

**Exact algorithms for electric vehicle-
routing problems with time windows**

G. Desaulniers, F. Errico,
S. Irnich, M. Schneider

G-2014-110

December 2014

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2014.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2014.

Exact algorithms for electric vehicle-routing problems with time windows

Guy Desaulniers^a

Fausto Errico^b

Stefan Irnich^c

Michael Schneider^d

^a GERAD & Polytechnique Montréal, Montréal (Québec)
Canada, H3C 3A7

^b École de technologie supérieure, Montréal (Québec)
Canada, H3C 1K3

^c Chair of Logistics Management, Gutenberg School of
Management and Economics, Johannes Gutenberg Univer-
sity Mainz, 55128 Mainz, Germany

^d Logistics Planning and Information Systems, Department
of Law and Economics, TU Darmstadt, 64289 Darmstadt,
Germany

guy.desaulniers@gerad.ca
fausto.errico@cirrelt.ca
irnich@uni-mainz.de
schneider@bwl.tu-darmstadt.de

December 2014

Les Cahiers du GERAD

G–2014–110

Copyright © 2014 GERAD

Abstract: Effective route planning for battery electric commercial vehicle (ECV) fleets has to take into account their limited autonomy and the possibility of visiting recharging stations during the course of a route. In this paper, we consider four variants of the electric vehicle-routing problem with time windows: (i) at most a single recharge per route is allowed, and batteries are fully recharged on visit of a recharging station, (ii) multiple recharges per route, full recharges only, (iii) at most a single recharge per route, and partial battery recharges are possible, and (iv) multiple, partial recharges. For each variant, we present exact branch-price-and-cut algorithms that rely on customized mono-directional and bi-directional labeling algorithms for generating feasible vehicle routes. In computational studies, we find that all four variants are solvable for instances with up to 100 customers and 21 recharging stations. This success can be attributed to the tailored resource extension functions (REFs) that enable efficient labeling with constant time feasibility checking and strong dominance rules, even if these REFs are intricate and rather elaborate to derive. The studies also highlight the superiority of the bi-directional labeling algorithms compared to the mono-directional ones. Finally, we find that allowing multiple as well as partial recharges both help to reduce routing costs and the number of employed vehicles in comparison to the variants with single and with full recharges.

Key Words: Vehicle routing, electric vehicles, recharging decisions, branch-price-and-cut, labeling algorithms.

1 Introduction

The utilization of battery electric commercial vehicles (ECVs) is steadily increasing, e.g., in the field of small-package shipping or the distribution of food and beverages (Pelletier et al., 2014), despite the fact that, when compared to conventional internal combustion engine vehicles, ECVs offer a limited driving range and are not competitive from a cost perspective due to their high acquisition costs (Davis and Figliozzi, 2013). This increase of ECV usage is due to the following advantages of the ECVs:

- ECVs produce minimal noise and no local greenhouse gas emissions. Therefore, they can be employed to meet emission targets of delivery fleets or to serve restricted inner-city areas with noise and emission limits.
- ECVs help logistics companies to promote a green image, an important competitive factor given the increasing number of socially and environmentally aware customers. Moreover, relative autonomy from fluctuating oil prices can be achieved.
- ECVs become attractive from a cost perspective due to heavy subsidies offered by several governments around the globe. In addition, governments and private companies are strongly investing to provide the required recharging infrastructure.

Effective route planning of an ECV fleet requires solving vehicle-routing problems (VRPs, see, e.g., Toth and Vigo, 2014) that take into account the limited driving range of ECVs and the possibility of visiting recharging stations during the course of a route. Several heuristic solution methods for such VRPs have recently been proposed in the literature (see, e.g., Erdogan and Miller-Hooks, 2012; Schneider et al., 2014a; Felipe et al., 2014). However, to the best of our knowledge, no exact solution method has been presented yet.

In this paper, we develop effective branch-price-and-cut-algorithms for four variants of the electric VRP with time windows (EVRPTW). The EVRPTW was introduced in (Schneider et al., 2014a) and includes recharging times that depend on the battery level on arrival at a recharging station, vehicle capacity, and customer time windows. The following four EVRPTW variants are addressed:

- (i) at most a single (S) recharge per route is allowed, and batteries are fully (F) recharged on visit of a recharging station (EVRPTW-SF);
- (ii) the variant considered in (Schneider et al., 2014a): multiple (M) recharges per route, full (F) recharges only (EVRPTW-MF);
- (iii) at most a single (S) recharge per route, and partial (P) battery recharges are possible (EVRPTW-SP); and
- (iv) multiple (M), partial (P) recharges (EVRPTW-MP).

We solve the four EVRPTW variants to optimality with eight branch-price-and-cut algorithms, two algorithms tailored to each specific variant. Branch-price-and-cut means that an extensive formulation (a set-partitioning model in our case) is linearly relaxed, the relaxation is solved using column generation, additional valid inequalities are added to strengthen the LP bound, and integer solutions are finally enforced by branching (see Desaulniers et al., 2005; Lübbecke and Desrosiers, 2005). The solution of the master program, i.e., the linear relaxation of the extensive formulation, starts with restricting the master to a small subset of variables. The optimization of this restricted master program (RMP) provides the necessary dual information needed to generate missing variables (columns) for the RMP. In our problem, as in many extensive formulations for vehicle routing and crew scheduling problems, the generation of variables uses a path representation of routes so that the column-generation subproblem is a variant of the elementary shortest-path problem with resource constraints (ESPPRC, see Irnich and Desaulniers, 2005). The column-generation process alternates between RMP reoptimization and solution of the ESPPRC until no more columns with a negative reduced cost exist.

While the master program for EVRPTW is standard, the four variants give rise to different ESPPRCs. The contribution of the paper at hand lies in the concise formulation of the different ESPPRC variants so that highly effective solution techniques can be applied. An important aspect is the modeling with as few as possible attributes (resource variables) in such a way that dominance rules allow the elimination of the majority of the partial paths constructed in the course of the ESPPRC labeling algorithm. For the variants with partial recharge (SP and MP), there is an immanent tradeoff between the amount recharged and the

time spent for recharging: longer recharging extends the driving range while it may prohibit the timely arrival at a later customer due to its service time window. Therefore, we require a label that models a tradeoff curve; tradeoffs between two resources were earlier discussed for flight synchronization (Ioachim et al., 1999), for the split delivery VRP with time windows (Desaulniers, 2010) and for inventory routing (Desaulniers et al., 2014b). Additional state-of-the-art solution principles for ESPPRC rely on bi-directional search (Righini and Salani, 2006) and bounding (Bode and Irnich, 2014). These techniques require a reversible labeling process, meaning that paths can be generated from a given endpoint (the destination depot) using backward propagation of labels. Interestingly, compared to the forward labeling process, the backward labeling for the EVRPTW variants with full recharges (SF and MF) is considerably harder from the conceptual and modeling point of view as well as from the computational perspective. Hence, it is not obvious whether bi-directional labeling is superior. In contrast, forward and backward labeling for the variants with partial recharges (SP and MP) are equally complex. The paper compares the performance of the mono-directional and bi-directional algorithms in EVRPTW branch-price-and-cut.

Finally, there exists another tradeoff within the branch-price-and-cut algorithm itself: since the solution of the ESPPRC is often prohibitively time consuming, relaxations of the ESPPRC are frequently solved instead, often in combination with the addition of valid inequalities for the master program. Therefore, we apply the *ng*-route relaxation (Baldacci et al., 2011), an adaptation of 2-path cuts (Kohl et al., 1999), and subset-row inequalities (Jepsen et al., 2008).

We present an overall scheme that uses all the algorithmic components in a well coordinated fashion. The computational experiments demonstrate that all four variants are solvable for instances with up to 100 customers and 21 recharging stations. This success can be attributed to the tailored resource extension functions (REFs) that enable efficient labeling with constant time feasibility checking and strong dominance rules, even if these REFs are intricate and rather elaborate to derive. The studies also highlight the superiority of the bi-directional labeling algorithms compared to the mono-directional ones and quantify the gains that can be achieved by allowing partial recharges and multiple recharges per route.

This paper is organized as follows. In the next section, we provide a short discussion of the literature on electric VRPs (EVRPs). In Section 3, the four variants of the EVRPTW studied in this paper are described in detail, and a mathematical formulation valid for all four variants is provided. In Section 4, we introduce the proposed branch-price-and-cut algorithms. The results of our computational experiments are presented in Section 5. Finally, conclusions are drawn in Section 6.

2 Literature

EVRPs are structurally similar to VRPs with intermediate replenishment facilities (Crevier et al., 2007; Muter et al., 2014) and VRPs with distance constraints (Laporte et al., 1985; Juan et al., 2014). However, neither heuristics nor exact solution methods for these problems are directly applicable for solving EVRPTW to optimality. We will instead borrow some solution techniques from exact VRPTW approaches (Desaulniers et al., 2014a). Moreover, EVRPs are related to the planning of recharging infrastructure (MirHassani and Ebrazi, 2013; Mak et al., 2014).

The EVRPs addressed in the literature cover different subsets of classical VRP constraints and are shortly discussed in the following. Conrad and Figliozzi (2011) present the recharging VRP that consists of routing a fleet of ECVs with limited driving range subject to vehicle capacity and time window constraints. Recharging is possible at certain customer locations and incurs a constant time penalty. The authors use an iterative route construction and improvement algorithm to study the impact of driving range, recharging times, and time window existence on modified Solomon (1987) instances. Moreover, bounds on the average tour lengths are computed. Erdogan and Miller-Hooks (2012) propose two heuristics for the green VRP, a routing problem in which vehicles can be refueled at dedicated stations that have to be visited en route. Neither capacity constraints nor time windows are considered. Montoya et al. (2014) address the green VRP as follows: first, giant tours are built by means of three randomized heuristics for the traveling salesman problem (TSP), then routes are generated by optimally splitting the giant tours, and finally, a green VRP solution is assembled by solving a set partitioning problem over the generated routes. Schneider et al. (2014b) develop an adaptive

variable neighborhood search (VNS) for VRPs with intermediate stops that is used to solve instances of the green VRP and of an EVRP with a maximal route duration constraint. Felipe et al. (2014) study an extension of the green VRP that considers different types of recharging stations with different costs and recharging speeds. As solution methods, a construction heuristic, several local search heuristics, and a simulated annealing algorithm are developed.

Schneider et al. (2014a) propose a metaheuristic hybrid of VNS and tabu search to address EVRPTW-MF. Hiermann et al. (2014) consider a combination of EVRPTW-MF and the fleet size and mix VRP with fixed costs. Here, an unlimited number of ECVs with different battery capacities, load capacities and acquisition costs but vehicle-independent routing costs are available. The problem is addressed by means of an adaptive large neighborhood search (ALNS) enhanced by a labeling algorithm. Finally, a VRP with a mixed fleet of electric and conventional vehicles is investigated by means of an ALNS in Goeke and Schneider (2014). For both types of vehicles, energy consumption depends on vehicle load, vehicle speed and gradients.

Among these works, Conrad and Figliozzi (2011); Erdogan and Miller-Hooks (2012), and Montoya et al. (2014) assume fixed recharging/refueling times, while Schneider et al. (2014a,b); Hiermann et al. (2014); Goeke and Schneider (2014) and Felipe et al. (2014) assume the recharging time to be a linear function of the amount of energy recharged. Among the latter works, all but Felipe et al. (2014) make the assumption of full recharges. None of the above works explicitly restricts the number of recharges per route to a single recharge, although this may often be possible in practical situations where the total route distance is limited by the working hours of the driver.

For a discussion of simplifications commonly used in EVRP models, we refer to (Schneider et al., 2014a; Goeke and Schneider, 2014). Finally, an overview of the field of goods distribution with ECVs can be found in (Pelletier et al., 2014).

3 Problem description and mathematical formulation

Let N be the set of customers that all require deliveries (collection from all customers is identical). Denote by q_i the demand of customer $i \in N$ and by $[e_i, \ell_i]$ the time window in which service has to start at this customer. A vehicle can arrive at a customer before the opening of its time window and wait to start service. We assume an unlimited fleet of identical ECVs with a storage capacity of Q and a battery capacity of B . At the beginning of the planning horizon, the ECVs are located in a single depot from which they start fully charged and to which they must return by the end of the planning horizon.

Let R be a set of recharging stations at which the vehicles can stop en route to recharge their battery. We assume that the battery recharging time is proportional to the amount of energy recharged. Traveling from one location i (the depot, a customer or a recharging station) to another location j incurs a cost c_{ij} , a travel time t_{ij} (that includes service time at i if $i \in N$), and an energy consumption b_{ij} .

There is certainly a cost for the energy consumed by the vehicle along its route. We consider the case with identical recharging costs at all stations R and at the depot, where vehicles are fully recharged at the end of the day. In this case, the recharging cost is irrelevant for routing decisions. The case of station-specific costs is more intricate to handle.

A vehicle route is a sequence of locations that starts and ends at the depot and visits a non-empty subset of customers and possibly some recharging stations. Its cost is given by the sum of the travel costs c_{ij} between the pairs of consecutive locations i and j that it visits. A route is feasible if (i) it is elementary with respect to the customers (recharging stations may be visited more than once); (ii) the total demand of the visited customers does not exceed the vehicle capacity; (iii) the battery charge level is always nonnegative along the route; and (iv) the customer time windows are respected. For the problem variants allowing partial recharges, the amount recharged at a station and thus the required recharging time is a variable. Testing whether there exists a feasible time schedule for a given route is therefore a non-trivial task in the EVRPTW-MP.

Note that the battery capacity constraint can also be expressed in terms of the time required to recharge the energy consumed. Indeed, let $h_{ij} = \alpha b_{ij}$ be the time required to recharge the consumed energy b_{ij} when

traveling between locations i and j , where $\alpha > 0$ is a proportionality factor. Furthermore, let $H = \alpha B$ be the total time required to recharge B units of energy. Let (i_1, i_2, \dots, i_k) be a subpath of a route whose extremities i_1 and i_k are either the depot or a recharging station and all intermediate locations i_2 to i_{k-1} are customer locations. The battery capacity constraint imposes that $\sum_{j=1}^{k-1} b_{i_j, i_{j+1}} \leq B$ or equivalently $\sum_{j=1}^{k-1} h_{i_j, i_{j+1}} \leq H$ holds. Neither bookkeeping of the consumed energy nor of the current battery level is therefore necessary, and we will work directly and exclusively with the required recharging times. This avoids numerical rounding errors that may otherwise occur due to time-to-energy and energy-to-time conversions with a fractional result.

The EVRPTW consists of finding a set of feasible vehicle routes such that each customer $i \in N$ is visited exactly once by a vehicle and the sum of the route costs is minimized. The four EVRPTW variants stated in Section 1 only differ by constraints imposed on battery recharges. The following mathematical formulation for the EVRPTW is valid for all of them. Let Ω be the set of feasible routes that depends on the variant considered. For each route $p \in \Omega$, denote by c_p its cost and by a_{pi} , $i \in N$, a binary parameter equal to 1 if route p visits customer i and 0 otherwise. With each route $p \in \Omega$, we associate a binary variable θ_p that takes value 1 if the route is part of the solution and 0 otherwise. Using this notation, the EVRPTW can be formulated as the following integer program:

$$\min \sum_{p \in \Omega} c_p \theta_p \quad (1a)$$

$$\text{s.t.} \quad \sum_{p \in \Omega} a_{pi} \theta_p = 1, \quad \forall i \in N \quad (1b)$$

$$\theta_p \in \{0, 1\}, \quad \forall p \in \Omega. \quad (1c)$$

Objective function (1a) seeks to minimize total routing costs. Set partitioning constraints (1b) ensure that each customer $i \in N$ is visited exactly once by a vehicle. Binary requirements (1c) restrict the domain of the route variables.

Note that for the variants that consider partial recharges (EVRPTW-SP and EVRPTW-MP), there might exist numerous feasible recharging plans for a route that visits one or several recharging stations. Because no recharging costs are considered, there is no need to distinguish between those plans and it is sufficient to know that at least one recharging plan exists for each such route. Consequently, the routes are not associated with recharging plans.

In practice, model (1) contains a huge number of variables, namely, one per feasible route in Ω . This number prohibits using a standard MIP solver or branch-and-bound algorithm for solving it. In the next section, we propose alternative solution algorithms based on the branch-price-and-cut paradigm.

4 Branch-price-and-cut algorithms

To solve the set-partitioning model (1), we develop two branch-price-and-cut algorithms (Barnhart et al., 1998; Desaulniers et al., 2005; Lübbecke and Desrosiers, 2005) for each problem variant. Because the procedure that generates the routes very much depends on the EVRPTW variant, we will mainly focus on this aspect in Section 4.1, while cutting planes and branching are discussed in Sections 4.2 and 4.3.

4.1 Column generation

In this section, we focus on the initial linear relaxation of the extensive formulation (1), i.e., without cuts or branching decisions. Recall that the subproblem aims at generating negative reduced cost columns (route variables) to be added to the current RMP. If no such columns exist, the algorithm stops and the computed solution to the current RMP is also optimal for the complete linear relaxation (for further details, see Desrosiers and Lübbecke, 2005).

For model (1), the column generation subproblem can be defined as follows. Let π_i for $i \in N$ be the dual variables associated with constraints (1b). Let \bar{c}_p , $p \in \Omega$, be the reduced cost of variable θ_p with respect to

these dual variables, i.e., $\bar{c}_p = c_p - \sum_{i \in N} a_{pi} \pi_i$. The subproblem can be stated as

$$\min_{p \in \Omega} \bar{c}_p. \quad (2)$$

The set of feasible routes in Ω can be implicitly represented in a directed graph $G = (V, A)$ with vertex set V and arc set A . The vertex set V is given by $V = \{o, d\} \cup N \cup R$, where o is a source and d a sink vertex, both associated with the depot. Demand q_i and time window $[e_i, \ell_i]$ are associated with each vertex $i \in N$. For all vertices $i \in R \cup \{o, d\}$, we define $q_i = 0$ and associate a nonrestrictive time window $[e_i, \ell_i]$ (note that our algorithms can be adapted to restrictive time windows). The arc set A contains all arcs (o, j) and (j, d) with $j \in N \cup R$ and all arcs $(i, j) \in (N \cup R)^2$ with $i \neq j$. With each arc $(i, j) \in A$, we associate the cost c_{ij} , the travel time t_{ij} that includes the service time at i if $i \in N$, and the required recharging time h_{ij} . Given these parameters, some arcs can be removed from A because they cannot be part of a feasible route, namely, the arcs (i, j) with $h_{ij} > H$, $q_i + q_j > Q$, or $e_i + t_{ij} > \ell_j$. We assume that the cost, travel time, and required recharging time matrices satisfy the triangle inequality.

A feasible route in Ω corresponds to an o - d path in G , in which any vertex $i \in N$ is visited at most once, i.e., elementarity is respected for the customer vertices. However, not all elementary o - d paths in G correspond to feasible routes as they may violate the time windows, the vehicle capacity, or the battery capacity. Additional constraints on the paths are therefore required to ensure that they represent feasible routes. We define such resource constraints below.

Subproblem (2) aims at finding a feasible route with minimum reduced cost. To compute the reduced cost of each o - d path in G , we replace the arc cost c_{ij} for each arc $(i, j) \in A$ by a modified cost $\bar{c}_{ij} = c_{ij} - \pi_i$, where we set $\pi_i = 0$ if $i \in R \cup \{o, d\}$. Then, the sum of the modified costs \bar{c}_{ij} of the arcs (i, j) of a path p is equal to its reduced cost \bar{c}_p . In this setting, the subproblem corresponds to an ESPPRC, in which elementarity is imposed only on the customer vertices and the path length is measured with respect to the modified arc costs \bar{c}_{ij} , $(i, j) \in A$.

Each EVRPTW variant induces a specific subproblem. The two single-recharge variants require a resource constraint to ensure that at most one vertex in R is visited in a path. Furthermore, because full battery recharges are more restrictive than partial battery recharges, these two recharge types must be handled differently. In consequence, we consider four variants of the subproblem called ESPPRC-SF, ESPPRC-MF, ESPPRC-SP, and ESPPRC-MP hereafter.

The ESPPRC on graph G can be solved by dynamic programming using a labeling algorithm (see Irnich and Desaulniers, 2005). In this algorithm, labels are used to represent partial paths that start at the origin vertex o . Starting from an initial label associated with vertex o , paths are constructed iteratively by extending this label and its descendants forwardly in G . The extension of a label along an arc is performed using REFs. Each generated label is checked for feasibility with respect to the resource constraints and infeasible labels are discarded. Furthermore, to avoid enumerating all feasible o - d -paths, a dominance criterion is applied to eliminate partial paths for which no completion to a full o - d -path with minimal reduced cost is possible. When the labels are extended in this way from o to d , we say that a mono-directional forward search is performed.

Alternatively, a bi-directional search (see Righini and Salani, 2006) can be applied and often yields lower computational times. In this case, a resource has to be selected, for which the value is non-decreasing along a path. Moreover, a midpoint $M \in [E, F]$ has to be chosen, where $[E, F]$ is the domain of the selected resource. For example, any $M \in [e_o, \ell_d]$ works for the time resource in the EVRPTW. Now, the algorithm proceeds in three steps. First, labels are extended forwardly in G until reaching the midpoint M . Second, labels are extended backwardly from d using backward REFs until reaching M . Finally, forward and backward labels are merged and checked for feasibility to yield complete o - d -paths.

In the following, we propose mono-directional and bi-directional labeling algorithms for all subproblem variants. We focus on the label components, the REFs, and the dominance rules.

4.1.1 Labeling algorithms for the full-recharge ESPPRCs

In this section, we describe the labeling algorithms for the ESPPRC-SF and ESPPRC-MF variants of subproblem (2). We first propose a mono-directional forward labeling algorithm for ESPPRC-SF, then provide a bi-directional algorithm for the same subproblem variant. Finally, we explain how both algorithms can be adapted to ESPPRC-MF.

Mono-directional search In a forward labeling algorithm for ESPPRC-SF, a partial path p from o to a vertex $i \in V$ is represented by a label $L_i = (T_i^{cost}, T_i^{load}, T_i^{rch}, T_i^{time}, T_i^{rt}, (T_i^{cust_n})_{n \in N})$, where the label components are:

T_i^{cost} : reduced cost of path p ;

T_i^{load} : total load delivered along path p ;

T_i^{rch} : number of recharges performed along path p ;

T_i^{time} : earliest service start time at vertex i ;

T_i^{rt} : cumulated required recharging time since the last recharge along path p (or since the beginning of p if p contains no recharge);

$T_i^{cust_n}$: number of times that customer $n \in N$ is visited along path p . Also set to 1 if customer n is not visited but is unreachable from p . A customer n is said to be unreachable if $T_i^{load} + q_n > Q$ or $T_i^{time} + t_{in} > \ell_n$ in which case it cannot be part of any feasible extension of path p .

In the initial label at vertex o , all components are set to 0 except T_o^{time} that is set to e_o . The extension of a label $L_i = (T_i^{cost}, T_i^{load}, T_i^{rch}, T_i^{time}, T_i^{rt}, (T_i^{cust_n})_{n \in N})$ along an arc $(i, j) \in A$ is performed using the following REFs

$$T_j^{cost} = T_i^{cost} + \bar{c}_{ij} \quad (3a)$$

$$T_j^{load} = T_i^{load} + q_j \quad (3b)$$

$$T_j^{rch} = T_i^{rch} + \begin{cases} 1 & \text{if } j \in R \\ 0 & \text{otherwise} \end{cases} \quad (3c)$$

$$T_j^{time} = \begin{cases} \max\{e_j, T_i^{time} + t_{ij} + T_i^{rt}\} & \text{if } i \in R \\ \max\{e_j, T_i^{time} + t_{ij}\} & \text{otherwise} \end{cases} \quad (3d)$$

$$T_j^{rt} = \begin{cases} h_{ij} & \text{if } i \in R \\ T_i^{rt} + h_{ij} & \text{otherwise} \end{cases} \quad (3e)$$

$$T_j^{cust_n} = \begin{cases} T_i^{cust_n} + 1 & \text{if } j = n \\ \max\{T_i^{cust_n}, U_n^{fw}(T_j^{load}, T_j^{time})\} & \text{otherwise,} \end{cases} \quad (3f)$$

where $U_n^{fw}(T_j^{load}, T_j^{time})$ is a function that sets the unreachability status of customer $n \in N$, i.e., $U_n^{fw}(T_j^{load}, T_j^{time})$ is equal to 1 if $T_j^{load} + q_n > Q$ or $T_j^{time} + t_{jn} > \ell_n$, and to 0 otherwise. The label $L_j = (T_j^{cost}, T_j^{load}, T_j^{rch}, T_j^{time}, T_j^{rt}, (T_j^{cust_n})_{n \in N})$ resulting from this extension is deemed feasible if $T_j^{load} \leq Q$, $T_j^{rch} \leq 1$, $T_j^{rt} \leq H$, $T_j^{time} \leq \ell_j$, and $T_j^{cust_n} \leq 1$ for all $n \in N$. If L_j is not feasible, it is deleted.

Given that all REFs (3) are non-decreasing functions, the following dominance rule can be applied (see Desaulniers et al., 1998).

Definition 4.1 Let $L^k = (T_k^{cost}, T_k^{load}, T_k^{rch}, T_k^{time}, T_k^{rt}, (T_k^{cust_n})_{n \in N})$, $k \in \{1, 2\}$, be two labels associated with paths ending at the same vertex. Label L^2 is said to be dominated by label L^1 if $T_1^r \leq T_2^r$ for all $r \in \{cost, load, rch, time, rt, (cust_n)_{n \in N}\}$ and at least one of these inequalities is strict.

Dominated labels are discarded. Furthermore, if two labels L^1 and L^2 are equal, then one of them can be discarded.

Bi-directional search In a bi-directional labeling algorithm for ESPPRC-SF, we can choose the time resource for defining a midpoint, e.g., $M = (\ell_d - e_o)/2$ or $\sum_{i \in V} (\ell_i - e_i)/(2|V|)$. We chose the first of these options for the numerical studies. Then, the forward step proceeds as above except that a label is not extended if its

T^{time} component exceeds M . In the backward step, a partial path p from a vertex $j \in V$ to d is associated with a label $L_j = (W_j^{cost}, W_j^{load}, W_j^{nrch}, W_j^{time}, W_j^{sl}, W_j^{avrt}, W_j^{rt}, (W_j^{cust_n})_{n \in N})$ whose components are:

- W_j^{cost} : reduced cost of path p ;
- W_j^{load} : total load delivered along path p except at the last vertex j ;
- W_j^{nrch} : negative number of recharges performed along path p except at the last vertex j ;
- W_j^{time} : latest service start time at vertex j that ensures time window feasibility at the subsequent vertices along path p ;
- W_j^{sl} : if a recharging station is visited along path p , this is the cumulated slack time at vertex j that can be used for recharging at the next recharging station without changing the latest service start time at j . Such slack time arises if the time window of a preceding customer is so restrictive that waiting cannot be avoided. Note that we will reuse this definition of resources also for the case of multiple recharges so that the term *next recharging station* refers to either the first and only or a later recharging station;
- W_j^{avrt} : if a recharging station is visited along path p , this is the maximum recharging time available at the next recharging station that ensures time window feasibility along p ;
- W_j^{rt} : cumulated required recharging time needed to recover the energy consumed up to the next recharge along path p (or until the end of p if no recharge is performed);
- $W_j^{cust_n}$: number of times that customer $n \in N$ is visited along path p except vertex j . Also set to 1 if customer n is not visited but is unreachable from p . In this case, a customer n is said to be unreachable if $W_j^{load} + q_j + q_n > Q$ or $W_j^{time} - t_{nj} < e_n$ in which case it cannot be part of any feasible backward extension of path p .

The backward labeling process starts with an initial label at vertex d whose components are set to 0 except that $W_d^{time} = \ell_d$, $W_d^{sl} = \infty$, and $W_d^{avrt} = H$. The backward extension of a label $L_j = (W_j^{cost}, W_j^{load}, W_j^{nrch}, W_j^{time}, W_j^{sl}, W_j^{avrt}, W_j^{rt}, (W_j^{cust_n})_{n \in N})$ along an arc $(i, j) \in A$ is performed using the following backward REFs as long as $W_j^{time} - t_{ij} > M$

$$W_i^{cost} = W_j^{cost} + \bar{c}_{ij} \quad (4a)$$

$$W_i^{load} = W_j^{load} + q_j \quad (4b)$$

$$W_i^{nrch} = W_j^{nrch} - \begin{cases} 1 & \text{if } j \in R \\ 0 & \text{otherwise} \end{cases} \quad (4c)$$

$$W_i^{time} = \begin{cases} \min\{\ell_i, W_j^{time} - t_{ij} - h_{ij}\} & \text{if } j \in R \\ \min\{\ell_i, W_j^{time} - t_{ij}\} & \text{if } j \notin R \text{ and } W_j^{nrch} = 0 \\ \min\{\ell_i, W_j^{time} - t_{ij} + \min\{0, W_j^{sl} - h_{ij}\}\} & \text{otherwise} \end{cases} \quad (4d)$$

$$W_i^{sl} = \begin{cases} \max\{0, W_j^{time} - t_{ij} - h_{ij} - \ell_i\} & \text{if } j \in R \\ \infty & \text{if } j \notin R \text{ and } W_j^{nrch} = 0 \\ \max\{W_j^{sl} - h_{ij}, 0\} + \\ \max\{0, W_j^{time} - t_{ij} + \min\{0, W_j^{sl} - h_{ij}\} - \ell_i\} & \text{otherwise} \end{cases} \quad (4e)$$

$$W_i^{avrt} = \begin{cases} \min\{H - h_{ij}, W_i^{time} - e_i + W_i^{sl}\} & \text{if } j \in R \\ H & \text{if } j \notin R \text{ and } W_j^{nrch} = 0 \\ \min\{W_j^{avrt} - h_{ij}, W_i^{time} - e_i + W_i^{sl}\} & \text{otherwise} \end{cases} \quad (4f)$$

$$W_i^{rt} = \begin{cases} h_{ij} & \text{if } j \in R \\ W_j^{rt} + h_{ij} & \text{otherwise} \end{cases} \quad (4g)$$

$$W_i^{cust_n} = \begin{cases} W_j^{cust_n} + 1 & \text{if } j = n \\ \max\{W_j^{cust_n}, U_n^{bw}(W_i^{load}, W_i^{time})\} & \text{otherwise,} \end{cases} \quad (4h)$$

where $U_n^{bw}(W_i^{load}, W_i^{time})$ is a function that sets the unreachability status of customer $n \in N$, i.e., $U_n^{bw}(W_i^{load}, W_i^{time})$ is equal to 1 if $W_i^{load} + q_i + q_n > Q$ or $W_i^{time} - t_{ni} < e_n$, and 0 otherwise. The label $L_i = (W_i^{cost}, W_i^{load}, W_i^{nrch}, W_i^{time}, W_i^{sl}, W_i^{avrt}, W_i^{rt}, (W_i^{cust_n})_{n \in N})$ resulting from this extension is deemed feasible if $W_i^{load} \leq Q - q_i$, $W_i^{nrch} \geq -1$, $W_i^{time} \geq e_i$, $W_i^{avrt} \geq 0$, $W_i^{rt} \leq H$, and $W_i^{cust_n} \leq 1$ for all $n \in N$. If L_i is not feasible, it is deleted.

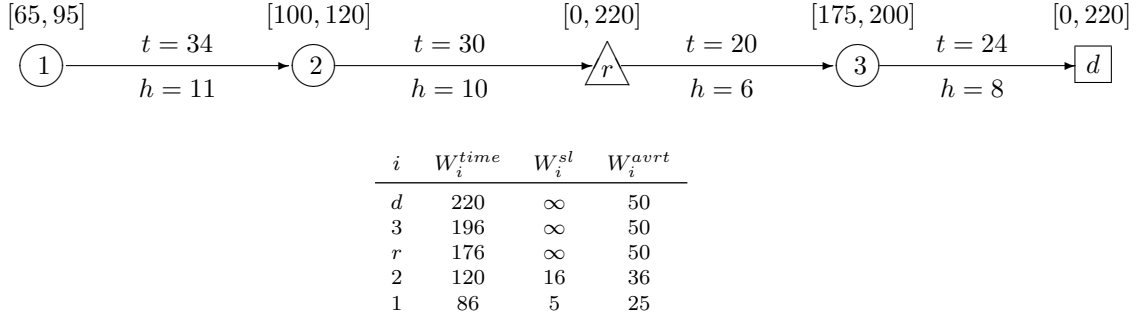


Figure 1: Computation of backward label components for a given path (ESPPRC-SF)

Figure 1 provides an example that describes the interaction between the resources $time$, sl and $avrt$. The top part shows a path from customer 1 to the destination vertex d , that visits customers 2 and 3 as well as a recharging station r . The time windows are depicted above each vertex. Travel time t and required recharging time h are given on each arc, and we assume the time to fully recharge an empty battery to be $H = 50$. The table in the bottom part of the figure indicates the W^{time} , W^{sl} , and W^{avrt} components of the backward labels generated along this path (from d to 1). The first row corresponds to the components of the initial label at vertex d . With the first two extensions, the W^{sl} and W^{avrt} components do not change given that no recharge occurs on the last two vertices. The W^{time} component is only affected by the travel times along these arcs. The extension along arc $(2, r)$ yields $W_2^{time} = \min \{\ell_2, W_r^{time} - t_{2,r} - h_{2,r}\} = \min \{120, 176 - 30 - 10\} = 120$, $W_2^{sl} = \max \{0, W_r^{time} - t_{2,r} - h_{2,r} - \ell_2\} = \max \{0, 176 - 30 - 10 - 120\} = 16$, and $W_2^{avrt} = \min \{H - h_{2,r}, W_r^{time} - t_{2,r} - h_{2,r} - e_2\} = \min \{50 - 10, 176 - 30 - 10 - 100\} = 36$. Indeed, even if the recharge at r was taking up to $W_2^{sl} = 16$ additional units of time, W_2^{time} would still be equal to 120. Furthermore, we observe that the time window at vertex 2 would still be respected even if up to $W_2^{avrt} = 36$ units of recharging time were additionally required at r , in which case W_2^{time} would be set at $e_2 = 100$. The last extension along arc $(1, 2)$ yields $W_1^{time} = \min \{95, 120 - 34 + \min \{0, 16 - 11\}\} = 86$, $W_1^{sl} = \max \{16 - 11, 0\} + \max \{0, 120 - 34 + \min \{0, 16 - 11\} - 95\} = 5$, and $W_1^{avrt} = \min \{36 - 11, 86 - 65 + 5\} = 25$. Here, the W^{time} component reduces only by 34 units along this arc although traveling along it takes 34 units of time and increases the recharging time at r by $h_{12} = 11$ units. This is because the time is compensated by the accumulated slack time which diminishes from 16 to 5. Finally, the W^{avrt} component also drops by $h_{12} = 11$ units to 25. Note that, given that there are 5 units of accumulated slack time, adding 25 units of recharging time at r would move backward W_1^{time} to $86 + 5 - 25 = 66$, which would still be feasible.

All REFs (3) are non-decreasing functions. This is trivial to see for all except the four interdependent resources $nrch$, $time$, sl , and $avrt$. The latter three of these resources depend on $nrch$ due to the distinction between the cases $j \notin R, W_j^{nrch} = 0$ and $j \notin R, W_j^{nrch} < 0$, i.e, the “otherwise” case in (4d), (4e), and (4f). Here, it is straightforward to check that the smaller values $W_j^{nrch} < 0$ produce not greater components W_i^{time} , W_i^{sl} , and W_i^{avrt} in comparison to $W_j^{nrch} = 0$. This is the reason why we defined W^{nrch} as a non-positive component, counting visits to recharging stations negatively. As a result, all four interdependent resources imply non-decreasing REFs. Moreover, they are bounded from below because of the feasibility conditions $W_i^{nrch} \geq -1$, $W_i^{time} \geq e_i$, and $W_i^{avrt} \geq 0$, and the definition of W_i^{sl} , which guarantees that $W_i^{sl} \geq 0$. Hence, the following dominance rule can be applied.

Definition 4.2 Let $L^k = (W_k^{cost}, W_k^{load}, W_k^{nrch}, W_k^{time}, W_k^{sl}, W_k^{avrt}, W_k^{rt}, (W_k^{cust_n})_{n \in N})$, $k \in \{1, 2\}$, be two backward labels associated with paths ending at the same vertex. Label L^2 is said to be dominated by label L^1 if $W_1^r \leq W_2^r$ for all resources $r \in \{cost, load, rt, (cust_n)_{n \in N}\}$, $W_1^r \geq W_2^r$ for $r \in \{nrch, time, sl, avrt\}$, and at least one of these inequalities is strict.

Once the forward labeling and the backward labeling processes are completed, forward and backward labels are joined together to form full o - d -paths. Let $L_i^{fw} = (T_i^{cost}, T_i^{load}, T_i^{rch}, T_i^{time}, T_i^{rt}, (T_i^{cust_n})_{n \in N})$ and $L_i^{bw} = (W_i^{cost}, W_i^{load}, W_i^{nrch}, W_i^{time}, W_i^{sl}, W_i^{avrt}, W_i^{rt}, (W_i^{cust_n})_{n \in N})$ be labels representing a forward and a

backward path ending at vertex i , respectively. Joining these two labels yields an o - d -path with reduced cost $T_i^{cost} + W_i^{cost}$ that is feasible if and only if the labels meet the conditions

$$T_i^{load} + W_i^{load} \leq Q \quad (5a)$$

$$T_i^{rch} - W_i^{nrch} \leq 1 \quad (5b)$$

$$T_i^{cust_n} + W_i^{cust_n} \leq 1 \quad \forall n \in N \setminus \{i\} \quad (5c)$$

as well as the following conditions, which we state according to three distinct cases concerning the merge point.

Case 1: the forward and backward paths are merged at a recharging station, i.e., $i \in R$,

$$T_i^{time} + T_i^{rt} \leq W_i^{time}; \quad (6a)$$

Case 2: the backward path does not include a recharging station, i.e., $i \notin R$ and $W_i^{nrch} = 0$,

$$T_i^{time} \leq W_i^{time} \quad \text{and} \quad T_i^{rt} + W_i^{rt} \leq H; \quad (6b)$$

Case 3: the backward path includes a recharging station, i.e., $i \notin R$ and $W_i^{nrch} < 0$,

$$T_i^{time} + \max\{0, T_i^{rt} - W_i^{sl}\} \leq W_i^{time}, \quad T_i^{rt} + W_i^{rt} \leq H, \quad \text{and} \quad T_i^{rt} \leq W_i^{avrt}. \quad (6c)$$

For the ESPPRC-SF (and for the EVRPTW-SP), we also implemented a simpler bi-directional labeling algorithm in which the forward and backward labels are extended until reaching a recharging station. The labels in both forward and backward steps only need to contain the *cost*, *load*, *time*, *rt* and $(cust_n)_{n \in N}$ components which are computed using simple REFs. Forward and backward labels are merged at recharging stations where the required recharging time can easily be computed. Preliminary tests showed that this approach is not competitive with the one proposed above. The midpoint definition by recharging station is less restrictive than the midpoint definition by the middle of the time horizon. Therefore, many more labels are generated in the forward and backward labeling steps.

Multiple recharge case To solve the ESPPRC-MF, the algorithms described above are modified as follows. For the mono-directional labeling algorithm, the T^{rch} component is not required anymore (including its corresponding tests). For the bi-directional algorithm, the T^{rch} component is not required in the forward labeling step. However, the W^{nrch} component is still needed in the backward step for the REFs (4d)–(4f) and to determine which case applies in conditions (6). On the other hand, the feasibility of a label does not depend on the value of this component and the merging condition (5b) does not have to be considered. Finally, the test on W^{nrch} can be omitted from the dominance rule in Definition 4.2. Given the usage of this component in the backward REFs, one should forbid the dominance of a label with this component equal to 0 by a label with a negative component. However, this situation is already considered by the W^{avrt} component which takes its maximal value H as long as no recharging station is visited. Note finally that all strictly negative values of W_i^{nrch} are equally good in the multiple-recharge case so that we can replace the REF (4c) by $W_i^{nrch} = \max\{-1, W_j^{nrch} - 1\}$ for $j \in R$, and $W_i^{nrch} = W_j^{nrch}$ otherwise, leading to a slightly stronger dominance according to Definition 4.2.

4.1.2 Labeling algorithms for the partial-recharge ESPPRCs

Now, we describe the labeling algorithms for the ESPPRC-SP and ESPPRC-MP variants of subproblem (2). Analogous to the previous section, we first present a mono-directional search algorithm for ESPPRC-SP, then a bi-directional one. Finally, we discuss how these algorithms can be generalized for ESPPRC-MP.

Mono-directional search When partial recharges are allowed, the main difficulty to overcome in a forward labeling algorithm is that the amount of energy to recharge at a station must be determined a posteriori, i.e., based on the travel performed after visiting this station. Consequently, the amount recharged at a station, which can be expressed by means of the corresponding recharging time as described in Section 3, must

be considered as a variable in the labeling algorithm. Therefore, the earliest service start time at vertices following a recharging station becomes a linear function of this variable.

In the proposed forward labeling algorithm for ESPPRC-SP, a partial path p from o to a vertex $i \in V$ is represented by a label $L_i = (T_i^{cost}, T_i^{load}, T_i^{rch}, T_i^{tMin}, T_i^{tMax}, T_i^{rtMax}, (T_i^{cust_n})_{n \in N})$. The definitions of the components T_i^{cost} , T_i^{load} , T_i^{rch} , and $(T_i^{cust_n})_{n \in N}$ are identical to the ones used in Section 4.1.1. In the following, we provide the definitions of the other components:

- T_i^{tMin} : earliest service start time at vertex i assuming that, if a recharging station is visited prior to i along p , a minimum recharge (ensuring battery feasibility up to i) is performed;
- T_i^{tMax} : earliest service start time at vertex i assuming that, if a recharging station is visited prior to i along p , a maximum recharge (ensuring time-window feasibility up to i) is performed;
- T_i^{rtMax} : For the definition of this resource, we make the artificial assumption that recharging is possible at all vertices. This assumption is only used to propagate the information forward along the path, but a real recharge never occurs at a customer. Using the assumption, T_i^{rtMax} denotes the maximum possible recharging time at vertex i assuming that, if a recharging station is visited prior to i along p , a minimum recharge (ensuring battery feasibility up to i) is performed.

If no recharging station is visited along the partial path p , then $T_i^{tMin} = T_i^{tMax}$ is the standard earliest arrival time and T_i^{rtMax} is the cumulated recharging time of the path. Otherwise, $T_i^{tMax} - T_i^{tMin}$ is equal to the maximum recharging time that can be added at the preceding recharge while maintaining path feasibility.

As illustrated in Figure 2, the three components T_i^{tMin} , T_i^{tMax} , and T_i^{rtMax} are sufficient to describe the relationship between the earliest service start time and the maximum possible recharging time. This relationship is depicted by the four line segments, each corresponding to a different label. The labels, which are denoted L^1 to L^4 , are associated with four paths ending at the same vertex and containing one visit to a recharging station (if no recharging visit is contained, the line segment reduces to a single point). For the moment, let us focus on the line segment associated with L^1 . The other segments will be useful when we discuss the dominance rule below. For this segment, the extremities $(T_1^{tMin}, T_1^{rtMax})$ and $(T_1^{tMax}, T_1^{rtMax} - (T_1^{tMax} - T_1^{tMin}))$ correspond to the situation in which a minimum or maximum recharge, respectively, is performed at the preceding recharging station while ensuring path feasibility. Thus, any point between these extremities corresponds to the recharge of an intermediate amount. Observe that spending one additional unit of recharging time reduces the maximum possible recharging time by 1 and increases the earliest service start time by 1, i.e., the slope of the line segment is -1 .

In the initial label at vertex o , all components are set to 0 except T_o^{tMin} and T_o^{tMax} that are both set to e_o . To extend a label $L_i = (T_i^{cost}, T_i^{load}, T_i^{rch}, T_i^{tMin}, T_i^{tMax}, T_i^{rtMax}, (T_i^{cust_n})_{n \in N})$ along an arc $(i, j) \in A$, we use the REFs presented in Section 4.1.1 for computing the components T_j^{cost} , T_j^{load} , T_j^{rch} , and $(T_j^{cust_n})_{n \in N}$ of the resulting label $L_j = (T_j^{cost}, T_j^{load}, T_j^{rch}, T_j^{tMin}, T_j^{tMax}, T_j^{rtMax}, (T_j^{cust_n})_{n \in N})$. The other components T_j^{tMin} , T_j^{tMax} , and T_j^{rtMax} represent the line segment that describes the relationship between the earliest service start time and the maximum possible recharging time. These components can be determined by extending the line segment associated with L_i using the following REFs:

$$T_j^{tMin} = \begin{cases} \max\{e_j, T_i^{tMin} + t_{ij}\} & \text{if } T_i^{rch} = 0 \\ \max\{e_j, T_i^{tMin} + t_{ij}\} + X_{ij}(T_i^{tMin}, T_i^{rtMax}) & \text{otherwise} \end{cases} \quad (7a)$$

$$T_j^{tMax} = \begin{cases} \min\{\ell_j, \max\{e_j, T_i^{tMin} + T_i^{rtMax} + t_{ij}\}\} & \text{if } i \in R \\ \min\{\ell_j, \max\{e_j, T_i^{tMax} + t_{ij}\}\} & \text{otherwise} \end{cases} \quad (7b)$$

$$T_j^{rtMax} = \begin{cases} T_i^{rtMax} + h_{ij} & \text{if } T_i^{rch} = 0 \\ \min\{H, \max\{0, T_i^{rtMax} - S_{ij}(T_i^{tMin})\} + h_{ij}\} & \text{otherwise.} \end{cases} \quad (7c)$$

In these REFs, $S_{ij}(T_i^{tMin}) = \max\{0, e_j - (T_i^{tMin} + t_{ij})\}$ is the slack time between the time window lower bound at vertex j and the earliest arrival time at j . If a recharge is performed before, this slack can be substituted by additional recharging time at the preceding recharging station. Furthermore, $X_{ij}(T_i^{tMin}, T_i^{rtMax}) = \max\{0, \max\{0, T_i^{rtMax} - S_{ij}(T_i^{tMin})\} + h_{ij} - H\}$ is the minimum recharging time

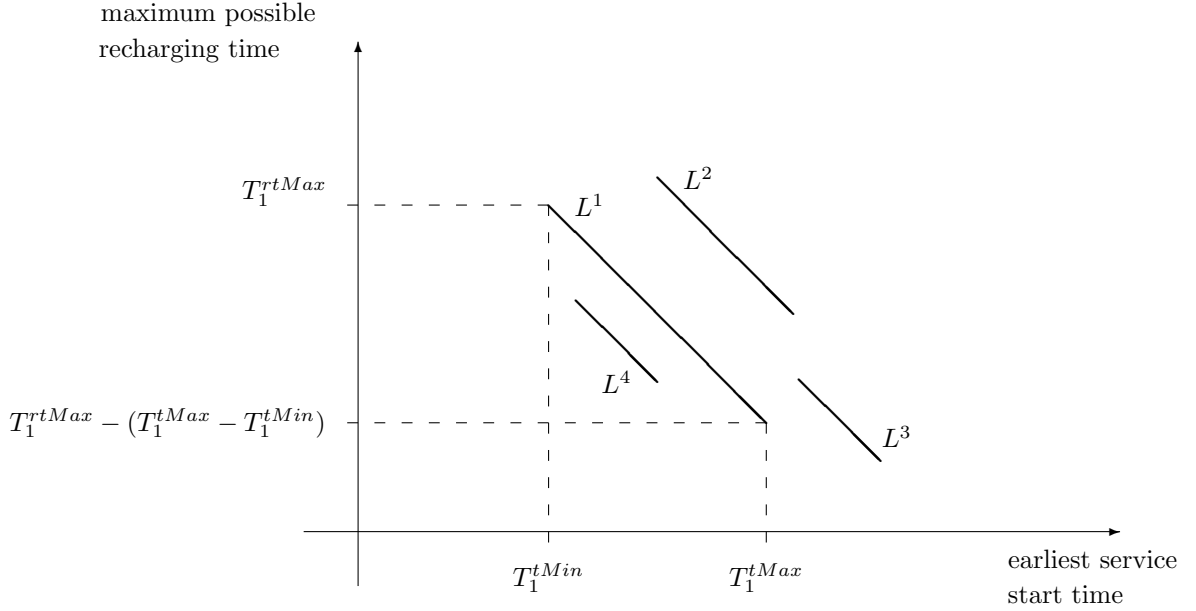


Figure 2: Relationship between earliest service start time and maximum possible recharging time

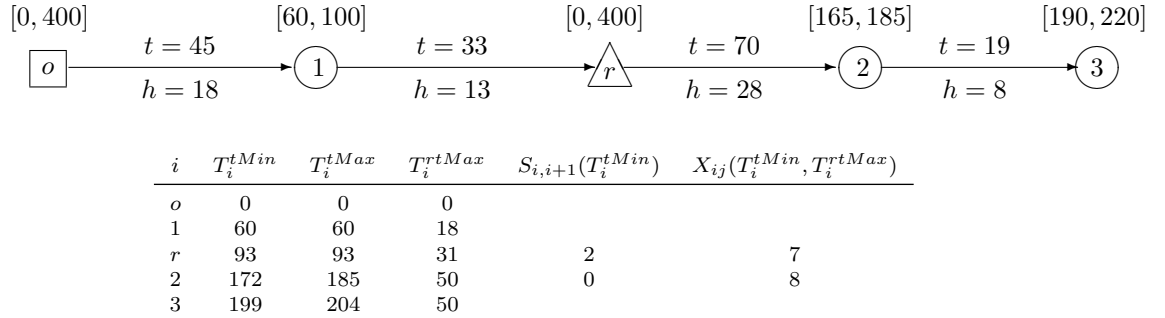


Figure 3: Computation of forward label components for a given path (ESPPRC-SP)

(besides the available slack time) that must be added at the preceding recharging station (if one exists) so that battery capacity is still respected after traveling arc (i, j) .

The resulting label L_j is feasible if $T_j^{load} \leq Q$, $T_j^{rch} \leq 1$, $T_j^{tMin} \leq \ell_j$, $T_j^{tMin} \leq T_j^{tMax}$, $T_j^{rtMax} \leq H$, and $T_j^{cust_n} \leq 1$ for all $n \in N$. Otherwise, L_j is rejected. Note that the condition $T_j^{tMin} \leq T_j^{tMax}$ is violated if either an additional recharging time at the preceding recharging station yields a time window violation or the required recharging time since the recharging station exceeds H . Concerning the latter case, it holds that $T_j^{tMax} - T_j^{tMin} \leq T_j^{rtMax} \leq H$ if $j \in R$ and this difference decreases by at least h_{uv} for each following extension along an arc (u, v) with $u \notin R$. Thus, $T_v^{tMin} \leq T_v^{tMax}$ will be violated at a certain vertex v as soon as the battery capacity is exceeded. If no recharge occurs ($T_i^{rch} = 0$), the condition $T_j^{rtMax} \leq H$ ensures that the battery capacity is respected.

Figure 3 illustrates the computation of the label components T_j^{tMin} , T_j^{tMax} , and T_j^{rtMax} using the proposed REFs for a path $(o, 1, r, 2, 3)$, in which 1, 2 and 3 are customers and r is a recharging station. Time windows, travel times and required recharging times are depicted as in Figure 1. Again, we assume that $H = 50$. The extensions along the arcs $(o, 1)$ and $(1, r)$ are straightforward. At station r , a maximum recharging time of $T^{rtMax} = 31$ may be consumed if the subsequent extensions require it. The extension along $(r, 2)$ consumes a travel time of $t_{r2} = 70$ and a required recharging time of $h_{r2} = 28$. Thus, the earliest arrival time at vertex 2 is $T_r^{tMin} + t_{r2} = 93 + 70 = 163$, providing a slack time of $S_{r2}(93) = \max\{0, 165 - 163\} = 2$ that

reduces the maximum possible recharging time from $T_r^{rtMax} = 31$ to 29 before increasing it to $29 + 28 = 57$. In this case, it exceeds $H = 50$ by $X_{r_2}(93, 31) = \max\{0, \max\{0, 31 - 2\} + 28 - 50\} = 7$. To respect the battery capacity, this extra recharging time must be added to the earliest service start time yielding $T_2^{tMin} = \max\{165, 93 + 70\} + 7 = 172$ and $T_2^{rtMax} = \min\{50, \max\{0, 31 - 2\} + 28\} = 50$. In this extension, the maximum earliest service start time $T_2^{tMax} = \min\{185, \max\{165, 93 + 31 + 70\}\} = 185$ indicates that a maximum additional recharging time of $T_2^{tMax} - T_2^{tMin} = 185 - 172 = 13$ can be added at r without violating the time window at 2. The final extension along arc $(2, 3)$ requires $h_{23} = 8$ of additional time units at r to ensure sufficient battery energy up to vertex 3, yielding $S_{23}(172) = \max\{0, 190 - (172 + 19)\} = 0$, $X_{23}(172, 50) = \max\{0, \max\{0, 50 - 0\} + 8 - 50\} = 8$, $T_2^{tMin} = \max\{190, 172 + 19\} + 8 = 199$, $T_2^{tMax} = \min\{220, \max\{190, 185 + 19\}\} = 204$, and $T_2^{rtMax} = \min\{50, \max\{0, 50 - 0\} + 8\} = 50$. Hence, 8 units of recharging time were added a posteriori at r , which increases the earliest service start time by 8 and reduces the gap between T_3^{tMax} and T_3^{tMin} to $204 - 199 = 5$ units. Overall, at station r , the recharging process takes exactly $S_{r_2}(93) + S_{23}(172) + X_{r_2}(93, 31) + X_{23}(172, 50) = 2 + 0 + 7 + 8 = 17$ time units.

Although all REFs (7) are non-decreasing (to see this, rewrite $-S_{ij}(T_i^{tMin}) = \min\{0, T_i^{tMin} + t_{ij} - e_j\}$ in (7c) and in the definition of X_{ij}), the standard dominance rule cannot be applied because of the feasibility condition $T_j^{tMin} \leq T_j^{tMax}$ and the relationship between the earliest service start time and maximum possible recharging time associated with a path.

A label L^1 dominates a label L^2 if the associated paths p_1 and p_2 end at the same vertex, $T_1^r \leq T_2^r$ for $r \in \{cost, load, rch, tMin, (cust_n)_{n \in N}\}$, and for every service start time $T_2 \in [T_2^{tMin}, T_2^{tMax}]$, there exists a service start time $T_1 \in [T_1^{tMin}, T_2]$ such that $T_1^{rtMax} - (T_1 - T_1^{tMin}) \leq T_2^{rtMax} - (T_2 - T_2^{tMin})$. The latter condition stipulates that, for every maximum possible recharging time $T_2^{rtMax} - (T_2 - T_2^{tMin})$ achievable by p_2 , path p_1 can achieve the same maximum possible recharging time or a lower one at the same service start time T_2 or earlier.

Figure 2 illustrates some examples. In this figure, label L^1 can dominate label L^2 because the above conditions on the components T^{tMin} , T^{tMax} , and T^{rtMax} are respected. However, L^1 cannot dominate L^3 because the path associated with L^3 can achieve a maximum possible recharging time that cannot be achieved by the path associated with L^1 . Moreover, L^1 cannot dominate L^4 because the path associated with L^4 can achieve a maximum possible recharging time that can also be achieved by the path associated with L^1 but only for a later service start time. Following these observations, we can state the dominance rule as follows.

Definition 4.3 Let $L^k = (T_k^{cost}, T_k^{load}, T_k^{rch}, T_k^{tMin}, T_k^{tMax}, T_k^{rtMax}, (T_k^{cust_n})_{n \in N})$, $k \in \{1, 2\}$, be two labels associated with paths ending at the same vertex. Label L^2 is said to be dominated by label L^1 if

$$T_1^r \leq T_2^r \quad \text{for all } r \in \{cost, load, rch, tMin, (cust_n)_{n \in N}\}, \quad (8a)$$

$$T_1^{rtMax} - (T_1^{tMax} - T_1^{tMin}) \leq T_2^{rtMax} - (T_2^{tMax} - T_2^{tMin}), \quad (8b)$$

$$T_1^{rtMax} - (T_2^{tMin} - T_1^{tMin}) \leq T_2^{rtMax}, \quad (8c)$$

and at least one of these inequalities is strict.

Condition (8b) ensures that the minimum value of the maximum possible recharging time that can be achieved by L^1 is less than or equal to that achieved by L^2 . Condition (8c) ensures that the maximum possible recharging time of L^1 for a service start time of T_2^{tMin} is less than or equal to the corresponding maximum possible recharging time of L^2 . These two conditions are equivalent to the requirement stated above: for every service start time $T_2 \in [T_2^{tMin}, T_2^{tMax}]$, there exists a service start time $T_1 \in [T_1^{tMin}, T_2]$ such that $T_1^{rtMax} - (T_1 - T_1^{tMin}) \leq T_2^{rtMax} - (T_2 - T_2^{tMin})$. Note that, if $T_1^{rch} = 0$, condition (8c) is not meaningful. However, in this case, it can only be violated if condition (8b) is also violated.

Bi-directional search For the ESPPRC-SP, the forward step of the bi-directional search labeling algorithm proceeds as described above. We design a backward labeling algorithm that is completely symmetric to the forward labeling algorithm. To this end, we define a reversed ESPPRC instance in which the underlying graph $G = (V, A)$ is replaced by the inverse graph $G' = (V, A')$ with $(i, j) \in A'$ if and only if $(j, i) \in A$, all time windows $[e_i, \ell_i]$ replaced by $[-\ell_i, -e_i]$, and all other coefficient $(c_{ij}, q_i, Q, t_{ij}, h_{ij}, H)$ kept. Moreover,

start and destination nodes are swapped, i.e., $o' = d$ and $d' = o$. We claim that for every feasible route of the given instance, the reversed route is feasible in the reversed instance, and vice versa.

First, this statement is certainly true for routes without recharging. Therefore, we only consider routes with one or several recharges in the following. (We use the plural *stations/recharges* to make the results reusable for the case of multiple recharges discussed later.) Second, in the partial recharge case considered here, it is always possible to reach the destination depot with all energy consumed, i.e., with an empty battery. Otherwise, one could have reduced the amount recharged at the last recharging station (minimum recharges).

Third, for a feasible route p in G with minimum recharges, the reversed route p' is feasible when the same amount of energy is recharged at the recharging stations. To prove the symmetry, assume the route $p = (r_0 - r_1 - r_2 - \dots - r_p - r_{p+1})$ with recharging stations $r_1, r_2, \dots, r_p \in R$ and $r_0 = o$ and $r_{p+1} = d$. Let b_1, b_2, \dots, b_{p+1} be the amount of energy consumed when traveling between $r_0 - r_1, r_1 - r_2, \dots, r_p - r_{p+1}$ and let f_1, f_2, \dots, f_p be the amount of energy recharged at the stations r_1, r_2, \dots, r_p , respectively. Of course, expressing energy consumption as recharging times is also possible here (H and h instead of B and b), however, we use energy consumption for clarity of the exposition. Then, the feasibility of p implies that for each recharging station $r_i, i \in \{1, \dots, p\}$, it holds that

$$0 \leq B - \sum_{j=1}^i b_j + \sum_{j=1}^{i-1} f_j \leq B - \sum_{j=1}^i b_j + \sum_{j=1}^i f_j \leq B,$$

where the left term is the battery level before and the right term the battery level after the visit of the respective recharging station. Reaching the destination depot empty implies $B = \sum_{j=1}^{p+1} b_j - \sum_{j=1}^p f_j$. Replacing B in the above inequality by this term and subtracting the resulting inequality from B yields

$$B \geq B - \sum_{j=i+1}^{p+1} b_j + \sum_{j=i}^p f_j \geq B - \sum_{j=i+1}^{p+1} b_j + \sum_{j=i+1}^p f_j \geq 0, \quad (9)$$

for $i \in \{1, \dots, p\}$. For the reversed route $p' = (r_{p+1} - r_p - \dots - r_2 - r_1 - r_0)$ with recharge quantities f_p, \dots, f_2, f_1 , the left term in inequality (9) is the battery level after visiting recharging station r_i and the right term is the one before reaching r_i . The proven bounds 0 and B ensure the battery feasibility of p' .

Finally, the identical recharging quantities at all recharging stations imply that all service and recharging times at all visits are identical in p and p' . Therefore, a feasible time schedule for p' exists if the route p is feasible.

Based on the discussed symmetry, the backward step of the bi-directional algorithm can be performed analogously to the forward step: A partial path p from a vertex $j \in V$ to d is associated with a label $L_j = (W_j^{cost}, W_j^{load}, W_j^{nrch}, W_j^{tMin}, W_j^{tMax}, W_j^{rtMax}, (W_j^{cust_n})_{n \in N})$. The components $W_j^{cost}, W_j^{load}, W_j^{nrch}$, and $(W_j^{cust_n})_{n \in N}$ are defined as in Section 4.1.1. The three components $W_j^{tMin}, W_j^{tMax}, W_j^{rtMax}$ play the same role as the components T^{tMin}, T^{tMax} and T^{rtMax} for the mono-directional forward labeling algorithm. Details on the initial label, the associated REFs, and the dominance rule can be found in the online appendix.

In the bi-directional algorithm, a forward label is not extended if its T^{tMin} component exceeds an a priori chosen midpoint $M \in [e_o, \ell_d]$. We used $M = (\ell_d - e_o)/2$ for the numerical studies. Accordingly, a backward label is not extended if its W^{tMin} component falls below M .

For the merging step of the bi-directional algorithm, let $L_i^{fw} = (T_i^{cost}, T_i^{load}, T_i^{rch}, T_i^{tMin}, T_i^{tMax}, T_i^{rtMax}, (T_i^{cust_n})_{n \in N})$ and $L_i^{bw} = (W_i^{cost}, W_i^{load}, W_i^{nrch}, W_i^{tMin}, W_i^{tMax}, W_i^{rtMax}, (W_i^{cust_n})_{n \in N})$ be labels representing a forward and a backward path ending at vertex i , respectively. Joining them yields an o - d -path with reduced cost $T_i^{cost} + W_i^{cost}$ that is feasible if and only if the labels satisfy the conditions

$$T_i^{load} + W_i^{load} \leq Q \quad (10a)$$

$$T_i^{rch} - W_i^{nrch} \leq 1 \quad (10b)$$

$$T_i^{cust_n} + W_i^{cust_n} \leq 1, \quad \forall n \in N \setminus \{i\} \quad (10c)$$

$$T_i^{tMin} + Z_i(T_i^{rtMax}, W_i^{rtMax}) \leq W_i^{tMin} \quad (10d)$$

$$i \in R \quad \text{or} \quad Z_i(T_i^{rtMax}, W_i^{rtMax}) \leq (T_i^{tMax} - T_i^{tMin}) + (W_i^{tMin} - W_i^{tMax}), \quad (10e)$$

where $Z_i(T_i^{rtMax}, W_i^{rtMax}) = \max\{0, T_i^{rtMax} + W_i^{rtMax} - H\}$ is the minimum recharging time to add at a recharging station to ensure that battery capacity is respected along the whole path. The interpretation of the conditions (10a)–(10c) is straightforward. Condition (10d) ensures that the time windows of the visited customers are met after adding the required minimum recharging time $Z_i(T_i^{rtMax}, W_i^{rtMax})$. Finally, condition (10e) stipulates that the minimum recharging time, if any, is available at the visited recharging station.

Multiple recharge case To handle the ESPPRC-MP, modifications must be made to the algorithms described above. For the forward mono-directional labeling algorithm, the definitions of the T^{tMin} , T^{tMax} , and T^{rtMax} components must include the possibility to have multiple visits to recharging stations, all with either minimum recharges or maximum recharges. Furthermore, the upper bound on the T^{rch} component is no longer necessary nor its corresponding test in the dominance rule stated in Definition 4.3. However, the component T^{rch} is needed to determine which case of the REFs (7) must be applied. For the bi-directional algorithm, the modifications described above are also valid for the forward labeling process. For the backward labeling phase, analogous modifications must be made for the W^{tMin} , W^{tMax} , and W^{rtMax} components. Finally, the merging condition (10b) must be dropped.

4.1.3 Acceleration strategies

We use two strategies to acceleration the solution process. Both aim at reducing the time spent solving the subproblem, which is NP-hard for all EVRPTW variants due to the elementarity requirements on the customers. The first strategy consists of relaxing the subproblem by allowing the generation of routes containing cycles, i.e., that visit a customer more than once. Several route relaxations relying on this principle have been proposed for the VRPTW (see Desaulniers et al., 2014a). Among them, the ng -route relaxation introduced by Baldacci et al. (2011) currently seems to be the most effective.

For the EVRPTW, we use the following ng -route relaxation. For each vertex $i \in N \cup R$, we define a neighborhood $NG_i \subset N$ that contains i and the ν closest customers to i which can be visited before i , where $\nu < |N|$ is a predefined parameter (set to 6 in the numerical studies). An ng -route allows to visit a customer i twice (or more often) if it visits at least one vertex j in between two visits to i such that $i \notin NG_j$. On the one hand, considering ng -routes in the set Ω (while redefining the parameter a_{pi} to be the number of times customer i is visited in route p) may yield weaker lower bounds to model (1). On the other hand, the subproblem becomes easier to solve if ν is sufficiently small. To address the subproblem, the computation of the $(T^{cust_n})_{n \in N}$ and $(W^{cust_n})_{n \in N}$ components in the algorithms presented above must depend on the neighborhoods NG_i , $i \in N \cup R$ (for details, see Desaulniers et al., 2014a).

The second acceleration strategy is to rapidly generate negative reduced cost columns using a graph of reduced size. More precisely, at each iteration of the column generation algorithm, the labeling algorithm is executed first on a simplified graph G that contains only a subset A' of the arcs in A . If it fails to find negative reduced cost columns, then the algorithm is executed again, but on the complete graph G . As suggested in (Desaulniers et al., 2008), the subset A' varies in each iteration: the arcs in A' are selected based on the arc modified costs \bar{c}_{ij} , $(i, j) \in A$, which depend on the current values of the dual variables of the RMP. First, for every vertex $i \in N \cup R$, we sort separately all incoming arcs and all outgoing arcs in increasing order of their modified cost and put them in separate ordered lists denoted I_i and O_i , respectively. An arc (i, j) is removed from A if (i) $i, j \in N \cup R$, (ii) the rank of (i, j) in list I_j is greater than a predefined parameter μ , and (iii) the rank of (i, j) in list O_i is also greater than μ . Thus, A' contains all arcs leaving the origin o , all arcs entering the destination d , and, for every vertex, at least μ incoming and μ outgoing arcs (unless there exist less than μ of these arcs initially). In the numerical studies, we used $\mu = 3$.

4.2 Cutting planes

To strengthen the linear relaxations in the branch-and-bound search tree, two types of cutting planes are applied: 2-path cuts and subset row inequalities. 2-path cuts were introduced by Kohl et al. (1999) for the VRPTW and can be defined as follows. Let $W \subseteq N \cup R$ be a subset of vertices in V that includes at least one customer and it is not possible to serve all customers in W on the same route, i.e., at least two vehicles are required to serve them. The corresponding 2-path inequality is given by $\sum_{p \in \Omega} n_p^W \theta_p \geq 2$, where n_p^W is equal to the number of times that route p enters into set W , i.e., the number of arcs $(i, j) \in A$ traversed in p such that $i \notin W$ and $j \in W$. To separate these inequalities, we first enumerate all subsets $W \subseteq N \cup R$ such that $|W|$ is less than or equal to a given parameter (we used a value of 10 in our studies), the flow entering W is less than two, and the vertices in W are connected in the support graph of the current linear relaxation solution. If $\sum_{i \in N \cap W} q_i > Q$, then a violated inequality is found. If not, we solve an elementary shortest path problem with time windows (ESPPTW) over $W \cup \{o, d\}$ with all costs set to -1 (this is equivalent to solving a TSP with time windows) to determine whether it is possible to visit all customers in W using a single route while respecting the customer time windows (Desaulniers et al., 2008). If this is not possible, a violated inequality is found. We also implemented a version of the separation algorithm that considered an ESPPTW with battery capacity. However, this approach turned out to be less effective as not many more cuts were found but separation time increased substantially. The violated 2-path inequalities that are identified are added to the RMP. The dual variable associated with the 2-path cut for subset W must be subtracted from \bar{c}_{ij} for all arcs $(i, j) \in A$ with $i \notin W$ and $j \in W$.

Subset row inequalities (Jepsen et al., 2008) are Chvátal-Gomory inequalities of rank 1 defined over subsets of the constraints (1b). As in (Jepsen et al., 2008; Desaulniers et al., 2008), we consider only the subsets involving three constraints (1b). Let $W \subset N$ be a subset of three customers. The corresponding subset row inequality is $\sum_{p \in \Omega} m_p^W \theta_p \leq 1$, where $m_p^W = \lfloor \beta_p^W / 2 \rfloor$ and β_p^W is equal to the number of visits to a customer in W along route p . If p is elementary, then m_p^W is equal to 1 if p visits two or three customers in W and 0 otherwise. In this case, the inequality specifies that at most one route visiting two or three customers in W can be part of a feasible integer solution. The subset row inequalities are separated by enumerating all subsets of three customers and checking for each subset whether the corresponding inequality is violated. Violated inequalities are added to the RMP. Contrary to the 2-path cuts, the dual variables associated with the subset row cuts cannot be integrated into the modified arc costs \bar{c}_{ij} . When solving the subproblem by a labeling algorithm, the dual variable associated with a subset row cut must be subtracted from the reduced cost of a partial path each time that it visits two customers defining the cut. In consequence, the labels must contain an additional component for each cut to count the number of visits to the associated customers. Accordingly, the dominance rule must be modified to take into account these additional components (for details, see Jepsen et al., 2008; Desaulniers et al., 2008, 2011).

The subset row cuts have proved to be very efficient for the VRPTW, substantially reducing the number of nodes to explore in the search tree and the total computational times. However, the treatment of their dual variables in the subproblem increases the difficulty of solving the subproblem to optimality. Therefore, we only separate these cuts if no violated 2-path cuts are found.

4.3 Branching

To derive integer solutions, we impose the following types of branching decisions in the branch-and-bound search tree: (i) on the total number of routes, (ii) on the total number of recharges, (iii) on the total number of recharges at a given recharging station, and (iv) on the total flow on an arc of graph G . Given a fractional-valued solution, these types of decisions are evaluated in the given order and the first type that can be imposed is selected. If the total number of recharges is fractional for several stations, we choose to branch on a station for which the fractional part of its total number of recharges is closest to 0.5. Similarly, if the arc flow is fractional for several arcs, we choose an arc for which the fractional part of its flow is closest to 0.5.

For every decision, two branches are created. Decisions of the first three types are imposed by adding an inequality to the RMP. The dual variable of this inequality alters the reduced cost of certain route variables. For the fourth decision type, the decisions are imposed in a different manner by removing arcs from the graph

G in the route-generation step. Moreover, all routes in RMP incompatible with this decision are removed. Details of implementing these decisions are discussed, e.g., in (Desaulniers et al., 1998, 2005).

The branch-and-bound search tree is explored using the local depth-first search strategy introduced in (Desaulniers et al., 2014a). This strategy chooses a node in the tree using the best-first criterion and explores, possibly partially, the subtree rooted at this node using a depth-first strategy. A node is evaluated in this subtree if the gap between the lower bound at its father node and the best lower bound of the unexplored nodes is within a given tolerance (we used a value of 10 in our studies). When the local exploration of the subtree is completed, the search strategy chooses the next node to explore using the best-first criterion before locally exploring its subtree. This process is repeated until completing the search tree exploration. This hybrid search strategy allows relatively fast linear relaxation reoptimizations in the depth-first phase and limits the total number of nodes evaluated using the best-first criterion.

5 Computational studies

In this section, we present computational experiments to analyze the effectiveness of the proposed branch-price-and-cut algorithms (Section 5.2), and to assess the benefits of allowing multiple and partial recharges, i.e., to compare the four EVRPTW problem variants (Section 5.3). Section 5.1 describes the instance sets used in our experiments. All algorithms were implemented in C/C++, compiled with GCC 4.4.7 using the CPLEX 12.4 library for solving linear programs. The experiments were performed on a standard PC with an Intel(R) Core(TM) i7-4770 CPU at 3.40 GHz, 16 GB of RAM, and running Linux, kernel version 3.8.13. The parameters of the algorithms were calibrated in preliminary studies and their final values have been presented in Section 4.

5.1 Benchmark instances

The EVRPTW benchmark set was introduced in (Schneider et al., 2014a) and is based on the VRPTW benchmark set of Solomon (1987). To each of the 100-customer Solomon instances, Schneider et al. (2014a) apply the following modifications to obtain an EVRPTW instance: (i) 21 randomly generated recharging stations are added; (ii) the battery capacity is suitably set; and (iii) the time windows of some customers are enlarged to ensure feasibility. The energy consumption b_{ij} along an arc $(i, j) \in A$ is set equal to the arc cost c_{ij} , and the proportionality factor α is chosen such that a complete battery recharge requires three times the average customer service time of the considered instance.

In our experiments, we only use the groups R1, C1, and RC1 of the EVRPTW instances, which are characterized by narrow time windows. The instances of the groups R2, C2, and RC2 have wide time windows and are not interesting for our analysis because the time window constraints can easily be satisfied and thus have only a minor influence on the recharging decisions. The resulting set of 29 instances could have been used as a test set for the EVRPTW-MF and the EVRPTW-MP without modifications. However, some of these instances are infeasible for the EVRPTW-SF and the EVRPTW-SP because two recharges are needed to visit certain customers. Therefore, we modify these instances by randomly relocating the critical customers closer to the depot. We additionally generate two sets of small and medium-sized instances, which are obtained by randomly extracting 25 and 50 customers, respectively, from each 100-customer instance and keeping the 21 recharging stations. Consequently, we end up with $3 \cdot 29 = 87$ instances.

5.2 Algorithmic performance

First, we investigate the performance of the different branch-price-and-cut algorithms with mono-directional (M) and bi-directional (B) labeling algorithms for the four EVRPTW variants (SF, SP, MF, MP). For the studies, the maximum run-time was set to 1 hour per instance. For each combination of solution algorithm and problem variant, Table 1 reports aggregated results over the sets of instances with the same size (given by the number of customers $|N|$): the number of instances solved to proven optimality within the time limit ($\#Opt$), and in the following columns, averages over these solved instances for the run-time in seconds (t), the relative integrality gap in percent (Δ), the percentage of this gap that is closed by the cuts (CC), the number

of 2-path cuts (#2PC), the number of subset row cuts (#SRC), and the number of branch-and-bound nodes explored (#Nodes). In addition, further aggregated results at different levels are provided.

Table 1: Performance of the branch-price-and-cut algorithms with mono-directional and bi-directional labeling for the different problem variants.

Variant	N	#Opt	t[s]	Δ [%]	CC[%]	#2PC	#SRC	#Nodes
Mono-directional								
SF	25	29/29	2.81	0.85	95.03	2.97	9.62	2.83
	50	26/29	147.55	1.48	88.00	5.96	52.62	21.12
	100	10/29	359.79	0.92	82.04	35.60	81.10	60.40
	<i>all</i>	65/87	115.63	1.12	89.96	9.18	37.82	19.00
SP	25	29/29	2.59	0.66	93.48	2.86	7.07	2.66
	50	25/29	130.20	1.37	90.77	6.88	39.84	13.64
	100	9/29	247.58	0.91	88.18	38.89	43.44	12.67
	<i>all</i>	63/87	88.23	0.98	91.40	9.60	25.27	8.44
MF	25	28/29	9.28	0.90	95.46	2.21	9.14	3.07
	50	27/29	136.77	1.09	93.43	7.59	58.78	54.37
	100	8/29	368.88	1.88	92.83	59.50	55.88	26.63
	<i>all</i>	63/87	109.58	1.11	94.15	11.79	36.35	28.05
MP	25	29/29	24.44	0.80	98.55	3.21	9.24	2.79
	50	23/29	371.87	0.99	84.36	7.00	108.87	155.65
	100	8/29	690.16	1.53	91.98	36.63	66.75	19.63
	<i>all</i>	60/87	246.38	0.97	91.53	9.12	55.10	63.63
<i>all</i>	<i>all</i>	251/348	138.49	1.05	91.76	9.93	38.43	29.29
Bi-directional								
SF	25	29/29	2.96	0.85	95.03	2.97	9.10	2.83
	50	27/29	169.76	1.56	86.88	4.89	52.59	20.19
	100	11/29	356.55	1.05	81.52	30.45	102.27	64.55
	<i>all</i>	67/87	128.23	1.17	89.23	8.25	41.93	19.96
SP	25	29/29	2.82	0.66	93.48	2.86	7.79	2.76
	50	27/27	112.90	1.50	90.33	6.11	48.33	20.07
	100	12/29	657.49	1.04	84.90	32.50	101.50	44.50
	<i>all</i>	68/87	162.06	1.06	90.33	9.38	40.43	17.00
MF	25	29/29	27.51	1.00	95.23	2.14	10.86	3.38
	50	27/29	60.46	1.09	92.32	7.52	61.04	62.96
	100	11/29	595.17	1.90	90.60	49.45	129.73	113.55
	<i>all</i>	67/87	133.99	1.18	93.14	12.07	50.60	45.48
MP	25	29/29	7.08	0.80	98.93	3.21	8.97	2.79
	50	26/29	380.55	1.17	85.68	7.08	85.08	158.38
	100	10/29	419.58	1.67	90.72	32.50	87.40	36.90
	<i>all</i>	65/87	219.93	1.08	91.62	9.26	51.48	70.28
<i>all</i>	<i>all</i>	267/348	160.61	1.13	91.07	9.75	46.04	37.86
M and B	<i>all</i>	<i>all</i>	267/348					

The results show that the bounds obtained at root node are tight with an average integrality gap of approximately 1.1%. On average, 91% of this gap is closed by the cutting planes. Many more subset row cuts are generated than 2-path cuts. The resulting gaps allow to prove optimality by exploring a relatively small number of branch-and-bound nodes. Overall, 518 out of 696 instances are solved within an average run-time of approximately 150 seconds. More precisely, more than 98% of the instances with 25 customers, approximately 90% of the instances with 50 customers, and approximately 27% of the 100-customer instances are solved. Moreover, we find that the EVRPTW-MP is the most challenging problem variant for both types of algorithms (M and B). This can be explained by the fact that the feasible space for this variant is significantly larger, which makes the subproblem harder to solve.

In addition, we are interested in a direct comparison of the performance of the branch-price-and-cut algorithms relying on mono-directional and bi-directional labeling. To this end, Table 2 reports the following aggregated results for each combination of problem variant and set of test instances of the same size: the number of instances solved by both types of branch-price-and-cut algorithms ($\#Common$) and the difference between the number of instances that can be solved by algorithms of type B and of type M ($\#Opt_B^+$). In the last two columns, averages over the instances solved to optimality are provided for the run-time in seconds obtained by the respective algorithm of type M (t_M), and for the relative deviation of the run-time of the algorithm of type B (Δt_B), which is calculated as $100 \cdot (t_B - t_M)/t_B$, where t_B (t_M) is the run-time of the algorithm of type B (M).

Table 2: Comparison between the algorithms using mono-directional and bi-directional labeling.

Variant	$ N $	$\#Common$	$\#Opt_B^+$	t_M [s]	Δt_B [%]
SF	25	29	0	2.81	13.61
	50	26	1	147.55	-55.51
	100	10	1	359.79	-53.01
	<i>all</i>	65	2	115.63	-24.29
SP	25	29	0	2.59	17.75
	50	25	2	130.20	-69.10
	100	9	3	247.58	-157.08
	<i>all</i>	63	5	88.23	-41.69
MF	25	28	1	9.28	-41.14
	50	27	0	136.77	-57.74
	100	8	3	368.88	-146.53
	<i>all</i>	63	4	109.58	-61.64
MP	25	29	0	24.44	-15.55
	50	23	3	371.87	-75.12
	100	8	2	690.16	-112.56
	<i>all</i>	60	5	246.38	-51.32
<i>all</i>	<i>all</i>	251	16	138.49	-44.49

The results indicate that, overall, the algorithms using bi-directional labeling are more effective than the algorithms with mono-directional labeling: on average, they are nearly twice as fast and are able to solve more instances to optimality, namely 267 instead of 251 out of the 348 test instances. Furthermore, all instances that were solved by algorithms of type M were also solved by algorithms of type B (this is additional information that cannot be derived from the table). However, we note that the advantage is rather slight for the small instances with 25 customers. Here, the algorithms of type B can only solve one additional instance for variant MF, but the average run-times are longer for variants SF and SM.

5.3 Effect of allowing partial and multiple recharges

We investigate the effects of allowing partial recharges by comparing the results obtained for the problem variants restricted to full recharges to those obtained with partial recharges. Because the algorithms relying on bi-directional labeling are able to solve more instances, we base the analysis on the respective results.

Table 3 reports aggregate results for each combination of instance set (with instances of the same size, i.e., $|N| = 25, 50, 100$) and number of recharges allowed per route in the considered problem variant (single vs. multiple). Column “ $\#Common$ ” reports the number of instances that were solved to optimality for both problem variants with partial (P) and full (F) recharges. The averages reported in the following columns are based on the solved instances. Here, the average cost (Cost), number of vehicles ($\#Vehicles$) and number of recharges per vehicle ($\#Rech./Veh.$) are reported for variant F. The results for variant P are given as percentage deviation from the result of variant F (Δ_P) and are computed in analogous fashion to Δt_B in Table 2. Moreover, aggregations at several levels are reported.

Table 3: Comparison between the results obtained for the full-recharge and partial-recharge variants.

#Rech. allowed	N	#Common	Cost		#Vehicles		#Rech./Veh.	
			F	Δ_P [%]	F	Δ_P [%]	F	Δ_P [%]
single	25	29	514.55	-0.93	5.41	-0.86	0.83	0.69
	50	26	809.65	-1.05	8.58	-4.42	0.82	4.30
	100	9	1177.54	-0.88	13.78	-0.47	0.63	-4.50
	<i>all</i>	64	727.67	-0.97	7.88	-2.25	0.79	1.43
multiple	25	29	505.67	-1.72	5.21	-2.70	1.06	5.15
	50	25	778.24	-2.04	8.32	-3.65	1.01	10.26
	100	8	1316.23	-2.21	15.00	-8.27	0.88	15.73
	<i>all</i>	62	720.17	-1.91	7.73	-3.80	1.02	8.57
<i>all</i>	<i>all</i>	126	723.98	-1.43	7.80	-3.02	0.90	4.94

For the cases with single recharge per route, we observe that allowing partial recharges leads to an average reduction of the routing costs of 0.97% and a decrease in the number of vehicles used of 2.25%. These savings are likely to happen because partial recharges allow to reduce recharging times in order to meet customer time windows. For the cases with multiple recharges per route, a similar but stronger tendency can be observed: on average, the routing costs reduce by 1.91%, the number of vehicles by 3.80%, and the number of recharges per route increases by 8.57%. In fact, combining the multiple-recharge option with partial recharges offers a much higher degree of flexibility resulting in lower costs and in more frequent but shorter recharging operations.

Next, we investigate the effect of allowing multiple recharges by comparing the results of the multiple-recharge variants to those of the single-recharge variants. Again, we base the analysis on the solutions of the algorithms relying on bi-directional labeling. Table 4 reports our findings in an analogous fashion to Table 3: now, results are aggregated according to the type of recharge (full vs. partial), and the differences between the variants with single (S) and multiple (M) recharges per route are reported.

Table 4: Comparison between the results obtained for the single-recharge and multiple-recharge variants.

#Rech. type	N	#Common	Cost		#Vehicles		#Rech./Veh.	
			S	Δ_M [%]	S	Δ_M [%]	S	Δ_M [%]
full	25	29	514.55	-1.65	5.41	-4.34	0.83	17.98
	50	26	806.44	-3.29	8.58	-3.20	0.82	17.07
	100	9	1218.76	-1.48	14.33	-1.30	0.65	19.31
	<i>all</i>	64	732.16	-2.29	7.95	-3.45	0.80	17.80
partial	25	29	509.82	-2.46	5.38	-6.18	0.83	22.64
	50	25	797.17	-3.79	8.20	-2.38	0.86	21.36
	100	7	1275.72	-2.43	14.57	-5.39	0.66	36.28
	<i>all</i>	61	715.47	-3.00	7.59	-4.53	0.82	23.68
<i>all</i>	<i>all</i>	125	724.02	-2.64	7.78	-3.98	0.81	20.67

We observe that the flexibility introduced by allowing multiple recharges per route yields, on average, a decrease of 2.64% in the routing costs and of 3.98% in the number of vehicles used, while the average number of recharges per vehicle increases considerably (by 20.67%). These gains, which are larger for the partial recharge case, are due to the possibility of scheduling the required recharges at several occasions along a route, exactly when spare time is available before serving customers. Compared to allowing partial recharges, allowing multiple recharges per route is clearly more profitable. On the other hand, this option might be less practical.

6 Conclusions

In this paper, we present effective branch-price-and-cut algorithms for four variants of the EVRPTW, which are defined according to the maximal number of recharges per route (single vs. multiple) and the type of recharge (partial vs. full). For each problem variant, mono-directional and bi-directional labeling algorithms for generating feasible routes are presented. Their efficiency results from complex REFs, that allow for constant time feasibility checking, and strong dominance rules. On average, only relatively few branch-and-bound nodes need to be explored due to the utilization of state-of-the-art techniques for reducing the integrality gap: almost elementary routes are generated using the ng-route relaxation and the remaining integrality gap is closed to a large extent by applying an adaptation of 2-path cuts and subset-row inequalities.

In numerical studies, we demonstrate that our algorithms are capable of solving instances with up to 100 customers and 21 recharging stations for each of the problem variants. Moreover, we show that, especially for larger instances with 50 and 100 customers, the bi-directional labeling is superior compared to the mono-directional one. Finally, we find that allowing multiple as well as partial recharges both help to reduce routing costs and the number of employed vehicles in comparison to the variants with single and with full recharges.

Appendix

A Backward labeling algorithm for the partial-recharge ESPPRCs

Recall from Section 4.1.2 that the backward labeling step for the partial-recharge ESPPRCs uses labels $L_j = (W_j^{cost}, W_j^{load}, W_j^{nrch}, W_j^{tMin}, W_j^{tMax}, W_j^{rtMax}, (W_j^{cust_n})_{n \in N})$, in which the components $W_j^{tMin}, W_j^{tMax}, W_j^{rtMax}$ play the same role as the components T^{tMin}, T^{tMax} and T^{rtMax} in the mono-directional forward labeling algorithm. We argued using the formal device of the reversed ESPPRC instance with inverse graph $G' = (V, A')$. Basically, the reversed instance uses reversed arcs and the negative values of all data related to points in time. This includes that time windows are defined as $[-\ell_i, -e_i]$ for $i \in V$. Then, we use the results obtained from the reversed instance and, in turn, replace all time related resources by their negative. This way, we derive the final formulas.

Hence, the components of the initial label at vertex d are set to 0 except for $W_d^{tMin} = W_d^{tMax} = \ell_d$. Further, we define the REFs for extending a label $L_j = (W_j^{cost}, W_j^{load}, W_j^{nrch}, W_j^{tMin}, W_j^{tMax}, W_j^{rtMax}, (W_j^{cust_n})_{n \in N})$ along an arc $(i, j) \in A$. The components $W_j^{cost}, W_j^{load}, W_j^{nrch}$, and $(W_j^{cust_n})_{n \in N}$ are not related to points in time and can therefore be propagated using the REFs (4a)–(4c) and (4h). For the other components, we use the forward REFs (7) for the reversed instance with the following modifications: we swap e and ℓ as well as i and j , multiply the time-related components T_i^{tMin} and T_i^{tMax} by -1 (note that T_i^{rtMax} is a duration and not a point in time and is therefore not multiplied by -1), use the equalities $-\max\{a, b\} = \min\{-a, -b\}$ and $-\min\{a, b\} = \max\{-a, -b\}$, and replace the symbols $-T^{tMin}$ by W^{tMin} , $-T^{tMax}$ by W^{tMax} , and $+T^{rtMax}$ by W^{rtMax} . The following REFs result

$$W_i^{tMin} = \begin{cases} \min\{\ell_i, W_j^{tMin} - t_{ij}\} & \text{if } W_j^{nrch} = 0 \\ \min\{\ell_i, W_j^{tMin} - t_{ij}\} - \hat{X}_{ij}(W_j^{tMin}, W_j^{rtMax}) & \text{otherwise} \end{cases} \quad (11a)$$

$$W_i^{tMax} = \begin{cases} \max\{e_i, \min\{\ell_i, W_j^{tMin} - W_j^{rtMax} - t_{ij}\}\} & \text{if } j \in R \\ \max\{e_i, \min\{\ell_i, W_j^{tMax} - t_{ij}\}\} & \text{otherwise} \end{cases} \quad (11b)$$

$$W_i^{rtMax} = \begin{cases} W_j^{rtMax} + h_{ij} & \text{if } W_j^{nrch} = 0 \\ \min\{H, \max\{0, W_j^{rtMax} - \hat{S}_{ij}(W_j^{tMin})\} + h_{ij}\} & \text{otherwise,} \end{cases} \quad (11c)$$

with $\hat{S}_{ij}(W_j^{tMin}) = \max\{0, W_j^{tMin} - t_{ij} - \ell_i\}$ and $\hat{X}_{ij}(W_j^{tMin}, W_j^{rtMax}) = \max\{0, \max\{0, W_j^{rtMax} - \hat{S}_{ij}(W_j^{tMin})\} + h_{ij} - H\}$.

The resulting label L_i is feasible if $W_i^{load} \leq Q$, $W_i^{nrch} \geq -1$, $W_i^{tMin} \geq e_i$, $W_i^{tMin} \geq W_i^{tMax}$, $W_i^{rtMax} \leq H$, and $W_i^{cust_n} \leq 1$ for all $n \in N$. Otherwise, L_i is rejected. Based on Definition 4.3 we can get the analogous dominance rule:

Definition A.1 Let $L^k = (W_k^{cost}, W_k^{load}, W_k^{nrch}, W_k^{tMin}, W_k^{tMax}, W_k^{rtMax}, (W_k^{cust_n})_{n \in N})$, $k \in \{1, 2\}$, be two labels associated with paths ending at the same vertex. Label L^2 is said to be dominated by label L^1 if

$$W_1^r \leq W_2^r \quad \text{for all } r \in \{cost, load, (cust_n)_{n \in N}\}, \quad (12a)$$

$$W_1^r \geq W_2^r \quad \text{for all } r \in \{nrch, tMin\}, \quad (12b)$$

$$W_1^{rtMax} - (W_1^{tMin} - W_1^{tMax}) \leq W_2^{rtMax} - (W_2^{tMin} - W_2^{tMax}), \quad (12c)$$

$$W_1^{rtMax} - (W_1^{tMin} - W_2^{tMin}) \leq W_2^{rtMax}, \quad (12d)$$

and at least one of these inequalities is strict.

Conditions (12c) and (12d) play the same role as conditions (8b) and (8c) in Definition 4.3.

References

- R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59:1269–1283, 2011.
- C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.
- C. Bode and S. Irnich. In-depth analysis of pricing problem relaxations for the capacitated arc-routing problem. *Transportation Science*, 2014. doi: 10.1287/trsc.2013.0507.
- R. G. Conrad and M. A. Figliozzi. The recharging vehicle routing problem. In T. Doolen and E. Van Aken, editors, *Proceedings of the 2011 Industrial Engineering Research Conference*, Reno, NV, 2011.
- B. Crevier, J.-F. Cordeau, and G. Laporte. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2):756–773, 2007.
- B.A. Davis and M.A. Figliozzi. A methodology to evaluate the competitiveness of electric delivery trucks. *Transportation Research Part E: Logistics and Transportation Review*, 49(1):8–23, 2013.
- G. Desaulniers. Branch-and-price-and-cut for the split delivery vehicle routing problem with time windows. *Operations Research*, 58(1):179–192, 2010.
- G. Desaulniers, J. Desrosiers, I. Ioachim, M. Solomon, F. Soumis, and D. Villeneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 57–93. Kluwer, Norwell, MA, 1998.
- G. Desaulniers, J. Desrosiers, and M. Solomon. *Column generation*. Springer, New York, NY, 2005.
- G. Desaulniers, F. Lessard, and A. Hadjar. Tabu search, partial elementarity, and generalized k -path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42:387–404, 2008.
- G. Desaulniers, J. Desrosiers, and S. Spoorendonk. Cutting planes for branch-and-price algorithms. *Networks*, 58(4):301–310, 2011.
- G. Desaulniers, O. Madsen, and S. Ropke. The vehicle routing problem with time windows. In P. Toth and D. Vigo, editors, *Vehicle routing: Problems, methods, and applications*, MOS-SIAM Series on Optimization, chapter 5, pages 119–159. SIAM, 2nd edition, 2014a.
- G. Desaulniers, J.G. Rakke, and L.C. Coelho. A branch-price-and-cut algorithm for the inventory-routing problem. *Les Cahiers du GERAD G-2014-19*, GERAD, HEC Montréal, Canada, 2014b.
- J. Desrosiers and M. Lübbecke. A primer in column generation. In G. Desaulniers, J. Desrosiers, and M. Solomon, editors, *Column Generation*, chapter 1, pages 1–32. Springer, 2005.
- S. Erdogan and E. Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012.
- Á. Felipe, M.T. Ortuño, G. Righini, and G. Tirado. 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, 2014.
- D. Goeke and M. Schneider. Routing a mixed fleet of electric and conventional vehicles. Technical Report LPIS-02/2014, Logistics Planning and Information Systems, TU Darmstadt, 2014.
- G. Hiermann, J. Puchinger, and R.F. Hartl. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. Working Paper, 2014. URL http://prolog.univie.ac.at/research/publications/downloads/Hie_2014_638.pdf.
- I. Ioachim, J. Desrosiers, F. Soumis, and N. Bélanger. Fleet assignment and routing with schedule synchronization constraints. *European Journal of Operational Research*, 119:75–90, 1999.

- S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, and M. Solomon, editors, *Column Generation*, chapter 2, pages 33–65. Springer, 2005.
- M. Jepsen, B. Petersen, S. Spoorendonk, and D. Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56:497–511, 2008.
- A. A. Juan, J. Goentzel, and T. Bektaş. Routing fleets with multiple driving ranges: Is it possible to use greener fleet configurations? *Applied Soft Computing*, 21:84–94, 2014.
- N. Kohl, J. Desrosiers, O.B.G. Madsen, M.M. Solomon, and F. Soumis. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33(1):101–116, 1999.
- G. Laporte, Y. Nobert, and M. Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5):1050–1073, 1985.
- M. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.
- H.-Y. Mak, Y. Rong, and Z.-J.M. Shen. Infrastructure planning for electric vehicles with battery swapping. *Management Science*, 59(7):1557–1575, 2014.
- S.A. MirHassani and R. Ebrazi. A flexible reformulation of the refueling station location problem. *Transportation Science*, 47(4):617–628, 2013.
- A. Montoya, C. Guéret, J.E. Mendoza, and J.G. Villegas. A modified multi-space sampling heuristic for the green vehicle routing problem. Technical Report LARIS-EA 7315, Laboratoire Angevin de Recherche en Ingénierie des Systèmes, Université d’Angers, 2014.
- I. Muter, J.-F. Cordeau, and G. Laporte. A branch-and-price algorithm for the multidepot vehicle routing problem with interdepot routes. *Transportation Science*, 48(3):425–441, 2014.
- S. Pelletier, O. Jabali, and G. Laporte. Goods distribution with electric vehicles: Review and research perspectives. Technical Report CIRRELT-2014-44, CIRRELT, Montréal, Canada, 2014.
- G. Righini and M. Salani. Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3(3):255–273, 2006.
- M. Schneider, A. Stenger, and D. Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520, 2014a.
- M. Schneider, A. Stenger, and J. Hof. An adaptive VNS algorithm for vehicle routing problems with intermediate stops. *OR Spectrum*, 2014b. doi: 10.1007/s00291-014-0376-5.
- M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 35:254–265, 1987.
- P. Toth and D. Vigo, editors. *Vehicle Routing: Problems, Methods, and Applications*. MOS-SIAM Series on Optimization. SIAM, Philadelphia, PA, 2nd edition, 2014.