result, we create one new label for each supporting point of the current SoC-function in order to explore the possibility of switching over to the new CS at that point.

Algorithm 1: ComputeRechargingPaths(i,j)

input : two nodes $i, j \in \widetilde{V}, i \neq j$
output: a set containing all the non-dominated recharging paths between i and j

$P \leftarrow \varnothing$ (P stores the non-dominated recharging paths)
if $\left(e_{ij} + \min_{l \in F \cup \{0\}} e_{li} + \min_{l \in F \cup \{0\}} e_{jl} \leqslant Q\right)$ **then**
  $P \leftarrow P \cup \{p_{ij}^0\}$
end
for $l \in F(i)$ **do**
  Use a label-correcting algorithm to compute all the non-dominated paths with respect of an initial SoC of e_{il}.
  for each non-dominated label at j **do**
    Let p be the recharging path associated with the label
    if $p \notin P$ **then**
      $P \leftarrow P \cup \{p\}$
    end
  end
end
return P

5. Computational experiments

We tested the different models presented in this document. We used Gurobi 7.0.2 to solve the MILP models through its Java API. All experiments were performed, using a single thread with 12 GB, on a cluster of 27 computers, each of which having 12 cores and two Intel(R) Xeon® X5675 3.07 GHz processors. We set a 3-hour time limit (the CPU times are reported in seconds and rounded to the nearest integer). In order to assess the

performance of the proposed formulations, we performed our tests on small-sized instances. We considered the 20 instances of the 120-instances testbed proposed by Montoya et al. (2017) that contains 10 customers.

5.1. CS replication-based formulation

First, we tested the efficiency of the MILP solver running the different models coming from the modeling alternatives presented in Section 3, which are based on the replication of CS nodes. The number of copies of each CS $i \in F$ is set to $\beta \in \mathbb{N}\backslash\{0\}$ (i.e. $\beta_i = \beta,\ \forall i \in F$). We refer to each model via "a_b" where a and b refer to the modeling of the SoC and time tracking (A: arc-based tracking constraints/ N: node-based tracking constraints) and of the charging function (C: CB piecewise linear constraints / M: MC piecewise linear constraints / R: R piecewise linear constraints). Note that formulation $[F^M]$ corresponds to notation N_C.

Table 1 presents the number of instances proven infeasible by the solver within the time limit according to the models and different values of β. In Table 2, we report for each formulation and each value of β the number of instances optimally solved to the number of instances with a feasible solution (#Opt), the average solution time (Time) for the instances solved to optimality, the average gap for the unsolved instances (Gap), and the gap of the solution with respect to the maximal lower bound found by the solver running any of the models (Gap to best LB). We compute the gap $(z - z^{LB})/z$ where z is the objective of the solution returned by the solver and z^{LB} is either the lower bound retrieved by the solver running the corresponding model (column "Gap") or the maximal lower bound found by any of the models (column "Gap to best LB"). Moreover, since average values do not provide sufficient information, Figure 5 shows the number of instances optimally solved according to the solution time for different models and some values of β. The detailed results of for all instances are reported in AppendixD.

Table 1: Number of instances proven infeasible by the solver within the time limit

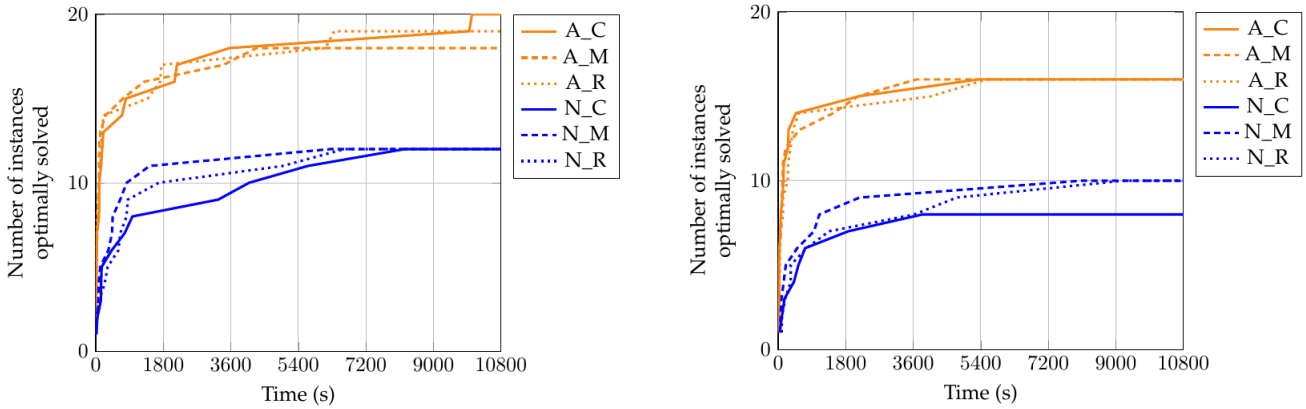
β	A_C	A_M	A_R	N_C	N_M	N_R
1	9	9	9	10	10	10
2	1	1	1	5	5	5
3	0	0	0	0	0	0
4	0	0	0	0	0	0

First, we observe that models with arc-based tracking constraints have difficulties to prove infeasibility of instances for small values of β. This may come from the larger number of variables they have (around two times more). Second, the results confirm that the value of β influences the feasibility of the instances. Since we do not know any procedure for fixing β, which guarantees optimality, we can only compare models for identical values of β. The best results are obtained when using arc-based tracking constraints: the number of optimal solutions is larger and the solution time is significantly reduced, compared to node-based tracking. Moreover, when the time limit is reached, the best solution returned by the arc-based models have a better quality. We also see that the modeling of the charging

function has no significant impact on the efficiency of the MILP solver. Finally, results tend to show the difficulty of optimally solving even small instances with CS replication-based models.

Table 2: Detailed computational results for the different models on the 10-customer instances

β		A_C	A_M	A_R	N_C	N_M	N_R
1	#Opt	10/10	10/10	10/10	10/10	10/10	10/10
	Time (s)	24	9	4	433	632	943
2	#Opt	15/15	15/15	15/15	11/15	11/15	11/15
	Time (s)	355	602	262	825	436	311
	Gap	-	-	-	22.6%	20.3%	17.8%
	Gap to best LB	-	-	-	0.0%	0.0%	0.0%
3	#Opt	20/20	18/20	19/20	12/20	12/20	12/20
	Time (s)	1511	586	963	1992	891	1408
	Gap	-	10.3%	15.5%	26.2%	25.5%	25.1%
	Gap to best LB	-	0.0%	0.0%	0.0%	0.1%	0.1%
4	#Opt	16/20	16/20	16/20	8/20	10/20	10/20
	Time (s)	574	541	746	963	1340	2077
	Gap	10.7%	13.7%	11.1%	22.9%	24.0%	25.9%
	Gap to best LB	8.1%	8.1%	8.1%	3.0%	3.6%	3.3%



(a) $\beta = 3$ (b) $\beta = 4$

Figure 5: Performance charts of the different CS replication-based formulations for different values of β

We also computed the average relative gap between the linear relaxation value and the best lower bound provided by the linear relaxation of any of the formulations. Using arc tracking variables yielded tighter formulations than using node tracking variables. More specifically, the gap between the values returned by these two types of models is around 50%. This is the main reason why a MILP solver running the arc-based models yields better performance than running the node-based ones. There is no influence of the piecewise linear constraints when using node-tracking. When using arc tracking, the R piecewise linear constraints can yield worst, equal, and better LP values than the other piecewise

linear constraints. Thus, we conclude that on the considered instances, the modeling of the charging function has very little impact on the results.

5.2. Recharging path-based formulation

We tested two recharging path-based formulations according to the modeling of the piecewise linear constraints: CB and R. In the two models, we use arc-based tracking constraints. We refer to formulation $[F^{path}]$ using notation Path_A_R and to the formulation using CB piecewise linear constraints with notation Path_A_C.

First, we tested the labeling algorithm with and without the exploitation of the dominance rule presented in Proposition 4.1. On our 20 instances, the average number of paths is reduced by a factor of 3.5 (on average 1071 paths without the dominance rule vs. 280 with). After applying the dominance rule, the average number of paths between each couple of nodes is approximately 2.5 for the 10-customer instances. In Table 3, we report for each formulation the number of instances solved to optimality (#Opt) and the average solution time (Time). for the sake of a better understanding of the results, Figure 6 shows the number of instances optimally solved according to the solution time. See AppendixD for the detailed results for each instance.

Table 3: Computational results on the 10-customer instances

	Path_A_C	Path_A_R
#Opt	20/20	20/20
Time (s)	415	439

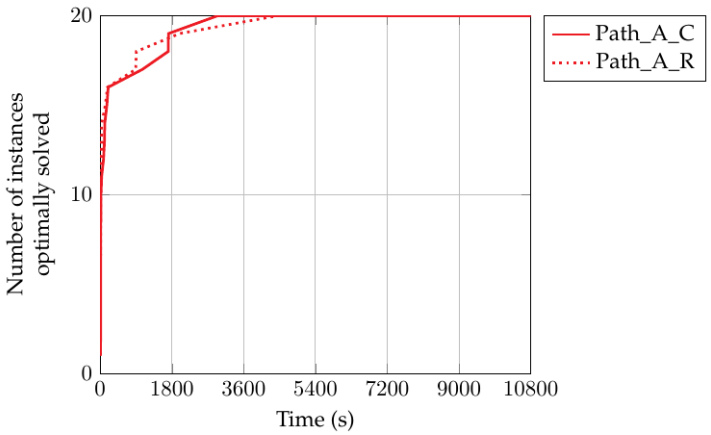


Figure 6: Performance charts of the different models

Results show that the MILP solver can optimally solve all the 10-customers instances in an average time of 7min. Nonetheless, the instances where customers are clustered seem to be the most difficult to solved (see the detailed results), and the solution time can increase subsequently (up to more than one hour).

We clearly see that the MILP solver tends to perform better on the recharging path-based models rather than on the CS replication-based models. Moreover, using the first type of model ensures to not cut off the optimal solution. Some preliminary tests on instances with 20 customers show that we are able to optimally solve (within the 3-hour time limit) nearly half of the 20-customer instances of the testbed of Montoya et al. (2017).

6. Conclusions and perspectives

In this study, we have introduced new MILP formulations for the e-VRP-NL. The first formulation of the problem was proposed by Montoya et al. (2017), where tracking the time and the SoC of each route was done using arc-based variables. We first proposed arc-based tracking variables. Considering the piecewise linear approximation of the charging functions, we proposed two additional alternative formulations, compared to Montoya et al. (2017). The intersection of these modeling components yields six formulations.

The majority of the eVRP literature, as well as the six previously mentioned models necessitate determining CS-replications. We have discussed and shown, that choosing a limited number of copies compromises the quality of the solution, by possibly eliminating optimal solutions or by simply yielding an unfeasible problem. Choosing a large number of copies yields large problems, which are difficult to solve even for small-sized instances. To overcome these drawbacks, we proposed another formulation of the problem, based on the concept of recharging paths between none-CS nodes. Using dominance rules to discard unpromising paths, this formulation significantly improves the results compared to the previously mentioned six formulations.

Experiments showed that the modeling of the piecewise linear approximation of the charging functions only has a limited impact on the results. However, tracking the time and the SoC of each route using arc-based variables (instead of node-based variables) drastically improves the results. This is mainly due to the stronger LP bound achieved by the arc-based formulations. The results of the recharging path-based models are significantly better, as all considered instances are solved to optimality in a much shorter time.

Future research could aim at developing an exact algorithm able to solve medium and some large-sized instances. Finally, one of the key assumptions in the E-VRP-NL is that CSs are able to simultaneously handle an unlimited number of EVs. In practice, however, each CS has a limited number of chargers. Handling capacitated CSs is therefore a timely and relevant challenge to address.

References

Andelmin, J. (2014). Electric vehicle routing with realistic recharging models. Master's thesis, Aalto University, Helsinki, Finland.

Ascheuer, N., Fischetti, M., and Grötschel, M. (2001). Solving the Asymmetric Travelling Salesman Problem with time windows by branch-and-cut. *Mathematical Programming*, 90(3):475–506.

Bartolini, E. and Andelmin, J. (2017). An exact algorithm for the green vehicle routing problem. *Transporation Science*.

Bruglieri, M., Pezzella, F., Pisacane, O., and Suraci, S. (2015). A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 47:221–228.

Croxton, K., Gendron, B., and Magnanti, T. (2003). A Comparison of Mixed-Integer Programming Models for nonconvex Piecewise Linear Cost Minimization Problems. *Management Science*, 49(3):1268–1273.

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.

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., 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.

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

Hiermann, G., Puchinger, J., Ropke, S., and Hartl, R. (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.

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

Koč, 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. (2016). Electric vehicle routing with mid-route recharging and uncertain charging station availability. In *INFORMS Annual Meeting 2016*.

Leggieri, V. and Haourari, M. (2017). A practical solution approach for the green vehicle routing problem. *Transportation Research Part C: Emerging Technologies*, 104:97–112.

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*. doi: 10.1016/j.trb.2017.02.004.

Roberti, R. and Wen, M. (2016). The electric traveling salesman problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 89:32–52.

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

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.

Schneider, M., Stenger, A., and Hof, J. (2015). An adaptive vns algorithm for vehicle routing problems with intermediate stops. *OR Spectrum*, 37(2):353–387.

Sweda, T., Dolinskaya, I., and Klabjan, D. (2017). Adaptive routing and recharging policies for electric vehicles. *Transportation Science*.

Vielma, J., Ahmed, S., and Nemhauser, G. (2010). Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions. *Operations Research*, 58(2):303–315.

Yang, J. and Sun, H. (2015). Battery swap station location-routing problem with capacitated electric vehicles. *Computers & Operations Research*, 55:217–232.

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

AppendixA. Upper bound on the number of CS copies

If we consider homogeneous CSs, an upper bound on the number of CS copies to consider is $2|I|$. Indeed, at worst we need an EV for each customer and a visit to a CS before and after serving it. If we consider heterogeneous CSs, an upper bound on the number of CS copies to consider is $4|I|$. Indeed, in that case a CS can be visited twice between the depot (or the customer) and the customer (or the depot). Let j be a slow CS and l a fast CS. Let i be a customer such that $t_{0j} < t_{0l}$, $e_{0j} \leqslant Q < e_{0l} \leqslant e_{0i}$, $t_{ji} < t_{li}$, and $e_{ji} < e_{li}$. The EV can go to j to charge the energy needed to reach l (i.e. $e_{jl} - (Q - e_{0j})$) and then go back for charging at j to have a SoC larger at the customer i.

AppendixB. Examples showing the non-optimality of the Montoya et al. (2017) procedure

In this section, we introduce 2 examples showing the non-optimality of the procedure introduced by Montoya et al. (2017).

In the first example (denoted Example 1), we have 1 depot (denoted 0), 2 customers (denoted 1 and 2), and 1 CS (denoted 3). The battery capacity of the EVs is 10 and T_{max} is set to 20. Tables B.4a and B.4b show the driving times and energy consumption between the nodes. Figure B.4c shows the charging function at the CS.

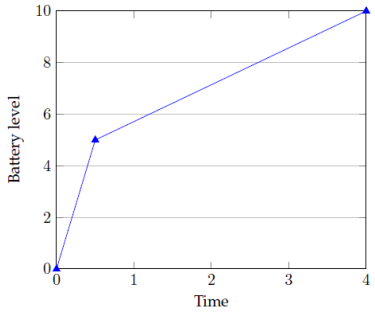
Table B.4: Data of Example 1.

	0	**1**	**2**	**3**
0	0	4	4	2
1	4	0	5	3
2	4	5	0	3
3	2	3	3	0

(a) Driving times.

	0	**1**	**2**	**3**
0	0	5	5	3
1	5	0	7	2.5
2	5	7	0	5
3	3	2.5	5	0

(b) Energy consumption.



(c) Charging function associated with CS 3

For $\beta = 0$ (i.e., the CS cannot be visited) and $\beta = 1$ (i.e., the CS can be visited at most once), the optimal solutions contains two routes $(0 \rightarrow 1 \rightarrow 0)$ and $(0 \rightarrow 2 \rightarrow 0)$ and the

total time is equal to 17. For $\beta = 2$ (i.e.,the CS can be visited at most twice), the optimal solution contains one route $(0 \to 2 \to 3 \to 1 \to 3 \to 0)$ and the total time is equal to 16.8. In that solution, we arrive at CS 3 with a 0-SoC. We charge the battery up to 50%-SoC and 30%-SoC during the first and second visits (it takes 0.5 and 0.3 time units, respectively).

Let us now consider the case where the charging function is linear. We introduce another example (denoted as Example 2). We have 1 depot (denoted 0), 4 customers (denoted 1, 2, 3, and 4), and 1 CS (denoted 5). The battery capacity of the EVs is 10 and T_{max} is set to 10. Tables B.5a and B.5b show the driving times and energy consumption between the nodes. Figure B.5c shows the charging function at the CS 5.

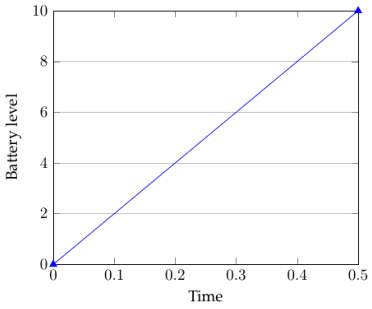
Table B.5: Data of Example 2.

	0	**1**	**2**	**3**	**4**	**5**
0	0	1	2	2	1	1
1	1	0	1	1	1	1
2	2	1	0	1	1	1
3	2	1	1	0	1	1
4	1	1	1	1	0	1
5	1	1	1	1	1	0

(a) Driving times.

	0	**1**	**2**	**3**	**4**	**5**
0	0	2.5	3	3	2.5	6
1	2.5	0	5	5.5	5	4
2	3	5	0	2	5.5	4
3	3	5.5	2	0	5	4
4	2.5	5	5.5	5	0	4
5	6	4	4	4	4	0

(b) Energy consumption.



(c) Charging function associated with CS 5

For $\beta = 0$ and $\beta = 1$, the optimal solutions contains two routes $(0 \to 1 \to 4 \to 0)$ and $(0 \to 2 \to 3 \to 0)$ and the total time is equal to 10. For $\beta = 2$, the optimal solution contains one route $(0 \to 1 \to 5 \to 2 \to 3 \to 5 \to 4 \to 0)$ and the total time is equal to 9.65. At the first visit to the CS 5, the arrival SoC of the EV is equal to 35% and we charge the battery up to 100%-SoC (it takes 0.325 time units). At the second visit to the CS 5, the arrival SoC of the EV is equal to 0% and we charge the battery up to 65%-SoC (it takes 0.325 time units).

Notice that if the charging function is assumed to be constantly equal to a value $\delta \geqslant 0$ in Example 2, the solutions remain the same. The solution for $\beta = 0$ or 1 has an objective

value equal to 10. The solution for $\beta = 2$ is equal to $9 + 2\delta$. So a better solution is found as soon as $\delta < 0.5$.

In conclusion, the procedure of Montoya et al. (2017) is not optimal for any approximation of the charging functions (constant, linear, and nonlinear).

AppendixC. Adaptation of the formulation $[F^{path}]$ to the use of a CB model

To write the formulation $[F^{path}]$ with CB piecewise linear constraints, we remove variables ϕ_{plk} and ω_{plk} for the variables z_{plk}, w_{plk}, α_{plk}, and λ_{plk} (their definition are identical to the one provided for the CS replication-based model). We also replace constraints (77)-(82) and (86)-(87) by the following constraints:

$$q_{pl} = \sum_{k \in B_{i(p,l)}} \alpha_{plk} a_{i(p,l)k}, \qquad \forall p \in P, \forall l \in L_p \tag{C.1}$$

$$s_{pl} = \sum_{k \in B_{i(p,l)}} \alpha_{plk} c_{i(p,l)k}, \qquad \forall p \in P, \forall l \in L_p \tag{C.2}$$

$$\sum_{k \in B_{i(p,l)}} \alpha_{plk} = \sum_{k \in B_{i(p,l)} \setminus \{0\}} z_{plk}, \qquad \forall p \in P, \forall l \in L_p \tag{C.3}$$

$$\sum_{k \in B_{i(p,l)} \setminus \{0\}} z_{plk} = x_p, \qquad \forall p \in P, \forall l \in L_p \tag{C.4}$$

$$\alpha_{pl0} \leqslant z_{pl1}, \qquad \forall p \in P, \forall l \in L_p \tag{C.5}$$

$$\alpha_{plk} \leqslant z_{plk} + z_{pl,k+1}, \qquad \forall p \in P, \forall l \in L_p, \forall k \in B_{i(p,l)} \setminus \{0, b_{i(p,l)}\} \tag{C.6}$$

$$\alpha_{plb_{i(p,l)}} \leqslant z_{plb_{i(p,l)}}, \qquad \forall p \in P, \forall l \in L_p \tag{C.7}$$

$$o_{pl} = \sum_{k \in B_{i(p,l)}} \lambda_{plk} a_{i(p,l)k}, \qquad \forall p \in P, \forall l \in L_p \tag{C.8}$$

$$d_{pl} = \sum_{k \in B_{i(p,l)}} \lambda_{plk} c_{i(p,l)k}, \qquad \forall p \in P, \forall l \in L_p \tag{C.9}$$

$$\sum_{k \in B_{i(p,l)}} \lambda_{plk} = \sum_{k \in B_{i(p,l)} \setminus \{0\}} w_{plk}, \qquad \forall p \in P, \forall l \in L_p \tag{C.10}$$

$$\sum_{k \in B_{i(p,l)} \setminus \{0\}} w_{plk} = x_p, \qquad \forall p \in P, \forall l \in L_p \tag{C.11}$$

$$\lambda_{i0} \leqslant w_{pl1}, \qquad \forall p \in P, \forall l \in L_p \tag{C.12}$$

$$\lambda_{plk} \leqslant w_{plk} + w_{pl,k+1}, \qquad \forall p \in P, \forall l \in L_p, \forall k \in B_{i(p,l)} \setminus \{0, b_{i(p,l)}\} \tag{C.13}$$

$$\lambda_{plb_{i(p,l)}} \leqslant w_{plb_{i(p,l)}}, \qquad \forall p \in P, \forall l \in L_p \tag{C.14}$$

$$\Delta_{pl} = d_{pl} - s_{pl}, \qquad \forall p \in P, \forall l \in L_p \tag{C.15}$$

$$z_{plk} \in \{0,1\}, w_{plk} \in \{0,1\}, \qquad \forall p \in P, \forall l \in L_p, \forall k \in B_i \setminus \{0\} \tag{C.16}$$

$$\alpha_{plk} \geqslant 0, \lambda_{plk} \geqslant 0, \qquad \forall p \in P, \forall l \in L_p, \forall k \in B_i \tag{C.17}$$

$$s_{pl} \geqslant 0, d_{pl} \geqslant 0 \qquad \forall i \in F' \tag{C.18}$$

AppendixD. Detailed computational results

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). The symbol "Inf" means that the instance has been proven infeasible, whereas the symbol "-" means that no feasible solution has been found by the solver but the instance has not been proven infeasible.

Table D.6: Detailed computational results on the 10-customer instances for the CS replication-based formulations ($\beta = 1$)

Instance	A_C			A_M			A_R		
	Obj	Bound	Time	Obj	Bound	Time	Obj	Bound	Time
tc0c10s2cf1	Inf	Inf	0	Inf	Inf	0	Inf	Inf	0
tc0c10s2ct1	17.30	17.30	1	17.30	17.30	1	17.30	17.30	2
tc0c10s3cf1	Inf	Inf	27	Inf	Inf	56	Inf	Inf	76
tc0c10s3ct1	15.80	15.80	1	15.80	15.80	1	15.80	15.80	1
tc1c10s2cf2	14.03	14.03	2	14.03	14.03	1	14.03	14.03	2
tc1c10s2cf3	Inf	Inf	0	Inf	Inf	0	Inf	Inf	0
tc1c10s2cf4	Inf	Inf	0	Inf	Inf	0	Inf	Inf	0
tc1c10s2ct2	15.76	15.76	8	15.76	15.76	10	15.76	15.76	9
tc1c10s2ct3	Inf	Inf	1852	Inf	Inf	1058	Inf	Inf	2831
tc1c10s2ct4	Inf	Inf	0	Inf	Inf	0	Inf	Inf	0
tc1c10s3cf2	14.03	14.03	1	14.03	14.03	1	14.03	14.03	1
tc1c10s3cf3	Inf	Inf	0	Inf	Inf	0	Inf	Inf	0
tc1c10s3cf4	19.95	19.95	2	19.95	19.95	2	19.95	19.95	2
tc1c10s3ct2	14.20	14.20	18	14.20	14.20	5	14.20	14.20	7
tc1c10s3ct3	18.02	18.02	11	18.02	18.02	3	18.02	18.02	3
tc1c10s3ct4	18.21	18.21	2	18.21	18.21	4	18.21	18.21	4
tc2c10s2cf0	Inf	Inf	304	Inf	Inf	259	Inf	Inf	203
tc2c10s2ct0	-	-	10800	-	-	10800	-	-	10800
tc2c10s3cf0	Inf	Inf	234	Inf	Inf	271	Inf	Inf	494
tc2c10s3ct0	16.51	16.51	195	16.51	16.51	65	16.51	16.51	8

Instance	N_C			N_M			N_R		
	Obj	Bound	Time	Obj	Bound	Time	Obj	Bound	Time
tc0c10s2cf1	Inf	Inf	1	Inf	Inf	1	Inf	Inf	0
tc0c10s2ct1	17.30	17.30	5	17.30	17.30	3	17.30	17.30	3
tc0c10s3cf1	Inf	Inf	1	Inf	Inf	1	Inf	Inf	4
tc0c10s3ct1	15.80	15.80	9	15.80	15.80	7	15.80	15.80	11
tc1c10s2cf2	14.03	14.03	13	14.03	14.03	27	14.03	14.03	57
tc1c10s2cf3	Inf	Inf	0	Inf	Inf	0	Inf	Inf	0
tc1c10s2cf4	Inf	Inf	0	Inf	Inf	0	Inf	Inf	0
tc1c10s2ct2	15.76	15.76	526	15.76	15.76	335	15.76	15.76	288
tc1c10s2ct3	Inf	Inf	50	Inf	Inf	5	Inf	Inf	4
tc1c10s2ct4	Inf	Inf	0	Inf	Inf	0	Inf	Inf	0
tc1c10s3cf2	14.03	14.03	16	14.03	14.03	26	14.03	14.03	30
tc1c10s3cf3	Inf	Inf	0	Inf	Inf	0	Inf	Inf	0
tc1c10s3cf4	19.95	19.95	4	19.95	19.95	4	19.95	19.95	4
tc1c10s3ct2	14.20	14.20	856	14.20	14.20	1313	14.20	14.20	915
tc1c10s3ct3	18.02	18.02	1762	18.02	18.02	2086	18.02	18.02	922
tc1c10s3ct4	18.21	18.21	13	18.21	18.21	14	18.21	18.21	6
tc2c10s2cf0	Inf	Inf	1	Inf	Inf	1	Inf	Inf	1
tc2c10s2ct0	Inf	Inf	111	Inf	Inf	254	Inf	Inf	472
tc2c10s3cf0	Inf	Inf	1	Inf	Inf	1	Inf	Inf	1
tc2c10s3ct0	16.51	16.51	1128	16.51	16.51	2509	16.51	16.51	7198

Table D.7: Detailed computational results on the 10-customer instances for the CS replication-based formulations ($\beta = 2$)

Instance	A_C			A_M			A_R		
	Obj	Bound	Time	Obj	Bound	Time	Obj	Bound	Time
tc0c10s2cf1	Inf	Inf	849	Inf	Inf	1739	Inf	Inf	704
tc0c10s2ct1	17.30	17.30	22	17.30	17.30	47	17.30	17.30	35
tc0c10s3cf1	25.50	25.50	162	25.50	25.50	148	25.50	25.50	94
tc0c10s3ct1	15.80	15.80	5	15.80	15.80	4	15.80	15.80	5
tc1c10s2cf2	14.03	14.03	2	14.03	14.03	2	14.03	14.03	2
tc1c10s2cf3	-	-	10800	-	-	10800	-	-	10800
tc1c10s2cf4	21.14	21.14	11	21.14	21.14	4	21.14	21.14	4
tc1c10s2ct2	15.75	15.75	281	15.75	15.75	46	15.75	15.75	58
tc1c10s2ct3	-	-	10800	-	-	10800	-	-	10800
tc1c10s2ct4	18.83	18.83	5	18.83	18.83	2	18.83	18.83	2
tc1c10s3cf2	14.03	14.03	3	14.03	14.03	4	14.03	14.03	2
tc1c10s3cf3	21.94	21.94	157	21.94	21.94	238	21.94	21.94	155
tc1c10s3cf4	19.90	19.90	19	19.90	19.90	28	19.90	19.90	28
tc1c10s3ct2	14.20	14.20	204	14.20	14.20	79	14.20	14.20	33
tc1c10s3ct3	18.02	18.02	49	18.02	18.02	88	18.02	18.02	28
tc1c10s3ct4	18.21	18.21	67	18.21	18.21	71	18.21	18.21	13
tc2c10s2cf0	-	-	10800	-	-	10800	-	-	10800
tc2c10s2ct0	18.84	18.84	1169	18.84	18.84	3053	18.84	18.84	872
tc2c10s3cf0	-	-	10800	-	-	10800	-	-	10800
tc2c10s3ct0	16.51	16.51	3168	16.51	16.51	5220	16.51	16.51	2600

Instance	N_C			N_M			N_R		
	Obj	Bound	Time	Obj	Bound	Time	Obj	Bound	Time
tc0c10s2cf1	Inf	Inf	3	Inf	Inf	2	Inf	Inf	2
tc0c10s2ct1	17.30	17.30	59	17.30	17.30	34	17.30	17.30	104
tc0c10s3cf1	25.50	25.50	14	25.50	25.50	13	25.50	25.50	68
tc0c10s3ct1	15.80	15.80	253	15.80	15.80	98	15.80	15.80	128
tc1c10s2cf2	14.03	14.03	125	14.03	14.03	121	14.03	14.03	108
tc1c10s2cf3	Inf	Inf	205	Inf	Inf	131	Inf	Inf	189
tc1c10s2cf4	21.14	21.14	15	21.14	21.14	9	21.14	21.14	14
tc1c10s2ct2	15.75	13.95	10800	15.75	14.66	10800	15.75	14.17	10800
tc1c10s2ct3	Inf	Inf	97	Inf	Inf	46	Inf	Inf	90
tc1c10s2ct4	18.83	18.83	9	18.83	18.83	10	18.83	18.83	7
tc1c10s3cf2	14.03	14.03	149	14.03	14.03	142	14.03	14.03	144
tc1c10s3cf3	21.94	21.94	4410	21.94	21.94	2907	21.94	21.94	239
tc1c10s3cf4	19.90	19.90	1206	19.90	19.90	92	19.90	19.90	125
tc1c10s3ct2	14.20	13.32	10800	14.20	13.5	10800	14.20	13.51	10800
tc1c10s3ct3	18.02	11.83	10800	18.02	12.18	10800	18.02	14.62	10800
tc1c10s3ct4	18.21	18.21	572	18.21	18.21	293	18.21	18.21	214
tc2c10s2cf0	Inf	Inf	296	Inf	Inf	327	Inf	Inf	31
tc2c10s2ct0	18.84	18.84	2264	18.84	18.84	1079	18.84	18.84	2276
tc2c10s3cf0	Inf	Inf	7753	Inf	Inf	281	Inf	Inf	2080
tc2c10s3ct0	16.51	10.16	10800	16.51	10.37	10800	16.51	10.34	10800

Table D.8: Detailed computational results on the 10-customer instances for the CS replication-based formulations ($\beta = 3$)

Instance	A_C			A_M			A_R		
	Obj	Bound	Time	Obj	Bound	Time	Obj	Bound	Time
tc0c10s2cf1	24.75	24.75	89	24.75	24.75	83	24.75	24.75	124
tc0c10s2ct1	17.30	17.30	94	17.30	17.30	60	17.30	17.30	80
tc0c10s3cf1	24.75	24.75	10026	24.75	24.75	4279	24.75	24.75	1751
tc0c10s3ct1	15.80	15.80	23	15.80	15.80	12	15.80	15.80	17
tc1c10s2cf2	14.03	14.03	6	14.03	14.03	6	14.03	14.03	15
tc1c10s2cf3	21.37	21.37	29	21.37	21.37	24	21.37	21.37	29
tc1c10s2cf4	21.10	21.10	15	21.10	21.10	9	21.10	21.10	28
tc1c10s2ct2	15.75	15.75	202	15.75	15.75	84	15.75	15.75	96
tc1c10s2ct3	18.17	18.17	26	18.17	18.17	43	18.17	18.17	27
tc1c10s2ct4	18.83	18.83	13	18.83	18.83	11	18.83	18.83	6
tc1c10s3cf2	14.03	14.03	7	14.03	14.03	5	14.03	14.03	26
tc1c10s3cf3	21.37	21.37	803	21.37	21.37	724	21.37	21.37	1715
tc1c10s3cf4	19.90	19.90	90	19.90	19.90	123	19.90	19.90	128
tc1c10s3ct2	14.20	14.20	172	14.20	14.20	88	14.20	14.20	203
tc1c10s3ct3	18.02	18.02	698	18.02	18.02	259	18.02	18.02	165
tc1c10s3ct4	18.21	18.21	142	18.21	18.21	59	18.21	18.21	32
tc2c10s2cf0	27.12	27.12	2099	27.12	27.12	3402	27.12	27.12	6352
tc2c10s2ct0	17.45	17.45	2162	17.45	17.45	1289	17.45	17.45	1385
tc2c10s3cf0	27.12	27.12	9953	27.12	23.66	10800	27.12	22.92	10800
tc2c10s3ct0	16.51	16.51	3567	16.51	15.21	10800	16.51	16.51	6124

Instance	N_C			N_M			N_R		
	Obj	Bound	Time	Obj	Bound	Time	Obj	Bound	Time
tc0c10s2cf1	24.75	24.75	15	24.75	24.75	13	24.75	24.75	14
tc0c10s2ct1	17.30	17.30	136	17.30	17.30	119	17.30	17.30	236
tc0c10s3cf1	24.75	24.75	3264	24.75	24.75	337	24.75	24.75	821
tc0c10s3ct1	15.80	15.80	156	15.80	15.80	66	15.80	15.80	317
tc1c10s2cf2	14.03	14.03	428	14.03	14.03	433	14.03	14.03	654
tc1c10s2cf3	21.37	21.37	8238	21.37	21.37	6249	21.37	21.37	6625
tc1c10s2cf4	21.10	21.10	143	21.10	21.10	79	21.10	21.10	86
tc1c10s2ct2	15.75	13.53	10800	15.75	13.73	10800	15.75	13.57	10800
tc1c10s2ct3	18.17	18.17	769	18.17	18.17	825	18.17	14.37	10800
tc1c10s2ct4	18.83	18.83	44	18.83	18.83	33	18.83	18.83	26
tc1c10s3cf2	14.03	14.03	977	14.03	14.03	451	14.03	14.03	605
tc1c10s3cf3	21.37	17.41	10800	21.37	17.62	10800	21.37	17.06	10800
tc1c10s3cf4	19.90	19.90	5653	19.90	19.90	652	19.90	19.90	855
tc1c10s3ct2	14.20	12.97	10800	14.20	13.18	10800	14.20	13.13	10800
tc1c10s3ct3	18.02	10.93	10800	18.12	10.78	10800	18.12	10.35	10800
tc1c10s3ct4	18.21	18.21	4085	18.21	18.21	1436	18.21	18.21	1674
tc2c10s2cf0	27.12	21.77	10800	27.12	21.16	10800	27.12	27.12	4978
tc2c10s2ct0	17.45	11.31	10800	17.45	11.40	10800	17.45	12.40	10800
tc2c10s3cf0	27.12	17.54	10800	27.12	19.11	10800	27.12	19.61	10800
tc2c10s3ct0	16.51	10.08	10800	16.51	9.88	10800	16.55	10.05	10800

Table D.9: Detailed computational results on the 10-customer instances for the CS replication-based formulations ($\beta = 4$)

Instance	A_C			A_M			A_R		
	Obj	Bound	Time	Obj	Bound	Time	Obj	Bound	Time
tc0c10s2cf1	24.75	24.75	150	24.75	24.75	206	24.75	24.75	131
tc0c10s2ct1	17.30	17.30	71	17.30	17.30	100	17.30	17.30	255
tc0c10s3cf1	24.75	21.88	10800	24.75	22.31	10800	24.75	22.27	10800
tc0c10s3ct1	15.80	15.80	29	15.80	15.80	23	15.80	15.80	44
tc1c10s2cf2	14.03	14.03	8	14.03	14.03	6	14.03	14.03	9
tc1c10s2cf3	21.37	21.37	38	21.37	21.37	81	21.37	21.37	129
tc1c10s2cf4	21.10	21.10	55	21.10	21.10	17	21.10	21.10	65
tc1c10s2ct2	15.75	15.75	267	15.75	15.75	125	15.75	15.75	394
tc1c10s2ct3	18.17	18.17	25	18.17	18.17	32	18.17	18.17	54
tc1c10s2ct4	18.83	18.83	28	18.83	18.83	20	18.83	18.83	15
tc1c10s3cf2	14.03	14.03	28	14.03	14.03	15	14.03	14.03	32
tc1c10s3cf3	21.37	21.37	5334	21.37	21.37	3684	21.37	20.91	10800
tc1c10s3cf4	19.90	19.90	111	19.90	19.90	82	19.90	19.90	265
tc1c10s3ct2	14.20	14.20	277	14.20	14.20	563	14.20	14.20	335
tc1c10s3ct3	18.02	18.02	473	18.02	18.02	1466	18.02	18.02	551
tc1c10s3ct4	18.21	18.21	142	18.21	18.21	116	18.21	18.21	41
tc2c10s2cf0	26.83	24.15	10800	26.77	23.80	10800	26.77	22.84	10800
tc2c10s2ct0	17.45	17.45	2147	17.45	17.45	2130	17.45	17.45	4069
tc2c10s3cf0	26.77	23.40	10800	26.77	21.85	10800	26.77	22.04	10800
tc2c10s3ct0	16.51	15.09	10800	16.51	13.96	10800	16.51	16.51	5552

Instance	N_C			N_M			N_R		
	Obj	Bound	Time	Obj	Bound	Time	Obj	Bound	Time
tc0c10s2cf1	24.75	24.75	40	24.75	24.75	66	24.75	24.75	90
tc0c10s2ct1	17.30	17.30	420	17.30	17.30	204	17.30	17.30	334
tc0c10s3cf1	24.75	18.97	10800	24.75	21.19	10800	24.75	21.15	10800
tc0c10s3ct1	15.80	15.80	541	15.80	15.8	59	15.80	15.80	105
tc1c10s2cf2	14.03	14.03	719	14.03	14.03	508	14.03	14.03	758
tc1c10s2cf3	21.37	18.67	10800	21.37	19.10	10800	21.37	18.58	10800
tc1c10s2cf4	21.10	21.10	168	21.10	21.10	147	21.10	21.10	337
tc1c10s2ct2	15.75	13.40	10800	15.75	13.50	10800	15.75	13.34	10800
tc1c10s2ct3	18.19	15.07	10800	18.17	18.17	2169	18.17	18.17	4761
tc1c10s2ct4	18.83	18.83	104	18.83	18.83	90	18.83	18.83	186
tc1c10s3cf2	14.03	14.03	1868	14.03	14.03	1099	14.03	14.03	1358
tc1c10s3cf3	21.37	16.83	10800	21.37	17.13	10800	21.37	16.74	10800
tc1c10s3cf4	19.90	19.90	3845	19.90	19.90	954	19.90	19.90	3629
tc1c10s3ct2	14.20	12.73	10800	14.20	12.85	10800	14.20	12.76	10800
tc1c10s3ct3	18.08	10.32	10800	18.08	10.49	10800	18.08	10.34	10800
tc1c10s3ct4	18.21	16.80	10800	18.21	18.21	8104	18.21	18.21	9210
tc2c10s2cf0	26.77	19.65	10800	26.77	18.15	10800	26.77	19.10	10800
tc2c10s2ct0	17.45	11.33	10800	17.45	10.69	10800	17.45	10.35	10800
tc2c10s3cf0	26.83	16.20	10800	26.77	17.56	10800	26.83	18.24	10800
tc2c10s3ct0	16.68	9.93	10800	16.54	9.99	10800	16.51	9.94	10800

Table D.10: Detailed computational results on the 10-customer instances for the recharging path-based formulations

Instance	Path_A_C		Path_A_R	
	Obj	Time	Obj	Time
tc0c10s2cf1	24.75	15	24.75	15
tc0c10s2ct1	17.30	20	17.30	20
tc0c10s3cf1	24.75	22	24.75	8
tc0c10s3ct1	15.80	15	15.80	8
tc1c10s2cf2	14.03	12	14.03	4
tc1c10s2cf3	21.37	108	21.37	32
tc1c10s2cf4	21.10	14	21.10	9
tc1c10s2ct2	15.75	202	15.75	184
tc1c10s2ct3	18.17	13	18.17	12
tc1c10s2ct4	18.83	6	18.83	4
tc1c10s3cf2	14.03	13	14.03	4
tc1c10s3cf3	21.37	169	21.37	43
tc1c10s3cf4	19.90	37	19.90	16
tc1c10s3ct2	14.20	117	14.20	129
tc1c10s3ct3	18.02	86	18.02	29
tc1c10s3ct4	18.21	14	18.21	13
tc2c10s2cf0	26.77	1715	26.77	894
tc2c10s2ct0	17.45	1062	17.45	1993
tc2c10s3cf0	26.77	1708	26.77	909
tc2c10s3ct0	16.51	2953	16.51	4459