# Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions

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

# Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions

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

[1] Université de Tours, LIFAT (EA 6300), ROOT (ERL CNRS 7002), Tours, France
[2] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy
[3] Canada Research Chair in Distribution Management, HEC Montréal, Montréal, Canada H3T 2A7

**Abstract**

Electric vehicle routing problems (E-VRPs) are receiving growing attention from the operations research community. Electric vehicles differ substantially from internal combustion engine vehicles, the main difference lying in their limited autonomy, which can be recovered at charging stations. Modeling the charging functions is a focal point of E-VRPs. Most of the research has focused on constant or linear charging functions. The E-VRP with nonlinear charging function (E-VRP-NL) was recently introduced to account for the more realistic nonlinear relationship between the time spent charging and the amount of energy charged. We propose two new formulations for this problem. We first develop an arc-based tracking of the time and the state of charge which, according to our experiments, outperforms the classical node-based tracking of these values. To avoid replicating the charging stations nodes, as done for both node and arc based formulations, we also introduce a path-based model. We develop an algorithm to generate a tractable number of these paths. This path-based model outperforms the classical models in our experiments. We also propose a new model, a heuristic, and an exact labeling algorithm for the problem of finding the optimal charging decisions for a given route. Extensive computational results show that charging decisions considerably impact the quality of the E-VRP-NL solutions. Indeed, we improve 23 out of 120 best known E-VRP-NL solutions by solely revising the charging decisions.

# 1 Introduction

Electric vehicles (EVs) offer a viable alternative to internal combustion engine vehicles (ICEVs), and their market share is steadily increasing, partly as a result of several governmental and private policies. In parallel, the study of problems rising from using EVs by operational researchers has gained importance in recent years, and has given a rise to a new class of problems called electric vehicle routing problems (E-VRPs). These problems differ from the classical vehicle routing problem (VRP) in several ways since the EV technological constraints need to be accounted for. Indeed, the driving autonomy of EVs is limited by their battery capacity, which can be recovered at charging stations (CSs). These are much scarcer than conventional refueling stations for ICEVs, which means that EVs often may need to perform detours to reach a CS. The latter is especially true in the context of mid-haul or long-haul routing (Montoya, 2016; Schiffer, 2017). In addition, charging the battery of an EV takes much longer than refilling the tank of an ICEV. Therefore, E-VRPs involve additional decisions such as where and how long to charge.

The charging infrastructure, which includes the CSs used to recuperate energy of EVs, can be either public or private. The use of the former by private companies may result in uncertainty with respect to CS availability. Therefore, the use of public infrastructure to charge EVs in a routing context has received little attention. We refer the reader to Sweda et al. (2017) and Kullman et al. (2018) for further details. Since the uncertainty with respect to CS availability is neglected in the majority of the E-VRP literature, we conclude that most of the research on E-VRPs implicitly assumes that the charging infrastructure is privately owned by the EV operator. We adopt this assumption in this paper as well.

Research on E-VRPs primarily started with the green vehicle routing problem (Green VRP) introduced by Erdoğan and Miller-Hooks (2012). In the Green VRP, a fleet of homogeneous alternative fuel vehicles must visit a set of customers from a single depot while traveling a minimum total distance, subject to a route duration constraint. The vehicles have a limited fuel tank capacity but can detour to CSs to extend their driving range. The tank is fully recharged in constant time at each visit to a CS, an assumption that is appropriate for the vehicles operating on alternative fuels such as biofuels and liquid nitrogen. With respect to EVs, such an assumption makes sense if batteries are replaced by fully loaded batteries at swapping stations. To our knowledge, however, battery swapping does not seem to be on the path of massive adoption. Indeed, the company Better Place that offered battery swapping services declared bankruptcy in 2013[1] and Tesla's only battery swapping station (part of a pilot program) shut down in 2016[2]. Several variants of the Green VRP have been studied. Thus 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 no longer assumed to be constant, it is linearly dependent on the state of charge (SoC) of the EV upon its arrival at the CS. Finally, no route duration limit is considered. In the remainder of this paper we use the term SoC to refer to the battery level in the EV.

The above problems assume 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 whenever a charging operation takes place is rather conservative. In view of the fact that the time needed to reach a fully charged battery may be substantial when charging operations occur en route, Felipe et al. (2014) introduced a partial recharging policy. Compared to a full charging policy, a partial charging policy allows savings in both energy consumption and time. The authors assumed a linear charging process and considered CSs with multiple technologies. Each technology has a distinct charging rate, which reflects the variety of chargers one can encounter in practice (e.g., slow, medium, and fast). They also considered capacitated vehicles and route duration constraints. Furthermore, Desaulniers et al. (2016) tackled a version of the E-VRPTW in which the

---

[1] https://en.wikipedia.org/wiki/Better_Place\_(company) - Retrieved 2018-04-10

[2] http://www.teslarati.com/tesla-shuts-down-battery-swap-program-for-superchargers/ - Retrieved 2018-04-10

fleet size is fixed and partial charging is allowed.

In reality, battery charging functions of EVs are not linear (see Pelletier et al. (2017) for further details). In order to model a more realistic charging process, Montoya et al. (2017) introduced the electric vehicle routing problem with nonlinear charging function (E-VRP-NL). In addition to considering a partial charging policy, the authors proposed a piecewise linear approximation of the charging process. They also showed that previously adopted linear charging functions yield infeasible routes with respect to route maximum-duration limit, when projected on the more realistic nonlinear charging functions. Moreover, even if the routes produced under linear charging functions are feasible, when subjected to nonlinear charging functions they tend to be of poor quality. The objective in the E-VRP-NL is to minimize the total time needed to serve all the customers, which is influenced by the charging decisions because it accounts for the time spent at CSs and for the nonlinear nature of the charging process. Using a temporal-based objective is all the more relevant since CSs may offer multiple technologies which impact the duration of the charging operations. To solve the E-VRP-NL, Montoya et al. (2017) developed a hybrid metaheuristic that combines an iterated local search heuristic with the solution of a set partitioning problem. Specifically, the first stage of the algorithm aims to build a high-quality and diverse pool of routes, while the second stage assembles solutions by selecting a subset of routes from the pool.

Research on the E-VRPs is now moving towards more accurate representations of EV characteristics which pose new modeling and algorithmic challenges. One of the most challenging issues when solving E-VRPs is making good charging decisions. In general, charging operations influence feasibility and in many cases they also increase costs, due to detours to CSs and the time spent at those CSs. Depending on the underlying E-VRP, deciding where to charge and how much to charge can be very complex. To better deal with these decisions, some authors have studied the fixed route vehicle charging problem (FRVCP). Given a fixed sequence of customers, the FRCVP consists in optimally inserting charging operations within it. As such, the FRVCP can be viewed as a subproblem of E-VRPs, which is pivotal to many heuristics that must compute the cost of a given route. Montoya et al. (2016) solved a FRVCP with the full charging policy and constant charging time. They formulated the problem as a constrained shortest path problem and solved it using the Pulse algorithm (Lozano and Medaglia, 2013). Roberti and Wen (2016) described a labeling algorithm to solve the FRVCP arising in the case where all the CSs have the same charging technology, the charging function is linear, and partial charging is allowed. Hiermann et al. (2016), Schiffer and Walther (2017), and Hiermann et al. (2018) considered the FRVCP as a sub-problem of the E-VRP-TW with single charging technology and linear charging function. They tackled a special version of the problem where at most one CS can be inserted between any pair of non-CS nodes. Note that Hiermann et al. (2016) also assumed a full charging policy. Finally, Montoya et al. (2017) considered the FRVCP as a subproblem of the E-VRP-NL. They presented a mixed integer linear programming (MILP) model in which they assume that an EV can visit at most one CS between any pair of non-CS nodes and a heuristic.

The scientific contribution of this paper is twofold. First, we propose two new mixed integer linear programming (MILP) models for the E-VRP-NL, which is central to the E-VRP literature in that it realistically models the EV charging operations via piecewise linear approximations of nonlinear functions, while accounting for various charging technologies and allowing partial recharge. The first model results from replacing the node-based tracking of the time and SoC from the Montoya et al. (2017) model by an arc-based tracking of these quantities. We demonstrate that this new formulation provides an equal or better LP relaxation bound. To avoid replicating the CS nodes, as was the case in both previously discussed models, we then define the problem on a multigraph, which allows us to derive a second model for the E-VRP-NL. We explain why this formulation better fits the problem from a modeling perspective, and we also demonstrate its superiority through extensive computational experiments. Building on the second new E-VRP-NL model, the second scientific contribution of this paper lies in the development of three new algorithms for the FRVCP: a direct approach that solves a MILP formulation of the problem, a heuristic algorithm, and an exact labeling algorithm. These

3

three algorithms are compared through an extensive set of computational experiments.

The remainder of this paper is organized as follows. We formally describe the E-VRP-NL in Section 2. In Section 3, we present a new MILP model for the problem based on its "classical" definition on a simple digraph. In Section 4, we introduce an alternative MILP model for the E-VRP-NL using the concept of paths visiting sequences of CSs. In Section 5, we describe a new MILP model, a heurisitic and an exact labeling algorithm for the FRVCP. We present computational results in Section 6, followed by conclusions in Section 7.

# 2 The electric vehicle routing problem with nonlinear charging function

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 and let $F$ be the set of CSs at which the vehicles can fully or partially recharge their battery. Each customer $i \in I$ has a service time $g_i$. The customers are served by an homogeneous fleet of EVs, which is unlimited. Each EV has a battery of capacity $Q$ (expressed in kWh) and a maximum tour duration $T_{max}$. At the start of the planning horizon, the EVs are located at a single depot (denoted by node 0), from which they leave fully charged. Traveling from a location $i$ (the depot, a customer, or a CS) to another location $j$ incurs a driving time $t_{ij} \geqslant 0$ and an energy consumption $e_{ij} \geqslant 0$. The triangular inequality holds for both the driving time and the energy consumption. Each CS $i$ has a charging technology (e.g., slow, moderate, fast) associated with a piecewise linear concave charging function $\Phi_i(\Delta)$ that maps, for an empty battery, the time $\Delta$ spent charging at $i$ to the SoC of the vehicle when it leaves $i$. If $q$ is the SoC of an EV when arriving at $i$, then the SoC of the EV upon departing from $i$ after charging during a time equal to $\Delta$ is given by $\Phi_i(\Delta + \Phi_i^{-1}(q))$. We denote by $B_i = \{0, \ldots, b_i\}$ the ordered set of breakpoints of the piecewise linear approximation of the charging curve of CS $i$. Let $c_{ik}$ and $a_{ik}$ be the charging time and the SoC of breakpoint $k \in B_i$ of the CS $i$. Let $\rho_{ik}$ be the slope of the segment joining the points $(c_{i,k-1}, a_{i,k-1})$ and $(c_{ik}, a_{ik})$ (i.e., $\rho_{ik} = (a_{ik} - a_{i,k-1})/(c_{ik} - c_{i,k-1})$. Figure 1 shows the piecewise linear approximation for a CS $i$ yielding a power of $22\,\text{kW}$ charging a vehicle equipped with a $16\,\text{kWh}$ battery.
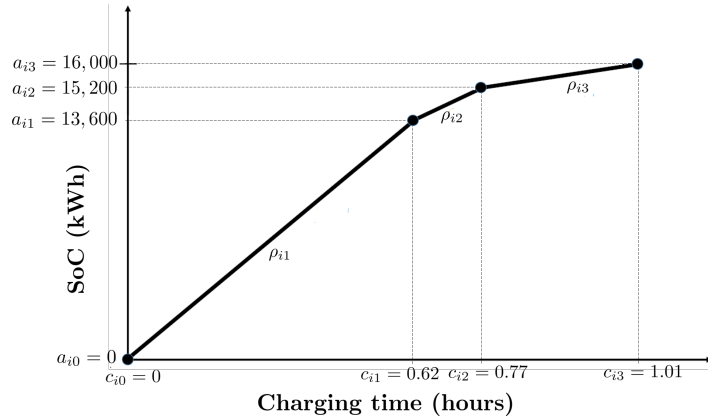


Figure 1: Piecewise linear approximation for a CS yielding a power of $22\,\text{kW}$ charging a $16\,\text{kWh}$ battery. Adapted from Montoya et al. (2017).

Feasible solutions to the E-VRP-NL satisfy the following conditions: 1) each customer is served exactly once by a single vehicle, 2) each route starts and ends at the depot, 3) each route satisfies the maximum-duration limit $T_{max}$, and 4) each route is energy-feasible (i.e., the battery level of an EV upon arriving at a location or departing from it lies between 0 and $Q$). The objective of the E-VRP-NL is to minimize the total time needed to serve all customers, including driving, service, and charging time.

# 3 Two CS replication-based models for the E-VRP-NL

The E-VRP-NL is defined on a 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 $V$. Node 0 represents the single depot. The set $F'$ contains $\beta_i$ copies of each CS $i \in F$ (i.e., $|F'| = \sum_{i \in F} \beta_i$). The value of $\beta_i$ corresponds to an upper bound on the number of times $i$ 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 the CS, i.e., each node in $F'$ can be restricted to be visited at most once. Therefore, the use of CS copies facilitates the tracking of the SoC and of the time. We denote by $F_i' \subseteq F'$ the set containing the $\beta_i$ 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 $\beta_i$. In the remainder of this paper, depending on the context, we refer to an element of $F'$ or $F_i'$ as a CS copy. Furthermore, the graph $G$ is not necessary complete, since some arcs cannot be traversed due to the SoC constraints. Therefore, we apply a preprocessing procedure to reduce its size (see A).

For the sake of completeness and clarity, we recall in §3.1 the model introduced in Montoya et al. (2017). In §3.2 we introduce an alternative formulation, which uses a different approach to model the time and SoC tracking.

## 3.1 First model for the E-VRP-NL

The formulation of Montoya et al. (2017) involves the following decision variables. The binary variable $x_{ij}$ is equal to 1 if and only if an EV travels on arc $(i, j) \in A$. The continuous variables $\tau_j$ and $y_j$ track the time and SoC of the EV when it departs from node $j \in V$. The continuous variables $\underline{q}_i$ and $\overline{q}_i$ specify (according to the piecewise linear approximation of the charging function of $i$) the SoC of an EV when it enters and leaves CS copy $i \in F'$. The variables $\underline{c}_i$ and $\overline{c}_i$ are the scaled start and end times for charging an EV according to the charging function of CS copy $i$. The continuous variable $\Delta_i = \overline{c}_i - \underline{c}_i$ represents the duration of the charging operation performed at CS copy $i$. The binary variables $\underline{w}_{ik}$ and $\overline{w}_{ik}$ are equal to 1 if and only if the SoC is between $a_{i,k-1}$ and $a_{ik}$, with $k \in B_i \backslash \{0\}$, when the EV enters and leaves CS copy $i$, respectively. Finally, the continuous variables $\underline{\lambda}_{ik}$ and $\overline{\lambda}_{ik}$ are the coefficients associated with the breakpoint $(c_{ik}, a_{ik})$ in the piecewise linear approximation, when the EV enters and leaves CS copy $i$, respectively. Specifically, the $\underline{\lambda}_{ik}$ variables enable the expression of $(\underline{c}_i, \underline{q}_i)$ as a convex combination of the breakpoints $(c_{ik}, a_{ik})$, where $k \in B_i$. Similarly, the $\overline{\lambda}_{ik}$ variables enable the expression of $(\overline{c}_i, \overline{q}_i)$ as a convex combination of the breakpoints $(c_{ik}, a_{ik})$, where $k \in B_i$. The Montoya et al. (2017) model for the E-VRP-NL, denoted as $[F_{Node}^{CS\_rep}]$, is as follows:

$$[F_{Node}^{CS\_rep}] \qquad \text{minimize} \sum_{(i,j) \in A} t_{ij} x_{ij} + \sum_{i \in F'} \Delta_i \qquad (1)$$

subject to

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

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

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

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

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

$$y_i \geqslant e_{i0} x_{i0} \qquad\qquad (i, 0) \in A \qquad (7)$$

$$y_i = \overline{q}_i \qquad\qquad i \in F' \qquad (8)$$

5

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

$$\underline{q}_i \leqslant \overline{q}_i \qquad\qquad i \in F' \tag{10}$$

$$\underline{q}_i = \sum_{k \in B_i} \underline{\lambda}_{ik} a_{ik} \qquad\qquad i \in F' \tag{11}$$

$$\underline{c}_i = \sum_{k \in B_i} \underline{\lambda}_{ik} c_{ik} \qquad\qquad i \in F' \tag{12}$$

$$\sum_{k \in B_i} \underline{\lambda}_{ik} = \sum_{k \in B_i \backslash \{0\}} \underline{w}_{ik} \qquad\qquad i \in F' \tag{13}$$

$$\sum_{k \in B_i \backslash \{0\}} \underline{w}_{ik} = \sum_{(i,j) \in A} x_{ij} \qquad\qquad i \in F' \tag{14}$$

$$\underline{\lambda}_{i0} \leqslant \underline{w}_{i1} \qquad\qquad i \in F' \tag{15}$$

$$\underline{\lambda}_{ik} \leqslant \underline{w}_{ik} + \underline{w}_{i,k+1} \qquad\qquad i \in F', k \in B_i \backslash \{0, b_i\} \tag{16}$$

$$\underline{\lambda}_{ib_i} \leqslant \underline{w}_{ib_i} \qquad\qquad i \in F' \tag{17}$$

$$\overline{q}_i = \sum_{k \in B_i} \overline{\lambda}_{ik} a_{ik} \qquad\qquad i \in F' \tag{18}$$

$$\overline{c}_i = \sum_{k \in B_i} \overline{\lambda}_{ik} c_{ik} \qquad\qquad i \in F' \tag{19}$$

$$\sum_{k \in B_i} \overline{\lambda}_{ik} = \sum_{k \in B_i \backslash \{0\}} \overline{w}_{ik} \qquad\qquad i \in F' \tag{20}$$

$$\sum_{k \in B_i \backslash \{0\}} \overline{w}_{ik} = \sum_{(i,j) \in A} x_{ij} \qquad\qquad i \in F' \tag{21}$$

$$\overline{\lambda}_{i0} \leqslant \overline{w}_{i1} \qquad\qquad i \in F' \tag{22}$$

$$\overline{\lambda}_{ik} \leqslant \overline{w}_{ik} + \overline{w}_{i,k+1} \qquad\qquad i \in F,' k \in B_i \backslash \{0, b_i\} \tag{23}$$

$$\overline{\lambda}_{ib_i} \leqslant \overline{w}_{ib_i} \qquad\qquad i \in F' \tag{24}$$

$$\Delta_i = \overline{c}_i - \underline{c}_i \qquad\qquad i \in F' \tag{25}$$

$$\tau_i + (t_{ij} + g_j)x_{ij} - T_{max}(1 - x_{ij}) \leqslant \tau_j \qquad\qquad (i,j) \in A : i \in V, j \in I \tag{26}$$

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

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

$$\tau_0 = 0 \tag{29}$$

$$\tau_l - \Delta_l \geqslant \tau_j - \Delta_j \qquad\qquad i \in F, j, l \in F_i', j < l \tag{30}$$

$$\sum_{(h,j) \in A} x_{hj} \geqslant \sum_{(h,l) \in A} x_{hl} \qquad\qquad i \in F, j, l \in F_i', j < l \tag{31}$$

$$x_{ij} \in \{0, 1\} \qquad\qquad (i,j) \in A \tag{32}$$

$$\tau_i \geqslant 0, y_i \geqslant 0 \qquad\qquad i \in V \tag{33}$$

$$\underline{w}_{ik} \in \{0, 1\}, \overline{w}_{ik} \in \{0, 1\} \qquad\qquad i \in F', k \in B_i \backslash \{0\} \tag{34}$$

$$\underline{\lambda}_{ik} \geqslant 0, \overline{\lambda}_{ik} \geqslant 0 \qquad\qquad i \in F', k \in B_i \tag{35}$$

$$\underline{q}_i \geqslant 0, \overline{q}_i \geqslant 0, \Delta_i \geqslant 0, \underline{c}_i \geqslant 0, \overline{c}_i \geqslant 0 \qquad\qquad i \in F'. \tag{36}$$

The objective function (1) minimizes the total driving and charging time. Constraints (2) ensure that each customer is served once, while constraints (3) state that every CS copy is visited at most once. Constraints (4) impose the flow conservation conditions. Constraints (5) and (6) track the SoC of an EV at each node. Constraints (7) ensure that if an EV travels between a customer or CS copy and the depot, then it has sufficient energy to reach its destination. Constraints (8) reset the battery tracking to $\overline{q}_i$ upon departure from CS copy $i$. Constraint (9) states that the SoC of an EV leaving the depot is $Q$. Constraints (10) couple the SoC of an

EV upon arrival at a CS with its SoC at departure. Constraints (11)–(17) define the SoC and the charging time of an EV upon arrival at CS copy $i$, based on the piecewise linear approximation of its charging function. Similarly, constraints (18)–(24) define the SoC and charging time of an EV upon departure from CS copy $i$. Constraints (25) define the time spent at any CS copy. Constraints (26) and (27) track the departure time from each node. The parameter $S_{max} = \max_{i \in F}\{c_{ib_i}\}$ corresponds to the maximum charging time, i.e., the time to fully charge an empty battery at the slowest CS. Constraints (28) and (29) ensure that the EVs return to the depot no later than $T_{max}$. Even if they are not necessary, constraints (30) and (31) avoid potential symmetries between the CS copies. These constraints ensure that the copies of CS $i$ are visited in the order in which 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)–(36) define the domains of the decision variables.

In order to strengthen the previous formulation, we add the following valid inequalities:

$$y_i \geqslant \min_{l \in F \cup \{0\}}\{e_{il}\} \qquad\qquad i \in V\backslash\{0\}. \qquad (37)$$

These inequalities state that the SoC of an EV upon departing from a node must be sufficient to reach the depot or the nearest CS.

## 3.2   Second model for the E-VRP-NL

We now introduce a modeling alternative for the E-VRP-NL. Formulation $[F_{Node}^{CS\_rep}]$ tracks the time and the SoC of the EVs with node-indexed variables. We therefore refer to constraints (5)–(9), (26)–(29), (31), (33), and (37) as *node-based tracking constraints.* Here we will model the time and SoC tracking by introducing arc-based variables. More precisely, we replace variables $\tau_j$ and $y_j$ with the continuous variables $\tau_{ij}$ and $y_{ij}$, which respectively represent the time and SoC of an EV when it departs from node $i \in V$ to travel on arc $(i, j) \in A$. If no vehicle travels on arc $(i, j)$, then both variables are 0. We model the time and SoC tracking through the following *arc-based tracking constraints* which replace the node-based tracking constraints in $[F_{Node}^{CS\_rep}]$. The second model for the E-VRP-NL, denoted as $[F_{Arc}^{CS\_rep}]$, is as follows:

$$[F_{Arc}^{CS\_rep}] \qquad\qquad \text{minimize} \sum_{(i,j) \in A} t_{ij} x_{ij} + \sum_{i \in F'} \Delta_i$$

subject to

$$(2) - (4), (10) - (25), (31), (32), (34) - (36)$$

$$y_{0j} = Q x_{0j} \qquad\qquad (0, j) \in A \qquad (38)$$

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

$$\sum_{(i,j) \in A} y_{ij} - \sum_{(i,j) \in A} e_{ij} x_{ij} = \underline{q}_j \qquad\qquad j \in F' \qquad (40)$$

$$\sum_{(j,l) \in A} y_{jl} = \overline{q}_j \qquad\qquad j \in F' \qquad (41)$$

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

$$\sum_{(i,j) \in A} (\tau_{ij} + (t_{ij} + g_j) x_{ij}) = \sum_{(j,l) \in A} \tau_{jl} \qquad\qquad j \in I \qquad (43)$$

$$\sum_{(i,j) \in A} (\tau_{ij} + t_{ij} x_{ij}) + \Delta_j = \sum_{(j,l) \in A} \tau_{jl} \qquad\qquad j \in F' \qquad (44)$$

$$\tau_{ij} \leqslant (T_{max} - t_{ij} - g_j - t_{j0}) x_{ij} \qquad\qquad (i, j) \in A : j \in I \qquad (45)$$

$$\tau_{ij} \leqslant (T_{max} - t_{ij} - \Delta_j^{min} - t_{j0}) x_{ij} \qquad\qquad (i, j) \in A : j \in F' \qquad (46)$$

$$\sum_{(j,h)\in A} \tau_{jh} - \Delta_j \geqslant \sum_{(l,h)\in A} \tau_{lh} - \Delta_l \qquad\qquad i \in F, j,l \in F'_i : j < l \qquad (47)$$

$$\tau_{ij} \geqslant 0, y_{ij} \geqslant 0 \qquad\qquad (i,j) \in A. \qquad (48)$$

Constraints (38) state that the EVs leave the depot fully charged. Constraints (39) track the SoC of an EV at each customer. Similarly, constraints (40) track the SoC of an EV when it arrives at a CS copy, and 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) and (44) track the departure time from each customer and CS copies, respectively. Constraints (45) and (46) couple the $\tau_{ij}$ and $x_{ij}$ variables. Specifically, if an EV traverses an arc $(i,j)$, then its departure time must guarantee that it returns to the depot without exceeding the tour maximum-duration limit. The parameter $\Delta_j^{min}$ represents the minimum duration of a charging operation at $j \in F'$, and is equal to the time needed to charge the energy quantity $\min_{l,l'\in V\backslash\{j\}:(l,j)\in A \wedge (j,l')\in A}\{e_{lj} + e_{jl'} - e_{ll'}\}$ if the EV has an empty battery. It is a lower bound on the time spent charging to recover the energy consumed to make the detour to $j$. Constraints (47) break symmetries created by the introduction of CS copies. Constraints (48) define the domains of the newly introduced decision variables.

In order to strengthen the previous formulation, we add the following valid inequalities:

$$y_{ij} \geqslant \left(e_{ij} + \min_{l\in F\cup\{0\}}\{e_{jl}\}\right) x_{ij} \qquad\qquad (i,j) \in A : \ i \neq 0. \qquad (49)$$

Constraints (49) state that if an EV traverses arc $(i,j)$, then its SoC when leaving $i$ must be sufficient to traverse the arc and then reach the closest CS or depot.

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)–(48), than on those based on node-based tracking constraints when there are few time windows. Therefore, there is ground to believe that reformulating the time and SoC tracking in $[F_{Node}^{CS\_rep}]$ using this new model may yield a better computational performance. Proposition 3.1 supports this claim for the two previously presented formulations.

**Proposition 3.1.** *Let $P_{arc}$ denote the polyhedron associated with the linear relaxation of $[F_{Arc}^{CS\_rep}]$, and let Let $P_{node}$ denote the polyhedron associated with the linear relaxation of $[F_{Node}^{CS\_rep}]$. Then $P_{arc} \subseteq P_{node}$.*

*Proof.* See B. ☐

## 3.3 Valid inequalities for the first two models

We introduce a new set of valid inequalities to strengthen the previous $[F_{Node}^{CS\_rep}]$ and $[F_{Arc}^{CS\_rep}]$ formulations. Let $ec_l^{min}$ be the minimum amount of energy that an EV must charge at CS copy $l$. This corresponds to the charge needed to recover the minimum energy consumed to make a detour to $l$. Specifically, $ec_l^{min} = \min_{i,j\in V\backslash\{l\}:(i,l)\in A \wedge (l,j)\in A}\{e_{il} + e_{lj} - e_{ij}\}$. Assuming an empty battery upon arrival at the CS, we can therefore state the following valid constraints:

$$\Delta_i \geqslant \Phi_i(ec_i^{min}) \sum_{(i,j)\in A} x_{ij} \qquad\qquad i \in F'. \qquad (50)$$

# 4 Path-based model without CS copies for the E-VRP-NL

A major drawback of the two previous formulations defined on a digraph is the need to replicate CS nodes. Indeed, to ensure that no optimal solutions are cut off, the number of copies may have to be very large, which affects the solution time. We can find examples where this value is equal to $4|I|$ (see C). However, working with an order of $|I|$ copies for each CS is impractical. Despite this limitation, the vast majority of MILP models

for E-VRPs listed in the literature are based on the CS-replication paradigm. To our knowledge, only Montoya et al. (2017) proposed a procedure to set the number of CS copies needed to be used in their model. They used the same number $\beta$ of CS copies for every CS. Starting with $\beta = 0$, they solved the MILP formulation to optimality with a time limit of $\bar{T}$ (set to 100 hours in their experiments); $\beta$ was then incremented and the resulting MILP was solved. If $\beta \geqslant 1$ and either the time limit was reached or the optimal solution obtained with $\beta = l$ had the same value as the one obtained with $\beta = l - 1$, they stopped the procedure and set $\beta = l - 1$. This procedure is not optimal when the time limit is reached. Moreover, even when an instance is solved to optimality, this procedure is not optimal (see D for examples).

We propose an alternative model that does not use CS copies. This new model is based on enumerating the sequence of CS nodes that can be visited by an EV between each pair of customer or depot nodes. Given two nodes (customer or depot) $i$ and $j$, we call a *CS path* (CSP) a simple path starting from $i$ visiting a sequence of CSs (possibly empty) and ending at $j$.

CSPs were introduced by Andelmin (2014) as "e-paths" in the context of the E-VRPTW with partial recharging. They were successfully used by Bartolini and Andelmin (2017) (as "refuel paths") and constituted the main building block of an exact algorithm for the Green VRP. They were also applied by Roberti and Wen (2016) (as "recharging paths") to the electric traveling salesman problem with time windows. Koç and Karaoglan (2016) and Leggieri and Haourari (2017) employed a similar concept. In the two latter studies, however, only one CS can be visited between any pair of nodes. This assumption may cut off optimal solutions to the Green VRP (see Bartolini and Andelmin (2017) for an example). Because in the Green VRP the vehicles are fully charged in constant time upon visiting a CS, the SoC of an EV after traversing a refuel path and the time spent traveling on a path can then be computed a priori. Allowing partial charging combined with the non-linearity of the charging functions and/or the consideration of multiple technologies (any one of the two latter is a sufficient condition), like in the E-VRP-NL, necessitates deciding how much to charge at each CS and thus tracking the time and SoC throughout the CSPs.

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

## 4.1 Third model for the E-VRP-NL

Our path-based formulation of the E-VRP-NL involves the following decisions variables. The binary variable $x_p$ is equal to 1 if and only if an EV travels on CSP $p \in P$. The continuous variables $\tau_p$ and $y_p$ track the time and SoC of an EV when it departs from node $o(p)$ to node $d(p)$ on CSP $p$. If no vehicle travels on the CSP, then both variables take the value of 0. For a path $p$, the continuous variables $\underline{q}_{pl}$ and $\overline{q}_{pl}$ specify the SoC of an EV when it enters and leaves $\mu_p(l)$, respectively. We model the piecewise linear approximation of the charging function following the same principle as in the first two models. The variables $\underline{c}_{pl}$ and $\overline{c}_{pl}$ are the scaled start and end times for charging an EV according to the charging function of CS $\mu_p(l)$. The continuous variable $\Delta_{pl}$ represents the duration of the charging operation performed at $\mu_p(l)$. The binary variables $\underline{w}_{plk}$ and $\overline{w}_{plk}$ are equal to 1 if and only if the SoC is between $a_{\mu_p(l),k-1}$ and $a_{\mu_p(l),k}$, with $k \in B_i \backslash \{0\}$, when the EV enters and leaves CS $\mu_p(l)$, respectively. The continuous variables $\underline{\lambda}_{plk}$ and $\overline{\lambda}_{plk}$ are the coefficients associated with the breakpoint $(c_{\mu_p(l),k}, a_{\mu_p(l),k})$ in the piecewise linear approximation, when the EV enters and leaves CS $\mu_p(l)$, respectively. The path-based formulation of the E-VRP-NL, denoted as $[F_{Arc}^{Path}]$, is as follows:

$$[F_{Arc}^{Path}] \qquad\qquad \min \sum_{p \in P} \left( t^p x_p + \sum_{l \in L_p} \Delta_{pl} \right) \tag{51}$$

subject to

$$\sum_{j \in \tilde{V}, i \neq j} \sum_{p \in P_{ij}} x_p = 1 \qquad i \in I \tag{52}$$

$$\sum_{j \in \tilde{V}, i \neq j} \sum_{p \in P_{ji}} x_p - \sum_{j \in \tilde{V}, i \neq j} \sum_{p \in P_{ij}} x_p = 0 \qquad i \in \tilde{V} \tag{53}$$

$$\sum_{j \in \tilde{V}, j \neq i} \sum_{p \in P_{ji}} \left( y_p - e^p x_p + \sum_{l \in L_p} (\overline{q}_{pl} - \underline{q}_{pl}) \right) = \sum_{j \in \tilde{V}, j \neq i} \sum_{p \in P_{ij}} y_p \qquad i \in I \tag{54}$$

$$y_p - e_{o(p), \mu_p(0)} x_p = \underline{q}_{p0} \qquad p \in P : |L_p| \neq 0 \tag{55}$$

$$\overline{q}_{p, l-1} - e_{\mu_p(l-1), \mu_p(l)} x_p = \underline{q}_{pl} \qquad p \in P, l \in L_p \backslash \{0\} \tag{56}$$

$$y_p = Q x_p \qquad i \in \tilde{V} \backslash \{0\}, p \in P_{0i} \tag{57}$$

$$y_p - e^p x_p + \sum_{l \in L_p} (\overline{q}_{pl} - \underline{q}_{pl}) \geqslant 0 \qquad i \in \tilde{V} \backslash \{0\}, p \in P_{i0} \tag{58}$$

$$y_p \leqslant Q x_p \qquad p \in P \tag{59}$$

$$\sum_{j \in \tilde{V} \backslash \{0\}, j \neq i} \sum_{p \in P_{ji}} \tau_p + \sum_{j \in \tilde{V}, j \neq i} \sum_{p \in P_{ji}} \left( t^p x_p + \sum_{l \in L_p} \Delta_{pl} \right) + g_i = \sum_{j \in \tilde{V}, j \neq i} \sum_{p \in P_{ij}} \tau_p \qquad i \in I \tag{60}$$

$$\tau_p + \sum_{l \in L_p} \Delta_{pl} \leqslant \left( T_{max} - t^p - g_{d(p)} - t_{d(p),0} \right) x_p \qquad p \in P \tag{61}$$

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

$$\underline{c}_{pl} = \sum_{k \in B_{\mu_p(l)}} \underline{\lambda}_{plk} c_{\mu_p(l)k} \qquad p \in P, l \in L_p \tag{63}$$

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

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

$$\underline{\lambda}_{pl0} \leqslant \underline{w}_{pl1} \qquad p \in P, l \in L_p \tag{66}$$

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

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

$$\overline{q}_{pl} = \sum_{k \in B_{\mu_p(l)}} \overline{\lambda}_{plk} a_{\mu_p(l)k} \qquad p \in P, l \in L_p \tag{69}$$

$$\overline{c}_{pl} = \sum_{k \in B_{\mu_p(l)}} \overline{\lambda}_{plk} c_{\mu_p(l)k} \qquad p \in P, l \in L_p \tag{70}$$

$$\sum_{k \in B_{\mu_p(l)}} \overline{\lambda}_{plk} = \sum_{k \in B_{\mu_p(l)} \backslash \{0\}} \overline{w}_{plk} \qquad p \in P, l \in L_p \tag{71}$$

$$\sum_{k \in B_{\mu_p(l)} \backslash \{0\}} \overline{w}_{plk} = x_p \qquad p \in P, l \in L_p \tag{72}$$

$$\overline{\lambda}_{i0} \leqslant \overline{w}_{pl1} \qquad p \in P, l \in L_p \tag{73}$$

$$\overline{\lambda}_{plk} \leqslant \overline{w}_{plk} + \overline{w}_{pl,k+1} \qquad p \in P, l \in L_p, k \in B_{\mu_p(l)} \backslash \{0, b_{\mu_p(l)}\} \tag{74}$$

$$\overline{\lambda}_{plb_{\mu_p(l)}} \leqslant \overline{w}_{plb_{\mu_p(l)}} \qquad p \in P, l \in L_p \tag{75}$$

$$\Delta_{pl} = \bar{c}_{pl} - \underline{c}_{pl} \qquad p \in P, l \in L_p \tag{76}$$

$$x_p \in \{0,1\} \qquad p \in P \tag{77}$$

$$\tau_p \geqslant 0, y_p \geqslant 0 \qquad p \in P \tag{78}$$

$$\underline{c}_{pl}, \bar{c}_{pl}, \underline{q}_{pl}, \bar{q}_{pl}, \Delta_{pl} \geqslant 0 \qquad p \in P, l \in L_p \tag{79}$$

$$\underline{w}_{plk} \in \{0,1\}, \overline{w}_{plk} \in \{0,1\} \qquad p \in P, l \in L_p, k \in B_i \backslash \{0\} \tag{80}$$

$$\underline{\lambda}_{plk} \geqslant 0, \overline{\lambda}_{plk} \geqslant 0 \qquad p \in P, l \in L_p, k \in B_i. \tag{81}$$

The objective function (51) minimizes the total driving and charging time. Constraints (52) ensure that each customer is served once. Constraints (53) impose the flow conservation conditions. Constraints (54) track the SoC of the EVs at each customer. Constraints (55) track the SoC of the battery upon arrival at the first CS of each CSP. For each CSP, constraints (56) couple the SoC of an EV upon departing from a CS with the SoC upon arriving at the next CS of the CSP. Constraints (57) state that every EV leaves the depot with a fully charged battery. Constraints (58) ensure that if the EV travels a path between a customer node and the depot, then it has sufficient energy to reach its destination. Constraints (59) couple the SoC tracking variable with the arc travel variables. Constraints (60) track the departure time from each node. Constraints (61) couple the time tracking variable with the arc travel variables, and impose the route maximum-duration limit. Constraints (62)–(75) are related to the piecewise linear approximation of the charging function. Specifically, constraints (62)–(68) define the SoC and the charging time of an EV upon arrival at each CS of the CSPs, and constraints (69)–(75) define the SoC and charging time of an EV upon departure from each CS of the CSPs. Constraints (76) define the time spent charging at every CS. Finally, constraints (77)–(81) define the domains of the decision variables.

## 4.2 Dominated paths

Theoretically the set $\widetilde{A}$ contains a very large number of arcs that grows quickly with the number of CSs and the number of customers. However, many of these arcs will not be part of an optimal solution. Hence, this section is devoted to identifying CSPs that can be eliminated using dominance rules. Defining these rules is a complex task since we need to account for the partial charging policy, the various charging technologies, and the piecewise linearity of the charging functions.

We first introduce the concept of *recharging path* (RP). We call a RP a CSP for which the SoC of the EV upon departing from each CS – except the last visited one – is known. Indeed, the SoC required at the end of a RP is unknown, because the destination of the EV after traveling the RP is itself unknown. We denote by $(p, \phi)$ a RP associated with CSP $p$ where $\phi$ is a vector specifying the SoC of the EV upon departing from each CS visited in position $l \in L_p \backslash \{n_p - 1\}$. Specifically, $\phi_l$ is the SoC of the EV upon departing from CS $\mu_p(l)$, with $l \in L_p \backslash \{n_p - 1\}$. We refer to $\phi$ as the *SoC target vector*.

Let us now assume that the SoC of the EV when departing from the origin of each CSP is known. We denote this SoC by $q$ and refer to it as the *initial SoC*. A SoC target vector is said to be feasible for a CSP with respect to $q$ if it leads to an energy-feasible path, that is a path in which the SoC is non-negative and less than $Q$ when the vehicle enters or leaves any of its nodes. Let $\widetilde{CS}$ be a CS other than the last on a CSP (i.e., a CS visited in position $l \in L_p \backslash \{n_p - 1\}$ when traveling a CSP $p$). Since the number of SoC targets at the $\widetilde{CS}$s of a RP is infinite, there is an unbounded number of RPs associated with a given CSP that contains two or more CSs. However, Zündorf (2014) showed that for a given sequence of CS visits between two non-CS nodes, we only need to consider a limited number of SoC targets for the $\widetilde{CS}$s. This is due to the piecewise linearity of the charging functions. Specifically, the number of SoC targets that should be considered at a $\widetilde{CS}$ is restricted by the number of breakpoints of its charging function. The possibilities are the following: 1) One can leave a $\widetilde{CS}$ with just enough energy to arrive to the next CS with an empty battery. 2) One can also stop charging at

a $\widetilde{\text{CS}}$ at any breakpoint of the charging function if the resulting SoC is larger than the SoC of the EV upon arriving at the $\widetilde{\text{CS}}$. Therefore, we apply these results to RPs, and by doing so we limit the number of RPs associated with every CSP.

We now define a dominance rule between two RPs. To this end, we exploit the concept of SoC functions introduced by Zündorf (2014) to solve an electric vehicle shortest path problem, which consists in finding a minimum duration route satisfying battery constraints and allowing the EV to recharge en route. A SoC function $SoC^q_{(p,\phi)}$, for a RP $(p,\phi)$ and a given initial SoC $q$, maps the duration of the path $t$ to the final SoC of the EV at destination $d(p)$.

Since we assume that all charging functions are piecewise linear, a SoC function can be defined by an ordered set of supporting points $\{(t_1, y_1), \cdots, (t_k, y_k)\}$ sorted in non-decreasing order of duration. These points are derived from the breakpoints of the charging function of the last CS in a RP according to $p$, $\phi$ and $q$. The value $t_1$ represents the minimal time required by an EV to travel on the RP according to the battery constraints (no undercharging or overcharging is allowed and the SoC is sufficient to travel on each arc) and $q$. A SoC function represented by $\{(t_1, y_1), \cdots, (t_n, y_n)\}$ is defined as follows:

$$SoC^q_{(p,\phi)}(t) = \begin{cases} -\infty & \text{if } 0 \leqslant t < t_1 \\ \dfrac{(t - t_{k-1})(y_k - y_{k-1})}{t_k} + y_{k-1} & \text{if } t_{k-1} \leqslant t < t_k, \quad k \in \{2, 3, ..., n\} \\ y_n & \text{if } t \geqslant t_n. \end{cases} \tag{82}$$
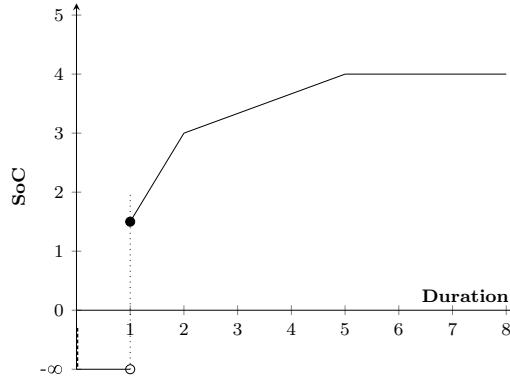


Figure 2: Example of a SoC function associated with the supporting points (1, 1.5), (2, 3), and (5, 4).

Figure 2 depicts a SoC function. In this example, the first supporting point of the curve is (1, 1.5). This means that the minimum duration to travel on the RP is equal to 1, and in this case, the SoC at the destination of the EV is equal to 1.5. If the SoC needed at $d(p)$ is 3, the duration of the corresponding path is equal to 2 (the duration is increased due to a larger amount of energy charged at a CS previously visited by the RP). A dominance rule between RPs having the same origin and destination can then be defined as follows:

**Definition 4.1.** *Let $(p_1, \phi_1)$ and $(p_2, \phi_2)$ be two RPs with the same origin $o = o(p_1) = o(p_2)$ and the same destination $d = d(p_1) = d(p_2)$, and let $q$ be the initial SoC of the EV at $o$. RP $(p_1, \phi_1)$ dominates RP $(p_2, \phi_2)$ with respect to $q$ if $SoC^q_{(p_1,\phi_1)}(t) \geqslant SoC^q_{(p_2,\phi_2)}(t)$ for every $t \geqslant 0$ and there exists $t' \geqslant 0$ such that $SoC^q_{(p_1,\phi_1)}(t') > SoC^q_{(p_2,\phi_2)}(t')$. We denote this dominance by $(p_1, \phi_1) \succ_q (p_2, \phi_2)$.*

This definition states that a RP dominates another only if for every possible duration the EV reaches the destination with higher or equal SoC. The dominance of RP $(p_1, \phi_1)$ over RP $(p_2, \phi_2)$ can be established by comparing the value of $SoC^q_{(p_1,\phi_1)}$ and $SoC^q_{(p_2,\phi_2)}$ for all abscissas of the supporting points of both functions.

For convenience, we also provide an equivalent definition to Definition 4.1 based on the comparison of the inverse of the SoC function. Let $\overline{\Theta}^q_{(p,\phi)}$ and $\underline{\Theta}^q_{(p,\phi)}$ be the maximum and minimum SoC that can be achieved at the destination of RP $(p,\phi)$ if the initial SoC is equal to $q$. The inverse of function $SoC^q_{(p,\phi)}$, denoted

$(SoC_{(p,\phi)}^q)^{-1}$, exists only in the interval $[\underline{\Theta}_{(p,\phi)}^q, \overline{\Theta}_{(p,\phi)}^q]$. We define its extension, denoted $\mathrm{InvSoC}_{(p,\phi)}^q$, over the interval $[0, Q]$ as follows:

$$\mathrm{InvSoC}_{(p,\phi)}^q(\tilde{q}) = \begin{cases} (SoC_{(p,\phi)}^q)^{-1}(\underline{\Theta}_{(p,\phi)}^q) & \text{if } 0 \leqslant \tilde{q} < \underline{\Theta}_{(p,\phi)}^q \\ (SoC_{(p,\phi)}^q)^{-1}(\tilde{q}) & \text{if } \underline{\Theta}_{(p,\phi)}^q \leqslant \tilde{q} \leqslant \overline{\Theta}_{(p,\phi)}^q \\ \infty & \text{if } \overline{\Theta}_{(p,\phi)}^q < \tilde{q} \leqslant Q \end{cases} . \tag{83}$$

Note that since it is not possible to achieve a SoC $0 \leqslant \tilde{q} < \underline{\Theta}_{(p,\phi)}^q$, the first case assumes that the time required to reach such a $\tilde{q}$ is equivalent to the time needed to reach $\underline{\Theta}_{(p,\phi)}^q$. An equivalent definition to Definition of 4.1 then reads:

**Definition 4.2.** *Let $(p_1, \phi_1)$ and $(p_2, \phi_2)$ be two RPs with the same origin $o = o(p_1) = o(p_2)$ and the same destination $d = d(p_1) = d(p_2)$, and let $q$ be the initial SoC of the EV. RP $(p_1, \phi_1)$ dominates RP $(p_2, \phi_2)$ with respect to $q$ if $InvSoC_{(p_1,\phi_1)}^q(\tilde{q}) \leqslant InvSoC_{(p_2,\phi_2)}^q(\tilde{q})$ for every $\tilde{q}$ and there exists $\tilde{q}'$ such that $InvSoC_{(p_1,\phi_1)}^q(\tilde{q}') < InvSoC_{(p_2,\phi_2)}^q(\tilde{q}')$. This is also denoted by $(p_1, \phi_1) \succ_q (p_2, \phi_2)$.*

This definition states that a RP dominates another only if for every possible SoC at the destination it has a shorter or equal duration. As in Definition 4.1, the dominance of RP $(p_1, \phi_1)$ over RP $(p_2, \phi_2)$ can be established by comparing the value of $\mathrm{InvSoC}_{(p_1,\phi_1)}^q$ and $\mathrm{InvSoC}_{(p_2,\phi_2)}^q$ for all abscissas of the supporting points of both functions.

**Definition 4.3.** *CSP $p_1$ is said to be non-dominated if there exists at least one RP $(p_1, \phi_1)$ and an initial SoC $q$, such that for every RP $(p_2, \phi_2)$ with the same origin $o = o(p_1) = o(p_2)$ and the same destination $d = d(p_1) = d(p_2)$ we have $(p_2, \phi_2) \nsucc_q (p_1, \phi_1)$ (i.e., $(p_2, \phi_2)$ does not dominate $(p_1, \phi_1)$ for an initial SoC $q$).*

The initial SoC $q$ can be restricted for a RP $(p, \phi)$ to take its value in the interval $[q_{(p,\phi)}^{MIN}, q_{(p,\phi)}^{MAX}]$. Let $\varrho_{(p,\phi)}(l)$ denote the SoC upon arrival at CS $\mu_p(l)$ for RP $(p, \phi)$ (with $l \in L_p \setminus \{0\}$). This quantity does not depend on the initial SoC $q$. The values of $q_{(p,\phi)}^{MIN}$ and $q_{(p,\phi)}^{MAX}$ can then be defined as follows:

$$q_{(p,\phi)}^{MIN} = \begin{cases} Q & \text{if } o(p) = 0 \\ e_{o(p),\mu_p(0)} & \text{if } n_p \neq 0 \wedge o(p) \neq 0 \\ e^p + \min_{l \in F \cup \{0\}} e_{d(p),l} & \text{if } n_p = 0 \wedge o(p) \neq 0 \end{cases}$$

$$q_{(p,\phi)}^{MAX} = \min\{Q - \min_{l \in F \cup \{0\}} e_{l,o(p)}, \min_{l \in L_p \setminus \{0\}} \{\varrho_{(p,\phi)}(l) + e_{o(p),\mu_p(l)}\}\}$$

The value $q_{(p,\phi)}^{MIN}$ corresponds to the minimum SoC needed at the origin to make RP $(p, \phi)$ energy-feasible. The first element in the definition of $q_{(p,\phi)}^{MAX}$ corresponds to the maximum SoC achievable at the origin of the RP. The second element considers that if the initial SoC allows the EV to reach a CS with a SoC larger than or equal to that achievable by visiting previous CSs, then the RP that goes directly from the origin to that CS dominates the current RP (because the driving time and energy consumption satisfy the triangular inequality).

It is impractical to compute the non-dominated RPs for every possible initial SoC using Definition 4.3, since there is an infinite number of possible initial SoCs (except at the depot). To resolve this issue, we show in Lemma 4.1 how given a SoC function and an initial SoC $q$, we are able to compute the SoC functions for other initial SoCs $q'$ for a given RP. Based on this result, we demonstrate in Proposition 4.1 how the dominance relationship between two different RPs with the same origin and destination can be efficiently verified.

**Lemma 4.1.** *Let $(p, \phi)$ be a RP and let $q \in [q_{(p,\phi)}^{MIN}, q_{(p,\phi)}^{MAX}]$. If the initial SoC at $o(p)$ is equal to $q'$ with $q \leqslant q' \leqslant q_{(p,\phi)}^{MAX}$, then the inverse of the SoC function $InvSoC_{(p,\phi)}^{q'}$ can be computed as follows:*

- *If $n_p \neq 0$ (i.e., $p$ contains at least one CS)*

$$\forall \, \tilde{q} \in [0, Q], \quad InvSoC_{(p,\phi)}^{q'}(\tilde{q}) = \begin{cases} InvSoC_{(p,\phi)}^q(\tilde{q}) - \lambda_{qq'}^p & \text{if } \tilde{q} \geqslant q' - e^p \\ InvSoC_{(p,\phi)}^q(q' - e^p) - \lambda_{qq'}^p & \text{if } \tilde{q} < q' - e^p \end{cases} \tag{84}$$

13

where $\lambda^p_{qq'} = \Phi^{-1}_{\mu_p(0)}(q' - e_{o(p),\mu_p(0)}) - \Phi^{-1}_{\mu_p(0)}(q - e_{o(p),\mu_p(0)})$ *corresponds to the difference in time spent charging $q' - q$ at the first CS of the path.*

- *If $n_p = 0$ (i.e., $p$ does not contain any CS)*

$$\forall \ \tilde{q} \in [0, Q], \quad InvSoC^{q'}_{(p,\phi)}(\tilde{q}) = \begin{cases} t^p & \text{if } q' - e^p \geqslant \tilde{q} \\ \infty & \text{if } q' - e^p < \tilde{q} \end{cases} \tag{85}$$

*Proof.* See B. $\qquad \square$

For each pair of values $q_1$ and $q_2$ such that $q^{MIN}_{(p_1,\phi_1)} \leqslant q_1 \leqslant q^{MAX}_{(p_1,\phi_1)}$ and $q^{MIN}_{(p_2,\phi_2)} \leqslant q_2 \leqslant q^{MAX}_{(p_2,\phi_2)}$, we define two functions, denoted by $\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$ and $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$. Intuitively, these functions represent the maximum and minimum differences in time for the two RPs to reach the same SoC at the destination. They are defined as follows:

$$\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) = \begin{cases} \max_{\tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)} \{ InvSoC^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - InvSoC^{q_2}_{(p_2,\phi_2)}(\tilde{q}) \} & n_{p_1} > 0 \wedge n_{p_2} > 0 \\ \max_{\tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)} \{ t^{p_1} - InvSoC^{q_2}_{(p_2,\phi_2)}(\tilde{q}) \} & n_{p_1} = 0 \wedge n_{p_2} > 0 \end{cases}$$

$$\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) = \begin{cases} \min_{\tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)} \{ InvSoC^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - InvSoC^{q_2}_{(p_2,\phi_2)}(\tilde{q}) \} & n_{p_1} > 0 \wedge n_{p_2} > 0 \\ \min_{\tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)} \{ InvSoC^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - t^{p_2} \} & n_{p_1} > 0 \wedge n_{p_2} = 0 \end{cases}$$

where $\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)$ is an interval equal to $[\max\{0, q - \min\{e^{p_1}, e^{p_2}\}\}, \min\{\overline{\Theta}^{q_1}_{(p_1,\phi_1)}, \overline{\Theta}^{q_2}_{(p_2,\phi_2)}\}]$.

We note that for a CSP with $n_p = 0$, since there are no charging decisions involved, there is a single RP. This RP cannot be dominated for every possible value of the initial SoC by other RPs, which are associated with other CSPs, because of the triangular inequality. We therefore never verify if a RP $p$ with $n_p = 0$ is dominated. However, such a RP may dominate other RPs. In this case, the initial SoC of RP $p$ with $n_p = 0$ must be such that the SoC at the destination is larger than the maximum SoC attainable at the destination by the other RPs. We also note that given two RPs $(p_1, \phi_1)$ and $(p_2, \phi_2)$ with the same origin and the same destination, such that $q^{MIN}_{(p_1,\phi_1)} \leqslant q^{MIN}_{(p_2,\phi_2)}$, the dominance for these two RPs must be verified only for $q \geqslant q^{MIN}_{(p_2,\phi_2)}$. Finally, we note that for a RP $(p, \phi)$ such that $n_p > 0$, reaching the maximum SoC at the destination is established when the EV is fully charged at the last CS of $p$. We note that this is independent of the value of the initial SoC $q$. Hence for simplicity we denote $\overline{\Theta}^q_{(p,\phi)}$ by $\overline{\Theta}_{(p,\phi)}$. Based on these observations, the dominance test between the two RPs (where $q^{MIN}_{(p_1,\phi_1)} \leqslant q^{MIN}_{(p_2,\phi_2)}$) needs to be carried for $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ (when the first element is less than or equal to the second element) where $\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$ and $\overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$ are defined as follows:

$$\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)} = \begin{cases} q^{MIN}_{(p_2,\phi_2)} & \text{if } n_{p_1} \geqslant 1 \wedge n_{p_2} \geqslant 1 \\ \max\{q^{MIN}_{(p_2,\phi_2)}, \overline{\Theta}_{(p_2,\phi_2)} + e_{o(p_1),d(p_1)}\} & \text{if } n_{p_1} = 0 \wedge n_{p_2} \geqslant 1 \\ \max\{q^{MIN}_{(p_2,\phi_2)}, \overline{\Theta}_{(p_1,\phi_1)} + e_{o(p_2),d(p_2)}\} & \text{if } n_{p_1} \geqslant 1 \wedge n_{p_2} = 0 \end{cases}$$
$$\overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)} = \min\{q^{MAX}_{(p_1,\phi_1)}, q^{MAX}_{(p_2,\phi_2)}\}$$

Observe that if $\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)} > \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$, then we cannot derive any dominance relationship. Moreover, if $n_{p_1} > 0$ and $\overline{\Theta}_{(p_1,\phi_1)} < \overline{\Theta}_{(p_2,\phi_2)}$, then the EV can have a higher SoC after traversing $(p_2, \phi_2)$ compared to the one obtained after traversing $(p_1, \phi_1)$, which means that there does not exist $q$ such that $(p_1, \phi_1) \succ_q (p_2, \phi_2)$. Similarly, if $n_{p_2} > 0$ and $\overline{\Theta}_{(p_2,\phi_2)} < \overline{\Theta}_{(p_1,\phi_1)}$, then there does not exist $q$ such that $(p_2, \phi_2) \succ_q (p_1, \phi_1)$. The following result allows us to compute intervals for the initial SoC for which there may exist a dominance between RPs.

**Proposition 4.1.** *Let $(p_1, \phi_1)$ and $(p_2, \phi_2)$ be two RPs with the same origin $o = o(p_1) = o(p_2)$ and the same destination $d = d(p_1) = d(p_2)$ such that $q^{MIN}_{(p_1,\phi_1)} \leqslant q^{MIN}_{(p_2,\phi_2)}$. For $k \in \{1, 2\}$, let $q_k$ be a parameter equal to $\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$ if $n_{p_k} = 0$ and $q^{MIN}_{(p_k,\phi_k)}$ otherwise. If $\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)} \leqslant \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$, the following statements hold:*

- **Case 1:** $n_{p_1} = 0$ or $\overline{\Theta}_{(p_1,\phi_1)} \geqslant \overline{\Theta}_{(p_2,\phi_2)}$:

  - if for an initial SoC $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \min\{e^{p_1}, e^{p_2}\}]$, we have $\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$ and $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we can state that $(p_1, \phi_1) >_q (p_2, \phi_2)$

  - if for an initial SoC $q \in [\min\{e^{p_1}, e^{p_2}\}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]]$, we have $\max\{\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2), \overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1)\} \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$ and $\min\{\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2), \underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1)\} < \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we can state that $(p_1, \phi_1) >_q (p_2, \phi_2)$

- **Case 2:** $n_{p_2} = 0$ or $\overline{\Theta}_{(p_2,\phi_2)} \geqslant \overline{\Theta}_{(p_1,\phi_1)}$

  - if for an initial SoC $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \min\{e^{p_1}, e^{p_2}\}]$, we have $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$ and $\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we can state that $(p_2, \phi_2) >_q (p_1, \phi_1)$.

  - if for an initial SoC $q \in [\min\{e^{p_1}, e^{p_2}\}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]]$, we have $\min\{\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2), \underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2)\} \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$ and $\max\{\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2), \overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2)\} > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we can state that $(p_2, \phi_2) >_q (p_1, \phi_1)$.

where $\Delta_p(q)$, $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, \alpha)$, and $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, \alpha)$ are defined as follows:

$$\Delta_p(q) = \begin{cases} \Phi^{-1}_{\mu_p(0)}(q - e_{o(p),\mu_p(0)}) & \text{if } n_p \geqslant 1 \\ 0 & \text{if } n_p = 0 \end{cases}$$

$$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, \alpha) = \begin{cases} InvSoC^{q_1}_{(p_1,\phi_1)}(\max\{0, q - \min\{e^{p_1}, e^{p_2}\}\}) - InvSoC^{q_2}_{(p_2,\phi_2)}(\max\{0, q - e^{p_2}\}) & \text{if } n_{p_1} \geqslant 1 \\ t^{p_1} - InvSoC^{q_2}_{(p_2,\phi_2)}(\max\{0, q - e^{p_2}\}) & \text{if } n_{p_1} = 0 \wedge \alpha = 1 \\ t^{p_1} - InvSoC^{q_2}_{(p_2,\phi_2)}(q - \min\{e^{p_1}, e^{p_2}\}) & \text{if } n_{p_1} = 0 \wedge \alpha = 2 \end{cases}$$

$$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, \alpha) = \begin{cases} InvSoC^{q_1}_{(p_1,\phi_1)}(\max\{0, q - e^{p_1}\}) - InvSoC^{q_2}_{(p_2,\phi_2)}(\max\{0, q - \min\{e^{p_1}, e^{p_2}\}\}) & \text{if } n_{p_2} \geqslant 1 \\ InvSoC^{q_1}_{(p_1,\phi_1)}(q - \min\{e^{p_1}, e^{p_2}\}) - t^{p_2} & \text{if } n_{p_2} = 0 \wedge \alpha = 1 \\ InvSoC^{q_1}_{(p_1,\phi_1)}(\max\{0, q - e^{p_1}\}) - t^{p_2} & \text{if } n_{p_2} = 0 \wedge \alpha = 2 \end{cases}$$

*Proof.* See B. $\qquad\square$

Using this proposition, we design an algorithm that enumerates all non-dominated CSPs between any pair of nodes $i, j \in \widetilde{V}$ belonging to the multigraph. We first build a directed graph $\hat{G}_{ij} = (N_{ij}, A_{ij})$, where $N_{ij} = \{i, j\} \cup F$. For each CS node $l \in F$, we create the arcs $(i, l)$ and $(l, j)$. For each pair of CSs $l, l'$, we also create the arc $(l, l')$. We also consider the arc $(i, j)$. We apply the preprocessing rules described in A on $\hat{G}_{ij}$. We then solve on this graph a multi-objective constrained shortest path problem from $i$ to $j$ where 1) the two objectives are the minimization of the total duration of the path (driving + charging time) and the maximization of the final SoC; 2) there is a constrained resource: the energy (i.e., the SoC of the EV must always be in the interval $[0, Q]$); and 3) the SoC upon leaving $i$ and the SoC needed upon reaching $j$ are unknown. To this end, we use a label-correcting algorithm where each label represents a RP. A SoC function is associated with each RP $(p, \phi)$ assuming that the initial SoC is equal to $q^{MIN}_{(p,\phi)}$. Non-dominated CSPs are then associated with Pareto optimal RPs. We outline the pseudo-code of our label-correcting algorithm in Algorithm 1.

We initialize the algorithm with one label at node $j$ representing a RP associated with arc $(i, j)$. We also initialize the algorithm with one label for each $l \in F$ such that $(i, l) \in A_{ij}$. This label represents a partial RP associated with partial CSP $(i,l)$. At each iteration of the algorithm, we select the label (among the unprocessed labels) with the minimum duration and we extend it along the arcs of $\hat{G}_{ij}$. Selecting this specific label is intended to reduce the number of labels the algorithm generates. If we extend a label along an arc $(l, l')$, then the new label consists of the translation of the SoC function along the vector $(t_{ll'}, -e_{ll'})$. Note that some supporting points of the new SoC function (after the extension along arc $(l, l')$) can have a negative

---

**Algorithm 1:** ComputeCSPs(*i*,*j*)

> **input** : $\hat{G}_{ij}$
>
> **output:** a set containing all the non-dominated CSPs between $i$ and $j$

1  **forall** $l \in N_{ij}\backslash\{i\}$ **do**  $L_1(l) \leftarrow \varnothing$ ($L_1(l)$ *stores the unprocessed labels associated with node l*) ;

2  **forall** $l \in N_{ij}\backslash\{i\}$ **do**  $L_2(l) \leftarrow \varnothing$ ($L_2(l)$ *stores the processed labels associated with node l*) ;

3  Add to $L_2(j)$ the RP associated with arc $(i,j)$

4  $R \leftarrow \varnothing$ ($R$ *stores nodes from* $N_{ij}\backslash\{i\}$ *with unprocessed labels*)

5  **forall** $l \in F$ such that $(i,l) \in A_{ij}$ **do**

6    Add to $L_1(l)$ the label corresponding to the RP starting at $i$ and going directly to $l$

7  **end**

8  $R \leftarrow \{l \in F : (i,l) \in A_{ij}\}$

9  **while** $R \neq \varnothing$ **do**

10   $l \leftarrow$ minElement($R$) *(gets from R the node which has associated the unprocessed label with the minimum total time)*

11   $(p_l, \phi_l) \leftarrow$ deleteMin($L_1(l)$) *(removes from $L_1(l)$ the label with the minimum total time)*

12   **if** $L_1(l) = \varnothing$ **then**  $R \leftarrow R\backslash\{l\}$ ;

13   Update($L_2(l), (p_l, \phi_l)$)

14   **forall** $(l,l') \in \hat{G}_{ij}$ **do**

15    **if** IsExtendable($(p_l, \phi_l), l'$) **then**  *(checks if RP $(p,\phi)$ can be extended to node l')*

16     $(p_{l'}, \phi_{l'}) \leftarrow$ Extend($(p_l, \phi_l), l'$) *(extends RP $(p_l, \phi_l)$ to node l')*

17     $K \leftarrow$ CreateLabels($(p_{l'}, \phi_{l'})$) *(if l' is a CS and $p_{l'}$ contains more than a CS, then it creates the relevant labels according to the supporting points of the SoC function associated with RP $(p_{l'}, \phi_{l'})$, otherwise K only contains $(p_{l'}, \phi_{l'})$)*

18     **forall** $(p, \phi) \in K$ **do**

19      Update($L_1(l'), (p, \phi)$) *(inserts RP $(p,\phi)$ in $L_1(l')$ if it is not dominated according to Proposition 4.1, and verifies if previously inserted labels become dominated)*

20     **end**

21     **if** $L_1(l')$ has changed and $l' \notin R$ **then**  $R \leftarrow R \cup \{l'\}$ ;

22    **end**

23   **end**

24 **end**

25 $P \leftarrow \varnothing$

26 **forall** $(p_j, \phi_j) \in L_2(j)$ **do**

27   **if** $p_j \notin P$ **then**  $P \leftarrow P \cup \{p_j\}$ ;

28 **end**

29 **return** $P$

---

y-coordinate, which implies violating a resource constraint. Therefore, we do not consider these points. If all the supporting points of the new SoC function have a negative y-coordinate, then traversing the arc is impossible. If $l'$ is a CS, we need to set the SoC target value at the previous CS (if there is any in the current path). We note that setting this SoC target value requires knowing the successive CS (if it exists). Stopping charging at the previous CS encountered in the path and starting charging at the current CS is only relevant at a supporting point of the current SoC function (Zündorf, 2014). Hence 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: Note that we can restrict the creation of new labels to the case where the slope of the charging function at the new CS is larger than that of the current SoC function. For each of these labels, the new SoC function is derived from the charging function associated with the new CS. Moreover, whenever we obtain a new label at a node, we check whether the corresponding partial RP is non-dominated before accepting it and we verify if previously generated labels associated with the same node become dominated. Specifically, using the result of Proposition 4.1 we can identify intervals for the SoC at the departure from $i$ for which the newly generated label is dominated by or dominates a processed or an unprocessed label associated with the same node. If a RP $(p,\phi)$ becomes dominated for every SoC $q \in [q^{MIN}_{(p,\phi)}, q^{MAX}_{(p,\phi)}]$, then it can be safely removed from the set of labels. The algorithm terminates when there are no more labels to set at the destination node.

To speed up the algorithm, we do not extend a label $(p, \phi)$ associated with a node $l$ to a node $l'$ if at least one of the following conditions is met: 1) the sum of the minimum duration associated with the SoC function at the current label (i.e., the x-coordinate of the first supporting point), the driving time $t_{ll'}$, and the quantity $t_{0i} + t_{l'j} + t_{j0}$ is larger than the sum of the duration limit $T_{max}$ and the quantity $\Phi^{-1}_{\mu_p(0)}(Q)$; 2) If $l'$ is a CS node, then while scanning the nodes of the route backwards we encounter the same node $l'$ without having scanned before a CS with a faster technology (note that a CS can be visited more than once in a path). The latter condition is due to the positivity of the energy consumption and driving time.

# 5 The fixed route vehicle charging problem

Building on the insights obtained by the previous sections, we now focus on the fixed route vehicle charging problem (FRVCP) which is a subproblem of the E-VRP-NL. We recall that FRVCPs are fundamental subproblems of E-VRP heuristics. Let $\Pi$ be a route that does not visit any CS. The objective of the FRVCP is to determine the charging operations (where to insert the CSs and how much to charge), in order to minimize the total route duration while satisfying the following conditions: the customers in the resulting route are visited in the initial order, and the resulting route is energy-feasible and satisfies the maximum-duration limit. We restrict our attention to routes that cannot be traveled by an EV without the need to recharge the battery. We therefore need to insert CSs in the route to make it energy-feasible. Figure 3 shows an example of the FRVCP with a route serving four customers and the possibility of visiting two CSs (as many times as necessary). In the FRVCP we allow an EV to visit several CSs between two successive non-CS nodes. Many studies focus on a more restricted version of the FRVCP in which an EV can visit at most one CS between two successive non-CS nodes. We denote this restricted version by FRVCP-1.
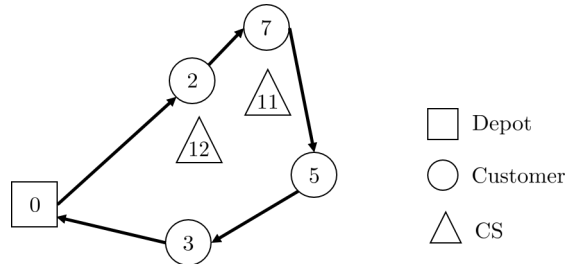


Figure 3: Example of a fixed route vehicle charging problem with route (0,2,7,5,3,0).

Here we propose three alternative algorithms to solve the FRVCP. Let $\Pi = (\pi(0), \pi(1), \cdots, \pi(i), \cdots, \pi(n(\Pi) - 1), \pi(n(\Pi)))$ be a fixed route, where $\pi(0)$ and $\pi(n(\Pi))$ represent the depot and the remaining $\pi(i)$ are the visited customers. We first describe in §5.1 a preprocessing procedure to reduce the size of the search space. We then present in §5.2 a MILP formulation for the problem. Contrary to the formulation by Montoya et al. (2017), our model accounts for visits to multiple CSs between any pair of non-CS nodes. Finally, we present in §5.3 an exact labeling algorithm for this formulation. Montoya et al. (2017) also introduced a greedy heuristic for the FRVCP. Contrary to their formulation, their heuristic allows visiting a number of CSs between two successive nodes of $\Pi$. Building on their work, we develop in §5.4 a new FRVCP heuristic.

## 5.1 Preprocessing procedure

Our preprocessing procedure works as follows. We first compute, according to $\Pi$ and $T_{max}$, the maximum time that is possible to spend detouring and charging at CSs. Using this value, we identify a priori the set of feasible CS insertions and the set of feasible consecutive CS visits between every pair of consecutive nodes of $\Pi$. We also remove additional infeasible connections between the nodes of $\Pi$ and the CSs. More specifically,

we perform the following preprocessing steps: 1) Let $\Delta t(\Pi) = T_{max} - \sum_{i=0}^{n(\Pi)-1} t_{\pi(i),\pi(i+1)}$ be the maximum time that is possible to spend detouring and charging at CSs; 2) Let $\pi(i)$ be the $i^{\text{th}}$ node in $\Pi$ and $j$ a CS. For $i \in \{0, \ldots, n(\Pi) - 1\}$, if $t_{\pi(i),j} + t_{j,\pi(i+1)} - t_{\pi(i),\pi(i+1)} > \Delta t(\Pi)$, then we forbid the insertion of CS $j$ between nodes $\pi(i)$ and $\pi(i+1)$; 3) Let $j, j'$ be two different CSs. If $t_{\pi(i),j} + t_{j,j'} + t_{j',\pi(i+1)} - t_{\pi(i),\pi(i+1)} > \Delta t(\Pi)$, then we forbid visiting CS $j'$ after $j$ between nodes $\pi(i)$ and $\pi(i+1)$; 4) Let $i, j$ be two nodes that belong to $\Pi$ or to $F$. If $\min_{l \in F \cup \{0\}}\{e_{li}\} + e_{ij} + \min_{l \in F \cup \{0\}}\{e_{jl}\} > Q$, then we forbid visiting node $j$ directly after node $i$ (since it violates the battery constraints).

## 5.2   A path-based model for the FRVCP

Building on the path-based formulation proposed in Section 4, we now propose a MILP formulation for the FRVCP. Let $P(\Pi)$ be the set of CSPs between two consecutive nodes of route $\Pi$. To generate these paths, we use the labeling algorithm described in Algorithm 1. The construction of the underlying graph takes into account the results of the preprocessing procedure. We use the same decisions variables as in formulation $[F_{Arc}^{Path}]$ but we define them only for paths belonging to $P(\Pi)$. The binary variable $x_p$ is equal to 1 if and only if an EV travels on CSP $p \in P(\Pi)$. For a path $p$, continuous variable $y_p$ track the time at which the EV leaves $o(p)$. The variables $\underline{c}_{pl}$ and $\overline{c}_{pl}$ are the scaled start and end times for charging an EV according to the charging function of CS $\mu_p(l)$. Continuous variables $\underline{q}_{pl}$ and $\overline{q}_{pl}$ specify (according to the piecewise linear approximation of the charging function) the SoC of an EV when it enters and leaves $\mu_p(l)$, i.e., the CS at position $l \in L_p$. The continuous variable $\Delta_{pl}$ represents the duration of the charging operation performed at $\mu_p(l)$. The binary variables $\underline{w}_{plk}$ and $\overline{w}_{plk}$ are equal to 1 if and only if the SoC is between $a_{\mu_p(l),k-1}$ and $a_{\mu_p(l),k}$, with $k \in B_i \backslash \{0\}$, when the EV enters and leaves CS $\mu_p(l)$, respectively. The continuous variables $\underline{\lambda}_{plk}$ and $\overline{\lambda}_{plk}$ are the coefficients associated with the breakpoint $(c_{\mu_p(l),k}, a_{\mu_p(l),k})$ in the piecewise linear approximation, when the EV enters and leaves CS $\mu_p(l)$, respectively. Let $e^p$ and $t^p$ be the energy consumption and the driving time associated with path $p \in P$. A formulation of the FRVCP for a route $\Pi$, denoted as $[FR^{Path}(\Pi)]$, is as follows:

$$[FR^{Path}(\Pi)] \qquad\qquad \text{minimize} \sum_{p \in P(\Pi)} \left( t^p x_p + \sum_{l \in L_p} \Delta_{pl} \right) \tag{86}$$

subject to

$$\sum_{p \in P_{\pi(i),\pi(i+1)}} x_p = 1, \qquad i \in \{0, \cdots, n(\Pi) - 1\} \tag{87}$$

$$\sum_{p \in P_{\pi(i-1),\pi(i)}} \left( y_p - e^p x_p + \sum_{l \in L_p} (\overline{q}_{pl} - \underline{q}_{pl}) \right) = \sum_{P_{\pi(i),\pi(i+1)}} y_p, \qquad i \in \{1, \cdots, n(\Pi) - 1\} \tag{88}$$

$$y_p - e_{o(p),\mu_p(0)} x_p = \underline{q}_{p0}, \qquad p \in P(\Pi) : |L_p| \neq 0 \tag{89}$$

$$\overline{q}_{p,l-1} - e_{\mu_p(l-1),\mu_p(l)} x_p = \underline{q}_{pl}, \qquad p \in P(\Pi), l \in L_p \backslash \{0\} \tag{90}$$

$$\sum_{p \in P_{\pi(n(\Pi)-1),\pi(n(\Pi))}} \left( y_p - e^p x_p - \sum_{l \in L_p} (\overline{q}_{pl} - \underline{q}_{pl}) \right) \geq 0, \tag{91}$$

$$y_p = Q x_p, \qquad p \in P_{\pi(0),\pi(1)} \tag{92}$$

$$y_p \leq Q x_p, \qquad p \in P(\Pi) \tag{93}$$

$$\sum_{p \in P(\Pi)} \left( t^p x_p + \sum_{l \in L_p} \Delta_{pl} \right) + \sum_{i=1}^{n(\Pi)-1} g_{\pi(i)} \leq T_{max} \tag{94}$$

$$\underline{q}_{pl} = \sum_{k \in B_{\mu_p(l)}} \underline{\lambda}_{plk} a_{\mu_p(l)k} \qquad p \in P(\Pi), l \in L_p \tag{95}$$

$$\underline{c}_{pl} = \sum_{k \in B_{\mu_p(l)}} \underline{\lambda}_{plk} c_{\mu_p(l)k} \qquad p \in P(\Pi), l \in L_p \tag{96}$$

$$\sum_{k \in B_{\mu_p(l)}} \underline{\lambda}_{plk} = \sum_{k \in B_{\mu_p(l)} \backslash \{0\}} \underline{w}_{plk} \qquad p \in P(\Pi), l \in L_p \tag{97}$$

$$\sum_{k \in B_{\mu_p(l)} \backslash \{0\}} \underline{w}_{plk} = x_p \qquad p \in P(\Pi), l \in L_p \tag{98}$$

$$\underline{\lambda}_{pl0} \leqslant \underline{w}_{pl1} \qquad p \in P(\Pi), l \in L_p \tag{99}$$

$$\underline{\lambda}_{plk} \leqslant \underline{w}_{plk} + \underline{w}_{pl,k+1} \qquad p \in P(\Pi), l \in L_p, k \in B_{\mu_p(l)} \backslash \{0, b_{\mu_p(l)}\} \tag{100}$$

$$\underline{\lambda}_{plb_{\mu_p(l)}} \leqslant \underline{w}_{plb_{\mu_p(l)}} \qquad p \in P(\Pi), l \in L_p \tag{101}$$

$$\overline{q}_{pl} = \sum_{k \in B_{\mu_p(l)}} \overline{\lambda}_{plk} a_{\mu_p(l)k} \qquad p \in P(\Pi), l \in L_p \tag{102}$$

$$\overline{c}_{pl} = \sum_{k \in B_{\mu_p(l)}} \overline{\lambda}_{plk} c_{\mu_p(l)k} \qquad p \in P(\Pi), l \in L_p \tag{103}$$

$$\sum_{k \in B_{\mu_p(l)}} \overline{\lambda}_{plk} = \sum_{k \in B_{\mu_p(l)} \backslash \{0\}} \overline{w}_{plk} \qquad p \in P(\Pi), l \in L_p \tag{104}$$

$$\sum_{k \in B_{\mu_p(l)} \backslash \{0\}} \overline{w}_{plk} = x_p \qquad p \in P(\Pi), l \in L_p \tag{105}$$

$$\overline{\lambda}_{i0} \leqslant \overline{w}_{pl1} \qquad p \in P(\Pi), l \in L_p \tag{106}$$

$$\overline{\lambda}_{plk} \leqslant \overline{w}_{plk} + \overline{w}_{pl,k+1} \qquad p \in P(\Pi), l \in L_p, k \in B_{\mu_p(l)} \backslash \{0, b_{\mu_p(l)}\} \tag{107}$$

$$\overline{\lambda}_{plb_{\mu_p(l)}} \leqslant \overline{w}_{plb_{\mu_p(l)}} \qquad p \in P(\Pi), l \in L_p \tag{108}$$

$$\Delta_{pl} = \overline{c}_{pl} - \underline{c}_{pl} \qquad p \in P(\Pi), l \in L_p \tag{109}$$

$$x_p \in \{0, 1\}, \qquad p \in P(\Pi) \tag{110}$$

$$y_p \geqslant 0 \qquad p \in P(\Pi) \tag{111}$$

$$\underline{c}_{pl}, \overline{c}_{pl}, \underline{q}_{pl}, \overline{q}_{pl}, \Delta_{pl} \geqslant 0 \qquad p \in P(\Pi), l \in L_p \tag{112}$$

$$\underline{w}_{plk} \in \{0, 1\}, \overline{w}_{plk} \in \{0, 1\} \qquad p \in P, l \in L_p, k \in B_i \backslash \{0\} \tag{113}$$

$$\underline{\lambda}_{plk} \geqslant 0, \overline{\lambda}_{plk} \geqslant 0 \qquad p \in P, l \in L_p, k \in B_i. \tag{114}$$

The objective function (86) minimizes the total driving and charging time in the route. Constraints (87) ensure the selection of a CSP between each pair of successive non-CS nodes. The remaining constraints are similar to those introduced for model $[F_{Arc}^{Path}]$.

## 5.3 An exact labeling algorithm for the FRVCP

We now present an exact labeling algorithm for the FRVCP. We apply it on a directed graph $G(\Pi)$ that contains the nodes of the initial route $\Pi$ and the CS nodes. To enforce the order of the visits of the nodes that belong to $\Pi$, we replicate the CS nodes in such a way that one copy of every CS is associated with one arc of the initial route $\Pi$. Specifically, for every value of $i$ between 0 and $n(\Pi) - 1$, we create a copy $l(i)$ of CS $l \in F$. We then create the arc $(\pi(i), \pi(i + 1))$, the arcs $(\pi(i), l(i))$ and $(l(i), \pi(i + 1))$ for each CS $l \in F$, and the arcs $(l(i), l'(i))$ for each pair of CS $l, l' \in F$. It is worth mentioning that we only add these arcs to the graph if they still exist after running the preprocessing routines introduced in §5.1. Figure 4 depicts the graph considered for the example of Figure 3.

The FRVCP can then be defined as a constrained shortest path problem from $\pi(0)$ to $\pi(n(\Pi))$ on $G(\Pi)$. The objective is to minimize the path duration while the SoC acts as a resource constraint (i.e., the SoC must always be in the interval $[0, Q]$). To solve this problem, we propose the label-setting algorithm described in Algorithm 3, which shares many similarities with Algorithm 1. Algorithm 3 uses RPs as labels and exploits
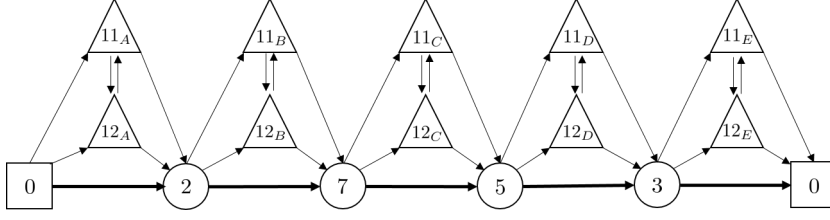
Figure 4: Graph considered when applying the labeling algorithm to solve the fixed route vehicle charging problem with route (0,2,7,5,3,0).

the concept of SoC functions presented in §4.2. Relaxing an arc and setting a label at a CS follows the same procedure as the one described in in §4.2. The algorithm terminates as soon as a label for node $n(\Pi)$ is set. The quantity of energy charged at the last CS of this RP is computed such that the EV reaches the depot with an empty battery.

To improve the efficiency of the algorithm, we apply two different strategies. We make use of lower bounds. Since we know the sequence of nodes to visit, we compute beforehand at each node $j \in G(\Pi)$ lower bounds using Dijkstra's algorithm. We first compute a lower bound $\overrightarrow{t_j}$ on the time and a lower bound $\overrightarrow{e_j}$ on the energy required to finish the route by running a backward search from $\pi(n(\Pi))$. If $\overrightarrow{e_j}$ is larger than $Q$, we can increase the value of $\overrightarrow{t_j}$ by assuming that the difference $Q - \overrightarrow{e_j}$ will be charged using the fastest charging rate. Specifically, we set the value of $\overrightarrow{t_j}$ to $\overrightarrow{t_j} + \max\{0, \overrightarrow{e_j} - Q\}/\rho^*$ where $\rho^*$ corresponds to the steepest slope for a segment of the piecewise linear approximation of the charging functions. We also compute a lower bound $\overleftarrow{e_j}$ on the energy consumption from the departing depot to node $j$ by running a forward search from $\pi(0)$. Then, we do not extend a label associated with a node $i$ to a node $j$ if one of the following four conditions is met: 1) the maximum SoC an EV can have upon departure from $j$ is strictly less than the energy needed to reach to the nearest CS according to the energy consumption, or is less than the SoC it could have attained by traveling on the route to $j$ without visiting any CS. Specifically, the difference between the maximum of the SoC function and the energy consumed $e_{ij}$ is strictly smaller than $\max\{\min_{l \in F \cup 0}\{e_{jl}\}, Q - \overleftarrow{e_j}\}$; 2) The sum of the minimum duration associated with the SoC function at the current label (i.e., the x-coordinate of the first supporting point), the driving time $t_{ij}$, and $\overrightarrow{t_j}$ is larger than the duration limit $T_{max}$; 3) If $j$ is a CS node, then while scanning the nodes of the route backward we encounter the same node $j$ without having scanned before the depot, a customer, or a CS with a faster technology; 4) If $j$ is a CS node, the value of the inverse of the SoC function evaluated in $\overrightarrow{e_j}$ is less than the sum of the minimum time associated with the current label and the time increase due to the detour to $j$.

Finally, to avoid enumerating all the RPs from $i$ to $j$, we use the dominance criterion introduced in Definition 4.1 (for a SoC upon departing to the depot equal to $Q$) to discard dominated RPs throughout the algorithm.

## 5.4   A heuristic algorithm for the FRVCP

The goal of this section is to design an effective and efficient heuristic algorithm, which could be periodically called inside a local search algorithm. Such an algorithm would prevent the search from ignoring regions of the search space that are feasible with respect to route duration constraints. The algorithm borrows some ideas from the FRVCP heuristic of Montoya et al. (2017). We describe the general framework of our heuristic in Algorithm 2. We refer the reader to E.2 for the algorithmic details on the different procedures of the heuristic. We note that this heuristic inserts CSs into $\Pi$.

Throughout the algorithm, we compute the SoC $Y(\Pi, i)$ of the EV upon arrival at node $i \in \Pi$ assuming the following heuristic (H) charging policy: at each CS, the EV is charged in such a way that its departure SoC is equal to the minimum between the battery capacity $Q$ and the energy required to reach the next depot or CS in $\Pi$. We then define the *energy deficit* at node $i$ as $\max\{0, -Y(\Pi, i)\}$.

The algorithm starts with procedure `locateCS1`(·) that scans the nodes in the route (see Algorithm 4). If the arc connecting two successive nodes does not exist (this rarely happens but it requires special attention), then the algorithm tries to insert a CS between these nodes by calling procedure `insertCS`(·) (see Algorithm 5). Inserting a CS between two nodes follows the same scheme throughout the algorithm. We evaluate the insertion of each CS by calling function `getLBTime`(·) which computes a lower bound on the duration to travel the resulting route. The lower bound value is the sum of two components: the driving time and a lower bound on the time to charge the energy deficit observed upon arrival at CSs and at the return to the depot. This latter bound is computed by dividing the energy deficit by the fastest charging rate of the instance. The energy deficit, computed using function `getEnergyDeficit`(·), is equal to $\sum_{i \in \Pi \cap (F \cup \{0\})} \max\{0, -Y(\Pi, i)\}$. We retain the insertion that leads to the minimum lower bound value. We forbid any insertion if after executing it the energy deficit computed at the inserted CS and at the next CS that is part of the route, if any, or the depot is larger than 0. Note that it may happen that the algorithm selects an insertion leading to an energy infeasible route rather than an insertion leading to an energy-feasible route. Throughout the execution of the algorithm, we therefore store the energy-feasible route with the minimum duration, assuming the H charging policy, returned by function `getTime`(·).

Whenever there exists an arc between each pair of successive nodes, we try to insert additional CSs by calling procedure `locateCS2`(·) (see Algorithm 6). Here, we iteratively test the insertion of a CS between two successive nodes in the current route and we proceed to the insertion yielding the minimum lower bound on the duration to travel the route. We try to insert additional CSs until we cannot improve the duration of the current route. Specifically, this procedure terminates when the lower bound on the duration of the current route is larger than or equal to the duration of the best energy-feasible route.

We then consider the best energy-feasible route computed during the previous steps of the algorithm. Calling procedure `defineChargingAmount`(·) (see Algorithm 7), we check whether it is possible to transfer some charge between the CSs of the route. Indeed, it may be profitable in some cases to charge at a CS more than the energy required to reach the next CS. To this end, we scan the CSs starting by the end of the route, and we try to transfer some energy from a CS to the CS that precedes it. A transfer of energy reduces the duration of the route if the slope of the charging function associated with the departure SoC of the EV at the previous CS is larger than the slope of the charging function associated with the arrival SoC of the EV at the current CS. Finally, we check whether the maximum-duration limit is exceeded by the optimized route.

---

**Algorithm 2:** ProcedureFRVCP($\Pi^0$)

    **input** : A route $\Pi^0 = \{\pi(0), \pi(1), \cdots, \pi(i), \cdots, \pi(j), \cdots, \pi(n(\Pi) - 1), \pi(n(\Pi))\}$

    **output:** A couple (f, $\Pi$) where the route $\Pi$ is an energy-feasible route whose duration does not exceed the duration limit
              $T_{max}$ if boolean f is equal to `true`, or the initial route $\Pi^0$ if boolean f is equal to `false`

**1**   Apply preprocessing rules on the multigraph induced by $\Pi$ and $F$

**2**   (f, $\Pi$, $\Pi^{best}$) ← `locateCS1`($\Pi^0$)

**3**   **if** f = true ∧ `getLBTime`($\Pi$) ≤ $T_{max}$ **then**

**4**       $\Pi^{best}$ ← `locateCS2`($\Pi$, $\Pi^{best}$)

**5**       **if** $\Pi^{best} \neq$ null **then**

**6**           (f, $\Pi^{best}$) ← `reviseChargingAmount`($\Pi^{best}$)

**7**           **if** *f=true* **then**

**8**              **return** (`true`, $\Pi^{best}$)

**9**           **end**

**10**     **end**

**11**   **end**

**12**   **return** (`false`, $\Pi^0$)

---

# 6 Computational experiments

We used Gurobi 7.5.0 (through its Java API) to solve the MILP models. All tests were performed on a single thread with 12 GB and on a cluster of 27 computers, each having 12 cores and two Intel(R) Xeon X5675 3.07 GHz processors. In all the tables, the CPU time is rounded to the nearest integer.

The first aim of our computational experiments is to assess and compare the performance of the CS replication-based and the path-based models, which are presented in Sections 3 and 4, for solving small size instances to the E-VRP-NL. These results are presented in §6.1. The second aim of our computational experiments is to assess the quality of the algorithms developed in Section 5 for the FRVCP. These results are presented in §6.2.

## 6.1 Results for E-VRP-NL formulations

For the E-VRP-NL we considered the twenty 10-customer and the twenty 20-customer instances of the Montoya et al. (2017) testbed (publicly available at `http://vrp-rep.org`). In these instances, both the energy consumption and travel times are obtained by multiplying the arc length by constant factors (consumption per unit distance and speed). It is worth noting that none of our results relies on the assumption that the energy consumption linearly depends on the travel time (this is only a characteristic of the instances). We ran the MILP models with a three-hour time limit.

### 6.1.1 CS replication-based models

We first tested the CS replication-based models $[F_{Node}^{CS\_rep}]$ and $[F_{Arc}^{CS\_rep}]$ presented in Section 3. The number of copies of each CS $i \in F$ was set to an integer value $\beta \geq 1$ (i.e., $\beta_i = \beta$, $\forall i \in F$).

Table 1 reports for each model and each value of $\beta$ the number of instances proven infeasible by the solver within the time limit (#Inf), the number of instances reaching the time limit without finding any feasible solution (#Unk), the number of instances with a feasible solution not proven to be optimal (#$\overline{\text{Opt}}$), the number of instances with a feasible solution proven to be optimal (#Opt), the average CPU time in seconds (Time (s)) for the instances in (#Opt), and the average gap (Gap) for the instances in (#$\overline{\text{Opt}}$). We compute the gap as $(z - z^{LB})/z$, where $z$ is the objective function value of the best integer solution returned by the solver, and $z^{LB}$ is the best lower bound retrieved by the solver running the corresponding model. The detailed results for all the tested instances are reported in F.

We first observe that for small values of $\beta$ the $[F_{Arc}^{CS\_rep}]$ model cannot always prove the infeasibility of some instances. This may be due to the relatively large number of variables it contains (around twice as many as the $[F_{Node}^{CS\_rep}]$ model for the tested instances). Second, the results confirm that the value of $\beta$ influences the feasibility of the instances. Since we do not know any procedure for fixing $\beta$, while guaranteeing optimality, we can only compare models for identical values of $\beta$. The best results are obtained with arc-based tracking constraints for which 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 has a better quality. Finally, our results illustrate the difficulty of optimally solving even small instances with the CS replication-based models.

Proposition 3.1 demonstrates that the LP relaxation value of $[F_{Arc}^{CS\_rep}]$ can never be less than that of $[F_{Node}^{CS\_rep}]$. In our experiments, we 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. The models with arc-based tracking variables yields tighter bounds than those with node-based tracking variables. Specifically, let $z_{Arc}^{LB}$ and $z_{Node}^{LB}$ correspond to the average LP relaxation of $[F_{Arc}^{CS\_rep}]$ and $[F_{Node}^{CS\_rep}]$. We observe that $(z_{Arc}^{LB} - z_{Node}^{LB})/z_{Arc}^{LB}$ is approximately 57%. This seems to explain why the solver yields a better performance running on arc-based models.

Table 1: Detailed computational results for the CS replication-based models on the 10-customer and 20-customer instances

| $|I|$ | $\beta$ | Model | #Inf | #Unk | #$\overline{\text{Opt}}$ | #Opt | Time (s) | Gap |
|---|---|---|---|---|---|---|---|---|
| 10 | 1 | $[F_{Arc}^{CS\_rep}]$ | 10 | 0 | 0 | 10 | 5 | - |
| | | $[F_{Node}^{CS\_rep}]$ | 10 | 0 | 0 | 10 | 252 | - |
| | 2 | $[F_{Arc}^{CS\_rep}]$ | 5 | 0 | 0 | 15 | 433 | - |
| | | $[F_{Node}^{CS\_rep}]$ | 5 | 0 | 4 | 11 | 699 | 31.8% |
| | 3 | $[F_{Arc}^{CS\_rep}]$ | 0 | 0 | 0 | 20 | 821 | - |
| | | $[F_{Node}^{CS\_rep}]$ | 0 | 0 | 8 | 12 | 1289 | 32.6% |
| | 4 | $[F_{Arc}^{CS\_rep}]$ | 0 | 0 | 4 | 16 | 215 | 14.4% |
| | | $[F_{Node}^{CS\_rep}]$ | 0 | 0 | 11 | 9 | 1423 | 34.0% |
| 20 | 1 | $[F_{Arc}^{CS\_rep}]$ | 8 | 1 | 5 | 6 | 632 | 16.4% |
| | | $[F_{Node}^{CS\_rep}]$ | 3 | 6 | 11 | 0 | 1274 | 24.8% |
| | 2 | $[F_{Arc}^{CS\_rep}]$ | 2 | 4 | 8 | 6 | 1323 | 8.8% |
| | | $[F_{Node}^{CS\_rep}]$ | 1 | 5 | 13 | 1 | 5663 | 24.0% |
| | 3 | $[F_{Arc}^{CS\_rep}]$ | 2 | 3 | 10 | 5 | 2233 | 9.8% |
| | | $[F_{Node}^{CS\_rep}]$ | 0 | 5 | 14 | 1 | 8706 | 28.9% |
| | 4 | $[F_{Arc}^{CS\_rep}]$ | 0 | 5 | 10 | 5 | 1200 | 10.9% |
| | | $[F_{Node}^{CS\_rep}]$ | 0 | 5 | 15 | 0 | - | 27.0% |

### 6.1.2 Path-based model

We also tested the path-based model $[F_{Arc}^{Path}]$ presented in Section 4. We first assessed the impact of the dominance rule described in §4.2. Applying this rule reduces the average number of paths by a factor of two for the 10-customer instances (on average 647 paths were generated without the dominance rule versus 270 with it), and by a factor of five for the 20-customer instances (on average 6512 paths were generated without the dominance rule versus 1314 with it). This leads to more tractable instances: the average number of paths between each couple of non-CS nodes is approximately 2.4 and 3.2 for the 10-customer and 20-customer instances, respectively. Table 2 reports the same measures as in Table 1 for the path-based model. See F for the detailed results for each instance.

Table 2: Computational results for the path-based model on the 10-customer and 20-customer instances

| $|I|$ | #Inf | #Unk | #$\overline{\text{Opt}}$ | #Opt | Time (s) | Gap |
|---|---|---|---|---|---|---|
| 10 | 0 | 0 | 0 | 20 | 229 | - |
| 20 | 0 | 0 | 15 | 5 | 489 | 14.0% |

Our results show that the MILP solver can optimally solve all the 10-customer instances within an average time of seven minutes. The solver fails to solve most of the instances with 20 customers due to the large size of the models. Doubling the number of customers from 10 to 20 multiplies the number of paths by five (from 270 to 1314, on average).

Figure 5 compares the CS replication-based and path-based models. Specifically, the figure shows the number of instances optimally solved according to the solution time for the different models. We conclude that the MILP solver performs better on the path-based model than on the CS replication-based models. Moreover, we recall that using the path-based model ensures that optimal solutions are not cut off.

## 6.2 Results for FRVCP algorithms

Since the FRVCP problem is typically solved within local search algorithms, we wanted to compare the speed of the algorithms we developed and the quality of the routes they provide. We tested the three algorithms described in Section 5 to solve the FRVCP: our formulation $[FR^{Path}(\Pi)]$ with the MILP solver, our labeling algorithm, and our heuristic (referred to as MILP_PATH, LABEL, and HEURISTIC). As a basis of comparison,

|  |  |
|---|---|
| (a) 10-customer instances | (b) 20-customer instances |
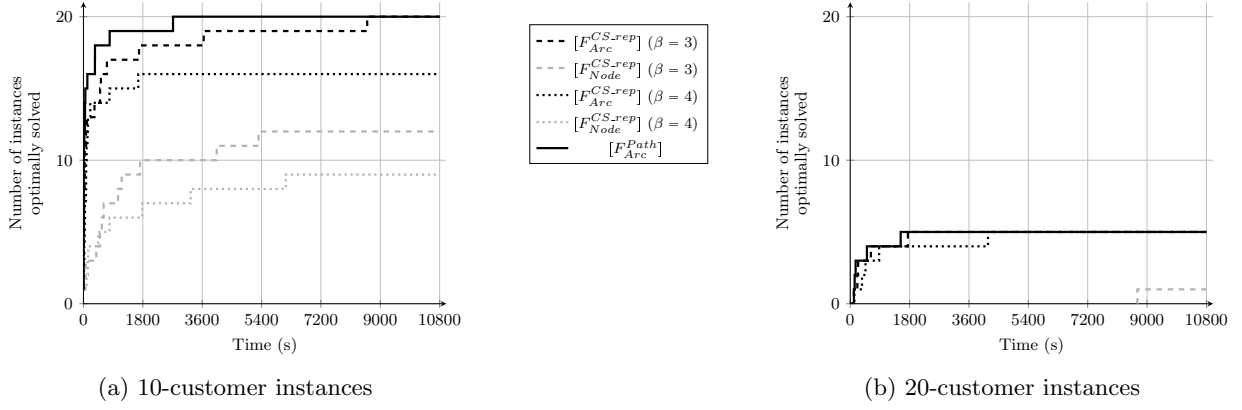
Figure 5: Performance charts of the different models on small size instances

we also implemented the FRVCP heuristic of Montoya et al. (2017) (referred to as HEURISTIC_M2017). Moreover, in order to assess the relevance of considering the insertion of multiple CSs between each pair of non-CS nodes of the initial route, we also solved the FRVCP-1. To this end, we adapted our path-based MILP formulation by only considering paths with at most one CS. We also adapted the graph on which we applied our labeling algorithm. These modifications yield a new MILP model and a new labeling algorithm (referred to as MILP_PATH_1 and LABEL_1) for this special version of the FRVCP. We also implemented the FRVCP formulation of Montoya et al. (2017) (referred to as MILP_M2017). No computational time limit was imposed on these tests.

We first tested the algorithms on the pool of routes used by Montoya et al. (2017) in the second phase of their metaheuristic. This pool consists of all routes encountered at each local optima solution found by an iterated local search procedure. The pool contains a total of 29,443 routes having between 3 and 15 nodes, which may visit between 2 and 24 CSs. For each algorithm, in Table 3 we report the number of routes for which the duration is below the duration limit $T_{max}$ (#Feas), the number of routes solved to optimality (#Opt), the average CPU time in milliseconds (Time (ms)), the average gap (Gap), and the maximum observed gap over all routes (Gap max).

Considering the detailed results presented in F.2 and F.3, we conclude that for all algorithms the number of CSs and the number of nodes have a relatively small impact on the solution time and quality. We note that the two exact algorithms for the FRVCP, i.e., MILP_PATH and LABEL, were able to solve all 29,443 routes to optimality within an average runtime of less than 100 milliseconds. LABEL is more than 90 times faster than MILP_PATH. HEURISTIC produces competitive results with a small average gap in a very short time. It also outperforms HEURISTIC_M2017 in terms of gap. However, both heuristics may fail to find existing feasible solutions. Such failures are due to the fact that feasibility often coincides with optimality. Since our heuristic found many more optimal solutions, compared to the FRVCP heuristic of Montoya et al. (2017), it found more feasible solutions. In particular, HEURISTIC failed to produce a feasible solution in 2.1% of the routes, whereas the HEURISTIC_M2017 failed to produce a feasible solution in 13.7% of the routes. Although LABEL outperforms HEURISTIC, the latter is easier to implement than the former which can make it attractive for some applications.

We observe that MILP_PATH_1 solves FRVCP-1 much faster than MILP_M2017. However, LABEL_1 substantially outperforms both of these formulations. Comparing FRVCP with FRVCP-1, we notice that restricting the insertion of at most one CS between any pair of non-CS nodes does not impact the quality of the solutions in most cases. Specifically, for 28,085 out of the 29,443 routes, the optimal objective function value considering either the FRVCP or the FRVCP-1 are the same. However, for the remaining 1358 routes, the charging decisions obtained by solving the FRVCP-1 are not optimal and in the worst case they can be quite far from the optimal ones. Specifically, let $z^*$ and $z_1^*$ correspond to the average optimal objective function

value obtained when solving the FRVCP and the FRVCP-1 on these 1358 routes. We observe that $(z_1^* - z^*)/z_1^*$ is approximately 0.01%. However, in the worst case (i.e., for a particular route), the deviation between the two optimal objective function values is 7.68%.

Table 3: Computational results for the different algorithms on 29,443 routes

| | FRVCP | | | | | FRVCP-1 | | | |
|---|---|---|---|---|---|---|---|---|---|
| Algorithm | #Feas | #Opt | Time (ms) | Gap | Gap max | Algorithm | #Feas | #Opt | Time (ms) |
| MILP_PATH | 29,443 | 29,443 | 96.7 | 0.00% | 0.00% | MILP_PATH_1 | 29,443 | 29,443 | 30.1 |
| LABEL | 29,443 | 29,443 | 1.0 | 0.00% | 0.00% | LABEL_1 | 29,443 | 29,443 | 0.6 |
| HEURISTIC | 28,815 | 23,792 | 0.2 | 0.19% | 14.00% | MILP_M2017 | 29,443 | 29,443 | 55.0 |
| HEURISTIC_M2017 | 25,398 | 19,104 | 0.2 | 0.83% | 36.32% | | | | |

We also tested our labeling algorithm of §5.3 on the 1426 routes making up the best solutions reported by Montoya et al. (2017) for 120 instances with 10, 20, 40, 80, 160, or 320 customers (20 instances for each instance size). We recall that the charging decisions for these routes were found using the FRVCP formulation of Montoya et al. (2017), which assumes at most one CS visit between two successive non-CS nodes. Therefore, we want to assess if the results can be improved by solving the more general version of the FRVCP, which allows multiple consecutive visits to CSs. We report in Table 4 the name of the instances for which a better solution has been identified, the number of routes (#R*) for which the charging decisions have been revised over the number of routes (#R) in the initial solution. We show the maximum gap improvement between a route that has been revised and the initial route (Max Gap R*). We also report the average gap with respect to the previous best known solution (Gap). We found 23 new best E-VRP-NL solutions. On these instances, taking optimal charging decisions improved the solutions of Montoya et al. (2017) by approximately 0.09%. This shows that charging decisions are critical in the E-VRP-NL. Therefore, there is ground to believe that the use of the labeling algorithm inside a metaheuristic algorithm deserves further study.

Table 4: E-VRP-NL solutions of (Montoya et al., 2017) improved by applying the labeling algorithm on each route

| Instance | #R*/#R | Max Gap R* | Gap | Instance | #R*/#R | Max Gap R* | Gap |
|---|---|---|---|---|---|---|---|
| tc0c40s8cf0 | 2/7 | -4.48% | -0.765% | tc1c160s24cf3 | 1/17 | -1.03% | -0.066% |
| tc2c40s5cf2 | 1/6 | -0.20% | -0.024% | tc1c320s24cf2 | 2/36 | -1.00% | -0.045% |
| tc2c40s8cf2 | 1/6 | -0.20% | -0.025% | tc1c320s24cf3 | 1/30 | -0.46% | -0.011% |
| tc0c80s8cf1 | 1/10 | -0.03% | -0.003% | tc1c320s38cf2 | 1/33 | -0.34% | -0.008% |
| tc1c80s12cf2 | 1/8 | -0.24% | -0.038% | tc1c320s38ct3 | 1/30 | -0.31% | -0.008% |
| tc2c80s8cf4 | 1/10 | -1.86% | -0.235% | tc2c320s24cf0 | 1/38 | -0.01% | -0.0004% |
| tc2c80s8ct3 | 1/9 | -0.08% | -0.009% | tc2c320s24ct4 | 1/32 | -2.57% | -0.096% |
| tc0c160s16cf4 | 3/18 | -0.43% | -0.074% | tc2c320s38cf4 | 1/32 | -0.19% | -0.006% |
| tc0c160s16ct4 | 3/18 | -0.31% | -0.052% | tc2c320s38ct0 | 2/41 | -0.31% | -0.006% |
| tc0c160s24cf4 | 3/18 | -0.44% | -0.076% | tc2c320s38ct1 | 1/28 | -0.02% | -0.001% |
| tc0c160s24ct4 | 4/18 | -3.37% | -0.197% | tc2c320s38ct4 | 1/32 | -0.09% | -0.004% |
| tc1c160s16cf3 | 1/17 | -5.64% | -0.347% | | | | |

The results presented in this section open an interesting research avenue. The only way to obtain an exact evaluation of a move in local search algorithms for the E-VRP-NL is to re-optimize the charging decisions (i.e., where and how much to charge) for the route(s) involved in the move. Specifically, one must decide the charging decisions such that each route has the lowest possible cost. This is a daunting task because it implies solving one (inter route moves) or more (intra route moves) FRVCPs. As Montoya et al. (2017) pointed out, exactly evaluating the moves using MILPs is intractable (our findings only confirm their observation). For this reason, most existing approaches use fast-to-compute proxy move evaluations. As a matter of fact, only the most sophisticated approaches use MILPs or exact algorithms at certain points (e.g., every time they reach a

local optimum) to improve the charging decisions. A notable exception is the multi-start local search heuristic proposed by Andelmin and Bartolini (2016) for the Green VRP. This heuristic works on a multigraph (where each arc represents a refuel path) avoiding the need to explicitly consider charging stations. Our results suggest that because of its good computational performance, our labeling algorithm may be used to exactly evaluate a large number of moves (if not all) in local search algorithms without penalizing the overall performance of the method.

# 7    Conclusions

We have compared three MILP formulations for the electric vehicle routing problem with nonlinear charging function (E-VRP-NL), two of which are new. The first formulation was proposed by Montoya et al. (2017), where tracking the time and the SoC of each route was achieved through the use of node-based variables. We have first proposed a new formulation that uses arc-based tracking variables. Computational experiments have shown that this alternative tracking strategy drastically improves the results. This can be explained by the fact that the arc-based formulation had a much tighter LP relaxation gap, compared to the node-based formulation. Most of the available E-VRP models, as well as the arc-based formulation, are based on pre-setting the number of CS copies. We have demonstrated that generating a limited number of copies compromises the quality of the solution by possibly eliminating optimal solutions or by yielding an infeasible model. With more copies, the models become larger and more difficult to solve, even for small-size instances. To overcome these drawbacks, we have proposed a third formulation for the E-VRP-NL based on the concept of CSPs between non-CS nodes. Using dominance rules to discard unpromising paths, we have shown through our computational experiments that this third formulation systematically yields better results than the first two formulations. All 10-customer instances could be solved to optimality within a much shorter time. As expected, solving our models using a commercial solver allows us to solve only small-size instances. However, most E-VRPs have been handled by node-based formulations, and for these cases the arc-based model provides a highly relevant alternative formulation. Furthermore, the introduction of the path-based model provides a new promising avenue for the development of exact and heuristic algorithms for the E-VRP-NL.

Building on the formulations we have developed for the E-VRP-NL, we have also introduced heuristic and exact algorithms to solve the fixed route vehicle charging problem (FRVCP). Given an initial energy infeasible route, this problem consists in finding the charging decisions that minimize the total duration of the route. We have proposed a new MILP formulation, an exact labeling algorithm, and a heuristic algorithm. Computational experiments have shown that taking optimal or near-optimal charging decisions is crucial in producing high quality E-VRP-NL solutions. Indeed, we have improved 23 out of 120 best known E-VRP-NL solutions by allowing the visit of multiple CSs between two non-CS nodes. Since the labeling algorithm is able to optimally solve the FRVCP within a very short time, it can be easily embedded in future E-VRP-NL heuristics.

# References

Andelmin, J. (2014). Electric vehicle routing with realistic recharging models. Master's thesis, - Department of Mathematics and Systems Analysis, Aalto University, Helsinki, Finland.

Andelmin, J. and Bartolini, E. (2016). A multi-start local search heuristic for the green vehicle routing problem based on a multigraph reformulation. Working paper available at `http://sal.aalto.fi/files/personnel/enrico.bartolini/green_heuristic_draft.pdf`.

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. *Transportation Science*, 51(4):1288–1303.

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., Ortuo, 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.

Hiermann, G., Hartl, R. F., Puchinger, J., and Vidal, T. (2018). Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*.

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

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

Kullman, N., Goodson, J., and Mendoza, J. E. (2018). Dynamic Electric Vehicle Routing with Mid-route Recharging and Uncertain Availability. In *ODYSSEUS 2018 - Seventh International Workshop on Freight Transportation and Logistics*, Cagliari, Italy.

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.

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

Montoya, A. (2016). *Electric Vehicle Routing Problems: models and solution approaches*. PhD thesis, Université dAngers, France.

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

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

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

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. (2017). *Logistics Networks with Intermediate Stops: Designing Innovative and Green Solutions*. PhD thesis, Rheinisch-Westfälische Technische Hochschule Aachen.

Schiffer, M. and Walther, G. (2017). An adaptive large neighborhood search for the location-routing problem with intra-route facilities. *Transportation Science*, 52(2):331–352.

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.

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

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

# A  Graph preprocessing

Some arcs from the arc set $A$ can be removed without cutting off the optimal solution. Specifically, we introduce the following preprocessing steps to reduce the size of the arc set:

1. Considering $i, j \in V$, if $\min_{l \in F \cup \{0\}}\{e_{li}\} + e_{ij} + \min_{l \in F \cup \{0\}}\{e_{il}\} > Q$ then we infer that the maximum possible SoC at $i$ the EV is not sufficient for traversing arc $(i, j)$ and reaching the nearest CS or the depot. Therefore, such an arc is removed from $A$.

2. Let $t_{ij}^{\star}$ be the duration of a shortest path from $i$ to $j$ in $G$ (after applying preprocessing rule 1) with respect to arc duration $t_{ij}$. Let $ec_{ij}^{min}$ be the minimum charging amount necessary to build a route traversing arc $(i, j)$. Then $ec_{ij}^{min} = \max\{0, (e_{0i} + e_{ij} + e_{j0}) - Q\}$. We also define $t_{ij}^{charge}$ as a lower bound on the 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

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

The value of $\rho^{*}$ corresponds to the steepest slope for a segment of the piecewise linear approximation of the charging functions (i.e., $\rho^{*} = \max_{l \in F}\{\rho_{l1}\}$). Using this parameter, we compute $t_{ij}^{charge}$ by exploring the 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 a violation of the route maximum-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 $A$.

# B Proofs

⋆ *Proof of Proposition 3.1.* Assume that the arc-based tracking constraints are satisfied by setting every variable $x_{ij}$, $y_{ij}$, $\underline{q}_j$, $\bar{q}_j$, $\tau_{ij}$, and $\Delta_j$ to $\boldsymbol{x_{ij}}$, $\boldsymbol{y_{ij}}$, $\boldsymbol{\underline{q}_j}$, $\boldsymbol{\bar{q}_j}$, $\boldsymbol{\tau_{ij}}$, and $\boldsymbol{\Delta_j}$, respectively. We now prove that the node-based tracking constraints are satisfied if: 1) the variables $x_{ij}$ are set to values $\boldsymbol{x_{ij}}$ for every $(i,j) \in A$, 2) the variables $\underline{q}_j$, $\bar{q}_j$, $\Delta_j$ are set to values $\boldsymbol{\underline{q}_j}$, $\boldsymbol{\bar{q}_j}$, $\boldsymbol{\Delta_j}$ for every $j \in F$, and 3) the variables $y_i$, $\tau_i$ are set $\sum_{(i,l)\in A} \boldsymbol{y_{il}}$ and $\sum_{(i,l)\in A} \boldsymbol{\tau_{il}}$ for every $i \in V\backslash\{0\}$. We also set the value of $\tau_0$ to $\boldsymbol{0}$ and the value of $y_0$ to $\boldsymbol{Q}$.

- Let $(i,j) \in A$ such that $j \in I$. Then,

    - if $i \in V\backslash\{0\}$,

$$y_i - y_j = \sum_{(i,l)\in A} \boldsymbol{y_{il}} - \sum_{(j,l)\in A} \boldsymbol{y_{jl}}$$

$$y_i - y_j = \sum_{(i,l)\in A} \boldsymbol{y_{il}} - \sum_{(l,j)\in A} (\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}}) \qquad \text{from constraints (39)}$$

$$y_i - y_j = e_{ij}\boldsymbol{x_{ij}} + \sum_{(i,l)\in A:l\neq j} \boldsymbol{y_{il}} - \sum_{(l,j)\in A:l\neq i} (\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}})$$

$$y_i - y_j \leqslant e_{ij}\boldsymbol{x_{ij}} + \sum_{(i,l)\in A:l\neq j} \boldsymbol{y_{il}}$$

$$y_i - y_j \leqslant e_{ij}\boldsymbol{x_{ij}} + \sum_{(i,l)\in A:l\neq j} (Q - \min_{l'\in F\cup\{0\}}\{e_{l'i}\})\boldsymbol{x_{il}} \qquad \text{from constraints (42)}$$

$$y_i - y_j \leqslant e_{ij}\boldsymbol{x_{ij}} + Q \sum_{(i,l)\in A:l\neq j} \boldsymbol{x_{il}}$$

$$y_i - y_j \leqslant e_{ij}\boldsymbol{x_{ij}} + Q(1 - \boldsymbol{x_{ij}}) \qquad \text{from constraints (2)}$$

and

$$y_i - y_j = \sum_{(i,l)\in A} \boldsymbol{y_{il}} - \sum_{(j,l)\in A} \boldsymbol{y_{jl}}$$

$$y_i - y_j = \sum_{(i,l)\in A} \boldsymbol{y_{il}} - \sum_{(l,j)\in A} (\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}}) \qquad \text{from constraints (39)}$$

$$y_i - y_j = e_{ij}\boldsymbol{x_{ij}} - \sum_{(l,j)\in A:l\neq i} \boldsymbol{y_{lj}} + \sum_{(i,l)\in A:l\neq j} \boldsymbol{y_{il}} + \sum_{(l,j)\in A:l\neq i} e_{lj}\boldsymbol{x_{lj}}$$

$$y_i - y_j \geqslant e_{ij}\boldsymbol{x_{ij}} - \sum_{(l,j)\in A:l\neq i} \boldsymbol{y_{lj}}$$

$$y_i - y_j \geqslant e_{ij}\boldsymbol{x_{ij}} - \sum_{(l,j)\in A:l\neq i} (Q - \min_{l'\in F\cup\{0\}}\{e_{l'l}\})\boldsymbol{x_{lj}} \qquad \text{from constraints (42)}$$

$$y_i - y_j \geqslant e_{ij}\boldsymbol{x_{ij}} - Q \sum_{(l,j)\in A:l\neq i} \boldsymbol{x_{lj}}$$

$$y_i - y_j \geqslant e_{ij}\boldsymbol{x_{ij}} - Q(1 - \boldsymbol{x_{ij}}). \qquad \text{from constraints (2)}$$

    - if $i = 0$,

$$y_0 - y_j = Q - \sum_{(j,l)\in A} \boldsymbol{y_{jl}}$$

$$y_0 - y_j = Q - \sum_{(l,j)\in A} (\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}}) \qquad \text{from constraints (39)}$$

$$y_0 - y_j = Q + e_{0j}\boldsymbol{x_{0j}} - \boldsymbol{y_{0j}} - \sum_{(l,j)\in A:l\neq 0} (\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}})$$

30

$$y_0 - y_j \leqslant e_{0j}\boldsymbol{x_{0j}} + Q - Q\boldsymbol{x_{0j}} \qquad\qquad \text{from constraints (38)}$$

$$y_0 - y_j \leqslant e_{0j}\boldsymbol{x_{0j}} + Q(1 - \boldsymbol{x_{0j}}) \qquad\qquad \text{from constraints (2)}$$

and

$$y_0 - y_j = Q - \sum_{(j,l)\in A} \boldsymbol{y_{jl}}$$

$$y_0 - y_j = Q - \sum_{(l,j)\in A} (\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}}) \qquad\qquad \text{from constraints (39)}$$

$$y_0 - y_j = Q + e_{0j}\boldsymbol{x_{0j}} - \sum_{(l,j)\in A} \boldsymbol{y_{lj}} + \sum_{(l,j)\in A:l\neq 0} e_{lj}\boldsymbol{x_{lj}}$$

$$y_0 - y_j \geqslant Q + e_{0j}\boldsymbol{x_{0j}} - Q \sum_{(l,j)\in A} \boldsymbol{x_{lj}} \qquad\qquad \text{from constraints (42)}$$

$$y_0 - y_j \geqslant Q + e_{0j}\boldsymbol{x_{0j}} - Q \qquad\qquad \text{from constraints (2)}$$

$$y_0 - y_j \geqslant e_{0j}\boldsymbol{x_{0j}}$$

$$y_0 - y_j \geqslant e_{0j}\boldsymbol{x_{0j}} - Q(1 - \boldsymbol{x_{0j}}).$$

Constraints (5) are therefore satisfied.

- Let $(i,j) \in A$ such that $j \in F'$. Then,

  - if $i \in V\backslash\{0\}$,

  $$y_i - \underline{q}_j = \sum_{(i,l)\in A} \boldsymbol{y_{il}} - \sum_{(l,j)\in A} (\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}}) \qquad\qquad \text{from constraints (40)}$$

  $$y_i - \underline{q}_j = e_{ij}\boldsymbol{x_{ij}} + \sum_{(i,l)\in A:l\neq j} \boldsymbol{y_{il}} - \sum_{(l,j)\in A:l\neq i} (\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}})$$

  $$y_i - \underline{q}_j \leqslant e_{ij}\boldsymbol{x_{ij}} + \sum_{(i,l)\in A:l\neq j} \boldsymbol{y_{il}}$$

  $$y_i - \underline{q}_j \leqslant e_{ij}\boldsymbol{x_{ij}} + Q \sum_{(i,l)\in A:l\neq j} \boldsymbol{x_{il}} \qquad\qquad \text{from constraints (42)}$$

  $$y_i - \underline{q}_j \leqslant e_{ij}\boldsymbol{x_{ij}} + Q(1 - \boldsymbol{x_{ij}}) \qquad\qquad \text{from constraints (3)}$$

  and

  $$y_i - \underline{q}_j = \sum_{(i,l)\in A} \boldsymbol{y_{il}} - \sum_{(l,j)\in A} (\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}}) \qquad\qquad \text{from constraints (40)}$$

  $$y_i - \underline{q}_j = e_{ij}\boldsymbol{x_{ij}} - \sum_{(l,j)\in A:l\neq i} \boldsymbol{y_{lj}} + \sum_{(i,l)\in A:l\neq j} \boldsymbol{y_{il}} + \sum_{(l,j)\in A:l\neq i} e_{lj}\boldsymbol{x_{lj}}$$

  $$y_i - \underline{q}_j \geqslant e_{ij}\boldsymbol{x_{ij}} - \sum_{(l,j)\in A:l\neq i} \boldsymbol{y_{lj}}$$

  $$y_i - \underline{q}_j \geqslant e_{ij}\boldsymbol{x_{ij}} - Q \sum_{(l,j)\in A:l\neq i} \boldsymbol{x_{lj}} \qquad\qquad \text{from constraints (42)}$$

  $$y_i - \underline{q}_j \geqslant e_{ij}\boldsymbol{x_{ij}} - Q(1 - \boldsymbol{x_{ij}}). \qquad\qquad \text{from constraints (3)}$$

  - if $i = 0$,

  $$y_0 - \underline{q}_j = Q - \sum_{(l,j)\in A} (\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}}) \qquad\qquad \text{from constraints (40)}$$

$$y_0 - \underline{q}_j = Q + e_{0j}\boldsymbol{x_{0j}} - \boldsymbol{y_{0j}} - \sum_{(l,j)\in A:l\neq 0}(\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}})$$

$$y_0 - \underline{q}_j \leqslant e_{0j}\boldsymbol{x_{0j}} + Q - Q\boldsymbol{x_{0j}} \qquad\qquad \text{from constraints (38)}$$

$$y_0 - \underline{q}_j \leqslant e_{0j}\boldsymbol{x_{0j}} + Q(1 - \boldsymbol{x_{0j}}) \qquad\qquad \text{from constraints (3)}$$

and

$$y_0 - \underline{q}_j = Q - \sum_{(l,j)\in A}(\boldsymbol{y_{lj}} - e_{lj}\boldsymbol{x_{lj}}) \qquad\qquad \text{from constraints (40)}$$

$$y_0 - \underline{q}_j = Q + e_{0j}\boldsymbol{x_{0j}} - \sum_{(l,j)\in A}\boldsymbol{y_{lj}} + \sum_{(l,j)\in A:l\neq 0}e_{lj}\boldsymbol{x_{lj}}$$

$$y_0 - \underline{q}_j \geqslant Q + e_{0j}\boldsymbol{x_{0j}} - Q\sum_{(l,j)\in A}\boldsymbol{x_{lj}} \qquad\qquad \text{from constraints (42)}$$

$$y_0 - \underline{q}_j \geqslant Q + e_{0j}\boldsymbol{x_{0j}} - Q \qquad\qquad \text{from constraints (3)}$$

$$y_0 - \underline{q}_j \geqslant e_{0j}\boldsymbol{x_{0j}}$$

$$y_0 - \underline{q}_j \geqslant e_{0j}\boldsymbol{x_{0j}} - Q(1 - \boldsymbol{x_{0j}}).$$

Constraints (6) are therefore satisfied.

- Let $i \in F'$, then

$$y_i = \sum_{(i,l)\in A}\boldsymbol{y_{il}}$$

$$y_i = \underline{q}_i. \qquad\qquad \text{from constraints (41)}$$

Constraints (8) are therefore satisfied.

- Let $(i,0) \in A$, then

$$y_i = \sum_{(i,l)\in A}\boldsymbol{y_{il}}$$

$$y_i \geqslant \boldsymbol{y_{i0}}$$

$$y_i \geqslant e_{i0}\boldsymbol{x_{i0}}. \qquad\qquad \text{from constraints (49)}$$

Constraints (7) are therefore satisfied.

- Let $(i,j) \in A$ such that $j \in I$. Then,

  − if $i \in V\backslash\{0\}$,

$$\tau_j - \tau_i = \sum_{(j,l)\in A}\boldsymbol{\tau_{jl}} - \sum_{(i,l)\in A}\boldsymbol{\tau_{il}}$$

$$\tau_j - \tau_i = \sum_{(l,j)\in A}(\boldsymbol{\tau_{lj}} + (t_{lj} + g_j)\,\boldsymbol{x_{lj}}) - \sum_{(i,l)\in A:}\boldsymbol{\tau_{il}} \qquad\qquad \text{from constraints (43)}$$

$$\tau_j - \tau_i = (t_{ij} + g_j)\,\boldsymbol{x_{ij}} + \sum_{(l,j)\in A:l\neq i}(\boldsymbol{\tau_{lj}} + (t_{lj} + g_j)\,\boldsymbol{x_{lj}}) - \sum_{(i,l)\in A:l\neq j}\boldsymbol{\tau_{il}}$$

$$\tau_j - \tau_i \geqslant (t_{ij} + g_j)\,\boldsymbol{x_{ij}} - \sum_{(i,l)\in A:l\neq j}\boldsymbol{\tau_{il}}$$

$$\tau_j - \tau_i \geqslant (t_{ij} + g_j)\,\boldsymbol{x_{ij}} - T_{max}\sum_{(i,l)\in A:l\neq j}\boldsymbol{x_{il}} \qquad\qquad \text{from constraints (45)}$$

$$\tau_j - \tau_i \geqslant (t_{ij} + g_j)\,\boldsymbol{x_{ij}} - T_{max}(1 - \boldsymbol{x_{ij}}). \qquad\qquad \text{from constraints (2)}$$

– if $i = 0$,

$$\tau_j - \tau_0 = \sum_{(j,l) \in A} \boldsymbol{\tau_{jl}}$$

$$\tau_j - \tau_0 = \sum_{(l,j) \in A} \left( \boldsymbol{\tau_{lj}} + (t_{lj} + g_j) \, \boldsymbol{x_{lj}} \right) \qquad \text{from constraints (43)}$$

$$\tau_j - \tau_0 = (t_{0j} + g_j) \, \boldsymbol{x_{0j}} + \boldsymbol{\tau_{0j}} + \sum_{(l,j) \in A: l \neq 0} \left( \boldsymbol{\tau_{lj}} + (t_{lj} + g_j) \, \boldsymbol{x_{lj}} \right)$$

$$\tau_j - \tau_0 \geqslant (t_{0j} + g_j) \, \boldsymbol{x_{0j}}$$

$$\tau_j - \tau_0 \geqslant (t_{0j} + g_j) \, \boldsymbol{x_{0j}} - T_{max}(1 - \boldsymbol{x_{0j}}). \qquad \text{from constraints (2)}$$

Constraints (26) are therefore satisfied.

- Let $(i, j) \in A$ such that $j \in F'$. Then,

    – if $i \in V \backslash \{0\}$,

$$\tau_j - \tau_i = \sum_{(j,l) \in A} \boldsymbol{\tau_{jl}} - \sum_{(i,l) \in A} \boldsymbol{\tau_{il}}$$

$$\tau_j - \tau_i = \sum_{(l,j) \in A} \left( \boldsymbol{\tau_{lj}} + t_{lj} \boldsymbol{x_{lj}} \right) + \boldsymbol{\Delta_j} - \sum_{(i,l) \in A} \boldsymbol{\tau_{il}} \qquad \text{from constraints (44)}$$

$$\tau_j - \tau_i = t_{ij} \boldsymbol{x_{ij}} + \boldsymbol{\Delta_j} + \sum_{(l,j) \in A: l \neq i} \left( \boldsymbol{\tau_{lj}} + t_{lj} \boldsymbol{x_{lj}} \right) - \sum_{(i,l) \in A: l \neq j} \boldsymbol{\tau_{il}}$$

$$\tau_j - \tau_i \geqslant t_{ij} \boldsymbol{x_{ij}} + \boldsymbol{\Delta_j} - \sum_{(i,l) \in A: l \neq j} \boldsymbol{\tau_{il}}$$

$$\tau_j - \tau_i \geqslant t_{ij} \boldsymbol{x_{ij}} + \boldsymbol{\Delta_j} - T_{max} \sum_{(i,l) \in A: l \neq j} \boldsymbol{x_{il}} \qquad \text{from constraints (45)}$$

$$\tau_j - \tau_i \geqslant t_{ij} \boldsymbol{x_{ij}} + \boldsymbol{\Delta_j} - T_{max}(1 - \boldsymbol{x_{ij}}) \qquad \text{from constraints (2)}$$

$$\tau_j - \tau_i \geqslant t_{ij} \boldsymbol{x_{ij}} + \boldsymbol{\Delta_j} - (T_{max} + S_{max})(1 - \boldsymbol{x_{ij}}).$$

    – if $i = 0$,

$$\tau_j - \tau_0 = \sum_{(j,l) \in A} \boldsymbol{\tau_{jl}}$$

$$\tau_j - \tau_0 = \sum_{(l,j) \in A} \left( \boldsymbol{\tau_{lj}} + t_{lj} \boldsymbol{x_{lj}} \right) + \boldsymbol{\Delta_j} \qquad \text{from constraints (44)}$$

$$\tau_j - \tau_0 = t_{0j} \boldsymbol{x_{0j}} + \boldsymbol{\Delta_j} + \boldsymbol{\tau_{0j}} + \sum_{(l,j) \in A: l \neq 0} \left( \boldsymbol{\tau_{lj}} + t_{lj} \boldsymbol{x_{lj}} \right)$$

$$\tau_j - \tau_0 \geqslant t_{ij} \boldsymbol{x_{0j}} + \boldsymbol{\Delta_j}$$

$$\tau_j - \tau_0 \geqslant t_{ij} \boldsymbol{x_{0j}} + \boldsymbol{\Delta_j} - (T_{max} + S_{max})(1 - \boldsymbol{x_{0j}}).$$

Constraints (27) are therefore satisfied.

- Let $j \in V \backslash \{0\}$. Then, if $j \in I$,

$$\tau_j + t_{j0} = \sum_{(j,l) \in A} \boldsymbol{\tau_{jl}} + t_{j0}$$

$$\tau_j + t_{j0} \leqslant \sum_{(j,l) \in A} \left( T_{max} - t_{jl} - p_l - t_{l0} \right) \boldsymbol{x_{jl}} + t_{j0}$$

33

$$\tau_j + t_{j0} \leqslant \sum_{(j,l) \in A} (T_{max} - t_{jl} - t_{l0})\, \boldsymbol{x_{jl}} + t_{j0} \qquad \text{from the triangular inequality}$$

$$\tau_j + t_{j0} \leqslant \sum_{(j,l) \in A} (T_{max} - t_{j0})\, \boldsymbol{x_{jl}} + t_{j0}$$

$$\tau_j + t_{j0} \leqslant T_{max} - t_{j0} + t_{j0} \qquad \text{from constraints (2)}$$

$$\tau_j + t_{j0} \leqslant T_{max}$$

and if $j \in F'$,

$$\tau_j + t_{j0} = \sum_{(j,l) \in A} \boldsymbol{\tau_{jl}} + t_{j0}$$

$$\tau_j + t_{j0} \leqslant \sum_{(j,l) \in A} \left(T_{max} - t_{jl} - t_{l0} - \Delta_j^{min}\right) \boldsymbol{x_{jl}} + t_{j0}$$

$$\tau_j + t_{j0} \leqslant \sum_{(j,l) \in A} (T_{max} - t_{j0})\, \boldsymbol{x_{jl}} + t_{j0} \qquad \text{from the triangular inequality}$$

$$\tau_j + t_{j0} \leqslant T_{max} - t_{j0} + t_{j0} \qquad \text{from constraints (2)}$$

$$\tau_j + t_{j0} \leqslant T_{max}.$$

Constraints (28) are therefore satisfied. By definition of the values $\bar{\tau}_j$, constraints (29) and (30) are also satisfied. The non-negativity constraints (33) are satisfied. Hence all the node-based tracking constraints are satisfied. $\qquad \square$

⋆ *Proof of Lemma 4.1.* Let us now consider a RP $(p, \phi)$ containing at least one CS. If the SoC of the EV upon departing from the origin is equal to $q'$ rather than $q \leqslant q'$, then the EV arrives at the first CS $\mu_p(0)$ with a SoC equal to $q' - e_{o(p), \mu_p(0)}$ rather than $q - e_{o(p), \mu_p(0)}$. The inverse of the SoC function $\mathrm{InvSoC}_{(p,\phi)}^q$ includes the time charging the EV from $q - e_{o(p), \mu_p(0)}$ up to $\phi(0)$, which is larger than or equal to $q' - e_{o(p), \mu_p(0)}$. Since the EV arrives with a larger SoC at $\mu_p(0)$ when starting with a SoC equal to $q'$, the time spent by the EV to reach a given SoC is reduced by the time spent at $\mu_p(0)$ charging the EV from $q - e_{o(p), \mu_p(0)}$ to $q' - e_{o(p), \mu_p(0)}$. This time is equal to $\lambda_{qq'}^p = \Phi_{\mu_p(0)}^{-1}(q' - e_{o(p), \mu_p(0)}) - \Phi_{\mu_p(0)}^{-1}(q - e_{o(p), \mu_p(0)})$. The inverse of the SoC function $\mathrm{InvSoC}_{(p,\phi)}^{q'}$ can therefore be built from the function $\mathrm{InvSoC}_{(p,\phi)}^q$ by subtracting $\lambda_{qq'}^p$ from the time needed to reach a given SoC. For a SoC that is smaller than the minimum SoC that can be reached at the destination, the time is consistently set (according to the definition of the inverse of the SoC function) to the time needed to reach that minimum.

Now consider a RP $(p, \phi)$ that does not contain any CS. The inverse of the SoC function is equal to $t^p$ if the SoC $\tilde{q}$ is less than or equal to the SoC of the EV upon arriving at the destination $d(p)$. Otherwise, the EV can never reached this SoC and therefore the value of the SoC function is set to $\infty$. $\qquad \square$

⋆ *Proof of Proposition 4.1.*

**Case 1:** If $n_{p_1} = 0$ or $\overline{\Theta}_{(p_1, \phi_1)} \geqslant \overline{\Theta}_{(p_2, \phi_2)}$, we have to consider two cases: $n_{p_1} > 0$ and $n_{p_1} = 0$ (note that $n_{p_2} > 0$ since we never check the dominance of a RP that does not contain any CS).

- **Case 1.1:** $n_{p_1} > 0$ $(q_1 = e_{o(p_1), \mu_{p_1}(0)}$ and $q_2 = e_{o(p_2), \mu_{p_2}(0)})$
  First, observe that if $q \in [\underline{Q}_{(p_2, \phi_2)}^{(p_1, \phi_1)}, \overline{Q}_{(p_2, \phi_2)}^{(p_1, \phi_1)}]$ is such that $\overline{T}_{(p_2, \phi_2)}^{(p_1, \phi_1)}(q, q_1, q_2) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

  $$\overline{T}_{(p_2, \phi_2)}^{(p_1, \phi_1)}(q, q_1, q_2) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$\Rightarrow \displaystyle\max_{\tilde{q}\in\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)}\{\mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q})-\mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q})\}\leqslant \Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

$\Rightarrow \forall\tilde{q}\in\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q),\quad \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q})-\mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q})\leqslant \Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

$\Rightarrow \forall\tilde{q}\in\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q),\quad \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q})-\Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})\leqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q})-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

$\Rightarrow \forall\tilde{q}\in\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q),\quad \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q})\leqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})\quad \text{from lemma 4.1}$

$\Rightarrow \forall\tilde{q}\in[\max\{0,q-\min\{e^{p_1},e^{p_2}\}\},\overline{\Theta}^{q_2}_{(p_2,\phi_2)}],\quad \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q})\leqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ \hfill (115)

Second, observe that if $q\in[\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)},\overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ is such that $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2)<\Delta_{p_1}(q)-\Delta_{p_2}(q)$, then we have:

$\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2)<\Delta_{p_1}(q)-\Delta_{p_2}(q)$

$\Rightarrow \displaystyle\min_{\tilde{q}\in\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)}\{\mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q})-\mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q})\}<\Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

$\Rightarrow \exists\tilde{q}'\in\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q),\quad \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}')-\mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}')<\Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

$\Rightarrow \exists\tilde{q}'\in\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q),\quad \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}')-\Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})<\mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}')-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

$\Rightarrow \exists\tilde{q}'\in\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q),\quad \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}')<\mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')\quad \text{from lemma 4.1}$

$\Rightarrow \exists\tilde{q}'\in[\max\{0,q-\min\{e^{p_1},e^{p_2}\}\},\overline{\Theta}^{q_2}_{(p_2,\phi_2)}],\quad \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}')<\mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$ \hfill (116)

Let $q\in[\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)},\min\{e^{p_1},e^{p_2}\}]$ be such that $\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2)\leqslant \Delta_{p_1}(q)-\Delta_{p_2}(q)$ and $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2)<\Delta_{p_1}(q)-\Delta_{p_2}(q)$. Equation (115) is equivalent to $\forall\tilde{q}\in[0,\overline{\Theta}_{(p_2,\phi_2)}]$, $\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q})\leqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ and Equation (116) is equivalent to $\exists\tilde{q}'\in[0,\overline{\Theta}_{(p_2,\phi_2)}]$, $\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}')<\mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. This means that $(p_1,\phi_1)>_q (p_2,\phi_2)$.

Let $q\in[\min\{e^{p_1},e^{p_2}\},\overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ be such that $\max\{\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2),\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,1)\}\leqslant \Delta_{p_1}(q)-\Delta_{p_2}(q)$ and $\min\{\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2),\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,1)\}<\Delta_{p_1}(q)-\Delta_{p_2}(q)$. We consider the two following subcases:

1. If $q$ is such that $\min\{e^{p_1},e^{p_2}\}<q\leqslant \max\{e^{p_1},e^{p_2}\}$, then

   – If $\min\{e^{p_1},e^{p_2}\}=e^{p_1}$, then

   $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,1)\leqslant \Delta_{p_1}(q)-\Delta_{p_2}(q)$

   $\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q-e^{p_1})-\mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(0)\leqslant \Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

   $\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q-e^{p_1})-\Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})\leqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(0)-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

   $\Rightarrow \forall\tilde{q}\in[0,q-e^{p_1}],\mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q-e^{p_1})-\Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})\leqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q})-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

   $\Rightarrow \forall\tilde{q}\in[0,q-e^{p_1}],\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q})\leqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})\quad \text{from lemma 4.1}$

   Since Equation (115) holds, we therefore have $\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q})\leqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q}\in[0,\overline{\Theta}_{(p_2,\phi_2)}]$. Observe that if $q$ is such that $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,1)<\Delta_{p_1}(q)-\Delta_{p_2}(q)$, then we have:

   $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,1)<\Delta_{p_1}(q)-\Delta_{p_2}(q)$

   $\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q-e^{p_1})-\mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q-e^{p_1})<\Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

   $\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q-e^{p_1})-\Phi^{-1}_{\mu_{p_1}(0)}(q-e_{o,\mu_{p_1}(0)})<\mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q-e^{p_1})-\Phi^{-1}_{\mu_{p_2}(0)}(q-e_{o,\mu_{p_2}(0)})$

$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \hfill (117)$

Since either Equation (116) or Equation (117) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$ such that $\text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_1, \phi_1) >_q (p_2, \phi_2)$.

– If $\min\{e^{p_1}, e^{p_2}\} = e^{p_2}$, then

$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$

$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_2}) - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) \leqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \leqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \leqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$

Since Equation (115) holds, we therefore have $\text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$. Observe that if $q$ is such that $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$

$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(0) - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) < \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(0) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \hfill (118)$

Since either Equation (116) or Equation (118) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$ such that $\text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_1, \phi_1) >_q (p_2, \phi_2)$.

2. If $q$ is such that $\max\{e^{p_1}, e^{p_2}\} < q \leqslant \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$, then

– If $\min\{e^{p_1}, e^{p_2}\} = e^{p_1}$, then

$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$

$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) \leqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \leqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \forall \tilde{q} \in [q - e^{p_2}, q - e^{p_1}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \leqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \forall \tilde{q} \in [q - e^{p_2}, q - e^{p_1}], \text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$

$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_1}], \text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q})$

Since Equation (115) holds, we therefore have $\text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$. Observe that if $q$ is such that $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$

$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_1}) < \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{119}$$

Since either Equation (116) or Equation (119) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$ such that $\mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_1,\phi_1) \succ_q (p_2,\phi_2)$.

– If $\min\{e^{p_1}, e^{p_2}\} = e^{p_2}$, then

$$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_2}) - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) \leqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \leqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in [q - e^{p_1}, q - e^{p_2}], \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \leqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in [q - e^{p_1}, q - e^{p_2}], \mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q})$$

Since Equation (115) holds, we therefore have $\mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$. Observe that if $q$ is such that $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) < \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) < \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) < \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{120}$$

Since either Equation (116) or Equation (120) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$ such that $\mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_1,\phi_1) \succ_q (p_2,\phi_2)$.

- **Case 1.2:** $n_{p_1} = 0$ $(q_2 = e_{o(p_2),\mu_{p_2}(0)})$
  First, observe that if $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ is such that $\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \max_{\tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)} \{t^{p_1} - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q})\} \leqslant -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad t^{p_1} - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}) \leqslant -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad t^{p_1} \leqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

$$\Rightarrow \forall \tilde{q} \in [\max\{0, q - \min\{e^{p_1}, e^{p_2}\}\}, \overline{\Theta}^{q_2}_{(p_2,\phi_2)}], \quad \mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}) \tag{121}$$

Second, observe that if $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ is such that $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \max_{\tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)} \{t^{p_1} - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q})\} < -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$\Rightarrow \exists \tilde{q}' \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad t^{p_1} - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') < -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \exists \tilde{q}' \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad t^{p_1} < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

$\Rightarrow \exists \tilde{q}' \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1}$

$\Rightarrow \exists \tilde{q}' \in [\max\{0, q - \min\{e^{p_1}, e^{p_2}\}\}, \overline{\Theta}^{q_2}_{(p_2,\phi_2)}], \quad \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad (122)$

Let $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \min\{e^{p_1}, e^{p_2}\}]$ be such that $\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$ and $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$. Equation (121) is equivalent to $\forall \tilde{q} \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$, $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ and Equation (122) is equivalent to $\exists \tilde{q}' \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$, $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. This means that $(p_1, \phi_1) \succ_q (p_2, \phi_2)$.

Let $q \in [\min\{e^{p_1}, e^{p_2}\}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ be such that $\max\{\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2), \overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1)\} \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$ and $\min\{\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2), \underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1)\} < \Delta_{p_1}(q) - \Delta_{p_2}(q)$. We consider the two following subcases:

1. If $q$ is such that $\min\{e^{p_1}, e^{p_2}\} \leqslant q \leqslant \max\{e^{p_1}, e^{p_2}\}$, then

   – If $\min\{e^{p_1}, e^{p_2}\} = e^{p_1}$, then

   $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$

   $\Rightarrow t^{p_1} - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(0) \leqslant -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

   $\Rightarrow t^{p_1} \leqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(0) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

   $\Rightarrow \forall \tilde{q} \in [0, q - e^{p_1}], \quad t^{p_1} \leqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

   $\Rightarrow \forall \tilde{q} \in [0, q - e^{p_1}], \quad \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$

   Since Equation (121) holds, we therefore have $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$. Observe that if $q$ is such that $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

   $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$

   $\Rightarrow t^{p_1} - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_1}) < -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

   $\Rightarrow t^{p_1} < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

   $\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \quad t^{p_1} < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

   $\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \quad \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \quad (123)$

   Since either Equation (122) or Equation (123) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$ such that $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_1, \phi_1) \succ_q (p_2, \phi_2)$.

   – If $\min\{e^{p_1}, e^{p_2}\} = e^{p_2}$, then

   $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$

   $\Rightarrow t^{p_1} - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) \leqslant -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

   $\Rightarrow t^{p_1} \leqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$

   $\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \quad \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$

   Since Equation (121) holds, we therefore have $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$. Observe that if $q$ is such that $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

   $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$

$$\Rightarrow t^{p_1} - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) < -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow t^{p_1} < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \quad t^{p_1} < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \quad \text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{124}$$

Since either Equation (122) or Equation (124) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$ such that $\text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_1, \phi_1) >_q (p_2, \phi_2)$.

2. If $q$ is such that $\max\{e^{p_1}, e^{p_2}\} \leqslant q \leqslant \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$, then

   – If $\min\{e^{p_1}, e^{p_2}\} = e^{p_1}$, then

   $$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

   $$\Rightarrow t^{p_1} - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) \leqslant -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

   $$\Rightarrow t^{p_1} \leqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

   $$\Rightarrow \forall \tilde{q} \in [q - e^{p_2}, q - e^{p_1}], \quad t^{p_1} \leqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

   $$\Rightarrow \forall \tilde{q} \in [q - e^{p_2}, q - e^{p_1}], \quad \text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

   $$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_1}], \quad \text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q})$$

   Since Equation (121) holds, we therefore have $\text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$. Observe that if $q$ is such that $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

   $$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

   $$\Rightarrow t^{p_1} - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_1}) < -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

   $$\Rightarrow t^{p_1} < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

   $$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \quad t^{p_1} < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

   $$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \quad \text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{125}$$

   Since either Equation (122) or Equation (125) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$ such that $\text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') < \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_1, \phi_1) >_q (p_2, \phi_2)$.

   – If $\min\{e^{p_1}, e^{p_2}\} = e^{p_2}$, then

   $$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) \leqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

   $$\Rightarrow t^{p_1} - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) \leqslant -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

   $$\Rightarrow t^{p_1} \leqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

   $$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \quad \text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

   Since Equation (121) holds, we therefore have $\text{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \leqslant \text{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$. Observe that if $q$ is such that $\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

   $$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 1) < \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

   $$\Rightarrow t^{p_1} - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) < -\Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

   $$\Rightarrow t^{p_1} < \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \quad t^{p_1} < \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \quad \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') < \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{126}$$

Since either Equation (122) or Equation (126) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_2,\phi_2)}]$ such that $\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') < \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_1, \phi_1) \succ_q (p_2, \phi_2)$.

**Case 2:** If $n_{p_2} = 0$ or $\overline{\Theta}_{(p_2,\phi_2)} \geqslant \overline{\Theta}_{(p_1,\phi_1)}$, we have to consider two cases: $n_{p_2} > 0$ and $n_{p_2} = 0$ (note that $n_{p_1} > 0$ since we never check the dominance of a RP that does not contain any CS).

- **Case 2.1:** $n_{p_2} > 0$ ($q_1 = e_{o(p_1),\mu_{p_1}(0)}$ and $q_2 = e_{o(p_2),\mu_{p_2}(0)}$)

First, observe that if $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ is such that $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Leftrightarrow \min_{\tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)} \{\mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q})\} \geqslant \Delta_{p_1}(q) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}) \geqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

$$\Rightarrow \forall \tilde{q} \in [\max\{0, q - \min\{e^{p_1}, e^{p_2}\}\}, \overline{\Theta}^{q_1}_{(p_1,\phi_1)}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \tag{127}$$

Second, observe that if $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ is such that $\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Leftrightarrow \max_{\tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)} \{\mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q})\} > \Delta_{p_1}(q) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') > \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1}$$

$$\Rightarrow \exists \tilde{q}' \in [\max\{0, q - \min\{e^{p_1}, e^{p_2}\}\}, \overline{\Theta}^{q_1}_{(p_1,\phi_1)}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \tag{128}$$

Let $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \min\{e^{p_1}, e^{p_2}\}]$ be such that $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$ and $\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$. Equation (127) is equivalent to $\forall \tilde{q} \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$, $\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ and Equation (128) is equivalent to $\exists \tilde{q}' \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$, $\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. This means that $(p_2, \phi_2) \succ_q (p_1, \phi_1)$.

Let $q \in [\min\{e^{p_1}, e^{p_2}\}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ be such that $\min\{\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2), \underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2)\} \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$ and $\max\{\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2), \overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2)\} > \Delta_{p_1}(q) - \Delta_{p_2}(q)$. We consider the two following sub-cases:

1. If $q$ is such that $\min\{e^{p_1}, e^{p_2}\} \leqslant q \leqslant \max\{e^{p_1}, e^{p_2}\}$, then

– If $\min\{e^{p_1}, e^{p_2}\} = e^{p_1}$, then

$$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_1}) \geqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_1}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_1}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

Since Equation (127) holds, we therefore have $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$. Observe that if $q$ is such that $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(0) > \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(0) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{129}$$

Since either Equation (128) or Equation (129) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$ such that $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_2, \phi_2) \succ_q (p_1, \phi_1)$.

– If $\min\{e^{p_1}, e^{p_2}\} = e^{p_2}$, then

$$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(0) - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) \geqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(0) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

Since Equation (127) holds, we therefore have $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$. Observe that if $q$ is such that $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(0) - \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) > \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(0) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > \text{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{130}$$

Since either Equation (128) or Equation (130) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$ such that $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_2, \phi_2) \succ_q (p_1, \phi_1)$.

2. If $q$ is such that $\max\{e^{p_1}, e^{p_2}\} \leqslant q \leqslant \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$, then

– If $\min\{e^{p_1}, e^{p_2}\} = e^{p_1}$, then

$$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_1}) \geqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in [q - e^{p_2}, q - e^{p_1}], \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in [q - e^{p_2}, q - e^{p_1}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_1}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$$

Since Equation (127) holds, we therefore have $\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$. Observe that if $q$ is such that $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) > \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [q - e^{p_2}, q - e^{p_1}], \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [q - e^{p_2}, q - e^{p_1}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1}$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \tag{131}$$

Since either Equation (128) or Equation (131) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$ such that $\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_2,\phi_2) \succ_q (p_1,\phi_1)$.

– If $\min\{e^{p_1}, e^{p_2}\} = e^{p_2}$, then

$$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) \geqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in [q - e^{p_1}, q - e^{p_2}], \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \forall \tilde{q} \in [q - e^{p_1}, q - e^{p_2}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$$

Since Equation (127) holds, we therefore have $\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$. Observe that if $q$ is such that $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_2}) - \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) > \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > \mathrm{InvSoC}^{q_2}_{(p_2,\phi_2)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_2}(0)}(q - e_{o,\mu_{p_2}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{132}$$

Since either Equation (128) or Equation (132) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$ such that $\mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_2,\phi_2) \succ_q (p_1,\phi_1)$.

• **Case 2.2:** $n_{p_2} = 0$ ($q_1 = e_{o(p_1),\mu_{p_1}(0)}$)

First, observe that if $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ is such that $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we

have:

$$\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Leftrightarrow \min_{\tilde{q}\in\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)} \{\text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - t^{p_2}\} \geqslant \Delta_{p_1}(q)$$

$$\Rightarrow \forall \tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - t^{p_2} \geqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)})$$

$$\Rightarrow \forall \tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant t^{p_2}$$

$$\Rightarrow \forall \tilde{q} \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

$$\Rightarrow \forall \tilde{q} \in [\max\{0, q - \min\{e^{p_1}, e^{p_2}\}\}, \overline{\Theta}^{q_1}_{(p_1,\phi_1)}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \tag{133}$$

Second, observe that if $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ is such that $\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Leftrightarrow \max_{\tilde{q}\in\Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q)} \{\text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - t^{p_2}\} > \Delta_{p_1}(q)$$

$$\Rightarrow \exists \tilde{q}' \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - t^{p_2} > \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)})$$

$$\Rightarrow \exists \tilde{q}' \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > t^{p_2}$$

$$\Rightarrow \exists \tilde{q}' \in \Omega^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q), \quad \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1}$$

$$\Rightarrow \exists \tilde{q}' \in [\max\{0, q - \min\{e^{p_1}, e^{p_2}\}\}, \overline{\Theta}^{q_1}_{(p_1,\phi_1)}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \tag{134}$$

Let $q \in [\underline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}, \min\{e^{p_1}, e^{p_2}\}]$ be such that $\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$ and $\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$. Equation (133) is equivalent to $\forall \tilde{q} \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$, $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ and Equation (134) is equivalent to $\exists \tilde{q}' \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$, $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. This means that $(p_2, \phi_2) \succ_q (p_1, \phi_1)$.

Let $q \in [\min\{e^{p_1}, e^{p_2}\}, \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}]$ be such that $\min\{\underline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2), \underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,2)\} \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$ and $\max\{\overline{T}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2), \overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,2)\} > \Delta_{p_1}(q) - \Delta_{p_2}(q)$. We consider the two following sub-cases:

1. If $q$ is such that $\min\{e^{p_1}, e^{p_2}\} \leqslant q \leqslant \max\{e^{p_1}, e^{p_2}\}$, then

   – If $\min\{e^{p_1}, e^{p_2}\} = e^{p_1}$, then

   $$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

   $$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - t^{p_2} \geqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)})$$

   $$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant t^{p_2}$$

   $$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_1}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant t^{p_2}$$

   $$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_1}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

   Since Equation (133) holds, we therefore have $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$. Observe that if $q$ is such that $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

   $$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q,q_1,q_2,2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

43

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - t^{p_2} > \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)})$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > t^{p_2}$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > t^{p_2}$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{135}$$

Since either Equation (134) or Equation (135) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$ such that $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_2,\phi_2) \succ_q (p_1,\phi_1)$.

- If $\min\{e^{p_1}, e^{p_2}\} = e^{p_2}$, then

$$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(0) - t^{p_2} \geqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)})$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(0) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant t^{p_2}$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant t^{p_2}$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

Since Equation (133) holds, we therefore have $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$. Observe that if $q$ is such that $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_2}) - t^{p_2} > \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)})$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > t^{p_2}$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > t^{p_2}$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{136}$$

Since either Equation (134) or Equation (136) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$ such that $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_2,\phi_2) \succ_q (p_1,\phi_1)$.

2. If $q$ is such that $\max\{e^{p_1}, e^{p_2}\} \leqslant q \leqslant \overline{Q}^{(p_1,\phi_1)}_{(p_2,\phi_2)}$, then

  - If $\min\{e^{p_1}, e^{p_2}\} = e^{p_1}$, then

$$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - t^{p_2} \geqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)})$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant t^{p_2}$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_1}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

Since Equation (133) holds, we therefore have $\text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$. Observe that if $q$ is such that $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - t^{p_2} > \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)})$$

$$\Rightarrow \text{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > t^{p_2}$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_1}], \text{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q})' > \text{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{137}$$

Since either Equation (134) or Equation (137) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$ such that $\mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_2, \phi_2) >_q (p_1, \phi_1)$.

- If $\min\{e^{p_1}, e^{p_2}\} = e^{p_2}$, then

$$\underline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) \geqslant \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - t^{p_2} \geqslant \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}))$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_1}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant t^{p_2}$$

$$\Rightarrow \forall \tilde{q} \in [q - e^{p_1}, q - e^{p_2}], \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) \geqslant t^{p_2}$$

$$\Rightarrow \forall \tilde{q} \in [q - e^{p_1}, q - e^{p_2}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}) \quad \text{from lemma 4.1}$$

$$\Rightarrow \forall \tilde{q} \in [0, q - e^{p_2}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q})$$

Since Equation (133) holds, we therefore have $\mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}) \geqslant \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q})$ for every $\tilde{q} \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$. Observe that if $q$ is such that $\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$, then we have:

$$\overline{U}^{(p_1,\phi_1)}_{(p_2,\phi_2)}(q, q_1, q_2, 2) > \Delta_{p_1}(q) - \Delta_{p_2}(q)$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_2}) - t^{p_2} > \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}))$$

$$\Rightarrow \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(q - e^{p_2}) - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > t^{p_2}$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \mathrm{InvSoC}^{q_1}_{(p_1,\phi_1)}(\tilde{q}') - \Phi^{-1}_{\mu_{p_1}(0)}(q - e_{o,\mu_{p_1}(0)}) > t^{p_2}$$

$$\Rightarrow \exists \tilde{q}' \in [0, q - e^{p_2}], \mathrm{InvSoC}^{q}_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^{q}_{(p_2,\phi_2)}(\tilde{q}') \quad \text{from lemma 4.1} \tag{138}$$

Since either Equation (134) or Equation (138) holds, there exists $\tilde{q}' \in [0, \overline{\Theta}_{(p_1,\phi_1)}]$ such that $\mathrm{InvSoC}^q_{(p_1,\phi_1)}(\tilde{q}') > \mathrm{InvSoC}^q_{(p_2,\phi_2)}(\tilde{q}')$. Therefore, we can conclude that $(p_2, \phi_2) >_q (p_1, \phi_1)$.

$\square$

# C   Example requiring a large number of CS copies

Figure 6 depicts an example in which a CS is visited $4|I|$ times in the optimal solution. In this example, we have one depot (denoted by 0), one customer (denoted by 1), and two CSs (denoted by 2 and 3). The service time at the customer is equal to 0.5, the battery capacity of the EVs is equal to 10, and $T_{max}$ is set to 40. The optimal solution to the E-VRP-NL contains one route $(0, 2, 3, 2, 1, 2, 3, 2, 0)$ and the total time is equal to 37.7. In this solution, the SoC of the EV upon arriving for the first time at CS 2 is equal to 1. To visit customer 1, the EV must necessarily leave CS 2 with a fully charged battery. Fully recharging the battery from 1 to full capacity at CS 2 takes 11.2 time units. However, by detouring to CS 3, the EV can obtain a fully charged battery in less time. Indeed, if the EV charges one energy unit at CS 2, 10 energy units at CS 3, and finally two energy units at CS 2, then the EV is fully charged in 11 time units (taking the round trip time to CS 3 into account). Moreover, after visiting customer 1, the EV arrives at CS 2 with an empty battery. To return to the depot, the EV must necessarily leave CS 2 with a SoC equal to 90%. Charging nine energy units at CS 2 takes in that case 10.4 time units. However, if the EV charges two energy units at CS 2, 10 energy units at CS 3, and finally one energy unit at CS 2, then the target SoC for the EV at CS 2 is obtained in 10.2 time units. In this example with only one customer, CS 2 is visited four times.

Figure 6: Example for which a CS is visited $4|I|$ times in the optimal solution
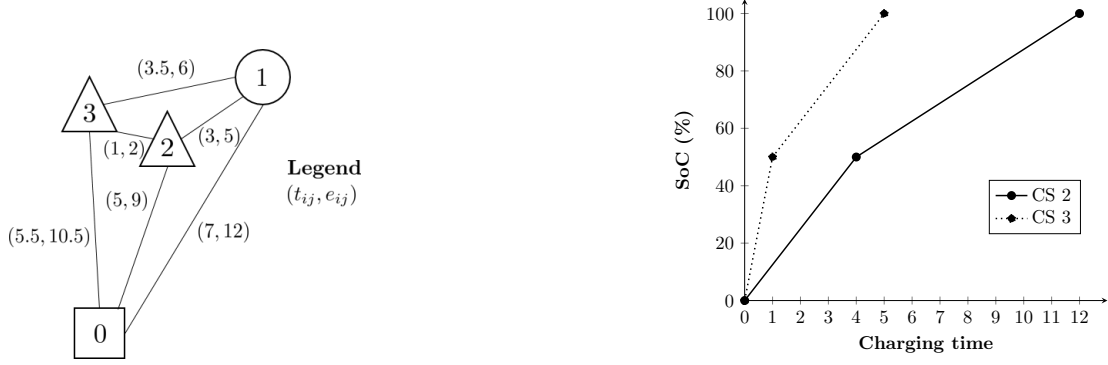


# D  Examples showing the non-optimality of the Montoya et al. (2017) procedure

Here we introduce two examples showing that the procedure proposed by Montoya et al. (2017) to set the number $\beta$ of CS copies is not necessarily optimal. We show this for both the case with nonlinear (example 1) and the case with linear (example 2) charging function approximations.

In the first example (denoted example 1), we have one depot (denoted by 0), two customers (denoted by 1 and 2), and one CS (denoted by 3). The service time at the customers is equal to 0.5, the battery capacity of the EVs is equal to 10 and $T_{max}$ is set to 20. Tables 5a and 5b show the driving time and energy consumption between the nodes. Figure 5c depicts the charging function at the CS.

Table 5: Data of example 1.

| $i\backslash j$ | 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 ($t_{ij}$).

| $i\backslash j$ | 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 ($e_{ij}$).
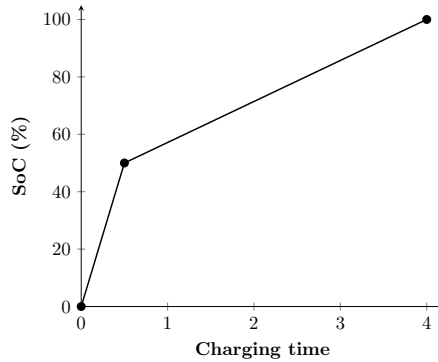


(c) Charging function associated with CS 3

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 $(0, 1, 0)$ and $(0, 2, 0)$ and the total time is equal to 17. According to the Montoya et al. (2017) procedure, $\beta$ will then be fixed to 0. However, for $\beta = 2$ (i.e., the CS can be visited at most twice), the optimal solution contains one route $(0, 2, 3, 1, 3, 0)$ and the total time is equal to 16.8. In this solution, the EV arrives at CS 3 with an empty battery. We charge the battery up to 50% and 30% during the first and

second visits, which takes 0.5 and 0.3 time units, respectively.

Now consider the case where the charging function is linear. We introduce another example (denoted as example 2). We have one depot (denoted by 0), four customers (denoted by 1, 2, 3, and 4), and one CS (denoted by 5). The service time at the customers is equal to 0.5, the battery capacity of the EVs is equal to 10 and $T_{max}$ is set to 10. Tables 6a and 6b show the driving times and energy consumption between the nodes. Figure 6c shows the charging function at the CS 5.

Table 6: Data of example 2.

| $i\backslash j$ | 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 $(t_{ij})$.

| $i\backslash j$ | 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 $(e_{ij})$.
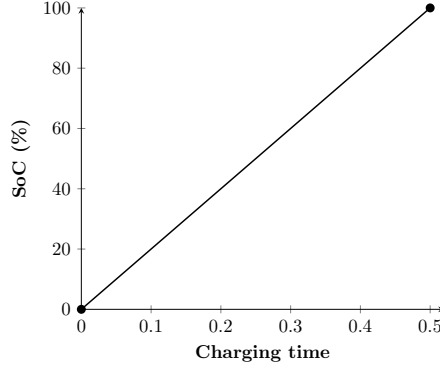


(c) Charging function associated with CS 5

For $\beta = 0$ and $\beta = 1$, the optimal solutions contains two routes $(0, 1, 4, 0)$ and $(0, 2, 3, 0)$ and the total time is equal to 10. According to the Montoya et al. (2017) procedure, $\beta$ will then be fixed to 0. However, for $\beta = 2$, the optimal solution contains one route $(0, 1, 5, 2, 3, 5, 4, 0)$ and the total time is equal to 9.65. At the first visit to the CS, the arrival SoC is 35% and we charge the battery up to 100%, which takes 0.325 time units. At the second visit to the CS, the arrival SoC of the EV is 0 and we charge the battery up to 65%, which takes 0.325 time units.

Note that if we assume a full charging policy and a constant charging time $\delta \geqslant 0$ 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 $9 + 2\delta$. Therefore a better solution is found as soon as $\delta < 0.5$.

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

# E    Algorithmic details

## E.1    Algorithmic details on the labeling algorithm for the FRVCP

---

**Algorithm 3:** `LabelingAlgo(Π)`

---

**input** : a route $\Pi$ that is not energy-feasible

**output:** a route $\Pi'$ that is energy-feasible if found, the initial route otherwise

**1** **forall** $l \in \Pi \cup \{F\}$ **do** $L_1(l) \leftarrow \varnothing$ *($L_1(l)$ stores the unprocessed labels associated with node $l$)* ;

**2** **forall** $l \in \Pi \cup \{F\}$ **do** $L_2(l) \leftarrow \varnothing$ *($L_2(l)$ stores the processed labels associated with node $l$)* ;

**3** Add to $L_1(\Pi(0))$ the label corresponding to a RP containing only the departing depot $\Pi(0)$

**4** $R \leftarrow \{\Pi(0)\}$ *($R$ stores nodes with unprocessed labels)*

**5** **while** $R \neq \varnothing$ **do**

**6** $\quad$ $l \leftarrow \texttt{minElement}(R)$ *(removes from $R$ the node associated with a RP $(p, \phi)$ with the minimum total time)*

**7** $\quad$ $(p_l, \phi_l) \leftarrow \texttt{deleteMin}(L_1(l))$ *(removes from $L_1(l)$ the label with the minimum total time)*

**8** $\quad$ **if** $l = \Pi(n(\Pi))$ **then**

**9** $\quad\quad$ **if** $p_l$ *contains a CS* **then**

**10** $\quad\quad\quad$ Set the SoC target value at the last CS of $p_l$ in such a way the EV arrives with an empty battery at $l$

**11** $\quad\quad$ **end**

**12** $\quad\quad$ Let $\Pi'$ be the route corresponding to label $(p_l, \phi_l)$

**13** $\quad\quad$ **return** $\Pi'$

**14** $\quad$ **end**

**15** $\quad$ **if** $L_1(l) = \varnothing$ **then** $R \leftarrow R\backslash\{l\}$ ;

**16** $\quad$ Update$(L_2(l), (p_l, \phi_l))$

**17** $\quad$ **forall** $(l, l') \in \hat{G}_{ij}$ **do**

**18** $\quad\quad$ **if** `IsExtendable`$((p_l, \phi_l), l')$ **then** *(checks if RP $(p, \phi)$ can be extended to node $l'$)*

**19** $\quad\quad\quad$ $(p_{l'}, \phi_{l'}) \leftarrow$`Extend`$((p_l, \phi_l), l')$ *(extends RP $(p_l, \phi_l)$ to node $l'$)*

**20** $\quad\quad\quad$ $K \leftarrow$`CreateLabels`$((p_{l'}, \phi_{l'}))$ *(if $l'$ is a CS and $p_{l'}$ contains more than a CS, then it creates relevant labels according to the supporting points of the SoC function associated with RP $(p_{l'}, \phi_{l'})$, otherwise $K$ only contains $(p_{l'}, \phi_{l'})$)*

**21** $\quad\quad\quad$ **forall** $(p, \phi) \in K$ **do**

**22** $\quad\quad\quad\quad$ `Update`$(L_1(l'), (p, \phi))$ *(inserts RP $(p, \phi)$ in $L_1(l')$ if it is not dominated according to the dominance rule introduced in Definition 4.1, and verifies if previously inserted labels become dominated)*

**23** $\quad\quad\quad$ **end**

**24** $\quad\quad\quad$ **if** $L_1(l')$ has changed and $l' \notin R$ **then** $R \leftarrow R \cup \{l'\}$ ;

**25** $\quad\quad$ **end**

**26** $\quad$ **end**

**27** **end**

**28** **return** $\Pi$

---

## E.2 Algorithmic details on the heuristic algorithm to solve the FRVCP

---

**Algorithm 4:** `locateCS1`$(\Pi^0)$

**input** : A route $\Pi^0 = \{\pi(0), \pi(1), \cdots, \pi(i), \cdots, \pi(n(\Pi) - 1), \pi(n(\Pi))\}$

**output:** (f, $\Pi$, $\Pi^{best}$) where $\Pi$ is the best route obtained if if boolean f is equal to `true`, or the initial route $\Pi^0$ otherwise. $\Pi^{best}$ represents the current best energy-feasible route with the minimum duration (under the H charging policy) that may have been computed during the procedure

1   $\Pi \leftarrow \Pi^0$

2   $\Pi^{best} \leftarrow$ `null`

3   $i \leftarrow 0, n \leftarrow n(\Pi)$

4   **while** $i < n$ **do**

5      **if** the arc connecting $\pi(i)$ and $\pi(i + 1)$ was eliminated during the preprocessing procedure **then**

6          $(f, \Pi, \Pi^{best}) \leftarrow \text{insertCS}(i, \Pi, \Pi^{best})$

7          **if** f = `false` **then**

8             **return** (`false`, $\Pi^0$, $\Pi^{best}$)

9          **end**

10         $n \leftarrow n + 1$

11      **else**

12          $i \leftarrow i + 1$

13      **end**

14   **end**

15   **return** (`true`, $\Pi$, $\Pi^{best}$)

---

---

**Algorithm 5:** `insertCS(`$i$`, `$\Pi^0$`, `$\Pi^{best}$`)`

---

**input** : A route $\Pi^0 = \{\pi(0), \pi(1), \cdots, \pi(i), \cdots, \pi(n-1), \pi(n)\}$

         A position $i$ where to insert a CS of set $F$

         The current best energy-feasible route $\Pi^{best}$

**output:** (f, $\Pi$, $\Pi^{best}$). $\Pi$ is the route yielding the minimum lower bound on the duration to travel it if f is equal to `true`, or the initial route $\Pi^0$ otherwise. $\Pi^{best}$ represents the current best energy-feasible route and may have been updated by the procedure

**1**    $\Pi \leftarrow \Pi^0$

**2**    $s \leftarrow$ `getEnergyDeficit(`$\Pi^0$`)`

**3**    $t^{best} \leftarrow \min\{T_{max}, $ `getTime(`$\Pi^{best}$`)`$\}$

**4**    $\tilde{t} \leftarrow +\infty$, f $\leftarrow$ `false`

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

**6**       **if** the arcs connecting $\pi(i)$ to $j$ and $j$ to $\pi(i+1)$ were not eliminated during the preprocess **then**

**7**           $\Pi' \leftarrow$ `copyAndInsert(`$\Pi^0$`, `$j$`, `$i$`)`

**8**           $s' \leftarrow$ `getEnergyDeficit(`$\Pi'$`)`

**9**           $\tilde{t}' \leftarrow$ `getLBTime(`$\Pi'$`)`

            ; /* Let $next(\Pi, j)$ be the first CS or depot node following node $j$ in route $\Pi$             */

**10**           **if** $\tilde{t}' < t^{best} \wedge s' < s \wedge \left( Y(\Pi', j) \geqslant 0 \vee Y(\Pi', next(\Pi', j)) \geqslant 0 \right)$ **then**

**11**              **if** $\tilde{t}' < \tilde{t}$ **then**

**12**                 $\tilde{t} \leftarrow \tilde{t}'$

**13**                 $\Pi \leftarrow \Pi'$

**14**                 f $\leftarrow$ true

**15**              **end**

**16**              **if** $s' = 0$ **then**

**17**                 $t' \leftarrow$ `getTime(`$\Pi'$`)`

**18**                 **if** $t' < t^{best}$ **then**

**19**                     $t^{best} \leftarrow t'$

**20**                     $\Pi^{best} \leftarrow \Pi'$

**21**                 **end**

**22**              **end**

**23**           **end**

**24**       **end**

**25**    **end**

**26**    **return** (f,$\Pi$,$\Pi^{best}$)

---

---

**Algorithm 6:** `locateCS2(`$\Pi^0$`, `$\Pi^{best}$`)`

**input** : A route $\Pi^0 = \{\pi(0), \pi(1), \cdots, \pi(i), \cdots, \pi(j), \cdots, \pi(n-1), \pi(n)\}$

The current best energy-feasible route $\Pi^{best}$

**output:** $\Pi^{best}$ where $\Pi^{best}$ represents the energy-feasible route with the minimum duration under the H charging rule

policy

1   $\Pi \leftarrow \Pi^0$

2   $t \leftarrow +\infty$

3   f $\leftarrow$ `false`

4   **while** true **do**

5      **for** $i = 0 \cdots n(\Pi) - 1$ **do**

6         (f', $\Pi'$, $\Pi^{best}$) $\leftarrow$ `insertCS(`$i$`,`$\Pi$`,`$\Pi^{best}$`)`

7         **if** f' = true **then**

8            f $\leftarrow$ true

9            $t' \leftarrow$ `getLBTime(`$\Pi'$`)`

10            **if** $t' < t$ **then**

11               $t \leftarrow t'$

12               $\overline{\Pi} \leftarrow \Pi'$

13            **end**

14         **end**

15      **end**

16      **if** f = true **then**

17         $\Pi \leftarrow \overline{\Pi}$

18      **else**

19         **return** $\Pi^{best}$

20      **end**

21   **end**

22   **return** null

---

---

**Algorithm 7:** `reviseChargingAmount(`$\Pi^0$`)`

**input** : A route $\Pi^0$ visiting $m$ CS(s) numbered in the order of visits from 1 to $m$ (let $f(i)$ be the $i$-th CS visited)

**output:** (f, $\Pi$) where $\Pi$ is an energy-feasible route derived from $\Pi^0$ after optimizing the charging decisions, and f is equal

to `true` if the maximum-duration limit is not exceeded, `false` otherwise.

1   $\Pi \leftarrow \Pi^0$

   ; /* Let $SoC_1(i)$, $SoC_2(i)$, and $ch(i)$ be the arrival and departure SoC and the amount of energy charged at the

   $i$-th CS visited                                                   */

   ; /* Let $b_1(i)$ and $b_2(i)$ be the segment associated with the SoC $SoC_1(i)$ and $SoC_2(i)$ for the $i$-th CS visited  */

2   $i \leftarrow m - 1$, $t \leftarrow$ `getTime(`$\Pi^{best}$`)`

3   **while** $i \geqslant 1$ **do**

4      **while** $\rho_{f(i),b_2(i)} > \rho_{f(i+1),b_1(i+1)} \wedge ch(i+1) > 0$ **do**

5         Transfer from the $(i+1)$-th to the $i-th$ CS of route $\Pi$ the amount of energy $\min\{ch(i+1), a_{f(i),b_2(i)} - SoC_2(i)\}$

6         Update accordingly $SoC_1(i+1), SoC_2(i), ch(i+1), b_2(i), b_1(i+1)$ and the duration $t$

7      **end**

8      $i \leftarrow i - 1$

9   **end**

10   **if** $t \leqslant T_{max}$ **then**

11      **return** (true, $\Pi$)

12   **end**

13   **return** (false, $\Pi^0$)

---

# F Detailed computational results

## F.1 E-VRP-NL

We represent each instance using the code $tc\gamma_1 c\gamma_2 s\gamma_3 c\gamma_4 \#$, where $\gamma_1$ is the method used to place the customers (i.e., 0) randomization, 1) combination of randomization and clustering, 2) clustering), $\gamma_2$ is the number of customers, $\gamma_3$ is the number of CSs, $\gamma_4$ 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 was proven infeasible, whereas the symbol "-" means that no feasible solution was found by the solver but the instance was not proven infeasible. The CPU time reported in column "Time (s)" in the following tables is in seconds and rounded to the nearest integer.

### F.1.1 CS replication-based models

Table 7: Detailed computational results on the 10-customer instances for the CS replication-based models ($\beta = 1$)

| Instance | $[F_{Arc}^{CS\_rep}]$ | | | | $[F_{Node}^{CS\_rep}]$ | | | |
|---|---|---|---|---|---|---|---|---|
| | Obj | Bound | LP Bound | Time (s) | Obj | Bound | LP Bound | Time (s) |
| tc0c10s2cf1 | Inf | Inf | Inf | 0 | Inf | Inf | 6.02 | 0 |
| tc0c10s2ct1 | 12.30 | 12.30 | 9.05 | 1 | 12.30 | 12.30 | 6.33 | 2 |
| tc0c10s3cf1 | Inf | Inf | 12.06 | 0 | Inf | Inf | 6.02 | 0 |
| tc0c10s3ct1 | 10.80 | 10.80 | 7.72 | 1 | 10.80 | 10.80 | 5.13 | 5 |
| tc1c10s2cf2 | 9.03 | 9.03 | 6.72 | 1 | 9.03 | 9.03 | 5.56 | 27 |
| tc1c10s2cf3 | Inf | Inf | Inf | 0 | Inf | Inf | 3.42 | 2 |
| tc1c10s2cf4 | Inf | Inf | Inf | 0 | Inf | Inf | 6.47 | 0 |
| tc1c10s2ct2 | 10.76 | 10.76 | 6.73 | 6 | 10.76 | 10.76 | 4.58 | 829 |
| tc1c10s2ct3 | Inf | Inf | 9.83 | 0 | Inf | Inf | 3.18 | 4 |
| tc1c10s2ct4 | Inf | Inf | Inf | 0 | Inf | Inf | 6.23 | 0 |
| tc1c10s3cf2 | 9.03 | 9.03 | 6.72 | 1 | 9.03 | 9.03 | 5.56 | 15 |
| tc1c10s3cf3 | Inf | Inf | Inf | 0 | Inf | Inf | 3.42 | 1 |
| tc1c10s3cf4 | 14.95 | 14.95 | 10.58 | 1 | 14.95 | 14.95 | 5.80 | 4 |
| tc1c10s3ct2 | 9.20 | 9.20 | 6.36 | 11 | 9.20 | 9.20 | 4.46 | 1107 |
| tc1c10s3ct3 | 13.02 | 13.02 | 7.14 | 4 | 13.02 | 13.02 | 3.16 | 1076 |
| tc1c10s3ct4 | 13.21 | 13.21 | 8.85 | 4 | 13.21 | 13.21 | 6.09 | 6 |
| tc2c10s2cf0 | Inf | Inf | 9.48 | 0 | Inf | Inf | 1.04 | 1 |
| tc2c10s2ct0 | Inf | Inf | 4.67 | 13 | Inf | Inf | 0.89 | 103 |
| tc2c10s3cf0 | Inf | Inf | 9.48 | 0 | Inf | Inf | 1.04 | 1 |
| tc2c10s3ct0 | 11.51 | 11.51 | 4.21 | 63 | 11.51 | 11.51 | 0.89 | 1859 |

Table 8: Detailed computational results on the 10-customer instances for the CS replication-based models ($\beta = 2$)

| Instance | [$F_{Arc}^{CS\text{-}rep}$] | | | | [$F_{Node}^{CS\text{-}rep}$] | | | |
|---|---|---|---|---|---|---|---|---|
| | Obj | Bound | LP Bound | Time (s) | Obj | Bound | LP Bound | Time (s) |
| tc0c10s2cf1 | Inf | Inf | 11.74 | 2 | Inf | Inf | 6.02 | 1 |
| tc0c10s2ct1 | 12.30 | 12.30 | 8.94 | 19 | 12.30 | 12.30 | 6.33 | 142 |
| tc0c10s3cf1 | 20.50 | 20.50 | 11.55 | 42 | 20.50 | 20.50 | 6.02 | 24 |
| tc0c10s3ct1 | 10.80 | 10.80 | 7.69 | 3 | 10.80 | 10.80 | 5.13 | 157 |
| tc1c10s2cf2 | 9.03 | 9.03 | 6.72 | 3 | 9.03 | 9.03 | 5.56 | 102 |
| tc1c10s2cf3 | Inf | Inf | 10.79 | 72 | Inf | Inf | 3.42 | 194 |
| tc1c10s2cf4 | 16.14 | 16.14 | 12.15 | 5 | 16.14 | 16.14 | 6.47 | 11 |
| tc1c10s2ct2 | 10.75 | 10.75 | 6.73 | 34 | 10.75 | 9.29 | 4.58 | 10800 |
| tc1c10s2ct3 | Inf | Inf | 8.52 | 38 | Inf | Inf | 3.18 | 82 |
| tc1c10s2ct4 | 13.83 | 13.83 | 10.68 | 5 | 13.83 | 13.83 | 6.23 | 9 |
| tc1c10s3cf2 | 9.03 | 9.03 | 6.72 | 3 | 9.03 | 9.03 | 5.56 | 184 |
| tc1c10s3cf3 | 16.94 | 16.94 | 10.71 | 81 | 16.94 | 16.94 | 3.42 | 5430 |
| tc1c10s3cf4 | 14.90 | 14.90 | 10.57 | 15 | 14.90 | 14.90 | 5.80 | 149 |
| tc1c10s3ct2 | 9.20 | 9.20 | 6.36 | 57 | 9.2 | 8.21 | 4.46 | 10800 |
| tc1c10s3ct3 | 13.02 | 13.02 | 7.07 | 36 | 13.08 | 6.80 | 3.16 | 10800 |
| tc1c10s3ct4 | 13.21 | 13.21 | 8.75 | 22 | 13.21 | 13.21 | 6.09 | 375 |
| tc2c10s2cf0 | Inf | Inf | 8.94 | 255 | Inf | Inf | 1.04 | 40 |
| tc2c10s2ct0 | 13.84 | 13.84 | 4.67 | 1165 | 13.84 | 13.84 | 0.89 | 3944 |
| tc2c10s3cf0 | Inf | Inf | 8.94 | 6156 | Inf | Inf | 1.04 | 340 |
| tc2c10s3ct0 | 11.51 | 11.51 | 4.2 | 654 | 11.51 | 5.17 | 0.89 | 10800 |

Table 9: Detailed computational results on the 10-customer instances for the CS replication-based models ($\beta = 3$)

| Instance | [$F_{Arc}^{CS\text{-}rep}$] | | | | [$F_{Node}^{CS\text{-}rep}$] | | | |
|---|---|---|---|---|---|---|---|---|
| | Obj | Bound | LP Bound | Time (s) | Obj | Bound | LP Bound | Time (s) |
| tc0c10s2cf1 | 19.75 | 19.75 | 11.74 | 21 | 19.75 | 19.75 | 6.02 | 9 |
| tc0c10s2ct1 | 12.30 | 12.30 | 8.94 | 21 | 12.30 | 12.30 | 6.33 | 389 |
| tc0c10s3cf1 | 19.75 | 19.75 | 11.55 | 1685 | 19.75 | 19.75 | 6.02 | 1718 |
| tc0c10s3ct1 | 10.80 | 10.80 | 7.69 | 9 | 10.80 | 10.80 | 5.13 | 500 |
| tc1c10s2cf2 | 9.03 | 9.03 | 6.72 | 4 | 9.03 | 9.03 | 5.56 | 559 |
| tc1c10s2cf3 | 16.37 | 16.37 | 10.79 | 39 | 16.37 | 16.37 | 3.42 | 5310 |
| tc1c10s2cf4 | 16.10 | 16.10 | 12.14 | 10 | 16.10 | 16.10 | 6.47 | 69 |
| tc1c10s2ct2 | 10.75 | 10.75 | 6.73 | 108 | 10.75 | 8.42 | 4.58 | 10800 |
| tc1c10s2ct3 | 13.17 | 13.17 | 8.48 | 11 | 13.17 | 13.17 | 3.18 | 1049 |
| tc1c10s2ct4 | 13.83 | 13.83 | 10.68 | 17 | 13.83 | 13.83 | 6.23 | 54 |
| tc1c10s3cf2 | 9.03 | 9.03 | 6.72 | 6 | 9.03 | 9.03 | 5.56 | 611 |
| tc1c10s3cf3 | 16.37 | 16.37 | 10.71 | 529 | 16.37 | 12.76 | 3.42 | 10800 |
| tc1c10s3cf4 | 14.90 | 14.90 | 10.57 | 37 | 14.90 | 14.90 | 5.80 | 1159 |
| tc1c10s3ct2 | 9.20 | 9.20 | 6.36 | 80 | 9.20 | 7.82 | 4.46 | 10800 |
| tc1c10s3ct3 | 13.02 | 13.02 | 7.07 | 504 | 13.02 | 4.90 | 3.16 | 10800 |
| tc1c10s3ct4 | 13.21 | 13.21 | 8.75 | 34 | 13.21 | 13.21 | 6.09 | 4040 |
| tc2c10s2cf0 | 22.12 | 22.12 | 8.94 | 3647 | 22.12 | 19.56 | 1.04 | 10800 |
| tc2c10s2ct0 | 12.45 | 12.45 | 4.67 | 334 | 12.45 | 7.14 | 0.89 | 10800 |
| tc2c10s3cf0 | 22.12 | 22.12 | 8.94 | 8604 | 22.12 | 16.95 | 1.04 | 10800 |
| tc2c10s3ct0 | 11.51 | 11.51 | 4.20 | 712 | 11.51 | 4.34 | 0.89 | 10800 |

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

| Instance | $[F_{Arc}^{CS\text{-}rep}]$ | | | | $[F_{Node}^{CS\text{-}rep}]$ | | | |
|---|---|---|---|---|---|---|---|---|
| | Obj | Bound | LP Bound | Time (s) | Obj | Bound | LP Bound | Time (s) |
| tc0c10s2cf1 | 19.75 | 19.75 | 11.74 | 96 | 19.75 | 19.75 | 6.02 | 25 |
| tc0c10s2ct1 | 12.30 | 12.30 | 8.94 | 81 | 12.30 | 12.30 | 6.33 | 133 |
| tc0c10s3cf1 | 19.75 | 17.68 | 11.55 | 10800 | 19.75 | 14.47 | 6.02 | 10800 |
| tc0c10s3ct1 | 10.80 | 10.80 | 7.69 | 37 | 10.80 | 10.80 | 5.13 | 445 |
| tc1c10s2cf2 | 9.03 | 9.03 | 6.72 | 12 | 9.03 | 9.03 | 5.56 | 791 |
| tc1c10s2cf3 | 16.37 | 16.37 | 10.79 | 82 | 16.37 | 13.86 | 3.42 | 10800 |
| tc1c10s2cf4 | 16.10 | 16.10 | 12.14 | 26 | 16.10 | 16.10 | 6.47 | 144 |
| tc1c10s2ct2 | 10.75 | 10.75 | 6.73 | 122 | 10.75 | 8.33 | 4.58 | 10800 |
| tc1c10s2ct3 | 13.17 | 13.17 | 8.48 | 41 | 13.17 | 13.17 | 3.18 | 6136 |
| tc1c10s2ct4 | 13.83 | 13.83 | 10.68 | 17 | 13.83 | 13.83 | 6.23 | 98 |
| tc1c10s3cf2 | 9.03 | 9.03 | 6.72 | 10 | 9.03 | 9.03 | 5.56 | 1784 |
| tc1c10s3cf3 | 16.37 | 16.37 | 10.71 | 1663 | 16.37 | 11.95 | 3.42 | 10800 |
| tc1c10s3cf4 | 14.90 | 14.90 | 10.57 | 76 | 14.90 | 14.90 | 5.80 | 3250 |
| tc1c10s3ct2 | 9.20 | 9.20 | 6.36 | 137 | 9.20 | 7.66 | 4.46 | 10800 |
| tc1c10s3ct3 | 13.02 | 13.02 | 7.07 | 207 | 13.08 | 5.28 | 3.16 | 10800 |
| tc1c10s3ct4 | 13.21 | 13.21 | 8.75 | 45 | 13.21 | 11.56 | 6.09 | 10800 |
| tc2c10s2cf0 | 21.77 | 19.62 | 8.94 | 10800 | 21.77 | 12.70 | 1.04 | 10800 |
| tc2c10s2ct0 | 12.45 | 12.45 | 4.67 | 789 | 12.45 | 6.32 | 0.89 | 10800 |
| tc2c10s3cf0 | 21.77 | 16.86 | 8.94 | 10800 | 21.83 | 11.98 | 1.04 | 10800 |
| tc2c10s3ct0 | 11.51 | 9.83 | 4.20 | 10800 | 11.56 | 4.90 | 0.89 | 10800 |

Table 11: Detailed computational results on the 20-customer instances for the CS replication-based models ($\beta = 1$)

| Instance | $[F_{Arc}^{CS\text{-}rep}]$ | | | | $[F_{Node}^{CS\text{-}rep}]$ | | | |
|---|---|---|---|---|---|---|---|---|
| | Obj | Bound | LP Bound | Time (s) | Obj | Bound | LP Bound | Time (s) |
| tc0c20s3cf2 | Inf | Inf | Inf | 0 | Inf | Inf | 6.89 | 61 |
| tc0c20s3ct2 | Inf | Inf | 11.65 | 460 | - | - | 6.96 | 10800 |
| tc0c20s4cf2 | Inf | Inf | Inf | 0 | Inf | Inf | 6.86 | 3763 |
| tc0c20s4ct2 | Inf | Inf | 10.89 | 10800 | - | - | 6.62 | 10800 |
| tc1c20s3cf1 | 20.75 | 17.76 | 14.00 | 10800 | 20.78 | 14.68 | 8.02 | 10800 |
| tc1c20s3cf3 | 16.80 | 16.10 | 9.81 | 10800 | 16.84 | 11.55 | 5.20 | 10800 |
| tc1c20s3cf4 | Inf | Inf | 13.21 | 18 | Inf | Inf | 9.52 | 0 |
| tc1c20s3ct1 | 24.42 | 19.23 | 14.85 | 10800 | 24.42 | 16.20 | 8.02 | 10800 |
| tc1c20s3ct3 | 12.60 | 12.60 | 9.35 | 189 | 12.67 | 10.12 | 5.45 | 10800 |
| tc1c20s3ct4 | 16.24 | 16.24 | 13.62 | 33 | 16.24 | 14.55 | 10.18 | 10800 |
| tc1c20s4cf1 | 19.29 | 15.82 | 11.85 | 10800 | 19.68 | 12.97 | 7.63 | 10800 |
| tc1c20s4cf3 | 16.80 | 16.80 | 9.81 | 6474 | 16.91 | 10.02 | 5.19 | 10800 |
| tc1c20s4cf4 | 19.09 | 19.09 | 13.18 | 226 | 19.09 | 18.15 | 9.52 | 10800 |
| tc1c20s4ct1 | 23.19 | 17.59 | 12.74 | 10800 | 23.19 | 14.54 | 7.65 | 10800 |
| tc1c20s4ct3 | 14.43 | 14.43 | 9.69 | 553 | 14.43 | 10.54 | 5.23 | 10800 |
| tc1c20s4ct4 | 17.14 | 17.14 | 13.00 | 51 | 17.14 | 16.41 | 10.08 | 10800 |
| tc2c20s3cf0 | Inf | Inf | 11.32 | 1 | - | - | 2.03 | 10800 |
| tc2c20s3ct0 | Inf | Inf | 11.56 | 12 | - | - | 1.95 | 10800 |
| tc2c20s4cf0 | Inf | Inf | 11.13 | 813 | - | - | 2.03 | 10800 |
| tc2c20s4ct0 | Inf | Inf | 11.43 | 18 | - | - | 1.94 | 10800 |

Table 12: Detailed computational results on the 20-customer instances for the CS replication-based models $(\beta = 2)$

| Instance | $[F_{Arc}^{CS\_rep}]$ | | | | $[F_{Node}^{CS\_rep}]$ | | | |
|---|---|---|---|---|---|---|---|---|
| | Obj | Bound | LP Bound | Time (s) | Obj | Bound | LP Bound | Time (s) |
| tc0c20s3cf2 | Inf | Inf | 14.89 | 4 | - | - | 6.89 | 10800 |
| tc0c20s3ct2 | 17.08 | 15.34 | 11.64 | 10800 | 17.31 | 12.45 | 6.96 | 10800 |
| tc0c20s4cf2 | Inf | Inf | 14.54 | 6 | Inf | Inf | 6.86 | 2841 |
| tc0c20s4ct2 | 16.99 | 14.92 | 10.87 | 10800 | 17.21 | 11.97 | 6.62 | 10800 |
| tc1c20s3cf1 | 17.53 | 17.53 | 13.91 | 8909 | 17.63 | 14.25 | 8.02 | 10800 |
| tc1c20s3cf3 | 16.44 | 14.21 | 9.72 | 10800 | 16.53 | 10.07 | 5.19 | 10800 |
| tc1c20s3cf4 | 17.00 | 17.00 | 13.21 | 87 | 17.00 | 17.00 | 9.41 | 8484 |
| tc1c20s3ct1 | 19.49 | 19.00 | 14.74 | 10800 | 19.71 | 15.73 | 8.02 | 10800 |
| tc1c20s3ct3 | 12.60 | 12.60 | 9.24 | 1086 | 12.60 | 9.29 | 5.45 | 10800 |
| tc1c20s3ct4 | 16.21 | 16.21 | 13.35 | 94 | 16.21 | 14.44 | 10.17 | 10800 |
| tc1c20s4cf1 | 16.40 | 15.42 | 11.8 | 10800 | 16.48 | 12.92 | 7.63 | 10800 |
| tc1c20s4cf3 | 16.44 | 14.30 | 9.72 | 10800 | 16.94 | 10.04 | 5.19 | 10800 |
| tc1c20s4cf4 | 17.00 | 17.00 | 13.17 | 174 | 17.01 | 15.18 | 9.41 | 10800 |
| tc1c20s4ct1 | 18.36 | 17.25 | 12.6 | 10800 | 18.98 | 13.53 | 7.65 | 10800 |
| tc1c20s4ct3 | 14.43 | 13.42 | 9.53 | 10800 | 14.43 | 10.43 | 5.23 | 10800 |
| tc1c20s4ct4 | 17.00 | 17.00 | 12.78 | 222 | 17.00 | 15.73 | 10.02 | 10800 |
| tc2c20s3cf0 | - | - | 11.18 | 10800 | - | - | 2.03 | 10800 |
| tc2c20s3ct0 | - | - | 11.04 | 10800 | - | - | 1.95 | 10800 |
| tc2c20s4cf0 | - | - | 11.02 | 10800 | - | - | 2.03 | 10800 |
| tc2c20s4ct0 | - | - | 10.97 | 10800 | - | - | 1.94 | 10800 |

Table 13: Detailed computational results on the 20-customer instances for the CS replication-based models $(\beta = 3)$

| Instance | $[F_{Arc}^{CS\_rep}]$ | | | | $[F_{Node}^{CS\_rep}]$ | | | |
|---|---|---|---|---|---|---|---|---|
| | Obj | Bound | LP Bound | Time (s) | Obj | Bound | LP Bound | Time (s) |
| tc0c20s3cf2 | Inf | Inf | 14.89 | 9074 | - | - | 6.89 | 10800 |
| tc0c20s3ct2 | 17.25 | 15.06 | 11.63 | 10800 | 17.33 | 12.32 | 6.96 | 10800 |
| tc0c20s4cf2 | Inf | Inf | 14.54 | 3619 | - | - | 6.86 | 10800 |
| tc0c20s4ct2 | 16.99 | 14.76 | 10.87 | 10800 | 17.04 | 10.25 | 6.62 | 10800 |
| tc1c20s3cf1 | 17.49 | 16.98 | 13.91 | 10800 | 17.49 | 14.31 | 8.02 | 10800 |
| tc1c20s3cf3 | 16.44 | 14.04 | 9.72 | 10800 | 16.48 | 9.94 | 5.19 | 10800 |
| tc1c20s3cf4 | 17.00 | 17.00 | 13.21 | 110 | 17.00 | 17.00 | 9.41 | 8706 |
| tc1c20s3ct1 | 19.38 | 18.78 | 14.73 | 10800 | 20.06 | 15.36 | 8.02 | 10800 |
| tc1c20s3ct3 | 12.60 | 12.60 | 9.24 | 1748 | 12.60 | 9.24 | 5.45 | 10800 |
| tc1c20s3ct4 | 16.21 | 16.21 | 13.24 | 239 | 16.21 | 14.16 | 10.17 | 10800 |
| tc1c20s4cf1 | 16.38 | 15.62 | 11.8 | 10800 | 16.40 | 12.96 | 7.63 | 10800 |
| tc1c20s4cf3 | 16.44 | 13.83 | 9.72 | 10800 | 16.48 | 9.67 | 5.19 | 10800 |
| tc1c20s4cf4 | 17.00 | 17.00 | 13.17 | 215 | 17.87 | 14.83 | 9.41 | 10800 |
| tc1c20s4ct1 | 18.02 | 17.17 | 12.59 | 10800 | 18.45 | 13.64 | 7.65 | 10800 |
| tc1c20s4ct3 | 14.43 | 13.34 | 9.52 | 10800 | 14.43 | 10.41 | 5.23 | 10800 |
| tc1c20s4ct4 | 17.00 | 17.00 | 12.78 | 626 | 18.36 | 14.41 | 10.02 | 10800 |
| tc2c20s3cf0 | - | - | 11.18 | 10800 | - | - | 2.03 | 10800 |
| tc2c20s3ct0 | 25.79 | 20.90 | 10.91 | 10800 | 25.84 | 10.17 | 1.95 | 10800 |
| tc2c20s4cf0 | - | - | 11.02 | 10800 | - | - | 2.03 | 10800 |
| tc2c20s4ct0 | - | - | 10.87 | 10800 | - | - | 1.94 | 10800 |

Table 14: Detailed computational results on the 20-customer instances for the CS replication-based models ($\beta = 4$)

| Instance | $[F_{Arc}^{CS-rep}]$ | | | | $[F_{Node}^{CS-rep}]$ | | | |
|---|---|---|---|---|---|---|---|---|
| | Obj | Bound | LP Bound | Time (s) | Obj | Bound | LP Bound | Time (s) |
| tc0c20s3cf2 | - | - | 14.89 | 10800 | - | - | 6.89 | 10800 |
| tc0c20s3ct2 | 17.08 | 14.96 | 11.63 | 10800 | 17.31 | 12.19 | 6.96 | 10800 |
| tc0c20s4cf2 | - | - | 14.54 | 10800 | - | - | 6.86 | 10800 |
| tc0c20s4ct2 | 16.99 | 14.63 | 10.87 | 10800 | 17.72 | 10.16 | 6.62 | 10800 |
| tc1c20s3cf1 | 17.49 | 16.94 | 13.91 | 10800 | 17.49 | 14.25 | 8.02 | 10800 |
| tc1c20s3cf3 | 16.44 | 13.99 | 9.72 | 10800 | 16.95 | 9.36 | 5.19 | 10800 |
| tc1c20s3cf4 | 17.00 | 17.00 | 13.21 | 126 | 17.00 | 16.82 | 9.41 | 10800 |
| tc1c20s3ct1 | 18.94 | 18.42 | 14.73 | 10800 | 19.68 | 14.93 | 8.02 | 10800 |
| tc1c20s3ct3 | 12.60 | 12.60 | 9.24 | 4177 | 12.67 | 9.29 | 5.45 | 10800 |
| tc1c20s3ct4 | 16.21 | 16.21 | 13.14 | 357 | 16.23 | 13.70 | 10.17 | 10800 |
| tc1c20s4cf1 | 16.38 | 15.17 | 11.8 | 10800 | 16.38 | 12.91 | 7.63 | 10800 |
| tc1c20s4cf3 | 16.80 | 13.41 | 9.72 | 10800 | 16.89 | 10.30 | 5.19 | 10800 |
| tc1c20s4cf4 | 17.00 | 17.00 | 13.17 | 463 | 17.00 | 14.99 | 9.41 | 10800 |
| tc1c20s4ct1 | 17.81 | 16.60 | 12.59 | 10800 | 17.87 | 13.73 | 7.65 | 10800 |
| tc1c20s4ct3 | 14.43 | 13.28 | 9.51 | 10800 | 14.50 | 10.40 | 5.23 | 10800 |
| tc1c20s4ct4 | 17.00 | 17.00 | 12.78 | 876 | 17.17 | 14.31 | 10.02 | 10800 |
| tc2c20s3cf0 | - | - | 11.18 | 10800 | - | - | 2.03 | 10800 |
| tc2c20s3ct0 | 25.79 | 20.85 | 10.91 | 10800 | 26.24 | 10.13 | 1.95 | 10800 |
| tc2c20s4cf0 | - | - | 11.02 | 10800 | - | - | 2.03 | 10800 |
| tc2c20s4ct0 | - | - | 10.87 | 10800 | - | - | 1.94 | 10800 |

### F.1.2 Path-based model

Table 15: Detailed computational results on the 10-customer instances for the path-based model

| Instance | #CSPs | $[F_{Arc}^{Path}]$ | | | |
|---|---|---|---|---|---|
| | | Obj | Bound | LP Bound | Time (s) |
| tc0c10s2cf1 | 197 | 19.75 | 19.75 | 12.26 | 11 |
| tc0c10s2ct1 | 245 | 12.30 | 12.30 | 9.36 | 10 |
| tc0c10s3cf1 | 197 | 19.75 | 19.75 | 12.26 | 11 |
| tc0c10s3ct1 | 358 | 10.80 | 10.80 | 8.25 | 7 |
| tc1c10s2cf2 | 192 | 9.03 | 9.03 | 6.81 | 4 |
| tc1c10s2cf3 | 204 | 16.37 | 16.37 | 11.26 | 56 |
| tc1c10s2cf4 | 253 | 16.10 | 16.10 | 12.74 | 10 |
| tc1c10s2ct2 | 283 | 10.75 | 10.75 | 6.99 | 118 |
| tc1c10s2ct3 | 249 | 13.17 | 13.17 | 9.01 | 4 |
| tc1c10s2ct4 | 259 | 13.83 | 13.83 | 11.15 | 4 |
| tc1c10s3cf2 | 192 | 9.03 | 9.03 | 6.81 | 4 |
| tc1c10s3cf3 | 218 | 16.37 | 16.37 | 11.25 | 48 |
| tc1c10s3cf4 | 336 | 14.90 | 14.90 | 11.17 | 13 |
| tc1c10s3ct2 | 448 | 9.20 | 9.20 | 6.61 | 45 |
| tc1c10s3ct3 | 350 | 13.02 | 13.02 | 7.33 | 27 |
| tc1c10s3ct4 | 317 | 13.21 | 13.21 | 8.99 | 10 |
| tc2c10s2cf0 | 238 | 21.77 | 21.77 | 10.96 | 348 |
| tc2c10s2ct0 | 265 | 12.45 | 12.45 | 4.94 | 794 |
| tc2c10s3cf0 | 238 | 21.77 | 21.77 | 10.96 | 350 |
| tc2c10s3ct0 | 366 | 11.51 | 11.51 | 4.41 | 2716 |

Table 16: Detailed computational results on the 20-customer instances for the path-based model

| Instance | #CSPs | $[F_{Arc}^{Path}]$ | | | |
|---|---|---|---|---|---|
| | | Obj | Bound | LP Bound | Time (s) |
| tc0c20s3cf2 | 939 | 27.49 | 22.9 | 15.34 | 10800 |
| tc0c20s3ct2 | 1284 | 17.08 | 14.93 | 11.66 | 10800 |
| tc0c20s4cf2 | 1036 | 27.47 | 22.97 | 15.34 | 10800 |
| tc0c20s4ct2 | 1350 | 16.99 | 14.98 | 11.40 | 10800 |
| tc1c20s3cf1 | 1189 | 17.49 | 17.13 | 14.08 | 10800 |
| tc1c20s3cf3 | 1491 | 16.48 | 13.28 | 9.92 | 10800 |
| tc1c20s3cf4 | 1257 | 17.00 | 17.00 | 14.95 | 111 |
| tc1c20s3ct1 | 1255 | 18.94 | 18.55 | 14.98 | 10800 |
| tc1c20s3ct3 | 1373 | 12.60 | 12.60 | 9.39 | 1529 |
| tc1c20s3ct4 | 1104 | 16.21 | 16.21 | 13.91 | 163 |
| tc1c20s4cf1 | 1304 | 16.38 | 16.05 | 12.58 | 10800 |
| tc1c20s4cf3 | 1607 | 16.84 | 13.19 | 9.92 | 10800 |
| tc1c20s4cf4 | 1322 | 17.00 | 17.00 | 14.69 | 134 |
| tc1c20s4ct1 | 1516 | 18.02 | 16.67 | 13.72 | 10800 |
| tc1c20s4ct3 | 1645 | 14.43 | 13.60 | 9.69 | 10800 |
| tc1c20s4ct4 | 1330 | 17.00 | 17.00 | 14.54 | 506 |
| tc2c20s3cf0 | 1437 | 24.68 | 18.31 | 11.95 | 10800 |
| tc2c20s3ct0 | 1318 | 25.80 | 21.29 | 11.74 | 10800 |
| tc2c20s4cf0 | 1353 | 24.67 | 17.78 | 11.81 | 10800 |
| tc2c20s4ct0 | 1195 | 26.02 | 20.45 | 11.46 | 10800 |

## F.2 FRVCP

The average CPU time reported in column "Time (ms)" in the following tables is in milliseconds and rounded up to the nearest tenth.

Table 17: Computational results for the different algorithms on 29,443 routes according to the number of CSs

| $|F|$ | Algorithm | #Feas | #Opt | Gap | Time (ms) | $|F|$ | Algorithm | #Feas | #Opt | Gap | Time (ms) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | MILP_PATH | 440 | 440 | 0.000% | 14.2 | 8 | MILP_PATH | 5316 | 5316 | 0.000% | 178.4 |
| | LABEL | 440 | 440 | 0.000% | 0.8 | | LABEL | 5316 | 5316 | 0.000% | 0.7 |
| | HEURISTIC | 440 | 435 | 0.006% | 0.3 | | HEURISTIC | 5225 | 4466 | 0.174% | 0.3 |
| | HEURISTIC_M2017 | 418 | 339 | 0.599% | 0.2 | | HEURISTIC_M2017 | 4412 | 3221 | 1.453% | 0.2 |
| 3 | MILP_PATH | 1569 | 1569 | 0.000% | 27.7 | 12 | MILP_PATH | 3218 | 3218 | 0.000% | 76.6 |
| | LABEL | 1569 | 1569 | 0.000% | 0.7 | | LABEL | 3218 | 3218 | 0.000% | 1.0 |
| | HEURISTIC | 1559 | 1475 | 0.055% | 0.3 | | HEURISTIC | 3202 | 2687 | 0.199% | 0.3 |
| | HEURISTIC_M2017 | 1498 | 1229 | 0.692% | 0.2 | | HEURISTIC_M2017 | 2996 | 2110 | 0.959% | 0.2 |
| 4 | MILP_PATH | 1079 | 1079 | 0.000% | 27.1 | 16 | MILP_PATH | 7861 | 7861 | 0.000% | 69.0 |
| | LABEL | 1079 | 1079 | 0.000% | 0.7 | | LABEL | 7861 | 7861 | 0.000% | 0.9 |
| | HEURISTIC | 1078 | 1047 | 0.025% | 0.3 | | HEURISTIC | 7613 | 5971 | 0.236% | 0.2 |
| | HEURISTIC_M2017 | 1046 | 876 | 0.807% | 0.1 | | HEURISTIC_M2017 | 6424 | 4894 | 0.690% | 0.2 |
| 5 | MILP_PATH | 1982 | 1982 | 0.000% | 24.0 | 24 | MILP_PATH | 7978 | 7978 | 0.000% | 123.1 |
| | LABEL | 1982 | 1982 | 0.000% | 0.6 | | LABEL | 7978 | 7978 | 0.000% | 1.5 |
| | HEURISTIC | 1948 | 1763 | 0.154% | 0.2 | | HEURISTIC | 7750 | 5948 | 0.228% | 0.3 |
| | HEURISTIC_M2017 | 1737 | 1422 | 0.903% | 0.1 | | HEURISTIC_M2017 | 6867 | 5013 | 0.534% | 0.3 |

Table 18: Computational results for the different algorithms on 29,443 routes according to the number of nodes in the initial route (including the depot at the beginning and at the end)

| $|n(\Pi)|$ | Algorithm | #Feas | #Opt | Gap | Time (ms) | $|n(\Pi)|$ | Algorithm | #Feas | #Opt | Gap | Time (ms) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | MILP_PATH | 58 | 58 | 0.000% | 19.7 | 10 | MILP_PATH | 5828 | 5828 | 0.000% | 99.9 |
| | LABEL | 58 | 58 | 0.000% | 0.5 | | LABEL | 5828 | 5828 | 0.000% | 1.0 |
| | HEURISTIC | 58 | 51 | 0.285% | 0.1 | | HEURISTIC | 5631 | 4427 | 0.193% | 0.2 |
| | HEURISTIC_M2017 | 58 | 49 | 0.926% | 0.1 | | HEURISTIC_M2017 | 4843 | 3505 | 0.541% | 0.2 |
| 4 | MILP_PATH | 298 | 298 | 0.000% | 23.4 | 11 | MILP_PATH | 4991 | 4991 | 0.000% | 166.6 |
| | LABEL | 298 | 298 | 0.000% | 0.6 | | LABEL | 4991 | 4991 | 0.000% | 1.0 |
| | HEURISTIC | 298 | 278 | 0.090% | 0.2 | | HEURISTIC | 4905 | 4250 | 0.105% | 0.2 |
| | HEURISTIC_M2017 | 292 | 246 | 1.303% | 0.1 | | HEURISTIC_M2017 | 4640 | 3658 | 0.284% | 0.2 |
| 5 | MILP_PATH | 710 | 710 | 0.000% | 29.4 | 12 | MILP_PATH | 3942 | 3942 | 0.000% | 77.5 |
| | LABEL | 710 | 710 | 0.000% | 0.7 | | LABEL | 3942 | 3942 | 0.000% | 1.1 |
| | HEURISTIC | 706 | 613 | 0.250% | 0.2 | | HEURISTIC | 3907 | 3579 | 0.040% | 0.3 |
| | HEURISTIC_M2017 | 678 | 512 | 2.028% | 0.1 | | HEURISTIC_M2017 | 3813 | 3322 | 0.139% | 0.2 |
| 6 | MILP_PATH | 1227 | 1227 | 0.000% | 43.6 | 13 | MILP_PATH | 1523 | 1523 | 0.000% | 77.5 |
| | LABEL | 1227 | 1227 | 0.000% | 0.9 | | LABEL | 1523 | 1523 | 0.000% | 1.2 |
| | HEURISTIC | 1193 | 1010 | 0.298% | 0.3 | | HEURISTIC | 1515 | 1441 | 0.023% | 0.2 |
| | HEURISTIC_M2017 | 1158 | 825 | 2.431% | 0.2 | | HEURISTIC_M2017 | 1501 | 1425 | 0.020% | 0.2 |
| 7 | MILP_PATH | 2084 | 2084 | 0.000% | 59.5 | 14 | MILP_PATH | 318 | 318 | 0.000% | 55.2 |
| | LABEL | 2084 | 2084 | 0.000% | 0.9 | | LABEL | 318 | 318 | 0.000% | 1.3 |
| | HEURISTIC | 2051 | 1648 | 0.291% | 0.3 | | HEURISTIC | 317 | 306 | 0.012% | 0.2 |
| | HEURISTIC_M2017 | 1825 | 1159 | 1.911% | 0.2 | | HEURISTIC_M2017 | 318 | 301 | 0.012% | 0.2 |
| 8 | MILP_PATH | 3411 | 3411 | 0.000% | 76.8 | 15 | MILP_PATH | 23 | 23 | 0.000% | 16.8 |
| | LABEL | 3411 | 3411 | 0.000% | 0.9 | | LABEL | 23 | 23 | 0.000% | 1.9 |
| | HEURISTIC | 3318 | 2543 | 0.309% | 0.3 | | HEURISTIC | 23 | 22 | 0.007% | 0.2 |
| | HEURISTIC_M2017 | 2571 | 1659 | 1.753% | 0.2 | | HEURISTIC_M2017 | 23 | 23 | 0.000% | 0.2 |
| 9 | MILP_PATH | 5030 | 5030 | 0.000% | 104.0 | | | | | | |
| | LABEL | 5030 | 5030 | 0.000% | 0.9 | | | | | | |
| | HEURISTIC | 4893 | 3624 | 0.314% | 0.3 | | | | | | |
| | HEURISTIC_M2017 | 3678 | 2420 | 1.077% | 0.2 | | | | | | |

## F.3   FRVCP_1

The average CPU time reported in column "Time (ms)" in the following tables is in milliseconds and rounded up to the nearest tenth.

Table 19: Computational results for the different algorithms on 29,443 routes according to the number of CSs

| $|F|$ | Algorithm | #Feas | #Opt | Gap | Time (ms) | $|F|$ | Algorithm | #Feas | #Opt | Gap | Time (ms) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | MILP_PATH_1 | 440 | 440 | 0.000% | 9.9 | 8 | MILP_PATH_1 | 5316 | 5316 | 0.000% | 23.1 |
| | LABEL_1 | 440 | 440 | 0.000% | 0.3 | | LABEL_1 | 5316 | 5316 | 0.000% | 0.3 |
| | MILP_M2017 | 440 | 440 | 0.000% | 13.1 | | MILP_M2017 | 5316 | 5316 | 0.000% | 39.2 |
| 3 | MILP_PATH_1 | 1569 | 1569 | 0.000% | 14.4 | 12 | MILP_PATH_1 | 3218 | 3218 | 0.000% | 30.2 |
| | LABEL_1 | 1569 | 1569 | 0.000% | 0.4 | | LABEL_1 | 3218 | 3218 | 0.000% | 0.5 |
| | MILP_M2017 | 1569 | 1569 | 0.000% | 19.7 | | MILP_M2017 | 3218 | 3218 | 0.000% | 57.6 |
| 4 | MILP_PATH_1 | 1079 | 1079 | 0.000% | 14.8 | 16 | MILP_PATH_1 | 7861 | 7861 | 0.000% | 31.7 |
| | LABEL_1 | 1079 | 1079 | 0.000% | 0.3 | | LABEL_1 | 7861 | 7861 | 0.000% | 0.6 |
| | MILP_M2017 | 1079 | 1079 | 0.000% | 21.4 | | MILP_M2017 | 7861 | 7861 | 0.000% | 58.4 |
| 5 | MILP_PATH_1 | 1982 | 1982 | 0.000% | 14.0 | 24 | MILP_PATH_1 | 7978 | 7978 | 0.000% | 43.5 |
| | LABEL_1 | 1982 | 1982 | 0.000% | 0.3 | | LABEL_1 | 7978 | 7978 | 0.000% | 1.0 |
| | MILP_M2017 | 1982 | 1982 | 0.000% | 23.3 | | MILP_M2017 | 7978 | 7978 | 0.000% | 82.8 |

Table 20: Computational results for the different algorithms on 29,443 routes according to the number of nodes in the initial route (including the depot at the beginning and at the end)

| $|n(\Pi)|$ | Algorithm | #Feas | #Opt | Gap | Time (ms) | $|n(\Pi)|$ | Algorithm | #Feas | #Opt | Gap | Time (ms) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | MILP_PATH_1 | 58 | 58 | 0.000% | 8.1 | 10 | MILP_PATH_1 | 5828 | 5828 | 0.000% | 33.4 |
|  | LABEL_1 | 58 | 58 | 0.000% | 0.2 |  | LABEL_1 | 5828 | 5828 | 0.000% | 0.6 |
|  | MILP_M2017 | 58 | 58 | 0.000% | 14.1 |  | MILP_M2017 | 5828 | 5828 | 0.000% | 64.9 |
| 4 | MILP_PATH_1 | 298 | 298 | 0.000% | 9.0 | 11 | MILP_PATH_1 | 4991 | 4991 | 0.000% | 36.3 |
|  | LABEL_1 | 298 | 298 | 0.000% | 0.2 |  | LABEL_1 | 4991 | 4991 | 0.000% | 0.7 |
|  | MILP_M2017 | 298 | 298 | 0.000% | 18.7 |  | MILP_M2017 | 4991 | 4991 | 0.000% | 53.7 |
| 5 | MILP_PATH_1 | 710 | 710 | 0.000% | 11.1 | 12 | MILP_PATH_1 | 3942 | 3942 | 0.000% | 36.7 |
|  | LABEL_1 | 710 | 710 | 0.000% | 0.2 |  | LABEL_1 | 3942 | 3942 | 0.000% | 0.8 |
|  | MILP_M2017 | 710 | 710 | 0.000% | 28.6 |  | MILP_M2017 | 3942 | 3942 | 0.000% | 38.1 |
| 6 | MILP_PATH_1 | 1227 | 1227 | 0.000% | 14.1 | 13 | MILP_PATH_1 | 1523 | 1523 | 0.000% | 33.4 |
|  | LABEL_1 | 1227 | 1227 | 0.000% | 0.3 |  | LABEL_1 | 1523 | 1523 | 0.000% | 0.9 |
|  | MILP_M2017 | 1227 | 1227 | 0.000% | 41.3 |  | MILP_M2017 | 1523 | 1523 | 0.000% | 22.2 |
| 7 | MILP_PATH_1 | 2084 | 2084 | 0.000% | 19.6 | 14 | MILP_PATH_1 | 318 | 318 | 0.000% | 30.8 |
|  | LABEL_1 | 2084 | 2084 | 0.000% | 0.4 |  | LABEL_1 | 318 | 318 | 0.000% | 1.0 |
|  | MILP_M2017 | 2084 | 2084 | 0.000% | 58.7 |  | MILP_M2017 | 318 | 318 | 0.000% | 17.1 |
| 8 | MILP_PATH_1 | 3411 | 3411 | 0.000% | 24.7 | 15 | MILP_PATH_1 | 23 | 23 | 0.000% | 13.8 |
|  | LABEL_1 | 3411 | 3411 | 0.000% | 0.4 |  | LABEL_1 | 23 | 23 | 0.000% | 0.8 |
|  | MILP_M2017 | 3411 | 3411 | 0.000% | 67.6 |  | MILP_M2017 | 23 | 23 | 0.000% | 5.5 |
| 9 | MILP_PATH_1 | 5030 | 5030 | 0.000% | 30.2 |  |  |  |  |  |  |
|  | LABEL_1 | 5030 | 5030 | 0.000% | 0.5 |  |  |  |  |  |  |
|  | MILP_M2017 | 5030 | 5030 | 0.000% | 70.3 |  |  |  |  |  |  |

## F.4 E-VRP-NL

Table 21: New BKS obtained for the E-VRP-NL by applying the labeling algorithm on each route of the solutions reported by Montoya et al. (2017)

| Instance | Old BKS | New BKS |
|---|---|---|
| tc0c40s8cf0 | 31.282 | 31.045 |
| tc2c40s8cf2 | 27.147 | 27.141 |
| tc2c40s5cf2 | 27.543 | 27.536 |
| tc0c80s8cf1 | 45.227 | 45.225 |
| tc1c80s12cf2 | 29.544 | 29.532 |
| tc2c80s8cf4 | 49.287 | 49.171 |
| tc2c80s8ct3 | 32.315 | 32.312 |
| tc0c160s16cf4 | 82.924 | 82.863 |
| tc0c160s16ct4 | 82.365 | 82.323 |
| tc0c160s24ct4 | 80.956 | 80.796 |
| tc0c160s24cf4 | 81.442 | 81.380 |
| tc1c160s16cf3 | 71.757 | 71.509 |
| tc1c160s24cf3 | 68.558 | 68.512 |
| tc1c320s24cf2 | 152.131 | 152.063 |
| tc1c320s24cf3 | 117.475 | 117.462 |
| tc1c320s38cf2 | 141.631 | 141.620 |
| tc1c320s38ct3 | 116.075 | 116.065 |
| tc2c320s24cf0 | 182.454 | 182.453 |
| tc2c320s24ct4 | 121.936 | 121.820 |
| tc2c320s38cf4 | 122.325 | 122.318 |
| tc2c320s38ct0 | 190.974 | 190.963 |
| tc2c320s38ct1 | 94.534 | 94.533 |
| tc2c320s38ct4 | 121.662 | 121.657 |