Transportation Research Part C 157 (2023) 104374

Contents lists available at ScienceDirect

Transportation Research Part C

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

A route-based algorithm for the electric vehicle routing problem with multiple technologies

Dario Bezzi [a], Alberto Ceselli [b,*], Giovanni Righini [b]

[a] Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione, Università degli Studi di Bergamo, Italy
[b] Dipartimento di Informatica, Università degli Studi di Milano, Italy

ARTICLE INFO

Keywords:
Electric vehicles
Routing
Branch-and-price
Dynamic programming

ABSTRACT

We consider a variant of the electric vehicle routing problem: a fleet of identical vehicles of limited capacity needs to visit a set of customers with given demands. An upper limit is imposed on the duration of the routes. Vehicles have limited autonomy: they may need to stop en-route at recharge stations. Recharges can be partial and multiple recharge technologies are available at stations, providing energy at different costs and different recharge rates.

We present a new a branch-and-price algorithm, that relies on an extended formulation having one variable for each possible depot-to-depot route of each vehicle, implicitly encoding also recharge plans. We design ad-hoc pricing algorithms, which exploit a novel encoding of recharge plans, allowing for efficient bi-directional dynamic programming techniques.

Extensive computational results show our approach to clearly outperform previous ones from the literature, being able to solve instances with up to 30 customers, 5 stations, 7 vehicles and 3 technologies to proven optimality within some minutes on a standard PC.

1. Introduction

For the sake of reducing pollution produced by transportation activities, governments and industries are promoting the replacement of thermal engines with electric ones. In fact, recent evaluations (see for instance Scorrano et al., 2021) have shown that such a replacement would be both environment-friendly and appealing in terms of operating costs, also in the field of freight transportation. However, to grasp the environmental and economic advantages of the introduction of Electric Vehicles (EVs), their use must be suitably optimized. That is why the scientific literature on optimization and management of EVs has rapidly developed in the last years. Comprehensive surveys on EV routing have been published in Pelletier et al. (2016) and, more recently, in Kucukoglu et al. (2021) and Xiao et al. (2021). In particular, the Electric Vehicle Routing Problem (EVRP) consists of optimally routing a fleet of EVs to visit a given set of customers. Unlike traditional vehicles, EVs have limited autonomy, which forces to explicitly plan recharge stops along routes. This makes the combinatorial structure of EVRPs substantially different to their traditional counterparts. Since recharging EVs is much more time-consuming than refueling traditional vehicles, and compelling time constraints are likely to be imposed in freight transportation, realistic models must allow for partial recharges. This introduces a further complexity element, since the amounts of recharged energy are represented by additional continuous variables, originating mixed-integer programming models.

Exact optimization algorithms for the EVRP were developed for instance in Desaulniers et al. (2016) and Duman et al. (2021).

A stream of research focuses on refining the modeling of the non-linear charging process (Montoya et al., 2017; Froger et al., 2019; Lee, 2021). In Lam et al. (2022) the authors consider recharge amounts which are piecewise linear functions of charging time.

* Correspondence to: Alberto Ceselli, Via Celoria 18, 20133 Milano, Italy.
E-mail addresses: dario.bezzi@unibg.it (D. Bezzi), alberto.ceselli@unimi.it (A. Ceselli), giovanni.righini@unimi.it (G. Righini).

https://doi.org/10.1016/j.trc.2023.104374
Received 14 January 2023; Received in revised form 1 September 2023; Accepted 5 October 2023
Available online 20 October 2023

They also take into account the limited availability of chargers at stations, thereby integrating charger scheduling in the routing models. Other researchers focused on finding suitable simplifying assumptions, to limit the combinatorial explosion of the number of solutions. For instance, in Löffler et al. (2020) a single recharge is allowed in each route.

In all this body of literature a single recharge technology is assumed to be available in each station for partial recharges during vehicle routes. However, several recharge technologies have been developed so far (Yilmaz and Krein, 2012), each one with different recharge speed. Multiple recharge technologies are sometimes referred to also as *fast or super fast chargers*. The availability of multiple recharge technologies proves beneficial in applications like (Wang and Szeto, 2021), in which a fleet of EVs is used for rebalancing a commodity in a sharing system (in the specific case, bikes). Being able to charge the EVs faster, even at a higher price, might allow to make the commodity available to users quicker, and ultimately to meet more demand. The converse is also true: unless strict time conditions force fast recharges, savings in terms of energy network costs and battery lifetime can be obtained thanks to slower and cheaper recharges.

The EVRP with multiple recharge technologies has been considered in Felipe et al. (2014), who developed heuristics, and in Ceselli et al. (2021); both papers showed the significant savings achievable owing to the availability of several recharge technologies and partial recharges. In Keskin and Çatay (2018) an Adaptive Large Neighborhood Search (ALNS) heuristic was proposed for the same problem. The authors of Keskin et al. (2019) classified 49 EVRP papers according to fleet composition, objective function terms, presence of multiple recharge technologies and constraints such as capacities and time windows. Similar features have been used in more recent surveys like Erdelić and Carić (2019), Schiffer et al. (2019) and Qin et al. (2021).

In this paper we focus on the exact optimization of the EVRP with partial recharges and multiple technologies. The fleet consists of identical electric vehicles with a given capacity and each customer has an associated demand as in the classical capacitated VRP. The vehicles are equipped with a battery of given capacity and each arc traversal implies a given energy consumption. To avoid running out of energy, vehicles can visit recharge stations in given positions.

Partial recharges are allowed, the cost and time of the each recharge being proportional to the amount of recharged energy. At each station several recharge technologies can be available, with different energy price and recharge speed. A given fixed cost is also charged for every recharge operation, to represent the amortized cost of the battery, whose life time depends on the number of recharge cycles it is subject to. A maximum duration is imposed to all routes. A given service time is spent at customer nodes and when recharge stations are visited. Service times at customers represent pickup/delivery time, while service times at stations represent the duration of connection/disconnection operations that are independent of the amount of energy recharged. The objective is to minimize the overall energy consumption cost.

Considering partial recharges and multiple technologies together is of particular interest not only for its practical applications, but also from a methodological point of view. It is easy to see that each feature independently introduces a further decision level, thus complicating the problem with respect to the traditional VRP. Furthermore, even the combination of them breaks known properties of the model. For instance, when only one recharge technology is available, partial recharges have very limited impact in instances without time windows: as discussed in Desaulniers et al. (2016), at most one recharge for each vehicle is partial in optimal solutions. This is no longer true if multiple technologies are considered; see Ceselli et al. (2021).

Besides Felipe et al. (2014) a few authors proposed heuristic algorithms for variations of the EVRP with partial recharges and multiple technologies: among others, the authors of Sassi et al. (2014) proposed a MILP formulation and a local search heuristic extending later with Large Neighborhood and Tabu Search (Sassi et al., 2015); the authors of Li-ying and Yuan-bin (2015) proposed an adaptive variable neighborhood search.

The only attempt to develop an exact optimization algorithm for the EVRP with multiple recharge technologies is described in Ceselli et al. (2021). The algorithm is a branch-and-cut-and-price, built on a formulation where columns correspond to paths between recharge stations. Such an extended formulation lends itself to a highly parallel implementation of the pricing algorithm. A similar idea was also explored in Bruglieri et al. (2019) to develop a two-stage approach based on a MILP formulation.

The route-based formulation presented here is characterized by a complex mixed-integer pricing sub-problem and a very simple master problem, while the path-based formulation of Ceselli et al. (2021) has a pure integer pricing sub-problem and a much more involved mixed-integer master problem. Both allow for the pricing problem to be solved as a resource constrained shortest path problem by dynamic programming algorithms. The key difference is the following: while in Ceselli et al. (2021) columns correspond to paths, in which all resources are monotonically non-increasing, the route-based algorithm presented here requires to handle recharges along each route; therefore, battery charge is no longer monotone. This changes the structure of the pricing subproblem. Accordingly, the two formulations are equivalent in the discrete domain, but their linear relaxations are not, because the linear relaxation of the path-based formulation allows for convex combinations of paths that are not routes.

Main contributions. In this paper we present a new exact algorithm. It is a branch-and-price algorithm, based on a formulation where columns correspond to complete routes from the depot to the depot (Section 3). With respect to the path-based extended formulation used in Ceselli et al. (2021), the route-based extended formulation has the advantage of providing tighter lower bounds. As a drawback, the structure of pricing problem is much more involved. Therefore, we have designed specific exact pricing algorithms (Section 4). They rely on a novel encoding of partial recharge plans, besides partial routes, which allows us to devise dynamic programming techniques.

In order to assess the effectiveness of these techniques, and that of a full branch-and-price when specific branching rules are included (Section 5), we report on experiments using data-sets from the literature (Section 6). We finally summarize our findings and we outline some conclusions and perspectives (Section 7).

2. Problem definition

The EVRP problem with partial recharges and multiple technologies can be formulated as follows.

Let $G = (\mathcal{N} \cup \mathcal{R}, \mathcal{E})$ be a weighted undirected graph, where $\mathcal{N}$ is a set of n customers and $\mathcal{R}$ is a set of m recharge stations. A distinguished station, $R_0 \in \mathcal{R}$, is located at the depot where all vehicle routes start and end. A set $\mathcal{H} = \{0, \dots, H\}$ includes $H + 1$ recharge technologies. Each technology $h \in \mathcal{H}$ is characterized by a unit recharge time ρ_h and a unit recharge cost γ_h. Without loss of generality, we assume that faster technologies are more expensive, and we number them so that $h' < h'' \Leftrightarrow \rho_{h'} > \rho_{h''} \Leftrightarrow \gamma_{h'} < \gamma_{h''}$ $\forall h', h'' \in \mathcal{H}\backslash\{0\}$. We also assume that a single technology is available at each station; this is again without loss of generality, since such a condition can be enforced by simple station node replication. With an abuse of notation, for each $j \in \mathcal{R}$, we indicate by ρ_j and γ_j the unit recharge time and cost of the (single) technology available at j. Finally, we assume that a dummy technology $h = 0$ is available only at the depot, with $\rho_0 = 0$ and $0 < \gamma_0 \leq \gamma_h \forall h \in \mathcal{H}$. This allows modeling cheap overnight recharge.

A demand q_i and a service time s_i are associated with each customer $i \in \mathcal{N}$; all customers must be visited exactly once: split delivery is not allowed.

The available fleet consists of K identical vehicles with given capacity Q and equipped with batteries of given capacity B.

We make the assumption that each vehicle has initially empty battery. From a practical point of view, all optimal solutions imply that the vehicles return to the depot with no residual energy, thus leaving the battery empty for all subsequent routing plans. In fact, any amount of residual energy at the end of the route could have been avoided, improving the route in both cost and time. Formally, such an assumption is without loss of generality. Any instance with non-zero initial charge E for all vehicles can be reformulated into an equivalent instance with zero initial charge by introducing a dummy depot with no recharge cost and no recharge time, and to connect it only to the true depot by an arc with energy consumption $B - E$ and traveling time 0. Symmetrically, any instance requiring all vehicles to end with non-zero charge R, can be handled by introducing a dummy depot, that is reachable only from the true depot with an arc with energy consumption R and traveling time 0. Given the availability of technology $h = 0$, the best option is often to leave the depot with full charge. However, our models and algorithms do not rely on this assumption. In particular, if a route requires less than B units of energy to be traversed, then its cost is correctly computed according to the actual energy consumption (see next section) and it is represented by an initial recharge of less than B units of energy with technology 0.

Each edge $e \in \mathcal{E}$ requires d_e units of energy and t_e units of time to be traversed. No assumption is required on the triangle inequality.

While every customer must be visited once, stations can be visited at any time by vehicles, even more than once along the same route. During each recharge operation at station $j \in \mathcal{R}$, a vehicle can recharge any amount δ of energy up to its residual battery capacity. The time consumption for a recharge of an amount δ at a station $j \in \mathcal{R}$ is computed as $s_j + \rho_j \delta$, where s_j is the service time at station j; the recharge cost is given by $\gamma_j \delta$. The duration of each route, which is the sum of its traveling and recharge time, is required to be within a given limit T. The objective is to minimize the total recharge cost.

Routes. We define a *route* as an ordered sequence of vertices. The depot R_0 is the initial and final vertex of every route, and it never appears as an intermediate vertex. Customer vertices cannot appear more than once in feasible routes, whereas station vertices can.

For every route r, a corresponding *recharge plan* δ^r must be computed, specifying the total amount of energy δ^r_h recharged with each technology $h \in \mathcal{H}$ along route r, potentially at different recharge stations. For every route r, let us denote by $E_r \subseteq \mathcal{E}$ the multiset of edges traversed along it. Then we can define the following indicators for each route r and recharge plan δ^r:

- total capacity consumption: $\phi_r = \sum_{i \in r \cap \mathcal{N}} q_i$;
- total time consumption: $\tau_r = \sum_{e \in E_r} t_e + \sum_{k \in r} s_k + \sum_{h \in \mathcal{H}} \rho_h \delta^r_h$, where each term t_e occurs as many times as e occurs in E_r;
- total cost $c_r = \sum_{h \in \mathcal{H}} (\gamma_h \delta^r_h)$.

We define a route r to be *feasible* (with respect to a recharge plan δ^r) if and only if these three properties hold:

- Capacity constraint is respected: $\phi_r \leq Q$;
- Time constraint is respected: $\tau_r \leq T$;
- Battery constraint is respected: δ^r keeps the battery charge between 0 and B at any time.

Before stating the mathematical model used for branch-and-price, we outline a useful property of the EVRP.

Proposition 1. *Being the graph G undirected, every feasible route can be traversed in both directions at the same cost using the same recharge plan.*

Proof. The battery charge level along a feasible route can be represented by a piecewise linear function made by alternating ascending and descending segments. All ascending segments correspond to recharge operations, while all descending segments correspond to edge traversals (when energy is consumed). Their slopes correspond to the recharge speed and the consumption rate, respectively. An example is shown in the top part of Fig. 1. Consider now the piecewise linear function that is obtained by a 180 degrees rotation. This corresponds to a reversal of both axes. Therefore, according to the orientation of the time axis, the sequence of recharges and edge traversals is now reversed. The slope of all segments remains the same. Interpreting it as the battery charge profile of a route, it is made by the same ascending and descending segments, representing the same amounts of energy recharged at the same stations with the same recharge speeds and consumed along the same edges with the same consumption rates.

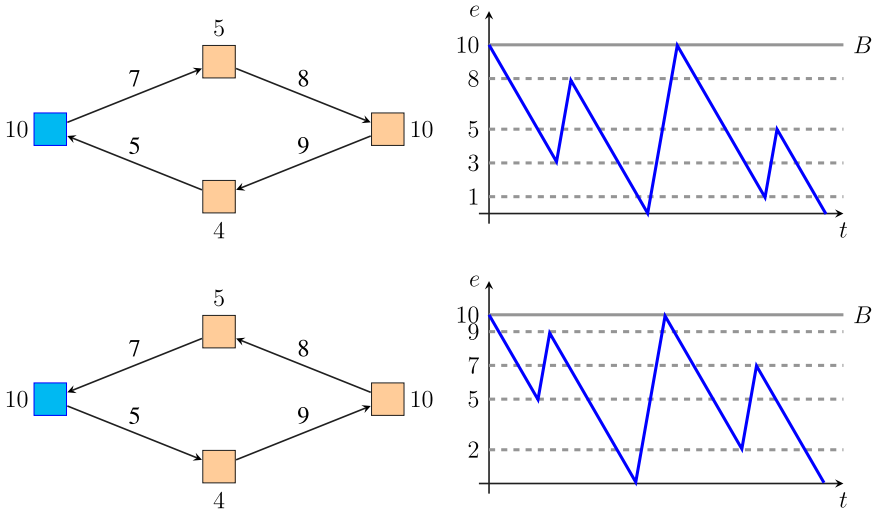

Fig. 1. Top: a feasible route and its battery charge profile. Bottom: the reversed route and the corresponding battery charge profile (traveling time on the horizontal axis, level of charge on the vertical axis). Every feasible route remains feasible when reversed. The battery charge profile of the reversed route is obtained by rotating the charge profile of the original route by 180 degrees. Dark square: depot; light squares: stations. Numerical labels indicate energy consumption along paths and energy recharge at stations.

Therefore, the rotated charge profile corresponds to the battery charge profile along the reversed route, with the same recharge plan (i.e. recharging the same amounts of energy at the same stations). An example is shown in the bottom part of Fig. 1.

In the original feasible route the battery charge level is guaranteed to be between 0 and B at any point and the residual battery capacity is given by the difference between B and the battery charge level. By construction, when the charge profile is rotated, the values 0 and B are swapped, so that the battery charge level along a route is equivalent to the residual battery capacity along the reversed route. Therefore, the battery charge level is guaranteed to be between 0 and B at any point along the reversed route. Hence, traveling along a feasible route in the opposite direction keeping the same recharge plan is feasible. □

3. Mathematical formulation

The branch-and-price algorithm illustrated in the remainder is based on an extended formulation where each column represents a feasible route.

3.1. Master problem

Let us define Ω as the set of all feasible routes. A binary variable x_r is associated with each feasible route $r \in \Omega$ and c_r indicates the route cost. Binary coefficients y_{ir} indicate whether each customer $i \in \mathcal{N}$ is visited or not along each route $r \in \Omega$.

With this notation the following master problem (MP) is obtained:

$$\text{minimize} \sum_{r\in\Omega} c_r x_r \tag{1}$$

$$\text{s.t.} \sum_{r\in\Omega} y_{ir} x_r \geq 1 \qquad \forall i \in \mathcal{N} \tag{2}$$

$$-\sum_{r\in\Omega} x_r \geq -K \tag{3}$$

$$x_r \in \{0,1\} \qquad \forall r \in \Omega \tag{4}$$

The objective function (1) asks for the minimization of the total cost of the selected routes. Covering constraints (2) impose that all customers be visited; constraint (3) allows to select at most K columns in the solution.

We indicate by LMP the linear relaxation of the master problem. We indicate by $\beta \geq 0$ the dual variables vector corresponding to the covering constraints (2) and by $\mu \geq 0$ the dual variable corresponding to constraint (3). With this notation, the expression of the reduced cost of a generic column r in the LMP is:

$$\bar{c}_r = c_r - \sum_{i\in\mathcal{N}} \beta_i y_{ir} + \mu.$$

At each node of a branch-and-bound tree the linear relaxation of the master problem is solved by column generation, where a restricted linear master problem ($RLMP$) with a subset $\Omega' \subset \Omega$ of columns is iteratively solved.

To guarantee its feasibility, the $RLMP$ is initially populated with a dummy column corresponding to a route visiting all customers at a very large cost. For this purpose we use a greedy heuristic algorithm to compute a route visiting all customers with no constraints on duration and capacity. Its cost c^{dummy}, its time consumption τ^{dummy} and its capacity consumption ϕ^{dummy} (set equal to $\sum_{i\in\mathcal{N}} q_i$) are guaranteed to be larger than or equal to the cost, time consumption and capacity consumption of any route in an optimal solution. Therefore $K \cdot c^{dummy}$ is used as the cost of the dummy column, while the input data T and Q are updated to $\min\{T, \tau^{dummy}\}$ and $\min\{Q, \phi^{dummy}\}$, respectively.

3.2. Pricing sub-problem

The pricing sub-problem requires to find a minimum cost closed walk from the depot to the depot, not visiting any customer vertex more than once and not consuming more than a given amount of capacity and time. The vehicle can visit recharge stations to comply with battery constraints. A reward $\beta_i \geq 0$ is obtained from visiting each customer $i \in \mathcal{N}$ and a fixed cost $\mu \geq 0$ is paid at the depot.

This problem is a variation of the Resource Constrained Elementary Shortest Path Problem (which is known to be NP-hard, as stated in Dror 1994), in which the elementary path constraints are imposed only on a subset of vertices, the resources are partly discrete (capacity) and partly continuous (time) and one of the resources (energy) is renewable.

4. Pricing algorithm

We solve the pricing sub-problem with a bi-directional dynamic programming (DP) algorithm, where states correspond to paths with an endpoint at the depot. In DP algorithms developed for similar purposes, encoding paths in DP labels is usually enough to fully define pricing solutions. In our case, instead, it is necessary to encode both the path and the recharge plan along it. Such a need is mainly motivated by the presence of multiple technologies. For instance, the resource extension functions modeling proposed in Desaulniers et al. (2016) does not translate in our setting. Intuitively, when a single technology exists, many recharge plans are equivalent, as recharging in one station or another among the visited ones does not affect the overall time and cost of the route. This is no longer true in the case of multiple technologies.

4.1. Solution encoding

As discussed above, for each path infinite feasible recharge plans may exist. They should be recorded in the piece of information associated with each DP label. Accordingly, we introduce the notion of feasible recharge polyhedron, to encode all feasible recharge plans associated with a partial path.

Definition 1. A *Feasible Recharge Polyhedron* (FRP) $\mathcal{P} = (\underline{\Delta}, \overline{\Delta}, \underline{\delta}, \overline{\delta})$ is defined in as many dimensions as the number of technologies H and it has the following special structure:

- the two scalars $\underline{\Delta}$ and $\overline{\Delta}$ represent the minimum and maximum total amount of recharged energy along the path;
- the two vectors $\underline{\delta} = \{\underline{\delta}_0, \dots \underline{\delta}_H\}$ and $\overline{\delta} = \{\overline{\delta}_0, \dots \overline{\delta}_H\}$ represent the minimum and maximum amounts of energy δ_h recharged for each $h \in \mathcal{H}$.

The *reference point* of $\mathcal{P}$ is the vertex of $\mathcal{P}$ with minimum cost. We indicate the coordinates of the reference point with a vector $\hat{\delta} = \{\hat{\delta}_0, \dots, \hat{\delta}_{\mathcal{H}}\}$ for any given $\mathcal{P}$. The value of the coordinates $\hat{\delta}$ of the reference point are computed as illustrated in Algorithm 1. Fig. 2 shows an example of a FRP for the case $\mathcal{H} = \{1, 2\}$.

In the DP algorithm, the states corresponding to a generic path from the depot to a vertex u are represented by labels $\mathcal{L} = (u, S, \phi, \mathcal{P}, \tau, \bar{c})$, where

- $u \in \mathcal{N} \cup \mathcal{R}$ is the last vertex reached by the path, that starts from the depot;
- $S \subseteq \mathcal{N}$ is the set of customers visited by the path;
- ϕ is the amount of capacity consumed along the path: $\phi = \sum_{i\in S} q_i$;
- $\mathcal{P}$ is the FRP representing the set of feasible recharge plans associated with the path;
- τ is the time needed to reach the *reference point* of $\mathcal{P}$ along the path;
- $\bar{c}$ is the reduced cost corresponding to the *reference point* of $\mathcal{P}$.

Every DP label represents an infinite set of states and each state corresponds to a point of the FRP.

The initial state is $\mathcal{L}_0 = (R_0, \emptyset, 0, \mathcal{P}^0, 0, 0)$, corresponding to a degenerate path including only the depot R_0. The initial polyhedron $\mathcal{P}^0$ is defined as follows:

$$\mathcal{P}^0 = \begin{cases} \underline{\Delta} = 0 \\ \overline{\Delta} = B \\ \underline{\delta_h} = 0 \ \forall h \in \mathcal{H} \\ \overline{\delta_h} = 0 \ \forall h \in \mathcal{H} \backslash \{0\} \\ \overline{\delta_0} = B \end{cases}$$

Therefore, the initial reference point has coordinates $\hat{\delta}_h = 0 \ \forall h \in \mathcal{H}$.

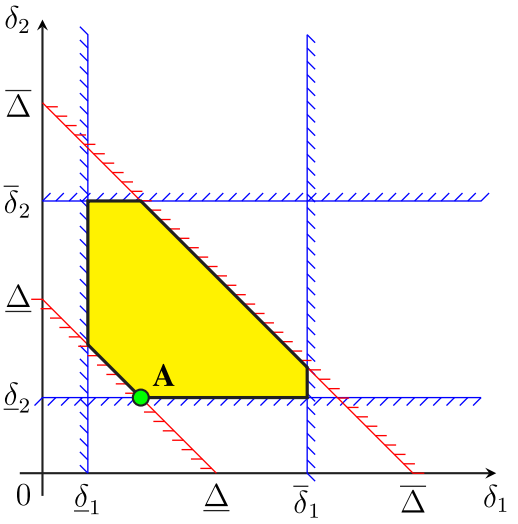

Fig. 2. The reference point of a feasible recharge polyhedron $\mathcal{P}$ is the vertex of $\mathcal{P}$ with minimum cost (point **A**, assuming $\gamma_1 < \gamma_2$).

4.2. Label extension

When a label $\mathcal{L}' = (i, S', \phi', \mathcal{P}', \tau', \overline{c}')$ of vertex i is tentatively extended along an edge $e = [i, j]$ to generate a new label for vertex j, the following conditions are tested:

- Elementarity: $j \notin S'$;
- Capacity: $\phi' + q_i/2 + q_j/2 \leq Q$ (assuming $q_j = 0\ \forall j \in \mathcal{R}$);
- Duration: $\tau' + s_i/2 + t_e + s_j/2 \leq T$;
- Energy: $\overline{\Delta}' - \underline{\Delta}' \geq d_e$.

A valid but stronger version of the last two feasibility conditions is

- Duration: $\tau' + s_i/2 + t_e + s_j/2 + t_{[0,j]} \leq T$,
- Energy: $\overline{\Delta}' - \underline{\Delta}' \geq d_e + m_j$,

where $t_{[0,j]}$ is the traveling time along the edge from j to the depot, while m_j is the amount of energy needed to reach the nearest recharge station from vertex j (it can be pre-computed in linear time for each vertex: $m_j = \min_{k \in \mathcal{R}} d_{[j,k]}$).

The following rule is used to avoid some useless extensions and to save computing time. If $j \in \mathcal{R}$ and no customer vertex has been visited after the last visit to j, then the extension towards j is forbidden. Keeping a boolean flag for each station is enough for this test. In general, a significant imbalance between recharge technologies could make a station-to-station loop necessary to achieve optimality or feasibility (see Appendix B), but this is quite unrealistic and never happens in our instances; we can safely assume every station-to-station loop to be suboptimal.

When the extension of a label $\mathcal{L}' = (i, S', \phi', \mathcal{P}', \tau', \overline{c}')$ to a vertex j along edge $e = [i, j]$ is feasible, a new label $\mathcal{L}'' = (j, S'', \phi'', \mathcal{P}'', \tau'', \overline{c}'')$ is generated according to the following rules.

First, the new FRP $\mathcal{P}''$ and its reference point $\hat{\delta}''$ are computed as indicated in Algorithm 1. Traveling along an edge e of length d_e increases $\underline{\Delta}$ by d_e. The first loop checks for possible updates to the minimum recharge needed in each station h, triggered by the increase to $\underline{\Delta}$, when recharge in all other stations are assumed to be performed at their upper bounds. The second loop is the update to the FRP in each coordinate, implied by the energy consumption along edge e. The final check allows to update to the FRP when a station is visited. Fig. 3 shows a graphical example of the effect of these FRP updates: when the vehicle consumes energy traveling along an edge e, $\underline{\Delta}$ (and possibly $\underline{\delta}_h$ for some h) increases, restricting the FRP (left). When the vehicle visits a station equipped with technology h, then $\overline{\Delta}$ and $\overline{\delta}_h$ increase, enlarging the FRP (right).

Finally, the following extension rules are applied to generate the new label:

- $S'' = \begin{cases} S' \cup \{j\} & \text{if } j \in \mathcal{N} \\ S' & \text{if } j \in \mathcal{R} \end{cases}$,
- $\phi'' = \phi' + q_i/2 + q_j/2$,
- $\tau'' = \tau' + s_i/2 + t_e + s_j/2 + \sum_{h \in \mathcal{H}} \rho_h(\hat{\delta}''_h - \hat{\delta}'_h)$,
- $\overline{c}'' = \overline{c}' - \beta_i/2 - \beta_j/2 + \sum_{h \in \mathcal{H}} \gamma_h(\hat{\delta}''_h - \hat{\delta}'_h)$.

Algorithm 1 Extension from $\mathcal{P}'$ to $\mathcal{P}''$ along edge e towards vertex j

// Bounds update //
$\overline{\Delta}'' \leftarrow \overline{\Delta}'$
$\underline{\Delta}'' \leftarrow \underline{\Delta}' + d_e$
for $h \in \mathcal{H}$ **do**
  $\overline{\delta}''_h \leftarrow \overline{\delta}'_h$
  $\underline{\delta}''_h \leftarrow \max\{\underline{\delta}'_h, \underline{\Delta}'' - \sum_{k \in \mathcal{H}: k \neq h} \overline{\delta}'_k\}$
// Reference point update //
$Z = d_e$
for $h \in \mathcal{H}$ **do**
  $\epsilon \leftarrow \min\{Z, \overline{\delta}''_h - \hat{\delta}'_h\}$
  $\hat{\delta}''_h \leftarrow \hat{\delta}'_h + \epsilon$
  $Z \leftarrow Z - \epsilon$
// Recharge (if j is a station vertex) //
if $j \in \mathcal{R}_h$ **then**
  $\overline{\Delta}'' \leftarrow \underline{\Delta}'' + B$
  $\overline{\delta}''_h \leftarrow \overline{\Delta}'' - \sum_{k \neq h} \underline{\delta}''_k$

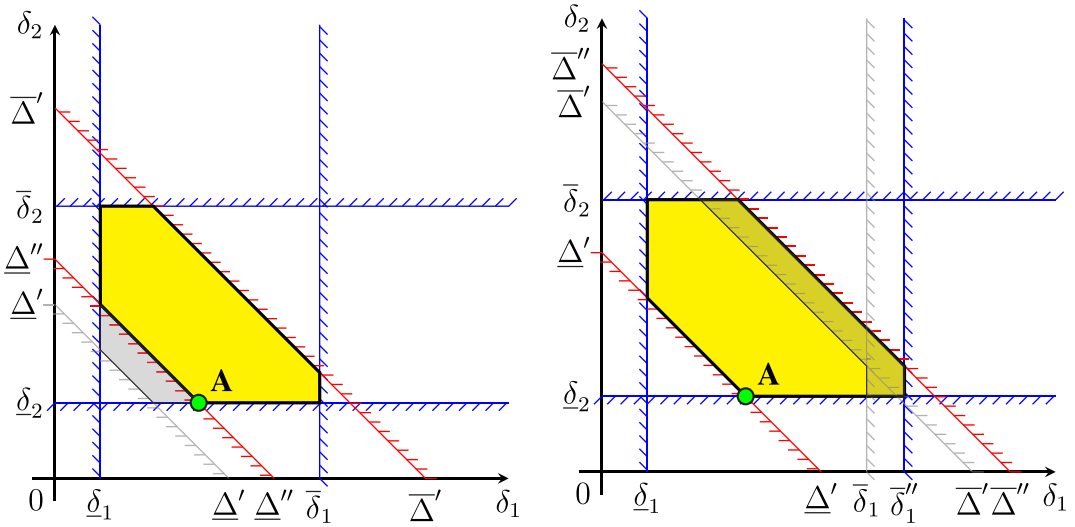

Fig. 3. Effect of the extensions of a label on the FPR. The FRP is restricted when extension traverses an edge (left), it is enlarged when extension results in visiting a station (right).

4.3. Dominance

The exponential number of labels to be generated is a primary source of inefficiency of the DP algorithm. For this reason, a dominance test is performed to fathom labels that cannot lead to an optimal solution, thus limiting the combinatorial explosion in the number of labels.

Efficient dominance rules for the RCESPP have been described in Irnich and Desaulniers (2005). We use the following dominance criteria: a label $\mathcal{L}' = (i, S', \phi', \mathcal{P}', \tau', \overline{c}')$ dominates a label $\mathcal{L}'' = (i, S'', \phi'', \mathcal{P}'', \tau'', \overline{c}'')$ if and only if the following inequalities hold and at least one of them is strict:

$$\begin{cases} S' \subseteq S'' \\ \phi' \leq \phi'' \\ \tau' \leq \tau'' \\ \overline{c}' \leq \overline{c}'' \\ \overline{\Delta}' - \underline{\Delta}' \geq \overline{\Delta}'' - \underline{\Delta}'' \\ \overline{\delta}'_h - \hat{\delta}'_h \geq \overline{\delta}''_h - \hat{\delta}''_h \quad \forall h \in \mathcal{H} \\ \hat{\delta}'_h - \underline{\delta}'_h \geq \hat{\delta}''_h - \underline{\delta}''_h \quad \forall h \in \mathcal{H} \end{cases}$$

Proposition 2. *For any feasible completion of a state of $\mathcal{L}''$, there exists a feasible completion of a state of $\mathcal{L}'$ whose cost is not larger.*

Proof. First we prove Proposition 2 for the reference points, disregarding the energy constraints. Let $\tilde{S}$ be the set of customers visited along a feasible completion of the state of $\mathcal{L}''$ corresponding to its reference point; this implies $\tilde{S} \cap S'' = \emptyset$; hence $S' \subseteq S''$ implies $\tilde{S} \cap S' = \emptyset$. Let $\tilde{\phi}$ be the capacity consumption along a feasible completion of the state of $\mathcal{L}''$ corresponding to its reference point; this implies $\tilde{\phi} + \phi'' \leq Q$; hence $\phi' \leq \phi''$ implies $\tilde{\phi} + \phi' \leq Q$. Let $\tilde{\tau}$ be the time consumption along a feasible completion of the state of $\mathcal{L}''$ corresponding to its reference point; this implies $\tilde{\tau} + \tau'' \leq T$; hence $\tau' \leq \tau''$ implies $\tilde{\tau} + \tau' \leq T$. Let $\tilde{c}$ be the cost of a feasible completion of the state of $\mathcal{L}''$ corresponding to its reference point; this implies that the final cost of the solution is $\tilde{c} + c''$; hence $c' \leq c''$ implies that the final cost of the solution obtained from the reference point of $\mathcal{L}'$ with the same completion is $\tilde{c} + c'$, which is not larger. Concerning the energy constraints, no comparison is needed between the two reference points, since both of them correspond to states with zero residual energy, by construction.

Now, we extend the dominance proof to the feasible recharge polyhedra that represent the energy constraints. The last three conditions are equivalent to $\mathcal{P}' \supseteq \mathcal{P}''$ when the reference points of the two polyhedra are made coincident. In turn, this implies that for each state in $\mathcal{P}''$ there exists a corresponding state in $\mathcal{P}'$ such that their difference in time, energy and cost is the same as for the two reference points: this is because the two corresponding states coincide when the reference points of the two polyhedra are made coincident. In turn, this guarantees that for every feasible solution that can be obtained from any state in $\mathcal{L}''$, another feasible and non-worse solution can be obtained with the same completion from a state in $\mathcal{L}'$. □

With this dominance test we can compare labels instead of states, where each label is a polyhedron containing an infinite number of states. This allows discarding dominated labels from further consideration without losing the optimality guarantee.

Observation 1. *The test $S' \subseteq S''$ takes $O(|\mathcal{N}|)$ time. However, memory word parallelism can be triggered very efficiently, if implemented by bitwise operations.*

By memory word parallelism we mean the simultaneous computation of bit values in a single memory cell performed by bitwise machine instructions. Although its worst case complexity remains linear, the algorithm consistently achieves a substantial speedup.

Labels are kept in buckets, one for each vertex of the graph. This allows to test for dominance only labels associated with the same vertex. The labels are extended following their generation order, as indicated in Algorithm 2.

Algorithm 2 Label extension

```
Create an empty bucket B_n for every node n ∈ N ∪ R
Queue ← {L_0}
B_0 ← {L_0}
i = 0
while i < size of Queue do
    L_i ← i^th label in Queue
    if L_i is not marked closed then
        for All nodes k reachable from L_i do
            Generate label L_k (extend L_i towards k)
            dominated = false
            for All labels ℓ in B_k do
                if ℓ dominates L_k then
                    dominated = true
                    break                                   ▷ switch to the next k
            if dominated = false then
                for All labels ℓ in B_k do
                    if L_k dominates ℓ then
                        Mark ℓ as closed in Queue
                        Remove ℓ from B_k
                Append {L_k} to Queue
                Insert {L_k} into B_k
    i = i + 1
```

Following (Feillet et al., 2004), unreachable customers are included in the subset S for each given path. These are customers that cannot be visited in any feasible extension of the path. We perform two simple checks for marking a customer as unreachable: either its demand does not fit into the remaining vehicle capacity, or the time to reach and service it does not fit into the remaining route time. One further condition is introduced by branching decisions, as explained in Section 5.

4.4. Bi-directional DP

In general, bi-directional DP requires backward extension rules for labels. However, here we consider symmetric graphs and hence the same paths can be traversed in either direction at the same (reduced) cost in the same time. In bi-directional DP, labels are never extended to the depot.

The extension stop criterion must guarantee that every feasible route can be generated by a pair of paths. For this purpose extensions are stopped as soon as half of the available amount of a suitably chosen *critical resource* has been consumed. The selection of the critical resource can heavily affect the overall algorithm performance.

In the EVRP, vehicle capacity and time can be chosen as critical resources, whereas energy cannot because it can be recharged. If capacity is selected as critical, then the stop condition $\phi' + q_i/2 + q_j/2 \le Q/2$ replaces the feasibility test $\phi' + q_i/2 + q_j \le Q$; if time is selected as critical, then the stop condition $\tau' + s_i/2 + t_e + s_j/2 \le T/2$ replaces the feasibility test $\tau' + s_i/2 + t_e + s_j \le T$. For the run-time selection of the critical resource we devised and tested some strategies, described hereafter.

4.4.1. Critical resource selection

Since we have precomputed a dummy route with capacity consumption ϕ^{dummy} and time consumption τ^{dummy}, as described in Section 3.1, the critical resource is guessed from it. Three different heuristic criteria were considered:

- Criterion (1) is simply based on total resource consumptions: time is selected as critical when it is tight along the dummy route (i.e. $\tau^{dummy} \ge T$), capacity otherwise.
- Criterion (2) compares the relative consumption of resources defined as follows:

$$\overline{Q} = Q/\phi^{dummy}$$

$$\overline{T} = T/\tau^{dummy}$$

  Time is selected as critical if $(\phi^{dummy} < Q) \vee (\overline{Q} > \overline{T})$, capacity otherwise.
- Criterion (3) is similar to (2), but the definition of $\overline{T}$ is

$$\overline{T} = KT/\tau^{dummy}.$$

In Section 6.2, some computational results on the effectiveness of these criteria are presented.

4.4.2. Joining paths

A numerical example of the join procedure is illustrated in Appendix A. Given the use of the FRP, it requires specific design choices. When two paths with labels $\mathcal{L}' = (i, S', \phi', \mathcal{P}', \tau', \overline{c}')$ and $\mathcal{L}'' = (j, S'', \phi'', \mathcal{P}'', \tau'', \overline{c}'')$ are joined along the edge $e = [i, j]$, they produce a route whose energy consumption is given by $\underline{\Delta}' + \underline{\Delta}'' + d_e$.

Additionally, we must take into account that one of the two joined paths must be traversed *to* the depot, instead of *from* the depot. The energy consumption along a reversed path is the same (see Proposition 1), but the recharge technology cannot be the same, because the initial (possibly cheap and instantaneous) recharge at the depot is not possible when traveling to the depot.

Therefore, a feasible route can be produced only if at least d_e units of additional energy can be recharged in the resulting route and a complete recharge of B units of energy at the depot can be cancelled. For each partial route the additional energy that can be recharged is given by $\overline{\Delta} - \underline{\Delta}$. Hence a necessary condition to produce a feasible route is $(\overline{\Delta}' - \underline{\Delta}') + (\overline{\Delta}'' - \underline{\Delta}'') \ge B + d_e$.

The energy needed to traverse edge e, implies also an increase in time and in (reduced) cost. These amounts can be quickly lower bounded by $d_e\rho^*$ (time) and $d_e\gamma^*$ (reduced cost), where ρ^* is the unit recharge time of the fastest technology along the route, i.e. $\rho^* = \min_{j \in \mathcal{R}: \hat{\delta}_j < \bar{\delta}_j} \rho_j$ and γ^* is the unit recharge cost of the cheapest technology along the route, i.e. $\gamma^* = \min_{j \in \mathcal{R}: \hat{\delta}_j < \bar{\delta}_j} \gamma_j$.

Therefore some units of energy recharged with technology 0 (available at the depot only) must be replaced using a different technology. In the worst case (routes whose overall consumption is larger than or equal to B) this implies the replacement of B units of energy. We call this amount *join energy*:

$$E^{join} = \max(0, \hat{\delta}'_0 + \hat{\delta}''_0 - B).$$

The additional time and reduced cost due to the join energy can be lower bounded by $E^{join}\rho^*$ (time) and $E^{join}(\gamma^* - \gamma_0)$ (cost), as before.

This lower bounding technique can be used to possibly discard a route immediately. Consider two labels $\mathcal{L}' = (i, S', \phi', \mathcal{P}', \tau', \overline{c}')$ and $\mathcal{L}'' = (j, S'', \phi'', \mathcal{P}'', \tau'', \overline{c}'')$ and the edge $e = [i, j]$. The following (necessary but not sufficient) conditions are tested to check whether the two corresponding paths can be joined to produce a feasible route:

- Elementarity: $S' \cap S'' = \emptyset$;
- Capacity: $\phi' + \phi'' \le Q - (q_i/2 + q_j/2)$;
- Energy: $(\overline{\Delta}' - \underline{\Delta}') + (\overline{\Delta}'' - \underline{\Delta}'') \ge B + d_e$;
- Duration: $\tau' + (E^{join} + d_e)\rho^* + \tau'' + s_i/2 + t_e + s_j/2 \le T$;
- Cost: $\overline{c}' + d_e\gamma^* + E^{join}(\gamma^* - \gamma_0) + \overline{c}'' - \beta_i/2 - \beta_j/2 < 0$.

When the route is not discarded by these tests based on lower bounds on duration and cost, its corresponding duration and cost must be computed exactly. For this purpose, we need to examine the FRP $\mathcal{P}^* = (\underline{\Delta}^*, \overline{\Delta}^*, \underline{\delta}^*, \overline{\delta}^*)$ produced when two paths with FRP $\mathcal{P}' = (\underline{\Delta}', \overline{\Delta}', \underline{\delta}', \overline{\delta}')$ and $\mathcal{P}'' = (\underline{\Delta}'', \overline{\Delta}'', \underline{\delta}'', \overline{\delta}'')$ are joined. It represents the set of all feasible recharge plans of the resulting route. Its computation is illustrated in Algorithm 3. In the first step the energy consumptions along the two paths are summed up. Then, the FRP is updated according to the additional consumption of energy d_e along the edge between the two path endpoints, as in Algorithm 1; the reference point coordinates are updated accordingly. Finally, the join energy is taken into account: the amount

Algorithm 3 Computing the FRP of a route. IN: $\mathcal{P}'$, $\mathcal{P}''$. OUT: $\mathcal{P}^*$.

1: // Sum of the energy consumptions along the paths //
2: $\underline{\Delta}^* = \underline{\Delta}' + \underline{\Delta}''$
3: $\overline{\Delta}^* = \overline{\Delta}' + \overline{\Delta}''$
4: $\overline{\delta}^* = \overline{\delta}' + \overline{\delta}''$
5: $\underline{\delta}^* = \underline{\delta}' + \underline{\delta}''$
6: $\hat{\delta}^* = \hat{\delta}' + \hat{\delta}''$
7: $\tau^* \leftarrow \tau' + s_i/2 + t_e + s_j/2 + \tau''$
8: $\overline{c}^* \leftarrow \overline{c}' + \beta_i/2 + \beta_j/2 + \overline{c}''$
9: // Energy and time consumption along edge e //
10: $\underline{\Delta}^* \leftarrow \underline{\Delta}^* + d_e$
11: **for** $h \in \mathcal{H}$ **do**
12: $\quad \underline{\delta}_h^* \leftarrow \max\{\underline{\delta}_h^*, \underline{\Delta}^* - \sum_{k \in \mathcal{H} \setminus \{h\}} \overline{\delta}_k^*\}$
13: // Reference point update //
14: $Z = d_e$
15: **for** $h \in \mathcal{H}$ **do**
16: $\quad \epsilon \leftarrow \min\{Z, \overline{\delta}_h^* - \hat{\delta}_h^*\}$
17: $\quad \hat{\delta}_h^* \leftarrow \hat{\delta}_h^* + \epsilon$
18: $\quad Z \leftarrow Z - \epsilon$
19: $\quad \tau^* \leftarrow \tau^* + \rho_h \epsilon$
20: $\quad \overline{c}^* \leftarrow \overline{c}^* + \gamma_h \epsilon$
21: // Join energy replacement //
22: $h \leftarrow 0$
23: **while** $(\hat{\delta}_0^* > B) \wedge (\overline{c}^* < 0) \wedge (h \leq H)$ **do**
24: $\quad h \leftarrow h + 1$
25: $\quad \epsilon \leftarrow \min\{\hat{\delta}_0^* - B, \overline{\delta}_h^* - \hat{\delta}_h^*\}$
26: $\quad \hat{\delta}_h^* \leftarrow \hat{\delta}_h^* + \epsilon$
27: $\quad \hat{\delta}_0^* \leftarrow \hat{\delta}_0^* - \epsilon$
28: $\quad \tau^* \leftarrow \tau^* + \rho_h \epsilon$
29: $\quad \overline{c}^* \leftarrow \overline{c}^* + (\gamma_h - \gamma_0)\epsilon$

of energy coming from technology 0, i.e. $\overline{\delta}_0^*$, is decreased to B if needed. For this purpose the technologies are scanned from the cheapest to the costliest and the maximum amount of replacement is done and the duration τ^* and the reduced cost $\overline{c}^*$ of the route are increased accordingly.

If Algorithm 3 exits the loop on line 23 with $\hat{\delta}_0^* \leq B$ and $\overline{c}^* < 0$, then a feasible recharge plan with negative reduced cost may exist; on the contrary, if the algorithm exits the loop with $\overline{c}^* \geq 0$, then no feasible recharge plan has a negative reduced cost.

4.4.3. Repairing infeasible recharge plans

It may happen that the updated reference point of the FRP $\mathcal{P}^*$ is not feasible, because the duration of the route exceeds the prescribed limit T. In this case another point in the FRP must be found, trading time for cost, i.e. replacing slower and cheaper recharges with faster and more expensive ones.

Finding the minimum cost point in the FRP $\mathcal{P}^*$ satisfying the time constraint is a linear programming problem. The candidate optimal points in the FRP are those for which the total amount of recharged energy equals the total consumption along the route. All these points lie on the same facet, say $\mathcal{F}$, of the FRP (for a graphical illustration, see facet $AECG$ in Figure 9 in Appendix A). The projection of facet $\mathcal{F}$ in the cost/time space is still a polyhedron, since cost and time linearly depend on the δ_h variables (see Figure 10 in Appendix A). In particular, the projection of $\mathcal{F}$ is a polyhedron, say C, whose vertices correspond to vertices of $\mathcal{F}$. The point corresponding to an optimal recharge plan must be sought along the piece-wise linear function made of Pareto-optimal points of C, i.e. solutions such that no other point exists in C with smaller values of both time consumption and cost. After the join step, the reference point is the bottom-most point of C, since it corresponds to the minimum cost recharge plan. If the reference point violates the maximum duration constraint but another feasible and optimal recharge plan exists, it is at the intersection between the Pareto frontier along the contour of C and the maximum duration constraint (see, for instance, point X in Figure 10). Finding such a point is the problem of finding a minimum cost point in a polyhedron with an additional linear constraint on time: this is a linear programming problem in two dimensions. Owing to its special structure (only two dimensions) and since C is a projection of $\mathcal{F}$, that is a facet of the FRP, that has a special structure, then the linear programming problem can be efficiently solved to optimality with an ad hoc algorithm, shown in Algorithm 4. The algorithm simply follows the Pareto-optimal frontier from vertex to vertex, until either the time constraint is satisfied or infeasibility is detected.

For this purpose, the algorithm uses two subsets D and I containing the technologies h for which the recharged amounts δ_h^* can be decreased or increased, respectively. The most profitable replacement between two technologies $k^- \in D$ and $k^+ \in I$ is selected

according to the minimum value of the ratio $\frac{\gamma_{k^+}-\gamma_{k^-}}{\rho_{k^-}-\rho_{k^+}}$, i.e. the ratio between the cost increase and the duration decrease. Referring to Figure 10, this means that the point corresponding to the recharge plan δ^* is moved upward and to the left along the most profitable segment on the frontier of C. All vertices of C correspond to vertices of $\mathcal{F}$, where the values of the recharged amounts are equal either to their maximum $\overline{\delta}_h$ or to their minimum $\underline{\delta}_h$, for each recharge technology $h \in \mathcal{H}$. This is why the next vertex of C is identified either by δ_{k^-} reaching its minimum or by δ_{k^+} reaching its maximum. When δ^* is modified, the values of duration τ^* and reduced cost c^* are updated accordingly. We remark that order matters: in the loop on line 15 of Algorithm 3, technologies are examined from the slowest and cheapest one ($h = 1$) to the fastest and most expensive one ($h = H$).

Algorithm 4 Technology replacement to make the route duration feasible

1: $D \leftarrow \{h \in \mathcal{H} : \hat{\delta}_h > \underline{\delta}_h\}$
2: $I \leftarrow \{h \in \mathcal{H} : \hat{\delta}_h < \overline{\delta}_h\}$
3: **for** $h \in \mathcal{H}$ **do**
4: $\quad \delta^*_h \leftarrow \hat{\delta}_h$
5: **while** $(\tau^* > T)$ and $(\overline{c}^* < 0)$ **do**
6: $\quad (k^-, k^+) \leftarrow \arg\min_{k^- \in D, k^+ \in I, k^- \neq k^+} \{\frac{\gamma_{k^+}-\gamma_{k^-}}{\rho_{k^-}-\rho_{k^+}}\}$
7: $\quad \epsilon \leftarrow \min\{\delta^*_{k^-} - \underline{\delta}_{k^-}, \overline{\delta}_{k^+} - \delta^*_{k^+}, \frac{\tau^*-T}{\rho_{k^-}-\rho_{k^+}}\}$
8: $\quad \delta^*_{k^-} \leftarrow \delta^*_{k^-} - \epsilon$
9: $\quad \delta^*_{k^+} \leftarrow \delta^*_{k^+} + \epsilon$
10: $\quad \tau^* \leftarrow \tau^* - (\rho_{k^-} - \rho_{k^+})\epsilon$
11: $\quad \overline{c}^* \leftarrow \overline{c}^* + (\gamma_{k^+} - \gamma_{k^-})\epsilon$
12: $\quad D \leftarrow D \cup \{k^+\}$
13: $\quad$ **if** $\delta^*_{k^-} = \underline{\delta}_{k^-}$ **then**
14: $\quad\quad D \leftarrow D \backslash \{k^-\}$
15: $\quad I \leftarrow I \cup \{k^-\}$
16: $\quad$ **if** $\delta_{k^+} *= \overline{\delta}_{k^+}$ **then**
17: $\quad\quad I \leftarrow I \backslash \{k^+\}$

When the algorithm terminates with $\tau^* \leq T$ and $\overline{c}^* < 0$, a feasible recharge plan with negative reduced cost has been found; on the contrary, when it terminates with $\overline{c}^* \geq 0$, no feasible recharge plan with negative reduced cost exists.

Remark. Although this feasibility repair step is needed to ensure the correctness of the pricing algorithm, in practice it is very unlikely to be necessary. In our computational tests we could never observe any occurrence.

4.5. Implementation and speed-up techniques

Pricing is the most time consuming step of the whole branch-and-price algorithm and in particular the join operation turns out to be the bottleneck. Hence, we devised and tested some rules to avoid as many tentative join operations as possible.

Duplicates avoidance. In general each route can be generated by joining several different pairs of paths. To avoid useless duplicates, we introduced an additional test, that depends on the selected critical resource: the unbalance in the resource consumption between the two paths must be minimum along the resulting route.

For instance, assume capacity has been selected as critical and let ϕ' and ϕ'' be the capacity consumptions in vertices i and j, respectively. Then $\mathcal{L}'$ and $\mathcal{L}''$ are joined through edge $e = [i, j]$ only if

$$|\phi' - \phi''| \leq (q_i + q_j)/2.$$

Alternatively, assume time has been selected as critical and let τ' and τ'' be the time consumptions in vertices i and j, respectively. Then $\mathcal{L}'$ and $\mathcal{L}''$ are joined through edge $e = [i, j]$ only if

$$|\tau' - \tau''| \leq s_i/2 + t_e + \rho^* d_e + s_j/2,$$

where ρ^* is the unit recharge time of the fastest technology along the route.

Label extension in stages. Instead of generating all labels first and then trying to combine them in pairs, it is possible and profitable to have these two phases interleaved. A threshold l_{max} is set for the number of labels to be generated at each stage. When the number of labels reaches l_{max}, the label extension phase freezes and the algorithm starts tentatively joining the labels obtained so far. If no route are found in this way, the value of l_{max} is raised and another stage is executed.

In our implementation we heuristically set $l_{max} = 200k^3$ for the kth stage.

Maximum number of routes. Another threshold s_{max} has been set for the number of routes corresponding to columns of negative reduced cost to be returned by the pricer. When the number of generated feasible routes reaches s_{max}, the pricer stops This is especially effective if the most promising path pairs are examined first, as explained below.

In our implementation we heuristically set $s_{max} = 1000$.

Labels sorting. The labels of each vertex are sorted by non-decreasing reduced cost. The most promising path pairs correspond to nearby vertices and to the first labels of the vertices. Hence, tentative join operations are done according to the following order: first, the edge set is sorted and edges are examined by non-decreasing distance. Then, for each edge $e = [i, j]$ the label lists of i and j are scanned according to their order.

5. Branching

When the optimal solution of the LMP is fractional, two branching policies are used.

Branching on the number of routes. The total number of routes in the solution is upper bounded by m in one branch and lower bounded by $m + 1$ in the other, where m is the integer part of the number of routes occurring in the fractional solution.

This branching rule is applied first.

Branching on customer pairs. Following the commonly used binary branching technique introduced by Ryan and Foster (1981), a pair of customer vertices $u, v \in \mathcal{N}$ is selected such that the sum $\sum_{r\in\Omega: y_{ur}=y_{vr}=1} x_r$ is closest to $1/2$. Then, u and v are forced to be visited in the same route in one branch and they are forced to be visited by different routes in the other.

It is worth noting that such a rule is not in general *robust* with respect to the structure of the pricing problem. That is the pricing problem changes after branching decisions are taken, which then requires to either map branching decisions in weak form as master constraints, or to consider more complex pricing problems.

We decided for the second option, adapting the DP algorithm to get bounds which are as strong as possible. In particular, branching decisions imposing u and v to be visited in different routes are easy to embed in our setting: when u (resp. v) is visited, v (resp. u) is marked as unreachable. No additional change is required. Instead, when u and v are imposed to be visited in the same route, the DP labels need to be enriched by additional resources. In details, we keep in each label a set of *open* nodes which are forced to be visited before completing the route; when u (resp. v) is visited, v (resp. u) is inserted in such a set. Dominance can occur only if the set of open nodes of the dominating label is a subset (possibly equal) of those of the dominated.

Finally, branching decisions are checked during join: those pairs of semi-routes violating them are simply discarded.

The Ryan-Foster rule is applied only to customer vertices: station vertices do not appear in the constraints of the LMP and they do not play any role concerning the integrality of the solution. As described in Ryan and Foster (1981), such a rule is applicable whenever the MP has set partitioning form, and the optimal solution of the LMP is fractional. The branching tree exploration strategy we have used mixes best bound estimates and depth first, according to policies which are embedded in the framework SCIP (Gamrath et al., 2020), that was used in our implementation. In our computational tests, the number of branchings needed to reach optimality never reached 20.

6. Computational results

6.1. Dataset

We did our experiments on three datasets. The full repository is available at Bezzi (2021) in xml format, compliant with the VRP-rep (Mendoza et al., 2014) specifications.

Remark. Input files include also two constants π and v, since energy and time consumption are assumed to be proportional to the distance along each edge through coefficients π and $1/v$, respectively. In other words, being l_e the length of an edge $e \in \mathcal{E}$, in our experiments we have $d_e = \pi l_e$ and $t_e = l_e/v$, although our algorithms do not rely on such a regularity.

Dataset A. This dataset was derived in Schneider et al. (2014) from the Solomon dataset, by relaxing the time windows constraints. It is composed by 36 instances with up to 15 customers and 5 stations with a single technology. Some of these instances are very small and not challenging: we solved them mainly to use them as benchmarks and for the sake of comparison between instances with single technology and multiple technologies.

Instances in dataset A are split in three classes: C (*clustered*), R (*random*) and RC (*random-clustered*), according to the distances between customers. The details of these instances are reported also in the first seven columns of Table 5 in Appendix C.

For some instances in this dataset we modified the number of vehicles with respect to the original value used in Schneider et al. (2014). In one case this was done to make the instance feasible, because the original one was not (Schneider, 2014). In some other cases we decreased the number of vehicles to the minimum value for which the instance was known to be feasible (Felipe et al., 2014).

Dataset B. It is composed by 20 instances with 10 customers, up to 5 vehicles, up to 9 stations and 3 technologies (see also the first four columns of Table 7 in Appendix C). Full details are given in Bezzi (2021).

Dataset C. It is composed by 10 instances directly adapted from the Solomon dataset (clustered instances): all instances have 30 customers, 7 vehicles, 5 stations and 3 technologies (see also the first four columns of Table 8 in Appendix C). Full details are given in Bezzi (2021).

Table 1
Comparison between critical resource selection criteria.

Criterion	Accuracy	Precision Q	Precision T	Recall Q	Recall T
A	0.19	0.10	0.31	0.15	0.22
B	0.69	1.00	0.31	0.64	1.00
C	0.81	1.00	0.56	0.74	1.00

We implemented the branch-and-price algorithm in C++, using the SCIP framework (Gamrath et al., 2020) version 7.0.1 (linking CPLEX 20.1 for the LP subproblems). SCIP was set to use Simplex algorithms for LPs. Preliminary tests with barrier algorithms for reducing potential stability issues did not lead to improvements. Upper bounds for the master problem were generated by general-purpose rounding heuristics embedded in SCIP. The algorithm was executed on a PC equipped with an AMD Ryzen 1950X 16-Core processor and 32 GB of RAM, running Linux Ubuntu 18.

6.2. Selection of the critical resource

The three criteria outlined in Section 4.4.1 were compared. Detailed results for dataset A are given in Table 5 of Appendix C. We do not report results for datasets B and C, because all algorithms always (correctly) predicted time as critical.

Accuracy, Precision and Recall metrics are reported in Table 1. We denote by CT (resp. WT) the case in which T was the correct choice, and the criterion correctly predicted T (resp. predicted T but was instead Q). Symmetrically, we denote by CQ (resp. WQ) the case in which Q was the correct choice, and the criterion correctly predicted Q (resp. predicted Q but was instead T). Accuracy is measured as (CT + CQ) / (CT + CQ + WT + WQ); precision on T is measured as CT / (CT+WT); recall on T is measured as CT / (CT+WQ). Precision and recall on Q are measured in the same way, replacing T with Q. Criterion C proved best in all metrics, therefore we decided to use it in all our tests.

6.3. Results

To better appreciate the pros and cons of the branch-and-price algorithm described so far, we present its results in comparison with those obtained by another branch-and-price algorithm devised for the same problem in Ceselli et al. (2021), where columns do not correspond to routes but to paths between recharge stations.

We remark that the problem addressed in Ceselli et al. (2021) is slightly different from the one considered here, because the number of vehicles to be used was fixed, rather than upper bounded. However, this minor difference does not hamper the conclusions that come from the comparison between the two algorithms.

Table 2 summarizes the results concerning dual bounds at the root node. For each Dataset we report, in turn, the number of instances solved within a timelimit of two hours, the average gap between the best known primal bound (which is often a proven optimal solution) and the dual bound (computed as primal–dual, divided by primal), and the average time to obtain it. The table compares the dual bounds obtained by the algorithm of Ceselli et al. (2021) with and without the use of capacity cuts (first two blocks, marked as 'path'), and those obtained by the route based formulation proposed in this paper (third block, marked as 'route'), as indicated in the leading row. It is interesting to note that in two instances of 15 nodes in dataset A our algorithm was unable to complete the column generation process within a time limit of two hours. We also note that average computing times on Dataset A are higher for the route based formulation, although the bounds are much tighter. The remaining results clearly show that our algorithm is consistently able to produce better bounds in shorter computing time.

In Tables 6, 7 and 8 of Appendix C we report the results obtained by the two algorithms, in their full branch-and-price process, with a timeout of two hours. Similarly, in Tables 9, 10 and 11 of Appendix C we report the primal bound, the dual bound and the gap obtained when solving the problem to proven optimality with two hours timeout. The results of Ceselli et al. (2021) are obtained by imposing an additional cutoff when optimality gap reaches 0.1%.

Average results are summarized in Table 3, in terms of number of solved instances, calls to the pricing algorithm, number of nodes explored in the branch-and-bound tree, computing time (excluding instances leading to timeout), gap after two hours of computation. The algorithm of Ceselli et al. (2021) was unable to find any feasible solution in three instances of dataset B and three instances of dataset C: these were excluded from the average gap computation on the corresponding column.

The results show that optimal solution is usually found by the route-based branch-and-price algorithm rather early, when only few nodes have been analyzed. Indeed, for the smallest instances the optimal solution was found at the root node, without branching at all. This is due to the relative strength of the dual bound (compared with the path-based formulation), that in turn requires few branching steps to achieve integrality. We also report that branching on number of routes has been applied only at the root node, because at non-root nodes the number of routes in the optimal fractional solution of the LMP was always integer.

It is interesting to note that the most time-consuming instances for the route-based formulation are not the largest ones in datasets B and C, but some 15 customers instances in dataset A (namely, A-C208-15, A-R209-15 and A-RC204-15). This is due to the loose bound on time and capacity. Indeed, on two of these three instances (A-R209-15 and A-RC204-15) the algorithm reached the time-out at the root node: they have redundant duration and capacity constraints, so that their optimal solution requires a single vehicle. These are the only instances for which the route-based algorithm does not outperform the path-based one.

Table 2
Column generation dual bounds and cutting strategies: average results.

Instances		Path, without capacity cuts			Path, with capacity cuts			Route		
D.S.	N	n.in.	Gap	Time (s)	n.in.	Gap	Time (s)	n.in.	Gap	Time (s)
A	5	12	17.1%	0.1	12	3.0%	0.1	12	0.0%	0.1
	10	12	13.5%	0.1	12	5.4%	0.2	12	0.1%	6.0
	15	12	11.8%	1.2	12	5.2%	1.9	10	0.6%	73.6
B	10	20	6.8%	141.5	20	6.5%	115.8	20	3.2%	0.1
C	30	10	15.8%	40.6	10	10.0%	41.9	10	1.5%	0.6

Table 3
Branch and price: average results.

Instances		Solved inst.		Pricer calls		B&B nodes		Time (s)		Gap	
D.S.	N	Path	Route	Path	Route	Path	Route	Path	Route	Path	Route
A	5	12	12	938.75	3.83	1836.33	1.00	7.85	0.03	0.02%	0.00%
	10	12	12	13 949.25	17.08	27 512.17	1.17	1015.56	4.53	0.21%	0.00%
	15	8	9	9335.92	66.17	12 134.83	2.25	753.35	62.79	7.95%	0.15%
B	10	12	20	1391.75	19.80	1847.05	3.80	2330.17	0.16	1.71%	0.00%
C	30	6	10	864.40	148.60	918.80	7.60	2448.83	5.90	0.51%	0.00%

For some instances in datasets B and C, such as C-3-N030, it looks inconsistent to have a primal bound (from the route-based formulation) smaller than a lower bound (from the path-based formulation). This is due to the mandatory usage of all vehicles in the path-based formulation: when the number of routes in the optimal solution is smaller than K, forcing the use of an extra vehicle worsens the optimal value.

7. Conclusions

In this paper we have described a branch-and-price algorithm for the EVRP problem with partial recharges and multiple technologies, employing bi-directional labeling algorithms for generating feasible routes.

In order to provide detailed computational insights we have considered data sets from the literature. The advantage of our method with respect to previous attempts emerges clearly: we have been able to solve all instances but two, with a computing effort which is a fraction of that of Ceselli et al. (2021). Still, the size of instances which can be solved to proven optimality (up to 30 customers) is small compared to those solvable for other vehicle routing problems. We report that additional techniques, such as advanced stabilization techniques, and the use of cuts (e.g. subset-row inequalities) might be beneficial, allowing to solve larger instances. Their actual impact, however, would need to be evaluated by proper algorithmic design and experiments.

Interestingly, the three instances left open by our algorithm are not among the largest ones. This confirms that the presence of continuous variables and multiple recharge technologies changes the structure of the problem, making it different in nature (and much more difficult) than the classical VRP, as already reported in the literature. The specific nature of such a difficulty, however, has still to be fully clarified.

Our computational tests gave other interesting insights. For instance, it appears that a path-based approach like that developed in Ceselli et al. (2021) could be promising to solve single-vehicle EVRP variants, i.e. the "electric" equivalent of the TSP, whereas route-based approaches like ours are more powerful when fleets of vehicles need to be optimized.

Many extensions are possible both to make the model more realistic and to develop better algorithms. On the model side, our framework might incorporate constraints that also appear in classical VRP variants (e.g. heterogeneous fleets, pick-up and delivery, loading constraints etc.), although some extensions would not be trivial. For instance, time windows associated with customers, as in Desaulniers et al. (2016), would imply constraints on arrival and departure time that would not correspond to simple upper or lower bounds on recharged amounts. The corresponding facets of the FRP would destroy the easy-to-manage FRP structure we have exploited. It is also worth mentioning some extensions that are specific to the Electric VRP:

- synchronization constraints on simultaneous use of capacitated stations by different vehicles;
- asymmetric graphs and negative cost arcs (corresponding to downhill roads, where energy can be accumulated instead of being consumed);
- variable (e.g. speed-dependent, time-dependent, load-dependent) energy consumption along the edges;
- time-dependent price of energy.

Exploiting parallel computing concepts can speed up pricing (Cabrera et al., 2020). In fact, on the side of possible improvements to the algorithms, we mention the development of a parallel implementation of the join procedure in the pricing algorithm: once the bi-directional extension phase has been performed, each pair of vertices could be examined independently.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors wish to thank two anonymous reviewers and the area editor, whose insightful comments helped to improve the paper. Appendix B originated from a comment of one of the referees, who is also credited for the example reported in Figure 11. The project has been partially funded by Regione Lombardia, grant agreement n. E97F17000000009, Project AD-COM.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.trc.2023.104374.

References

Bezzi, Dario, 2021. Electric Vehicle Routing Problem with Multiple Recharge Technologies (EVRP-MRT) Dataset. Technical Report, UNIMI Dataverse, http://dx.doi.org/10.13130/RD_UNIMI/JEA5XX.

Bruglieri, Maurizio, Mancini, Simona, Pezzella, Ferdinando, Pisacane, Ornella, 2019. A path-based solution approach for the green vehicle routing problem. Comput. Oper. Res. 103, 109–122.

Cabrera, Nicolás, Medaglia, Andrés L., Lozano, Leonardo, Duque, Daniel, 2020. An exact bidirectional pulse algorithm for the constrained shortest path. Networks 76 (2), 128–146.

Ceselli, Alberto, Felipe, Ángel, Ortuño, M. Teresa, Righini, Giovanni, Tirado, Gregorio, 2021. A branch-and-cut-and-price algorithm for the electric vehicle routing problem with multiple technologies. In: Operations Research Forum, Vol. 2. Springer, pp. 1–33.

Desaulniers, Guy, Errico, Fausto, Irnich, Stefan, Schneider, Michael, 2016. Exact algorithms for electric vehicle-routing problems with time windows. Oper. Res. 64 (6), 1388–1405.

Dror, Moshe, 1994. Note on the complexity of the shortest path models for column generation in VRPTW. Oper. Res. 42 (5), 977–978.

Duman, Ece Naz, Taş, Duygu, Çatay, Bülent, 2021. Branch-and-price-and-cut methods for the electric vehicle routing problem with time windows. Int. J. Prod. Res. 1–22.

Erdelić, Tomislav, Carić, Tonči, 2019. A survey on the electric vehicle routing problem: variants and solution approaches. J. Adv. Transp. 2019.

Feillet, Dominique, Dejax, Pierre, Gendreau, Michel, Gueguen, Cyrille, 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks Int. J. 44 (3), 216–229.

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

Froger, Aurélien, Mendoza, Jorge E., Jabali, Ola, Laporte, Gilbert, 2019. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. Comput. Oper. Res. 104, 256–294.

Gamrath, Gerald, Anderson, Daniel, Bestuzheva, Ksenia, Chen, Wei-Kun, Eifler, Leon, Gasse, Maxime, Gemander, Patrick, Gleixner, Ambros, Gottwald, Leona, Halbig, Katrin, Hendel, Gregor, Hojny, Christopher, Koch, Thorsten, Le Bodic, Pierre, Maher, Stephen J., Matter, Frederic, Miltenberger, Matthias, Mühmer, Erik, Müller, Benjamin, Pfetsch, Marc E., Schlösser, Franziska, Serrano, Felipe, Shinano, Yuji, Tawfik, Christine, Vigerske, Stefan, Wegscheider, Fabian, Weninger, Dieter, Witzig, Jakob, 2020. The SCIP Optimization Suite 7.0. Technical Report, Optimization Online.

Irnich, Stefan, Desaulniers, Guy, 2005. Shortest path problems with resource constraints. In: Column Generation. Springer, pp. 33–65.

Keskin, Merve, Çatay, Bülent, 2018. A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. Comput. Oper. Res. 100, 172–188.

Keskin, Merve, Laporte, Gilbert, Çatay, Bülent, 2019. Electric vehicle routing problem with time-dependent waiting times at recharging stations. Comput. Oper. Res. 107, 77–94.

Kucukoglu, Ilker, Dewil, Reginald, Cattrysse, Dirk, 2021. The electric vehicle routing problem and its variations: A literature review. Comput. Ind. Eng. 161, 107650.

Lam, Edward, Desaulniers, Guy, Stuckey, Peter J., 2022. Branch-and-cut-and-price for the electric vehicle routing problem with time windows, piecewise-linear recharging and capacitated recharging stations. Comput. Oper. Res. 145, 105870.

Lee, Chungmok, 2021. An exact algorithm for the electric-vehicle routing problem with nonlinear charging time. J. Oper. Res. Soc. 72 (7), 1461–1485.

Li-ying, Wang, Yuan-bin, Song, 2015. Multiple charging station location-routing problem with time window of electric vehicle.. J. Eng. Sci. Technol. Rev. 8, 190–201.

Löffler, Maximilian, Desaulniers, Guy, Irnich, Stefan, Schneider, Michael, 2020. Routing electric vehicles with a single recharge per route. Networks 76 (2), 187–205.

Mendoza, Jorge, Hoskins, Maxim, Guéret, Christelle, Pillac, Victor, Vigo, Daniele, 2014. VRP-REP: the vehicle routing community repository. In: Third Meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization. VeRoLog. Oslo, Norway.

Montoya, Alejandro, Guéret, Christelle, Mendoza, Jorge E., Villegas, Juan G., 2017. The electric vehicle routing problem with nonlinear charging function. Transp. Res. B 103, 87–110.

Pelletier, Samuel, Jabali, Ola, Laporte, Gilbert, 2016. Goods distribution with electric vehicles: Review and research perspectives. Transp. Sci. 50, 3–22.

Qin, Hu, Su, Xinxin, Ren, Teng, Luo, Zhixing, 2021. A review on the electric vehicle routing problems: Variants and algorithms. Front. Eng. Manag. 8 (3), 370–389.

Ryan, David M., Foster, Brian A., 1981. An integer programming approach to scheduling. Comput. Sched. Public Transp. Urban Passeng. Veh. Crew Sched. 269–280.

Sassi, Ons, Cherif, Wahiba Ramdane, Oulamara, Ammar, 2014. Vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs. Working paper or preprint.

Sassi, Ons, Cherif-Khettaf, Wahiba Ramdane, Oulamara, Ammar, 2015. Iterated tabu search for the mix fleet vehicle routing problem with heterogenous electric vehicles. In: Modelling, Computation and Optimization in Information Systems and Management Sciences: Proceedings of the 3rd International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences-MCO 2015-Part I. Springer, pp. 57–68.

Schiffer, Maximilian, Schneider, Michael, Walther, Grit, Laporte, Gilbert, 2019. Vehicle routing and location routing with intermediate stops: A review. Transp. Sci. 53 (2), 319–343.

Schneider, Michael, 2014. Personal communication.
Schneider, Michael, Stenger, Andreas, Goeke, Dominik, 2014. The electric vehicle-routing problem with time windows and recharging stations. Transp. Sci. 48 (4), 500–520.
Scorrano, Mariangela, Danielis, Romeo, Giansoldati, Marco, 2021. Electric light commercial vehicles for a cleaner urban goods distribution. Are they cost competitive? Res. Transp. Econ. 85, 101022.
Wang, Yue, Szeto, W.Y., 2021. The dynamic bike repositioning problem with battery electric vehicles and multiple charging technologies. Transp. Res. C 131, 103327.
Xiao, Yiyong, Zhang, Yue, Kaku, Ikou, Kang, Rui, Pan, Xing, 2021. Electric vehicle routing problem: A systematic review and a new comprehensive model with nonlinear energy recharging and consumption. Renew. Sustain. Energy Rev. 151, 111567.
Yilmaz, Murat, Krein, Philip T., 2012. Review of battery charger topologies, charging power levels, and infrastructure for plug-in electric and hybrid vehicles. IEEE Trans. Power Electron. 28 (5), 2151–2169.