

Branch-price-cut-and-check for electric vehicle routing and charging scheduling on a capacitated grid with adjustable power

D. Yamín, G. Desaulniers, J. E. Mendoza

G-2026-63

October 2026

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Citation suggérée : D. Yamín, G. Desaulniers, J. E. Mendoza (October 2026). Branch-price-cut-and-check for electric vehicle routing and charging scheduling on a capacitated grid with adjustable power, Rapport technique, Les Cahiers du GERAD G- 2026-63, GERAD, HEC Montréal, Canada.

Suggested citation: D. Yamín, G. Desaulniers, J. E. Mendoza (October 2026). Branch-price-cut-and-check for electric vehicle routing and charging scheduling on a capacitated grid with adjustable power, Technical report, Les Cahiers du GERAD G-2026-63, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (<https://www.gerad.ca/fr/papers/G-2026-63>) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Before citing this technical report, please visit our website (<https://www.gerad.ca/en/papers/G-2026-63>) to update your reference data, if it has been published in a scientific journal.

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.

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.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2026
– Bibliothèque et Archives Canada, 2026

Legal deposit – Bibliothèque et Archives nationales du Québec, 2026
– Library and Archives Canada, 2026

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Télec. : 514 340-5665
info@gerad.ca
www.gerad.ca

Branch-price-cut-and-check for electric vehicle routing and charging scheduling on a capacitated grid with adjustable power

Daniel Yamín ^a

Guy Desaulniers ^b

Jorge E. Mendoza ^c

^a *Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, USA*

^b *Département de mathématiques et de génie industriel, Polytechnique Montréal & GERAD, Montréal (Qc), Canada*

^c *Department of Operations and Logistics Management, HEC Montréal & CIRRELT, Montréal (Qc), Canada*

guy.desaulniers@polymtl.ca

October 2026
Les Cahiers du GERAD
G–2026–63

Copyright © 2026 Yamín, Desaulniers, Mendoza

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract : Electricity grid capacity is rapidly becoming a major bottleneck in the large-scale integration of electric vehicles (EVs). With this motivation, we address a new variant of the EV routing and charging scheduling problem that incorporates a capacitated power grid and adjustable charging power. The problem consists in determining a set of minimum-cost EV routes during a given day and the preceding overnight on-site charging schedule. Routes must satisfy load, time window, and battery capacity constraints, while charging schedules must respect charger availability, grid capacity, and non-preemptive charging constraints. Piecewise-linear recharging functions and alternative travel paths between locations (e.g., minimizing travel cost or energy consumption) are also considered. We develop an exact branch-price-cut-and-check algorithm in which the master problem enforces customer covering, charger availability, and grid capacity, and a backward labeling algorithm generates columns integrating a route and an adjustable-power charging schedule. A dedicated checking problem verifies whether a set of routes admits a feasible charging schedule, enabling early fathoming of branch-and-bound nodes or the generation of feasibility cuts. Computational experiments show that our method solves instances with up to 50 customers and three charging power levels to optimality. Adjustable charging power can significantly reduce routing costs and allow the construction of feasible charging schedules in instances where none exist under fixed power, at the expense of increased computational effort. Our experiments also demonstrate the computational benefits of the checking procedure and the integrated route-and-schedule column representation.

Keywords: Electric vehicle routing; charging scheduling; capacitated grid; adjustable power; branch-price-cut-and-check; no-good cuts

1 Introduction

Electric vehicles (EVs) play a central role in the transition from fossil fuels to renewable energy sources in urban logistics and public transportation systems. Compared with internal combustion engine vehicles, EVs produce lower noise levels and zero local greenhouse gas emissions, making them particularly well suited to dense urban areas subject to stringent environmental regulations. Declining battery costs, volatile oil prices, and government incentive schemes have further enhanced the business case for EVs and accelerated their adoption. At the same time, this rapid and large-scale adoption has introduced a variety of challenges, ranging from infrastructure limitations to operational and economic considerations.

Among these challenges, the increase in electricity demand on power distribution networks has emerged as a critical issue, placing significant strain on energy systems (Hartvigsson et al. 2022). To ensure power system reliability and operational safety, energy providers implement time-of-use (TOU) electricity tariffs, enforce hard limits on the power that can be drawn from the grid, and impose facility-related demand charges (FRD) based on peak grid power consumption. These measures impose operational constraints on EV operators in urban logistics and public transportation systems, affecting their charging operations (Pelletier et al. 2018).

To cope with increasingly restrictive grid capacity constraints and rising electricity costs during peak hours, EV operators have invested in upgrading their grid connection capacity or in on-site electricity generation from renewable sources such as wind or solar power (Löbel et al. 2024). However, these solutions typically require substantial capital investment and, in some cases, facility relocation. As a more cost-effective and readily implementable alternative, EV operators are adopting active charging management systems, which enable dynamic adjustment of charger power to comply with grid constraints and facilitate *peak shaving*, i.e., a more even redistribution of power demand over time (He et al. 2020).

Active charging management systems fundamentally transform EV charging operations by introducing explicit charging power decisions. In this setting, vehicle routes impose strict time and energy requirements, while depot charging is constrained by charger availability and an overall grid capacity limit, making routing and charging decisions highly interdependent. Specifically, operators must trade off the number of vehicles charged simultaneously against the energy delivered to each vehicle, directly affecting route feasibility and operating costs. Charging many vehicles at low power distributes energy across a larger share of the fleet, whereas charging fewer vehicles at higher power concentrates energy to enable longer routes or earlier departures. Adjustable charging power thus enables richer interactions between routing and charging scheduling decisions.

Despite the intimate relationship between routing and charging scheduling, their interaction remains largely unexplored. The literature has primarily studied electric vehicle charging scheduling problems (E-VCSPs) and electric vehicle routing problems (E-VRPs) in isolation. E-VCSPs focus on constructing feasible EV duties (i.e., sequences of trips and recharging operations) covering a predefined set of timetabled trips, typically under depot-level charger availability and power grid constraints. However, these models treat routing decisions exogenously. In contrast, E-VRPs determine minimum-cost EV routes to serve a set of customers while accounting for battery limitations and recharging requirements. Recharging typically occurs *en route* at public charging stations, which means there is no charging scheduling. However, many EV operators deliberately avoid off-site charging due to uncertainty in charger availability, driver idle time, and higher daytime energy tariffs (Kullman et al. 2021). Instead, they rely on overnight depot charging before vehicle departure. While this reduces exposure to the highest TOU price periods, grid capacity limits remain critical operational constraints that must be explicitly managed.

Hoping to bridge the gap between E-VRPs and E-VCSPs, Yamín et al. (2025) introduced the multigraph-based Electric Vehicle Routing and Overnight Charging Scheduling Problem, denoted as mE-VRSPTW. The problem consists in determining a set of minimum-cost routes along with a pre-

ceding overnight charging schedule that ensures route feasibility. It incorporates E-VRP characteristics, such as battery capacity constraints, a piecewise-linear recharging function, and non-preemptive recharging. Recharging takes place at the depot with a limited number of chargers. To capture trade-offs between travel time and energy consumption in road networks, the problem also considers alternative travel paths (i.e., arcs) between locations. More specifically, in their experiments, they considered the two arcs minimizing the travel time and the energy consumption. To solve the mE-VRSPWTW, Yamín et al. (2025) developed an exact branch-price-and-cut algorithm where the variables are associated with a combination of a route and a charging schedule. Despite narrowing the gap between E-VRPs and E-VCSPs, the mE-VRSPWTW assumes that chargers always operate at full capacity (i.e., at a fixed charging power) and neglects grid capacity constraints, limiting its applicability in settings with active charging management, a rapidly expanding practice.

In this paper, we extend the mE-VRSPWTW by incorporating grid capacity constraints and adjustable charging power. In contrast to the original formulation, the resulting problem, denoted mE-VRSPWTW-AP, (i) allows the charging power delivered to an EV to vary within a discrete set of levels, (ii) imposes a capacity constraint on the total power drawn from the electricity grid at any time, and (iii) restricts the number of power adjustments during a single recharging operation. The latter aims to help preserve the health of the batteries (Pelletier et al. 2018).

Inspired by the algorithm of Yamín et al. (2025), we develop an exact branch-price-cut-and-check (BPC-C) algorithm in which the master problem enforces customer covering, charger availability, and grid capacity constraints, while a backward labeling algorithm generates columns encoding the combination of a route (i.e., a sequence of customers) and an adjustable-power charging schedule, which we refer to as the integrated route-and-schedule column representation. The labeling algorithm accurately models multiple piecewise-linear recharging functions (one for each charging power level) and leverages problem-specific dominance rules. Furthermore, we devise a dedicated checking problem that determines whether a set of routes admits a feasible charging schedule, enabling early fathoming of branch-and-bound nodes or the generation of feasibility cuts.

We report computational experiments on benchmark instances adapted from the literature with up to 50 customers and three charging power levels. Our algorithm solves all instances to optimality, requiring about 8 minutes for the 25-customer instances and 25 minutes for the 50-customer instances, on average. Our results show that, compared to a fixed-power setting, adjustable charging power can reduce the total cost by up to 11% and allow the construction of feasible charging schedules in instances where none exist under fixed power. However, in the adjustable charging power setting, the computational time increases by approximately a factor of eight, reflecting the added complexity. Moreover, the results indicate that generating columns that embed both routing and scheduling information yields substantially tighter linear programming bounds (up to 18% stronger) than a decomposition that considers only the routing component and adds no-good cuts whenever the resulting routing solution does not admit a feasible charging schedule. In this latter approach, weak bounds can render the method computationally intractable.

The rest of this paper is organized as follows. Section 2 reviews the literature most closely related to our work. Section 3 provides a detailed definition of the mE-VRSPWTW-AP. Section 4 describes the proposed BPC-C algorithm. Section 5 presents the results of our computational experiments. Finally, we conclude this paper and suggest future research avenues in Section 6.

2 Related literature

This work lies at the intersection of electric vehicle routing and on-site charging scheduling. In this section, we provide a concise review of the literature that is directly related to the studied problem. For more comprehensive reviews of E-VRPs, we refer to Yamín et al. (2025); Wang et al. (2026); for E-VCSPs, we refer to Perumal et al. (2022); Dolgui et al. (2025).

2.1 Electric vehicle routing problems

The battery charging process is non-linear but can be well approximated by a piecewise-linear function as shown by Montoya et al. (2016). Employing this approximation, they propose a mixed-integer linear program (MILP) for the E-VRP that captures non-linear charging, multiple charging technologies, multiple recharging stops, and partial recharges. Their model relies on a replication of the charging station vertices and imposes at most one visit per replica. While this facilitates tracking travel time, distance, and state of charge, it may lead to intractable models due to symmetry. Their computational results highlight the importance of modeling charging non-linearity, as neglecting it can yield infeasible or excessively costly solutions.

Froger et al. (2022) tackle an E-VRP that incorporates a piecewise-linear charging function, heterogeneous charging technologies, multiple recharging visits, partial recharges, and a limited number of chargers. They propose a path-based MILP formulation in which charging station capacity is enforced through sequencing variables, and solve the resulting model using an off-the-shelf MILP solver. Their computational study demonstrated that failing to explicitly model charging station capacity can yield infeasible routing solutions in practice.

Extending the work of Froger et al. (2022), Lam et al. (2022) introduce customer time windows and develop a specialized exact method. They propose a BPC-C algorithm that decomposes routing and charging decisions: column generation is used to construct routes, while constraint programming is used to check whether a corresponding feasible charging schedule exists. When no such schedule is found, a logic-based Benders cut is added to exclude the routing solution from the search space. To strengthen these cuts, the authors devise a conflict-analysis mechanism.

Lera-Romero et al. (2024) study an E-VRP with time-dependent travel times and speed-dependent energy consumption. In this setting, routes must trade off traversing an arc during less congested periods—which reduces travel time and may improve time-window feasibility—against potentially higher energy consumption, which negatively affects driving range. To solve the problem, they propose a branch-price-and-cut algorithm with tailored preprocessing rules, a dominance-based labeling algorithm, and specialized branching constraints. Their computational experiments revealed that more than 60% of the solutions obtained from a time-independent model become infeasible when evaluated in a time-dependent context.

Recently, Wang et al. (2025) extended the E-VRP to the heterogeneous fleet case with time-dependent waiting times at charging stations. In this context, each vehicle type is associated with a unique nonlinear charging function at each type of charging station. They first derive a path-flow formulation that can be solved exactly using a generic MILP solver for small-sized instances. To tackle practical cases, they propose an algorithmic framework in which an iterated local search generates a pool of high-quality routes, and a set-partitioning model selects a subset of these routes to construct a feasible solution. Subsequently, Wang et al. (2026) study a different variant that incorporates charging-station-specific electricity prices, charging technologies, and waiting times. As in their earlier work, they develop an exact MILP approach relying on a path-flow formulation for small-sized instances and a metaheuristic framework for larger instances.

2.2 Electric vehicle charging scheduling problems

The studies discussed so far primarily address problems in which vehicles recharge off-site, i.e., at public charging stations outside the depot. In such models, the recharging decisions concern when, where, and how much to recharge, possibly given exogenous information about charging station congestion. Nevertheless, as previously mentioned, many EV operators now prefer on-site charging to mitigate uncertainty in charger availability and reduce the risk of driver idle time associated with off-site charging. Consequently, more attention has been devoted in the literature to the depot-based charging scheduling problems.

Pelletier et al. (2018) formulate one of the first E-VCSPs that integrates a piecewise-linear recharging function, TOU electricity tariffs, battery degradation costs, multiple charging technologies, grid capacity constraints, and FRD charges. They propose a MILP formulation and demonstrate how the model can be used to evaluate alternative charging plans that balance trade-offs among these components (e.g., TOU tariffs vs. battery degradation).

Abdelwahed et al. (2020) study an E-VCSP in the context of electric bus (EB) networks. Given fixed charger locations and a predetermined timetable, the problem consists in determining when and where each bus should charge in order to minimize TOU tariffs. They assume a linear recharging function and propose two MILP formulations based on different time-discretization schemes: a minute-based discrete-time optimization (DTO) model and a more compact discrete-event optimization (DEO) model that restricts charging decisions to bus arrival and departure events. Experiments show that much better computation times can be achieved when relying on the DEO formulation compared to the DTO model, while both approaches yield similar solutions.

He et al. (2020) address the charging scheduling problem for a fast-charging battery EB system with on-route chargers. They propose a mathematical program that determines both when a bus should charge and the charging power level in order to minimize total electricity costs, including TOU tariffs and FRD charges, while respecting battery charge limits and fixed timetables. The charging power is modeled as a nonnegative decision variable bounded by the charger's maximum capacity. Because the original formulation results in a nonlinear and nonconvex optimization problem, the authors reformulate it as a tractable linear program using discretization and linearization techniques. Computational experiments on a real-world bus network show that coordinated charging and power control can substantially reduce peak power demand and total charging costs.

Wu et al. (2022) tackle a multi-depot E-VCSP in which vehicle-to-trip assignments and charging plans must be determined. The objective is to first minimize operational costs—including transportation costs, waiting time costs, and TOU tariffs—and second to minimize FRD charges. Station-specific peak-load restrictions are imposed, and linear recharging functions are employed. Using a time-expanded network, they formulate the problem as a biobjective MILP, which is subsequently reformulated as a set-partitioning problem over trip chains (i.e., sequences that start and end at a depot and may include service trips and charging stops). Based on this representation, they develop a tailored branch-and-price algorithm. Their results show that ignoring grid constraints can lead to infeasible or overly optimistic schedules.

de Vos et al. (2024) study an E-VCSP explicitly accounting for limited charging station capacity and partial charging. They consider a multi-depot setting with heterogeneous vehicle types and discretize battery state-of-charge levels to construct a connection-based network that tracks both trip compatibility and charging decisions. The problem is formulated as a path-based set covering model with additional constraints to enforce charging station capacities over discretized time blocks. The linear relaxation is solved via column generation, and integer solutions are obtained using a price-and-branch heuristic or a diving heuristic. To address the loss of accuracy induced by discretization, the authors also develop a dual network that provides valid lower bounds. Computational experiments on real-world instances show that the diving heuristic produces high-quality solutions within reasonable computational times.

In contrast to E-VRPs, the routing component in E-VCSPs is typically fixed, simplified, or entirely absent. As a result, MILP formulations solved with commercial solvers typically scale well in practice (Perumal et al. 2022). Furthermore, although several studies incorporate grid capacity constraints and facility-related demand charges, only a limited number explicitly model adjustable charging power. Existing approaches typically rely on simplifying assumptions, such as linear recharging functions or preemptive charging, that can lead to suboptimal or even infeasible schedules in practice.

2.3 Positioning of our work

The mE-VRSPTW introduced by Yamín et al. (2025) bridges the gap between E-VRPs and E-VCSPs by considering a joint EV routing and charging scheduling problem. Relative to E-VCSPs, the model explicitly incorporates routing decisions. Relative to E-VRPs, it assumes on-site charging rather than off-site charging, and employs a multigraph representation that captures trade-offs between energy consumption and travel distance or travel time on road networks.

The mE-VRSPTW-AP further narrows the gap between E-VRPs and E-VCSPs by simultaneously considering routing decisions and on-site charging scheduling with grid capacity constraints and adjustable power. From this perspective, our work is the first to incorporate both adjustable charging power and explicit grid capacity constraints within an E-VRP framework. This shift transforms the charging-scheduling component from a single-scalar feasibility check, driven by charger availability, into a joint decision over power levels, periods, and adjustments that is co-determined with routing through grid capacity and the piecewise-linear recharging functions. This extension aligns with the recent trend of enriching E-VRPs with features that were initially studied in E-VCSPs (e.g., TOU tariffs). From an E-VCSP perspective, existing models that account for adjustable charging power typically neglect routing decisions and rely on formulations that fail to capture important operational constraints (e.g., non-preemptive recharging or piecewise-linear charging functions). These limitations are unsurprising given the relatively early stage of this research stream. To address this gap, we devise an exact solution algorithm capable of solving instances of practical size while incorporating state-of-the-art E-VRP features, such as piecewise-linear recharging functions, together with adjustable charging power and grid capacity constraints.

3 Problem statement

The mE-VRSPTW-AP is defined on a directed multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} := \{0\} \cup \mathcal{C}$, vertex 0 represents the depot, and $\mathcal{C} := \{1, \dots, C\}$ is the customer set. The arc set $\mathcal{A}$ is a multiset, with $\mathcal{A}_{(i,j)} := \{(i,j)^k \mid k = 1, \dots, K_{(i,j)}\}$ representing $K_{(i,j)}$ parallel travel options from $i \in \mathcal{V}$ to $j \in \mathcal{V}$.

We let $\mathbb{N} := \{0, 1, 2, \dots\}$. The depot houses a homogeneous fleet of EVs that is large enough to service all customers. Each vehicle has a load capacity $Q \in \mathbb{N}$ and a battery that stores an amount of energy between 0 and $E \in \mathbb{N}$. Vehicles can leave the depot no earlier than $\underline{T} \in \mathbb{N}$ and must return by $\overline{T} \in \mathbb{N}$.

With each vertex $i \in \mathcal{V}$, we associate a load $q_i \in \{0, 1, \dots, Q\}$ to be picked up and a service time window $[t_i, \bar{t}_i]$, where $t_i, \bar{t}_i \in [\underline{T}, \overline{T}]$ and $t_i \leq \bar{t}_i$. Vehicles may arrive before t_i and wait to start service. We also set $q_0 = 0$ and $[t_0, \bar{t}_0] = [\underline{T}, \overline{T}]$.

With each arc $(i,j)^k \in \mathcal{A}$, we associate a distance (or travel cost) $c_{(i,j)^k} \geq 0$, a travel time $t_{(i,j)^k} \geq 0$ (including any service time at i), and an energy consumption $e_{(i,j)^k} \geq 0$. For each pair $(i,j) \in \mathcal{V}^2$, we denote by $\underline{c}_{(i,j)}$, $\underline{t}_{(i,j)}$, and $\underline{e}_{(i,j)}$ the minimum distance, travel time, and energy consumption when moving from i to j . We assume that the triangle inequality holds for these minimum values.

At the depot, B homogeneous chargers are available, allowing at most B vehicles to charge simultaneously. Each charger can deliver power from the discrete set $\mathcal{W} = \{w_1, \dots, w_M\}$ with $0 < w_1 < \dots < w_M$, where w_1 and w_M denote the minimum and maximum power of a charger.

We discretize charging decisions over the set of periods $\mathcal{T} := \{1, \dots, \tilde{T}\}$, where each period $t \in \mathcal{T}$ corresponds to the semi-open interval $[(t-1)\tau, t \cdot \tau)$, with $\tau \in \mathbb{N}$ denoting the discretization unit. As illustrated in Figure 1, the charging scheduling problem involves time-discretized decisions made over the periods in $\mathcal{T}$, whereas the routing problem is defined over a continuous-time horizon $[\underline{T}, \overline{T}]$.

Depending on the application, the last charging period $\tilde{T}$ can be defined loosely so that charging is possible after the routing time horizon starts (i.e., $0 < \underline{T} < \tau \tilde{T}$), enabling vehicles to delay their

departure from the depot for additional charging. In other cases, $\tilde{T}$ may instead be set before the routing time horizon begins, for example, due to workforce shift constraints or TOU electricity tariffs that restrict charging to a specific time window.

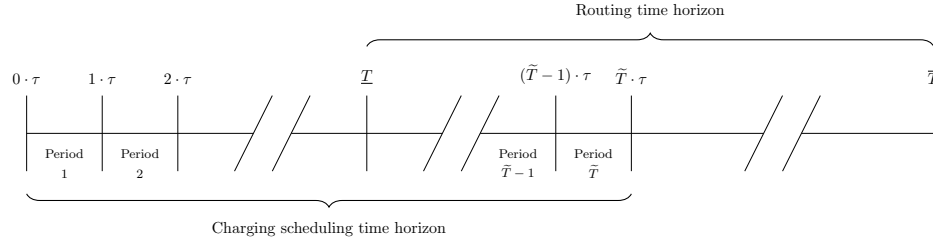


Figure 1: Routing and charging scheduling time horizons. Adapted from Yamín et al. (2025).

At the start of the planning horizon, each vehicle has zero energy, corresponding to a minimal state of charge. Battery recharging follows a concave piecewise-linear function f_w that depends on the power $w \in \mathcal{W}$ delivered by the charger (see Figure 2). All such functions share the same P pieces, indexed by the ordered set $\mathcal{P} := \{1, \dots, P\}$. Piece $p \in \mathcal{P}$ spans the energy interval $[\phi_{p-1}, \phi_p]$, with $0 = \phi_0 < \phi_1 < \dots < \phi_P = E$, and is associated with an efficiency factor $\theta_p \in (0, 1]$, where $1 = \theta_1 > \theta_2 > \dots > \theta_P > 0$. The functions differ only in their slopes, i.e., the energy charged per time unit, which equals $w \cdot \theta_p$ for each $w \in \mathcal{W}$ and $p \in \mathcal{P}$.

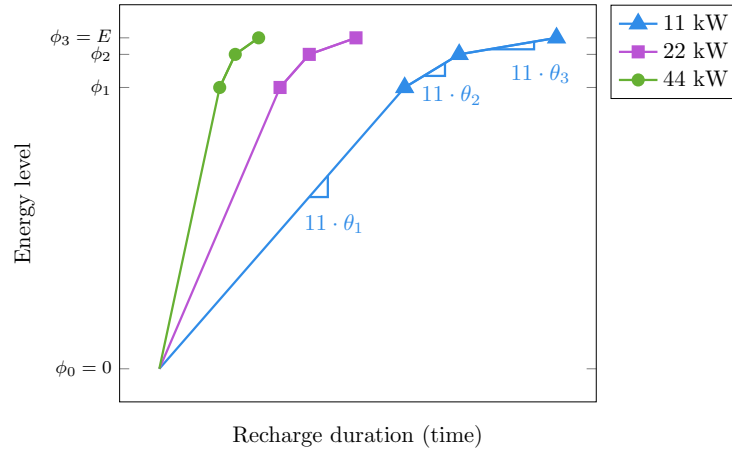


Figure 2: Recharging functions for charging power levels of 11, 22, and 44 kW.

Given a charging power $w \in \mathcal{W}$ and an energy level $e \in [0, E]$, the inverse recharging function $f_w^{-1} : [0, E] \rightarrow \mathbb{R}_+$ computes the time required to reach energy level e from level 0 when the charger delivers power w . It is defined as follows:

$$f_w^{-1}(e) = g(w, e) := \sum_{p=1}^{p'} \frac{1}{w \cdot \theta_p} \left(\min\{e, \phi_p\} - \phi_{p-1} \right),$$

where $p' \in \mathcal{P}$ satisfies $\phi_{p'-1} \leq e \leq \phi_{p'}$.

As for charging scheduling, each EV may charge its battery at most once and at a single charger. During this process, the delivered power may be adjusted at most N times, as frequent changes in charging power may deteriorate battery health. Finally, the total power drawn from the grid is limited by its capacity $W \in \mathbb{N}$ at all times.

In the mE-VRSPTW-AP, the EVs must be routed and their preceding overnight charging must be scheduled such that: (1) each customer is serviced within its time window by a single EV, (2) each

route starts and ends at the depot, (3) vehicle capacity and battery capacity are not exceeded along every route, (4) the charging capacity at the depot is always respected, (5) an EV can recharge at a single charger and for a number of consecutive periods (preemption is prohibited), (6) the power during a charging operation is adjusted at most N times, and (7) the total power drawn from the electricity grid at any time during the charging horizon is less than or equal to W . The objective is to minimize the total distance traveled by the EVs. Conditions (1)–(5) correspond to the original mE-VRSPW, while conditions (6)–(7) capture the adjustable charging power extension.

Illustrative example. In Figures 3 and 4, we present the routing and charging scheduling components of a feasible solution to a toy mE-VRSPW-AP instance with $C = 4$ customers, two EVs, a load capacity $Q = 10$, a battery capacity $E = 20$, and a demand $q_i = 5$ for each customer i . Figure 3 specifies the time windows at each vertex, together with the distance, travel time, and energy consumption of the used arcs, as well as a subset of unused arcs. The depot has $B = 2$ chargers, with possible charging power levels $\mathcal{W} = \{1, 1.5, 2\}$. The discretization unit is $\tau = 1$, and recharging follows the piecewise-linear functions shown in Figure 4(b), with efficiency factors $\theta_1 = 1$, $\theta_2 = 0.6$, and $\theta_3 = 0.15$. During a recharge, the charging power may be adjusted once (i.e., $N = 1$), and the maximum grid capacity is $W = 2.5$.

Figure 3 displays the routes in this solution: customers 1 and 2 are visited by EV 1 (blue arcs), whereas customers 3 and 4 by EV 2 (green arcs). By departing from the depot at times 15 and 17, respectively, both EVs service their customers within the prescribed time windows. They both pickup a total load of 10, respecting vehicle capacity. Figure 4 shows a feasible charging schedule. EV 1 charges for 8 periods at power $w = 2$ and for an additional 3 periods at power $w = 1.5$, yielding over 19 units of energy. EV 2 charges for 3 periods at power $w = 1$ and for 4 additional periods at power $w = 2$, resulting in exactly 11 units of energy. Consequently, the batteries of both EVs are sufficiently charged for completing their routes. Moreover, the charging power is adjusted once for each EV, and the grid capacity constraint is never violated, being tight during periods 9 to 11.

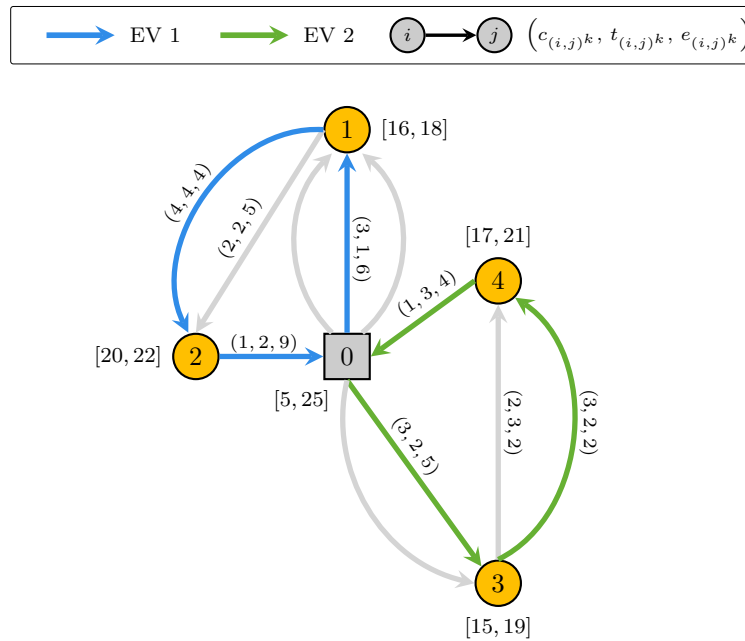


Figure 3: Routing component of the illustrative solution. Adapted from Yamín et al. (2025).

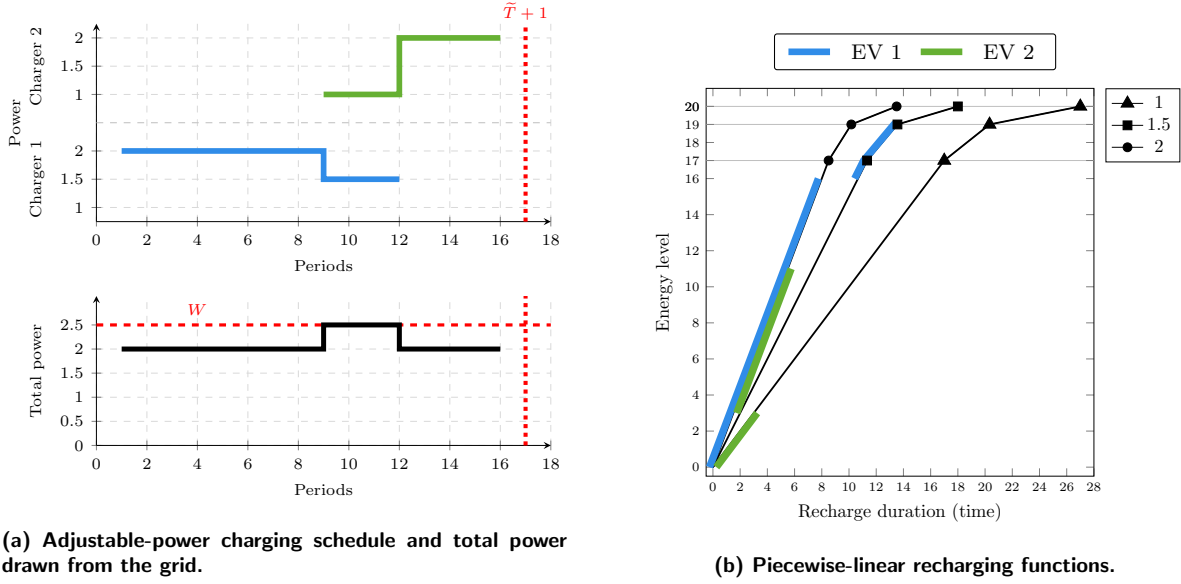


Figure 4: Charging component of the illustrative solution.

3.1 Set-partitioning formulation

We define a *route* by its visited customers, its traversed arcs, its charging periods, and the charging power used in each period. Let $a_i^r \in \{0, 1\}$ indicate whether route r visits customer $i \in \mathcal{C}$, $a_{(i,j)^k}^r \in \{0, 1\}$ whether it traverses arc $(i, j)^k \in \mathcal{A}$, $b_t^r \in \{0, 1\}$ whether it charges during period $t \in \mathcal{T}$, and $w_t^r \in (\{0\} \cup \mathcal{W})$ the charging power used in period $t \in \mathcal{T}$. Note that $b_t^r = 0$ if and only if $w_t^r = 0$. The total traveled distance defines the cost c_r of a route r , that is, $c_r := \sum_{(i,j)^k \in \mathcal{A}} c_{(i,j)^k} a_{(i,j)^k}^r$.

A route r is load-feasible if the total demand of its visited customers does not exceed vehicle capacity. It is time-feasible if service at each visited customer starts no later than the corresponding time window upper bound. Finally, it is energy-feasible if charging is non-preemptive, the charging power is adjusted at most N times, and the recharged energy is sufficient to complete the route.

Let $\mathcal{R}$ denote the set of all load-, time-, energy-feasible routes, and let $\lambda_r \in \{0, 1\}$ indicate whether route $r \in \mathcal{R}$ is selected in the solution. The mE-VRSPTW-AP can then be formulated as the following set-partitioning model:

$$\min \sum_{r \in \mathcal{R}} c_r \lambda_r \quad (1a)$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}} a_i^r \lambda_r = 1 \quad \forall i \in \mathcal{C}, \quad (1b)$$

$$\sum_{r \in \mathcal{R}} b_t^r \lambda_r \leq B \quad \forall t \in \mathcal{T}, \quad (1c)$$

$$\sum_{r \in \mathcal{R}} w_t^r \lambda_r \leq W \quad \forall t \in \mathcal{T}, \quad (1d)$$

$$\lambda_r \in \{0, 1\} \quad \forall r \in \mathcal{R}. \quad (1e)$$

The objective function (1a) aims at minimizing the overall routing costs. Constraints (1b) ensure that each customer is visited exactly once. Constraints (1c) impose charger availability, whereas constraints (1d) enforce the maximum grid power capacity. Finally, constraints (1e) establish the binary domain of the variables. Model (1) is the so-called *integer master problem*.

4 Branch-price-cut-and-check algorithm

To solve the integer master problem (1), we devise a BPC-C algorithm (see Figure 5). At each node of the branch-and-bound (BB) tree, we start by solving the linear relaxation of (1), which is called the master problem (MP). Because constraints (1b), together with $\lambda_r \geq 0$, imply $0 \leq \lambda_r \leq 1$ for all $r \in \mathcal{R}$, no upper bounds are explicitly imposed on the λ_r variables in the MP. During the BB search, valid inequalities and branching constraints are progressively added to the MP.

The MP is solved via a column generation (CG) algorithm that alternates between a master problem restricted to a subset of route variables, referred to as the restricted master problem (RMP; §4.1), and a pricing problem (PP; §4.2). Out of the known routes, the RMP selects those used in a linear relaxation solution. For its part, the PP searches for promising routes (including their charging schedules) to be added to the RMP, or to certify that none exist. Accordingly, the PP enforces the intra-route constraints and is solved using a labeling algorithm (§4.3). When the PP fails to find a route with a negative reduced cost, the current (and possibly fractional) RMP solution is optimal for the MP, and its objective value provides a valid lower bound.

If the MP solution value exceeds the current primal bound or the solution is integer, the corresponding node in the BB tree can be fathomed by bound or by feasibility, respectively. Otherwise, if the routing component of the solution is fractional, we strengthen the linear relaxation by adding valid inequalities (§4.4) or enforce integrality in the routing decisions through branching (§4.5). Alternatively, if the routing component is integer and the charging-scheduling component is fractional, a standard branch-price-and-cut (BPC) algorithm would continue branching to enforce integrality in the remaining fractional decisions. Instead, at this point, the “check” procedure of our BPC-C algorithm determines whether a feasible charging schedule exists for the current integer routing solution by solving a checking problem (CP; §4.6). If such a schedule exists, the routing solution produced by the MP and the charging schedule returned by the CP together form a feasible solution to (1), allowing the node to be fathomed by feasibility; otherwise, the routing solution is proven infeasible and is excluded from the search space by a *no-good* cut.

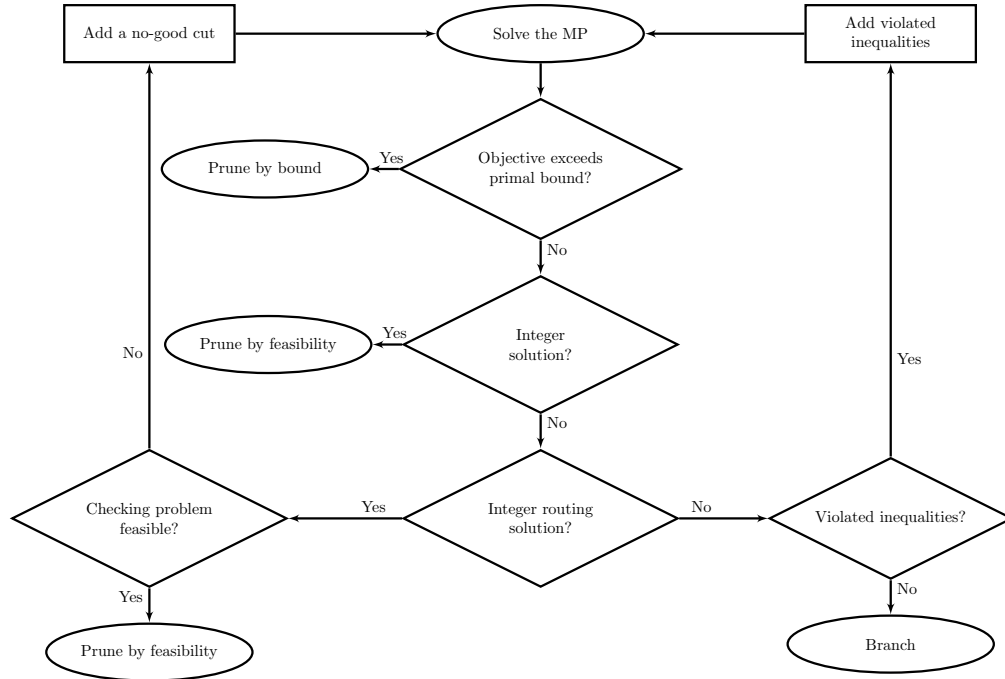


Figure 5: Solving a node in the BPC-C algorithm.

Our BPC-C algorithm builds on the BPC framework developed in Yamín et al. (2025) for the mE-VRSPTW, and extends it along three lines to accommodate adjustable charging power, grid capacity, and bounded power adjustments. First, the labeling algorithm that solves the PP must now track, in addition to the resources used in the fixed-power setting, the active piece of the recharging function and the number of power adjustments performed; correctly handling transitions between pieces within a single charging period is essential, as a naive treatment underestimates charging progress and can render feasible solutions unreachable. Second, the dominance rule of the labeling algorithm is redesigned: under adjustable power, two labels with comparable reduced cost and energy may differ in their remaining flexibility to switch power, and dominance must reflect this. Third, the MP alone no longer determines whether a routing solution admits a feasible charging schedule, because grid capacity couples the schedules of different routes; our dedicated CP, supported by a problem-specific power-requirement bound, resolves this question, updates the primal bound when a feasible schedule exists, and produces combinatorial Benders cuts otherwise. The remainder of this section describes each of these components in detail.

4.1 The restricted master problem

As mentioned above, the RMP is the MP restricted to a subset of routes $\mathcal{R}' \subseteq \mathcal{R}$. Solving it produces a primal solution and a complementary dual solution $(\alpha_i)_{i \in \mathcal{C}}$, $(\beta_t)_{t \in \mathcal{T}}$, and $(\omega_t)_{t \in \mathcal{T}}$ associated with constraints (1b), (1c), and (1d), respectively, where $\alpha_i \in \mathbb{R}$, $\beta_t \leq 0$, and $\omega_t \leq 0$. Extending the RMP primal solution by setting $\lambda_r = 0$ for all $r \in \mathcal{R} \setminus \mathcal{R}'$ is optimal for the MP if every such route has a nonnegative reduced cost. Consequently, the PP searches for routes with a negative reduced cost to add to $\mathcal{R}'$, or proves that none exist, in which case CG terminates.

4.2 The pricing problem

The PP considers the routing and charging decisions of a single vehicle while enforcing all intra-route constraints. Given a dual solution to the RMP, the PP seeks routes $r \in \mathcal{R}$ with a negative reduced cost, i.e.,

$$c_r - \sum_{i \in \mathcal{C}} \alpha_i a_i^r - \sum_{t \in \mathcal{T}} \beta_t b_t^r - \sum_{t \in \mathcal{T}} \omega_t w_t^r < 0. \quad (2)$$

We define the PP on a directed multigraph $\mathcal{G}_P = (\mathcal{V}_P, \mathcal{A}_P)$. The vertex set is $\mathcal{V}_P = \{s\} \cup \mathcal{T} \cup \{\underline{0}\} \cup \mathcal{C} \cup \{\bar{0}\}$, where s is a dummy source vertex, $\underline{0}$ and $\bar{0}$ are the source and sink depot vertices, each charging period $t \in \mathcal{T}$ is represented by a charging-period vertex, and each customer $i \in \mathcal{C}$ is represented by a customer vertex. The arc set is $\mathcal{A}_P = \mathcal{A} \cup \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$, where

- $\mathcal{A}_1 := \{(t, \underline{0})^k \mid t \in \mathcal{T}, k = 1, \dots, M\}$ is the multiset of arcs from charging-period vertices to the source depot vertex,
- $\mathcal{A}_2 := \{(t, t+1)^k \mid t \in \mathcal{T}, t < \tilde{T}, k = 1, \dots, M\}$ is the multiset of arcs between successive charging-period vertices, and
- $\mathcal{A}_3 := \{(s, t)^k \mid t \in \mathcal{T}, k = 1\}$ is the set of arcs from the dummy source vertex to charging-period vertices.

The parallel arcs in $\mathcal{A}_1 \cup \mathcal{A}_2$ model the different charging powers. With this representation, each arc $(i, j)^k \in \mathcal{A}_P$ is assigned a modified cost $r_{(i, j)^k}$ as follows:

$$r_{(i, j)^k} = \begin{cases} c_{(i, j)^k} - \alpha_i, & (i, j)^k \in \mathcal{A}, \\ -\beta_i - \omega_i w_k, & (i, j)^k \in \mathcal{A}_1 \cup \mathcal{A}_2, \\ 0, & (i, j)^1 \in \mathcal{A}_3, \end{cases} \quad (3)$$

where $\alpha_{\underline{0}} = 0$. Figure 6 maps the routes of the solution in Figures 3 and 4 onto the corresponding pricing problem multigraph $\mathcal{G}_P$. For clarity, some arcs in $\mathcal{A}_P$ are omitted. Also, the thickness of

the charging arcs (those in $\mathcal{A}_1 \cup \mathcal{A}_2$) is related to the charging power; for example, the thickest arcs correspond to the highest charging power of 2.

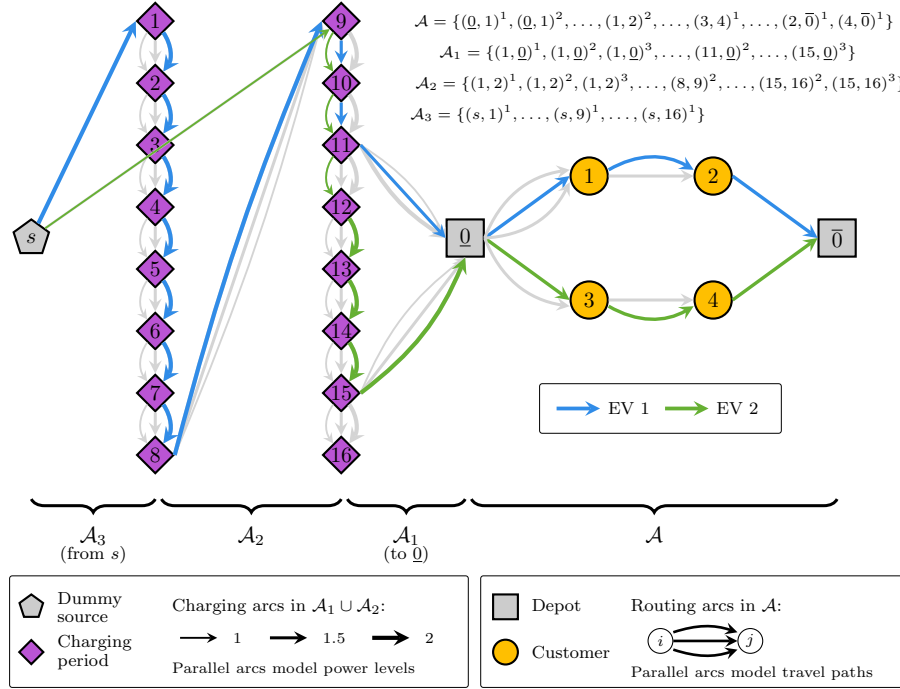


Figure 6: Illustrative pricing problem multigraph for the mE-VRSP-TW-AP. The charging arcs in $\mathcal{A}_1 \cup \mathcal{A}_2$ are parallel arcs associated with the charging power levels; thicker arcs represent higher charging power.

The PP is an Elementary Shortest Path Problem with Resource Constraints (ESPPRC) from s to $\bar{0}$, with arc costs given by (3). Solving the ESPPRC is strongly NP-hard, and it is therefore common to relax the elementarity requirement (Costa et al. 2019). Such relaxations are valid since constraints (1b) forbid nonelementary routes in integer solutions.

We adopt the *ng*-relaxation (Baldacci et al. 2011), which offers a good trade-off between computational effort and bound quality. In this relaxation, a neighborhood $\mathcal{N}_i \subseteq \mathcal{C}$ containing the Δ closest customers to i and customer i itself is defined for each customer $i \in \mathcal{C}$. A cycle involving customer j is allowed if and only if there exists a customer i in the cycle for which $j \notin \mathcal{N}_i$. Following Contardo et al. (2015), the neighborhoods $\mathcal{N}_i$ are initialized with a small Δ and iteratively enlarged whenever the PP produces only nonelementary routes, by adding the cycle-starting customer to the neighborhood of each customer in the cycle. The process stops when a negative reduced cost elementary route is found or a maximum neighborhood size $\bar{\Delta}$ is reached.

4.3 Labeling algorithm

The *ng*-SPPRC is solved using a backward labeling algorithm that builds feasible paths in $\mathcal{G}_P$ from the sink depot vertex $\bar{0}$ to the source vertex s . The algorithm initializes with the trivial subpath at $\bar{0}$ at time $\bar{T}$ and iteratively extends subpaths along incoming arcs, applying a dominance rule to avoid enumerating all feasible subpaths. The labeling process continues until no more subpaths can be extended or at least 400 complete s -to- $\bar{0}$ paths with a negative reduced cost are found, which correspond to feasible routes that can be added to $\mathcal{R}'$ in the RMP. Each subpath from an intermediate vertex $i \in \mathcal{V}_P$ to the depot sink $\bar{0}$ is represented by a *label* ℓ_i storing:

- the cumulative reduced cost $r(\ell_i) \in \mathbb{R}$,
- the remaining load capacity $q(\ell_i) \in \{0, 1, \dots, Q\}$,

- the latest time for starting service $t(\ell_i) \in [\underline{T}, \overline{T}]$,
- the remaining energy capacity $e(\ell_i) \in [0, E]$,
- the number of power adjustments $n(\ell_i) \in \{0, 1, \dots, N\}$,
- the power used on the last traversed arc $w(\ell_i) \in \mathcal{W}$,
- the amount of energy that still needs to be recharged $\bar{e}(\ell_i) \in [0, E]$, and
- the current piece of the recharging function $p(\ell_i) \in \mathcal{P}$.

For each customer $c \in \mathcal{C}$, the label also contains a resource $\mathbb{I}_c^u(\ell_i) \in \{0, 1\}$ indicating whether c is unreachable in any extension of the subpath due to resource limitations, and a resource $\mathbb{I}_c^{ng}(\ell_i) \in \{0, 1\}$ indicating whether visiting c would violate the ng -path cycling restrictions.

The labeling algorithm is initialized with a label $\ell_{\bar{0}}$ at the sink depot vertex, with $r(\ell_{\bar{0}}) = 0$, $q(\ell_{\bar{0}}) = Q$, $t(\ell_{\bar{0}}) = \overline{T}$, $e(\ell_{\bar{0}}) = E$, and $\mathbb{I}_c^u(\ell_{\bar{0}}) = \mathbb{I}_c^{ng}(\ell_{\bar{0}}) = 0$ for all $c \in \mathcal{C}$. The remaining resources are only relevant when exploring charging-period vertices and are therefore initialized when the propagation reaches the source depot $\underline{0}$. At each iteration, an unprocessed label ℓ_j at vertex j is selected and extended backward along each incoming arc $(i, j)^k \in \mathcal{A}_P$ to generate new labels. We chose to use this backward search because it allows to determine for every enumerated route its latest feasible departure time from the depot and its required energy before extending labels to charging-period vertices.

Algorithm 1 details the label extension to a customer or the source depot. A label is first discarded if the customer is unreachable or its visit induces a violation of the ng -path cycling restrictions (Line 1). Lines 2–5 update the resource consumptions. The label is discarded in Line 6 if the latest service start time precedes the time window opening or if the energy capacity is exceeded. Lines 9–11 update unreachable customers using lower bounds on travel time and energy consumption, while Lines 12–13 identify customers that cannot be visited due to the ng -path cycling restrictions. When the source depot $\underline{0}$ is reached, Lines 15–20 initialize the number of power adjustments, the power used on the last traversed arc, the amount of energy that still needs to be recharged, and the current piece of the recharging function. Finally, Line 21 returns the extended label.

When a label is extended to a charging period vertex or the dummy source vertex, Algorithm 2 describes the corresponding label extension procedure. Since the charging end time is fixed by the arc traversed from a charging period vertex to the depot source vertex $\underline{0}$, Line 1 discards the label if this time exceeds the route's latest start time. Line 2 checks that the battery is sufficiently charged when a label reaches the dummy source vertex or, conversely, that charging does not continue if the battery is already sufficiently charged. Line 3 handles the case in which the dummy source vertex is reached with sufficient energy, in which case the current label can be returned, as no further updates are necessary. Otherwise, Lines 4–15 update the resource consumptions. When the power of the extension arc differs from that of the previously traversed arc, the number of power adjustments is increased by one in Line 7. Lines 10–15 update the amount of energy that still needs to be recharged and the current piece of the recharging function, considering two cases: either the recharge occurs entirely within the same piece of the recharging function (Lines 10–12), or the recharge starts in one piece and ends in the preceding one (recall that we perform a backward search; Lines 14–15).

Figure 7 illustrates this second case. The correct computation of the remaining energy (in green) accounts for the change in the current piece of the recharging function. The incorrect computation (in red) ignores this transition, thereby underestimating the charging power and yielding longer charging schedules than necessary, which may incorrectly deem the solutions infeasible. Our computation assumes that at most one transition between pieces of the recharging function can occur within a single charging period. This assumption is not restrictive as a general expression can be derived to handle multiple transitions within a single period. However, violating this assumption indicates that the chosen time discretization lacks sufficient resolution.

Line 17 checks whether the number of power adjustments exceeds the maximum allowed or whether the remaining charging time is less than the time required to recharge the remaining energy. This latter time is computed using the maximum power when adjustments are still available (thus providing a

Algorithm 1: Label extension to a customer or the source depot.

Input: ℓ_j , label representing a subpath starting at vertex j ; $(i, j)^k \in \mathcal{A}$, arc along which the extension is performed.

Output: ℓ_i , label representing a subpath starting at vertex i .

```

1  /* check whether the extension is feasible */
2  if  $\mathbb{I}_i^u(\ell_j) = 1 \vee \mathbb{I}_i^{ng}(\ell_j) = 1$  then exit
3  /* update the resource consumptions */
4   $r(\ell_i) \leftarrow r(\ell_j) + r_{(i,j)^k}$ 
5   $q(\ell_i) \leftarrow q(\ell_j) - q_i$ 
6   $t(\ell_i) \leftarrow \min\{t(\ell_j) - t_{(i,j)^k}, \bar{t}_i\}$ 
7   $e(\ell_i) \leftarrow e(\ell_j) - e_{(i,j)^k}$ 
8  /* check whether the extension is actually feasible */
9  if  $t(\ell_i) < \underline{t}_i \vee e(\ell_i) < 0$  then exit
10 /* mark customers that are unreachable by resource or cycling constraints */
11 if  $i \in \mathcal{C}$  then
12   for  $c \in \mathcal{C}$  do
13     /* unreachable customers */
14      $\mathbb{I}_c^u(\ell_i) \leftarrow \mathbb{I}_c^u(\ell_j)$ 
15     if  $c \neq i \wedge \mathbb{I}_c^u(\ell_i) = 0 \wedge (q(\ell_i) < q_c \vee t(\ell_i) - \underline{t}_{(c,i)} < \underline{t}_c \vee e(\ell_i) < \underline{e}_{(c,i)})$  then
16        $\mathbb{I}_c^u(\ell_i) \leftarrow 1$ 
17     end
18     /* ng-path cycling restrictions */
19     if  $c = i \vee (\mathbb{I}_c^{ng}(\ell_j) = 1 \wedge c \in \mathcal{N}_i)$  then  $\mathbb{I}_c^{ng}(\ell_i) \leftarrow 1$ 
20     else  $\mathbb{I}_c^{ng}(\ell_i) \leftarrow 0$ 
21   end
22 end
23 /* initialize values for the charging period vertices exploration */
24 if  $i = 0$  then
25    $n(\ell_i) \leftarrow -1$ 
26    $w(\ell_i) \leftarrow 0$ 
27    $\bar{e}(\ell_i) \leftarrow E - e(\ell_i)$ 
28   for  $p \in \mathcal{P}$  do
29     if  $\phi_{p-1} < \bar{e}(\ell_i) \wedge \bar{e}(\ell_i) \leq \phi_p$  then
30        $p(\ell_i) \leftarrow p$ 
31     end
32   end
33 end
34 return  $\ell_i$ 

```

lower bound), or the current power when no further adjustments are possible (thus providing an exact value). Line 18 discards labels with a nonnegative reduced cost, as this value is nondecreasing over the remainder of the exploration. Finally, Line 19 returns the extended label.

When the extension of a label is feasible, we check dominance among labels in the same vertex to discard non-useful subpaths. If the new label is dominated by an existing one, the former is discarded. Conversely, if an existing label is dominated by the new one, we discard the existing label. When two labels dominate each other, we keep one of them to guarantee exactness. We use two different dominance rules: one for a customer vertex (Definition 1) and another for charging period vertex (Definition 2).

Definition 1. At a customer vertex $i \in \mathcal{C}$, a label ℓ'_i is dominated by another label ℓ_i if the following conditions hold:

$$r(\ell_i) \leq r(\ell'_i), \quad (4a)$$

$$q(\ell_i) \geq q(\ell'_i), \quad (4b)$$

$$t(\ell_i) \geq t(\ell'_i), \quad (4c)$$

$$e(\ell_i) \geq e(\ell'_i), \quad (4d)$$

$$\max\{\mathbb{I}_c^u(\ell_i), \mathbb{I}_c^{ng}(\ell_i)\} \leq \max\{\mathbb{I}_c^u(\ell'_i), \mathbb{I}_c^{ng}(\ell'_i)\} \quad \forall c \in \mathcal{C}. \quad (4e)$$

Algorithm 2: Label extension to a charging period vertex or the dummy source vertex.

Input: ℓ_j , label representing a subpath starting at vertex j ; $(i, j)^k \in \mathcal{A}_P \setminus \mathcal{A}$, arc along which the extension is performed.

Output: ℓ_i , label representing a subpath starting at vertex i .

/ check whether the charging end time is feasible */*

1 **if** $j = 0 \wedge i > \left\lfloor \frac{t(\ell_j)}{\tau} \right\rfloor$ **then** **exit**

/ check whether the recharge has been completed */*

2 **if** $(i = s \wedge \bar{e}(\ell_j) \neq 0) \vee (i \neq s \wedge \bar{e}(\ell_j) = 0)$ **then** **exit**

3 **if** $i = s \wedge \bar{e}(\ell_j) = 0$ **then** **return** ℓ_i

/ update the resource consumptions */*

4 $r(\ell_i) \leftarrow r(\ell_j) + r_{(i,j)^k}$

5 $w(\ell_i) \leftarrow w_k$

6 **if** $w(\ell_i) \neq w(\ell_j)$ **then**

7 $n(\ell_i) \leftarrow n(\ell_j) + 1$

8 **else**

9 $n(\ell_i) \leftarrow n(\ell_j)$

10 **end**

11 **if** $g(w(\ell_i), \bar{e}(\ell_j)) - \tau \geq g(w(\ell_i), \phi_{p(\ell_j)-1})$ **then**

12 $p(\ell_i) \leftarrow p(\ell_j)$

13 $\bar{e}(\ell_i) \leftarrow \bar{e}(\ell_j) - w(\ell_i) \cdot \theta_{p(\ell_i)} \cdot \tau$

14 **else**

15 $p(\ell_i) \leftarrow \max\{1, p(\ell_j) - 1\}$

16 $\bar{e}(\ell_i) \leftarrow \phi_{p(\ell_j)-1} - w(\ell_i) \cdot \theta_{p(\ell_i)} \cdot (g(w(\ell_i), \phi_{p(\ell_j)-1}) - g(w(\ell_i), \bar{e}(\ell_j)) + \tau)$

17 **end**

18 $\bar{e}(\ell_i) \leftarrow \max\{\bar{e}(\ell_i), 0\}$

/ check whether the extension is feasible */*

19 **if** $n(\ell_i) > N \vee (n(\ell_i) < N \wedge i - 1 < \left\lceil \frac{g(w_M, \bar{e}(\ell_i))}{\tau} \right\rceil) \vee (n(\ell_i) = N \wedge i - 1 < \left\lceil \frac{g(w(\ell_i), \bar{e}(\ell_i))}{\tau} \right\rceil)$ **then** **exit**

20 **if** $r(\ell_i) \geq 0$ **then** **exit**

21 **return** ℓ_i

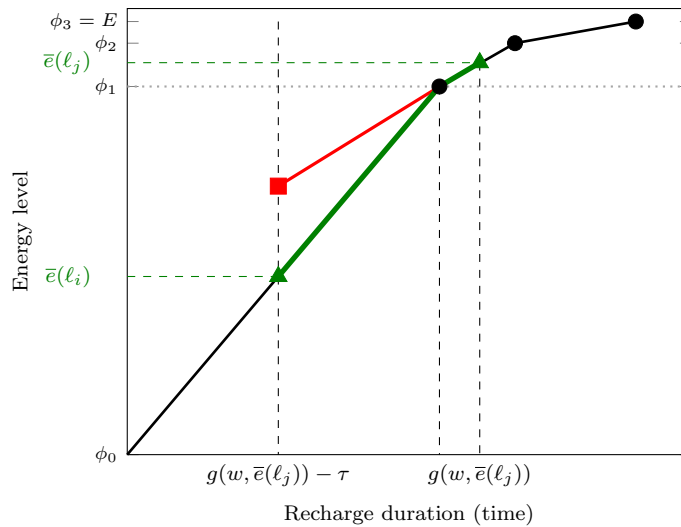


Figure 7: Transitioning between pieces of the recharging function within a charging period.

Definition 2. At a charging period vertex $i \in \mathcal{T}$, a label ℓ'_i is dominated by another label ℓ_i if the following conditions hold:

$$r(\ell_i) \leq r(\ell'_i), \quad (5a)$$

$$\bar{e}(\ell_i) \leq \bar{e}(\ell'_i), \quad (5b)$$

$$n(\ell_i) < n(\ell'_i) \vee (n(\ell_i) = n(\ell'_i) \wedge w(\ell_i) = w(\ell'_i)). \quad (5c)$$

Conditions (4) and (5) imply that the dominant label ℓ_i can be extended along any feasible extension of the dominated label ℓ'_i to yield an equal or lower reduced cost path. At a charging-period vertex, this implies that, in the remainder of the exploration, the dominant label's subpath can adjust the charging power more times than that of the dominated label, or the same number of times provided that both are currently using the same power, resulting in Condition (5c).

Heuristic labeling algorithm. We derive a heuristic labeling algorithm from the exact one by applying two modifications. First, we adopt a simplified dominance rule that ignores Condition (4e) in Definition 1. Second, we allow charging only at the maximum power w_M by removing all arcs $(i, j)^k \in \mathcal{A}_1 \cup \mathcal{A}_2$ with $k \neq M$. The exact labeling algorithm is invoked only if the heuristic fails to identify a route with a negative reduced cost.

4.4 Valid inequalities

We incorporate cutting planes that have proven effective in strengthening lower bounds in vehicle routing. At the root node, we strengthen the MP with the all-customer rounded capacity inequality:

$$\sum_{r \in \mathcal{R}} \lambda_r \geq \left\lceil \sum_{i \in \mathcal{C}} q_i / Q \right\rceil. \quad (6)$$

The MP is further tightened dynamically with three-customer subset-row cuts (SRCs) (Jepsen et al. 2008) that correspond to rank-1 Chvátal–Gomory cuts obtained from constraints (1b). For a customer triplet $S \subseteq \mathcal{C}$ with $|S| = 3$, the 3-SRC is

$$\sum_{r \in \mathcal{R}} \left\lfloor \frac{1}{2} \sum_{i \in S} a_i^r \right\rfloor \lambda_r \leq 1. \quad (7)$$

Like in Jepsen et al. (2008), separation is performed by enumeration. The SRCs are non-robust cuts (see Costa et al. (2019)), which means that handling their dual variables in the PP requires modifying the labeling algorithm. For details, see Jepsen et al. (2008) and Costa et al. (2019).

4.5 Branching rules

A solution expressed in terms of the route variables λ_r , $r \in \mathcal{R}$, admits an equivalent representation in terms of the arc-flow variables $x_{(i,j)^k}$, $(i, j)^k \in \mathcal{A}_P$. Given a fractional MP solution, we apply branching rules to enforce integrality in its *routing component*. Specifically, via branching, we enforce integer flow on the arcs in $\mathcal{A}$ (the routing component), although the flow on arcs in $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ (the charging scheduling component) may remain fractional.

To achieve this, first, we branch on the number of vehicles by modifying the lower bound in (6) or by imposing an upper bound on its left-hand side. When the number of vehicles used is integer, we branch on arc-flow variables $x_{(i,j)^k}$ with $(i, j)^k \in \mathcal{A}$ by selecting the one whose value is closest to 0.5. These branching decisions are imposed locally in the PP by removing arcs from $\mathcal{A}_P$ (see Yamín et al. (2025) for details).

One could derive branching rules on arcs $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ to obtain a feasible solution to (1) via branching alone. In fact, we implemented such branching rules initially. However, preliminary computational experiments showed that they led to very large branch-and-bound trees, with negligible improvements in the lower bound. This behavior can be explained by the limited impact of branching on these arcs when multiple charging schedules are feasible for a given set of routes, creating symmetry and rendering this approach inefficient, especially with adjustable charging power, which further amplifies this effect. As a result, the algorithm frequently reached the time limit or exhausted the available memory, and we turned our attention to the alternative approach presented next.

4.6 The checking problem

When the MP solution is fractional but none of the above branching rules can be enforced, the solution consists of integer routes with fractional charging schedules. Let $\bar{\mathcal{R}}$ denote the corresponding set of integer customer routes (without charging schedules). The goal of the checking problem (CP) is to determine whether there exists a feasible charging schedule for the customer routes in $\bar{\mathcal{R}}$. If such a schedule exists, then the current node of the BB tree can be pruned by feasibility and the incumbent solution updated. Otherwise, the routes in $\bar{\mathcal{R}}$ constitute an infeasible routing solution for model (1) and can be removed from the search space by a *no-good cut*.

For each route $r \in \bar{\mathcal{R}}$, let E_r be the energy required to serve it and let $\mathcal{T}^r := \{1, \dots, \bar{T}_r\} \subseteq \mathcal{T}$ be the set of charging periods during which it can charge, where $\bar{T}_r$ denotes the last charging period that allows the vehicle to depart on time. We derive a bound on the route's power requirement.

Proposition 1. For a route $r \in \bar{\mathcal{R}}$ with energy requirement $E_r \in (0, E]$, consider a charging schedule over consecutive periods in $\mathcal{T}^r$, with power $w_t \in \mathcal{W}$ in each charging period and $w_t = 0$ otherwise. Let $p' \in \mathcal{P}$ satisfy $\phi_{p'-1} \leq E_r \leq \phi_{p'}$, and define $W_r := \sum_{p=1}^{p'} \frac{\min\{E_r, \phi_p\} - \phi_{p-1}}{\theta_p \tau}$. The charging schedule reaches the energy level E_r if and only if $\sum_{t \in \mathcal{T}^r} w_t \geq W_r$.

Proof. ($\Rightarrow$) Suppose that the charging schedule reaches energy level E_r . For each piece $p \leq p'$, the amount of energy that must be charged while the battery energy lies in p is $E_r^{(p)} := \min\{E_r, \phi_p\} - \phi_{p-1}$. The battery energy evolves continuously over the period and may traverse multiple pieces. Let $\varphi_{t,p} \in [0, 1]$ denote the fraction of period t spent actively charging while the battery energy lies in piece p , for $p \leq p'$. These fractions are determined by the selected powers and the resulting battery-energy trajectory. By construction, $\sum_{p=1}^{p'} \varphi_{t,p} \leq 1$ for each period t . During the fraction $\varphi_{t,p}$ of period t spent in piece p , charging at power w_t increases the battery energy at a rate at most $\theta_p w_t$ per unit of time. Hence, the energy charged in piece p during period t is at most $\theta_p w_t \varphi_{t,p} \tau$. Summing over all charging periods yields $E_r^{(p)} \leq \sum_{t \in \mathcal{T}^r} \theta_p w_t \varphi_{t,p} \tau$, which implies $\sum_{t \in \mathcal{T}^r} w_t \varphi_{t,p} \geq \frac{E_r^{(p)}}{\theta_p \tau}$. Summing these inequalities over all pieces $p = 1, \dots, p'$ and exchanging the summation order gives

$$\sum_{t \in \mathcal{T}^r} w_t \left(\sum_{p=1}^{p'} \varphi_{t,p} \right) \geq \sum_{p=1}^{p'} \frac{E_r^{(p)}}{\theta_p \tau}.$$

Since $\sum_{p=1}^{p'} \varphi_{t,p} \leq 1$ for each $t \in \mathcal{T}^r$, it follows that $\sum_{t \in \mathcal{T}^r} w_t \geq \sum_{p=1}^{p'} \frac{E_r^{(p)}}{\theta_p \tau} = W_r$.

($\Leftarrow$) Conversely, suppose that $\sum_{t \in \mathcal{T}^r} w_t \geq W_r$. Apply the selected powers until E_r is reached or the schedule ends. Suppose, by contradiction, that the schedule ends with less than E_r units of energy. Let $\hat{E}_r^{(p)}$ denote the energy actually charged in piece $p \leq p'$, whereas $E_r^{(p)}$ is the amount required in that piece to reach E_r . Because the final energy is below E_r , $0 \leq \hat{E}_r^{(p)} \leq E_r^{(p)}$ for all $p \leq p'$, with strict inequality for at least one piece. Using the fractions $\varphi_{t,p}$ defined above, the energy actually charged in piece p is $\hat{E}_r^{(p)} = \sum_{t \in \mathcal{T}^r} \theta_p w_t \varphi_{t,p} \tau$ for all $p \leq p'$. Since every charging period is used in full and the battery energy remains below E_r , we have $\sum_{p=1}^{p'} \varphi_{t,p} = 1$ in every charging period. Together with $w_t = 0$ outside those periods, this gives

$$\tau \sum_{t \in \mathcal{T}^r} w_t = \tau \sum_{t \in \mathcal{T}^r} w_t \left(\sum_{p=1}^{p'} \varphi_{t,p} \right) = \sum_{p=1}^{p'} \frac{\hat{E}_r^{(p)}}{\theta_p} < \sum_{p=1}^{p'} \frac{E_r^{(p)}}{\theta_p} = \tau W_r,$$

where the strict inequality follows from $\theta_p > 0$ and the unmet energy requirement in at least one piece. This contradicts $\sum_{t \in \mathcal{T}^r} w_t \geq W_r$. $\square$

Proposition 1 provides a necessary and sufficient condition on the sum of charging powers for a schedule to reach the energy requirement of route $r \in \bar{\mathcal{R}}$. Using this result, we formulate the CP as

an integer program over a *charging scheduling network* $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ (see Figure 8). The vertex set is $\mathcal{V} := \{s\} \cup (\mathcal{T} \times \mathcal{W}) \cup \{0\}$, where s is a dummy source vertex, each charging period-power combination $(t, w) \in \mathcal{T} \times \mathcal{W}$ is represented by a vertex, and 0 denotes the source depot vertex, i.e., the end of the charging schedule. The arc set is $\mathcal{A} := \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4$, where

- $\mathcal{A}_1 := \{((t, w), 0) \mid (t, w) \in \mathcal{T} \times \mathcal{W}\}$ represents charging end times,
- $\mathcal{A}_2 := \{((t, w), (t+1, w)) \mid (t, w) \in \mathcal{T} \times \mathcal{W}, t < \tilde{T}\}$ represents one-period charging with no power adjustment in the following period,
- $\mathcal{A}_3 := \{((t, w_k), (t+1, w_{k'})) \mid (t, w_k) \in \mathcal{T} \times \mathcal{W}, t < \tilde{T}, k' \neq k\}$ represents one-period charging with a power adjustment in the following period, and
- $\mathcal{A}_4 := \{(s, (t, w)) \mid (t, w) \in \mathcal{T} \times \mathcal{W}\}$ represents charging start times.

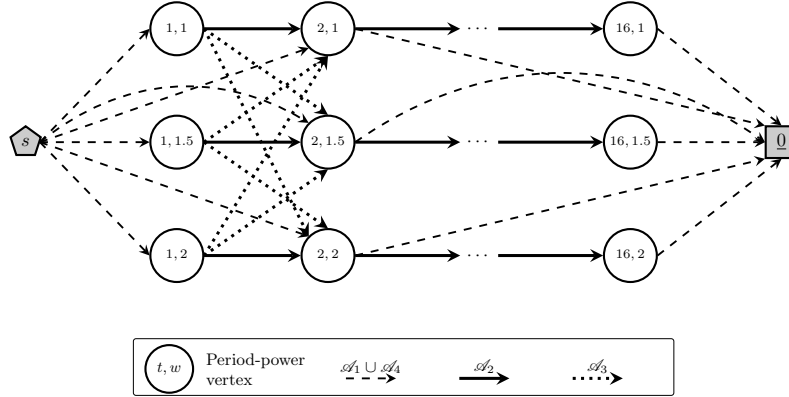


Figure 8: Illustrative charging scheduling network for the example in Figures 3 and 4. Not all arcs are depicted.

Any feasible charging schedule for route set $\overline{\mathcal{R}}$ can be mapped to an s -to- 0 flow of value $|\overline{\mathcal{R}}|$ in $\mathcal{G}$. Moreover, for each route $r \in \overline{\mathcal{R}}$, we define its subgraph $\mathcal{G}^r = (\mathcal{V}^r, \mathcal{A}^r)$ induced by the subset of vertices $\mathcal{V}^r := \{s\} \cup (\mathcal{T}^r \times \mathcal{W}) \cup \{0\} \subseteq \mathcal{V}$. Accordingly, $\mathcal{A}^r := \{(i, j) \in \mathcal{A} \mid i, j \in \mathcal{V}^r\}$.

Let $y_{i,j}^r \in \{0, 1\}$ be a flow variable associated with route $r \in \overline{\mathcal{R}}$ and arc $(i, j) \in \mathcal{A}^r$. Furthermore, let $\delta_r^+(i) := \{j \in \mathcal{V}^r \mid (i, j) \in \mathcal{A}^r\}$ and $\delta_r^-(j) := \{i \in \mathcal{V}^r \mid (i, j) \in \mathcal{A}^r\}$ denote the sets of outgoing and incoming arcs, respectively. The CP is the following feasibility integer program:

$$\min \quad 0 \quad (8a)$$

$$\text{s.t.} \quad \sum_{j \in \delta_r^+(s)} y_{s,j}^r = 1 \quad \forall r \in \overline{\mathcal{R}}, \quad (8b)$$

$$\sum_{j \in \delta_r^+(i)} y_{i,j}^r - \sum_{j \in \delta_r^-(i)} y_{j,i}^r = 0 \quad \forall i \in \mathcal{T}^r \times \mathcal{W}, r \in \overline{\mathcal{R}}, \quad (8c)$$

$$\sum_{r \in \overline{\mathcal{R}}: t \in \mathcal{T}^r} \sum_{w \in \mathcal{W}} \sum_{j \in \delta_r^+(t, w)} y_{(t, w), j}^r \leq B \quad \forall t \in \mathcal{T}, \quad (8d)$$

$$\sum_{r \in \overline{\mathcal{R}}: t \in \mathcal{T}^r} \sum_{w \in \mathcal{W}} \sum_{j \in \delta_r^+(t, w)} w \cdot y_{(t, w), j}^r \leq W \quad \forall t \in \mathcal{T}, \quad (8e)$$

$$\sum_{t \in \mathcal{T}^r} \sum_{w \in \mathcal{W}} \sum_{j \in \delta_r^+(t, w)} w \cdot y_{(t, w), j}^r \geq W_r \quad \forall r \in \overline{\mathcal{R}}, \quad (8f)$$

$$\sum_{(i, j) \in \mathcal{A}^r \cap \mathcal{A}_3} y_{i,j}^r \leq N \quad \forall r \in \overline{\mathcal{R}}, \quad (8g)$$

$$y_{i,j}^r \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}^r, r \in \overline{\mathcal{R}}. \quad (8h)$$

Constraints (8b) ensure that each route is assigned a charging schedule. Constraints (8c) enforce flow conservation in the network. Constraints (8d) and (8e) enforce the depot charger availability and the maximum grid power limit, respectively. Constraints (8f) ensure that each route receives sufficient energy, while Constraints (8g) limit the number of power adjustments. Finally, Constraints (8h) define the binary domain of the variables.

If CP (8) is feasible, then the customer routes in $\bar{\mathcal{R}}$ together with the corresponding charging schedule form a feasible solution to model (1). Otherwise, the routing solution is infeasible for the full problem and can be excluded from the search space.

No-good cuts Let $\bar{x}$ be the arc-flow (integer) solution over $\mathcal{A}$ defined by the routes in $\bar{\mathcal{R}}$. We can partition $\mathcal{A}$ into $X_0 := \{(i, j)^k \in \mathcal{A} : \bar{x}_{(i, j)^k} = 0\}$ and $X_1 := \{(i, j)^k \in \mathcal{A} : \bar{x}_{(i, j)^k} = 1\}$. A Combinatorial Benders' cut (Codato and Fischetti 2006) is given by:

$$\sum_{(i, j)^k \in X_0} x_{(i, j)^k} - \sum_{(i, j)^k \in X_1} x_{(i, j)^k} \geq 1 - |X_1|, \quad (9)$$

which removes the current integer routing solution from the search space by enforcing that any feasible solution differs in at least one arc-flow variable value. Cut (9) can be expressed in terms of the λ -variables by defining a new parameter $v^r := \sum_{(i, j)^k \in \mathcal{A}} \Theta_{(i, j)^k} a_{(i, j)^k}^r$ for each route $r \in \mathcal{R}$, with $\Theta_{(i, j)^k} := \left(\mathbf{1}\{(i, j)^k \in X_0\} - \mathbf{1}\{(i, j)^k \in X_1\} \right)$. The cut in the MP follows:

$$\sum_{r \in \mathcal{R}} v^r \lambda_r \geq 1 - |X_1|. \quad (10)$$

Let Υ be the set of no-good cuts (10) added to the MP. Now, the PP consists in identifying routes $r \in \mathcal{R}$ such that

$$c_r - \sum_{i \in \mathcal{C}} \alpha_i a_i^r - \sum_{t \in \mathcal{T}} \beta_t b_t^r - \sum_{t \in \mathcal{T}} \omega_t w_t^r - \sum_{\nu \in \Upsilon} \gamma_\nu v_\nu^r < 0,$$

where $\gamma_\nu \geq 0$ is the dual variable associated with cut $\nu \in \Upsilon$. These cuts are *robust*, meaning they do not increase the complexity of the PP, and only need to be accounted for in the modified arc cost as follows:

$$r_{(i, j)^k} = \begin{cases} c_{(i, j)^k} - \alpha_i - \sum_{\nu \in \Upsilon} \Theta_{(i, j)^k}^\nu \gamma_\nu, & (i, j)^k \in \mathcal{A}, \\ -\beta_i - \omega_i w_k, & (i, j)^k \in \mathcal{A}_1 \cup \mathcal{A}_2, \\ 0, & (i, j)^1 \in \mathcal{A}_3, \end{cases}$$

where parameter $\Theta_{(i, j)^k}^\nu$ is defined as $\Theta_{(i, j)^k}$ for the identified infeasible solution associated with the cut index ν . Intuitively, the reduced cost is *rewarded* by $-\gamma_\nu \leq 0$ when an arc $(i, j)^k \in X_0^\nu$ is used, whereas it is *penalized* by $\gamma_\nu \geq 0$ when an arc $(i, j)^k \in X_1^\nu$ is used.

5 Computational results

In this section, we report the computational results to assess the performance of the proposed BPC-C algorithm. It was implemented in Java using the Branch-and-Price framework of the Java OR library ([jORLib](#)) and the Java graph theory library ([JGraphT](#)) by Michail et al. (2020). To solve the RMPs, we used the IBM CPLEX solver, version 22.1. All experiments were run on a Linux server with an Intel Xeon Gold 6248R CPU @ 3.00 GHz, using 36 GB of Java heap memory. A 3-hour computational time limit was imposed for each instance. Our implementation extends the one available in Yamín et al. (2025).

5.1 Test instances

We adapt the instances in Yamín et al. (2025) to the adjustable charging power setting. These instances are publicly available at www.vrp-rep.org (VRP-REP-ID: 2024-0001; Mendoza et al. (2014)). The instances follow a four-parameter naming convention $DTm-n$, where D denotes the customer geographical distribution (R for random, C for clustered, and RC for mixed), T represents the tightness of the time windows (type 1 instances have tighter time windows than type 2), m is the two-digit instance number, and $n \in \{25, 50\}$ is the number of customers. The testbed comprises 112 instances, unevenly distributed across 12 groups, and includes the following information for each instance:

- the multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ and all associated parameters (e.g., customer demands and time windows, arc distances, travel times, and energy consumption),
- the vehicle load capacity Q and energy capacity E ,
- the number of chargers at the depot B_{old} ,
- the last charging period $\tilde{T}_{\text{old}}$, and
- a piecewise-linear recharging function based on a “slow” 11-kW charger.

To extend these instances to the adjustable power setting, we first adopt a time discretization of $\tau = 5$ min instead of 1 min. We consider $|\mathcal{W}| = 3$ discrete charging power levels, evenly spaced up to 10.79 kW, which corresponds to the effective power delivered by the 11-kW charger. In particular, $\mathcal{W} = \{3.59, 7.19, 10.79\}$ kW. We assume that the charging power can be adjusted at most $N = 3$ times. For each instance, we set the grid power capacity to $W := B_{\text{old}} \cdot 10.79$ kW and we increase the number of chargers by one, i.e., $B = B_{\text{old}} + 1$. This allows the same total grid power to be delivered as in the original testbed, while introducing additional charging flexibility through power adjustment across chargers. This modification also guarantees at least two chargers per instance and all B chargers can run simultaneously at minimum power. Finally, the recharging functions are defined by $(\phi_1, \phi_2, \phi_3) = (13.6, 15.2, 16)$ kWh and $(\theta_1, \theta_2, \theta_3) = (1.000, 0.529, 0.148)$, consistent with the empirical data in Montoya et al. (2017).

Under the above setting, the charging scheduling problem may be too loosely constrained to meaningfully evaluate adjustable charging power. Consequently, we shorten the charging horizon by redefining $\tilde{T}_{\text{old}}$ as follows. For each instance, we retrieve the set of routes $\bar{\mathcal{R}}$ (without charging schedules) in the optimal mE-VRSP-TW solution reported in Yamín et al. (2025) and their corresponding charging makespan $\bar{z}$. This value may be loose, as the algorithm delays route departures from the depot to allow additional recharging time and schedules charging close to departure times.

Next, we compute the minimum charging makespan for the same route set $\bar{\mathcal{R}}$ under the original testbed conditions but with B chargers (i.e., one additional charger), yielding a tighter makespan $\underline{z}$. We then define the charging horizon as $\tilde{T} = (1 - \mu)\underline{z} + \mu\bar{z}$, where, for each instance, μ is increased from 0 in increments of 0.2 until the resulting mE-VRSP-TW-AP instance can be proven feasible by our algorithm within three hours. Smaller values of μ produce shorter, more restrictive charging horizons and are therefore more challenging and informative in the adjustable-power setting. In §5.4, we analyze the effect of increasing μ , corresponding to progressively less constrained charging scenarios. Note that the minimum charging makespan associated with a given set of routes can be computed by solving (8) with minor modifications.

We remark that the instances are constructed using our algorithm. While an exogenous benchmark would be preferable, none is available for this problem. More importantly, adjustable charging power is meaningful only when charging schedules are sufficiently tight, which motivates using the algorithm to construct informative instances. Note that the instance construction is based on feasibility, not optimality.

5.2 Performance of the BPC-C algorithm

The computational performance of the BPC-C algorithm is summarized in Table 1 that reports average values for each instance group. Columns 2 and 3 present information related to the root node, namely, the lower bound (after adding cuts) and the number of separated SRCs. Columns 4 and 5 report the number of nodes explored in the BB tree and the number of times the CP was invoked. Column 6 shows the total computational time (in seconds), while Columns 7–9 report the proportion of this time spent on solving the RMP, the PP, and the CP, respectively. Finally, Column 10 provides the optimal objective value. We do not report the number of no-good cuts, as they were unnecessary for these instances. As discussed in §5.3 and §5.5, this is not always the case and is largely due to the strong bound provided by the integrated route-and-schedule representation.

Table 1: BPC-C algorithm computational performance.

Instance	LB	SRCs	Tree	CP Calls	Time				UB
					Total (s)	RMP (%)	PP (%)	CP (%)	
R1-25	495.2	15.4	39.5	1.2	44.2	3.7	33.7	58.9	497.1
C1-25	207.3	2.4	60.6	1.1	1,115.0	0.4	99.4	0.1	210.9
RC1-25	460.7	54.9	108.5	0.9	400.8	5.9	48.0	42.6	462.2
R2-25	444.4	12.1	113.9	1.0	897.9	4.6	50.8	41.8	444.9
C2-25	273.0	45.1	5.5	1.0	53.0	3.2	92.3	2.0	273.0
RC2-25	435.4	33.5	51.5	1.0	506.2	9.6	66.2	17.7	436.8
Average	386.0	27.3	63.3	1.0	502.2	4.6	65.7	27.2	387.5
R1-50	858.8	66.3	75.0	1.2	1,047.9	1.1	43.6	54.4	861.8
C1-50	376.8	8.6	18.3	1.0	611.5	0.9	97.2	1.1	382.4
RC1-50	818.1	107.2	43.5	1.0	2,735.6	0.9	84.6	14.0	819.6
R2-50	766.0	71.2	117.9	1.1	1,338.7	5.8	71.4	21.1	768.1
C2-50	485.2	86.2	8.8	1.0	2,615.0	2.0	95.9	1.7	485.3
RC2-50	705.4	30.6	50.0	1.0	526.1	2.6	64.1	31.2	706.6
Average	668.4	61.7	52.2	1.1	1,479.1	2.2	76.1	20.6	670.6

Within the time limit, the BPC-C algorithm solves to optimality all 112 instances. An average integrality gap of 0.4% indicates that the lower bounds are tight. This is partly due to the separated SRCs that help to yield relatively small BB trees, with 57 nodes on average. The CP is invoked, on average, once per run, yielding an incumbent solution that accelerates the exploration of the search tree.

The average running time for the 25-customer instances is less than 8.5 minutes, whereas for the 50-customer instances it is almost 25 minutes. The algorithm spends most of the time solving the PP, as is often the case in BPC algorithms for routing problems. However, in some cases, the CP can take a significant amount of time, as in the R1-25 and R1-50 instances, where the algorithm spends more time on it than on the PP. Still, this time is purposefully spent, as it ultimately leads to primal bound updates.

To complement the analysis, Table 2 presents key characteristics of the instances and solutions, reporting averages by instance group. Columns 2 and 3 show the number of chargers (which is constant within each group) and the last charging period, respectively. Column 4 reports the number of EVs used in the optimal solution, and Columns 5–9 report information on the associated charging schedule. Time in use represents the proportion of periods in the charging horizon during which at least one EV is charging. The charger and power utilization rates (Columns 6 and 7) correspond to the proportion of available chargers and grid capacity used during this time frame. Finally, charger and power time at capacity (Columns 8 and 9) represent the proportion of this time frame during which chargers and grid capacity are fully utilized.

The time in use exceeds 98%, indicating that chargers are almost continuously utilized, which is expected given the tight charging horizon. Moreover, power utilization is higher than charger

Table 2: Instance and solution characteristics.

Instance	B	$\tilde{T}$	EVs used	Time in use (%)	Utilization (%)		Time at capacity (%)	
					Charger	Power	Charger	Power
R1-25	4	31.6	6.0	100.0	92.7	98.6	72.7	93.8
C1-25	2	36.1	3.2	99.4	56.2	98.8	13.1	97.6
RC1-25	4	27.8	5.0	99.5	82.8	96.9	40.3	95.2
R2-25	2	84.1	4.8	99.7	68.2	98.8	36.7	97.5
C2-25	2	87.2	3.0	82.5	45.3	75.3	8.2	63.8
RC2-25	2	75.0	4.6	99.8	61.7	98.7	23.6	97.1
Average	2.7	57.0	4.4	96.8	67.8	94.5	32.4	90.8
R1-50	7	31.0	10.2	100.0	95.0	97.0	75.5	78.5
C1-50	3	35.6	5.2	100.0	85.0	97.0	58.3	90.6
RC1-50	5	32.0	7.6	100.0	89.7	94.6	63.5	77.6
R2-50	3	81.2	8.1	100.0	89.2	98.2	68.6	93.5
C2-50	2	113.9	5.4	97.0	57.6	93.6	18.1	88.2
RC2-50	3	55.5	6.4	100.0	86.0	97.9	58.6	91.0
Average	3.8	58.2	7.2	99.5	83.8	96.4	57.1	86.6

utilization. This stems from grid capacity constraints being more restrictive than charger availability constraints and sheds light on the role of adjustable charging power: either a few chargers operate at high or medium power levels, or all chargers operate at low power levels, while grid capacity is fully utilized. This trend is confirmed by the time at capacity: for chargers it is about 44%, whereas for power it exceeds 88%. Finally, the charging scheduling in C2-25 and C2-50 is loose. As discussed in §5.3, adjustable power cannot be effectively exploited in such cases. Although we could have generated tighter instances by refining the selection of μ , we deliberately chose not to, as this allows us to analyze the algorithm's performance and the diminishing benefits of adjustable power in such contexts.

Altogether, the proposed algorithm efficiently solves instances with up to 50 customers and three adjustable charging power levels to optimality. The significance of these results lies in the integration of a continuous-time VRP on a multigraph with a discrete-time charging scheduling problem with adjustable power levels, two problems that are already challenging when considered independently.

5.3 Impact of adjustable charging power on solution cost and computational time

In this section, we assess the impact of adjustable charging power on both solution cost and computational time. To this end, we create an additional copy of the 112 instances in which charging at each charger is allowed only at maximum power and the number of chargers is reduced by one, while keeping the same grid capacity W and charging scheduling horizon $\tilde{T}$. We refer to this additional copy as *fixed power* and the original set as *adjustable power*.

Table 3 compares the adjustable and fixed power settings by reporting average values for each instance group and considering only the feasible instances, since some fixed-power instances become infeasible (Column 2). For the adjustable power setting, Columns 3–7 report the number of generated columns at the root node, the nodes in the BB tree, the computational time (in seconds), the proportion of this time spent solving the CP, and the optimal objective value. Columns 8–12 report the same information for the fixed power setting. Column 13 reports the gap between the fixed power solution cost and the adjustable power solution cost. Finally, Column 14 reports the speedup obtained when solving the fixed power variant relative to the adjustable power setting. As we further discuss below, no-good cuts are required only for instance C108-50 in the fixed-power setting.

The additional flexibility provided by adjustable charging power can render some instances feasible that are otherwise infeasible under fixed power. In particular, 21 out of the 112 instances become infeasible in the latter setting. Due to the enlarged solution space in the adjustable power case, the

Table 3: Comparison between adjustable and fixed charging power settings.

Instance	Feasible	Adjustable Power ($ \mathcal{W} = 3, B$)					Fixed Power ($ \mathcal{W} = 1, B - 1$)					Gap (%)	Speedup (Geo. Mean)
		Columns	Tree	Time		UB	Columns	Tree	Time		UB		
				Total (s)	CP (%)				Total (s)	CP (%)			
R1-25	8/12	1,792.9	6.3	22.5	58.0	463.1	882.5	4.0	1.8	0.0	486.0	4.72	10.75
C1-25	9/9	2,232.9	60.6	1,115.0	0.1	210.9	1,625.6	1.2	47.5	0.0	214.4	1.48	22.35
RC1-25	5/8	1,735.4	96.2	287.7	44.7	448.4	646.0	1.4	3.1	0.0	460.5	2.10	22.93
R2-25	10/11	3,358.8	95.6	422.0	45.9	445.8	2,308.1	2.4	2.5	6.7	450.8	1.09	30.55
C2-25	8/8	3,124.1	5.5	53.0	2.0	273.0	2,329.3	1.0	18.6	0.0	273.2	0.07	2.12
RC2-25	6/8	4,362.5	32.0	103.8	22.9	405.0	2,369.2	7.7	4.0	3.2	407.5	0.45	24.85
Average	46/56	2,767.8	49.4	334.0	28.9	374.4	1,693.4	2.9	12.9	1.7	382.1	1.65	14.39
R1-50	9/12	3,778.7	33.7	896.4	53.2	825.4	2,118.8	19.9	53.7	0.0	857.2	3.74	17.54
C1-50	5/9	5,013.8	19.0	853.7	0.3	376.3	3,749.4	33.4	1,685.7	0.0	384.4	2.10	1.70
RC1-50	5/8	4,514.4	15.4	3,918.4	1.6	723.3	2,881.0	1.4	2,837.2	0.0	765.7	5.54	1.36
R2-50	11/11	6,648.9	117.9	1,338.7	21.1	768.1	4,942.8	15.9	311.7	0.9	773.0	0.58	9.38
C2-50	8/8	9,677.3	8.8	2,615.0	1.7	485.3	7,391.6	1.0	1,348.2	0.0	485.4	0.03	2.38
RC2-50	7/8	5,992.9	21.6	313.6	33.8	695.9	4,120.1	1.9	44.3	0.1	704.6	1.13	6.10
Average	45/56	5,937.6	36.1	1,656.0	18.6	645.7	4,200.6	12.2	1,046.8	0.2	661.7	2.19	4.20

problem also becomes computationally more challenging. This can be observed from the average number of generated columns and the BB nodes. For the adjustable power setting, these values are approximately 4,352 and 43, respectively, whereas for the fixed power setting they are 2,947 and 8. This difference illustrates the *controlled* combinatorial growth induced by adjustable charging power decisions.

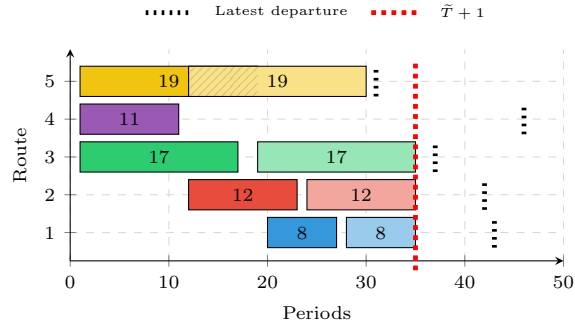
Consequently, the 25-customer fixed-power instances can be solved about 14.4 times faster than those with adjustable power, while for the 50-customer instances the speedup is 4.2. Despite this additional computational burden, the expanded solution space allows the algorithm to obtain lower-cost solutions. In particular, the fixed-power solution is 1.7% and 2.2% more expensive than the adjustable-power solution on the 25- and 50-customer instances, respectively. For instances with tighter charging schedules (see Table 2), the improvement is more pronounced, as adjustable power unlocks new, cheaper solutions (e.g., R1-50). Conversely, when the charging schedule is sufficiently loose, the benefits of adjustable power are incremental or nonexistent (e.g., C2-25 and C2-50).

Analysis of instance C108-50. To provide further insights into the benefits of adjustable charging power and the role of no-good cuts in our exact method, Figure 9 shows an intermediate fixed-power (fractional) solution encountered in the search tree, with an integer routing component and a total cost of 381.7. Figure 9(a) reports the cost, energy consumption, visited customers, charging periods, and value of each route in the solution. Figure 9(b) shows the corresponding fixed-power charging schedule, in which two chargers are available, along with the latest departure time of each route and last charging period $\tilde{T} = 35$.

In this case, the CP fails to find a feasible charging schedule. Ignoring the departure time of route 5, we must partition the charging times of the routes $\{19, 11, 17, 12, 8\}$ into two subsets (corresponding to the two chargers), each with total duration of at most 35 charging periods. Since $19 + 17 = 36 > 35$, routes 5 and 3 must be assigned to different chargers. Consider the possible assignments of the remaining routes. If route 2 is placed with route 5, we obtain $19 + 12 = 31$, but the remaining routes require $17 + 11 + 8 = 36 > 35$, which cannot be accommodated on the other charger. If route 4 is instead assigned with route 5, we obtain $19 + 11 = 30$, but the remaining routes require $17 + 12 + 8 = 37 > 35$, which again cannot be accommodated on the other charger. Finally, if route 1 is placed with route 5, we obtain $19 + 8 = 27$, but the remaining routes require $17 + 12 + 11 = 40 > 35$, which is also infeasible. This example illustrates why no-good cuts are necessary in our exact algorithm and why the checking problem does not always produce a feasible charging schedule. As a reference, Figure 10 shows the optimal fixed-power solution, in which more expensive routes are selected to accommodate the charging schedule, yielding a total cost of 382.7.

#	Cost	Energy	Customers	Charging periods	Value
1	48.9	698	[20, 24, 25, 27, 29, 30, 28, 26, 22, 21]	[20, 27] [28, 35]	0.5 0.5
2	59.2	993	[5, 3, 7, 8, 10, 11, 9, 6, 4, 2, 1]	[12, 23] [24, 35]	0.5 0.5
3	110.1	1,428	[32, 33, 31, 35, 37, 38, 39, 36, 34]	[1, 17] [19, 35]	0.5 0.5
4	59.7	963	[43, 42, 41, 40, 44, 46, 45, 48, 50, 49, 47]	[1, 11]	1.0
5	103.8	1,513	[13, 17, 18, 19, 15, 16, 14, 12, 23]	[1, 19] [12, 30]	0.5 0.5

(a) Routing.

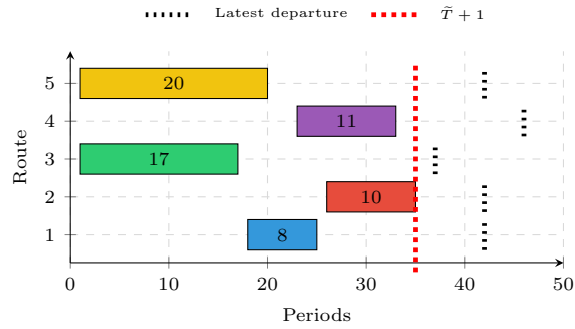


(b) Charging schedule.

Figure 9: Intermediate fixed-power (fractional) C108-50 solution.

#	Cost	Energy	Customers	Charging periods	Value
1	50.7	699	[20, 24, 25, 27, 29, 30, 28, 26, 23, 22, 21]	[18, 25]	1.0
2	66.4	899	[5, 3, 7, 8, 10, 11, 9, 6, 4, 2, 1]	[26, 35]	1.0
3	110.1	1,428	[32, 33, 31, 35, 37, 38, 39, 36, 34]	[1, 17]	1.0
4	59.7	963	[43, 42, 41, 40, 44, 46, 45, 48, 50, 49, 47]	[23, 33]	1.0
5	95.8	1,539	[13, 17, 18, 19, 15, 16, 14, 12]	[1, 20]	1.0

(a) Routing.



(b) Charging schedule.

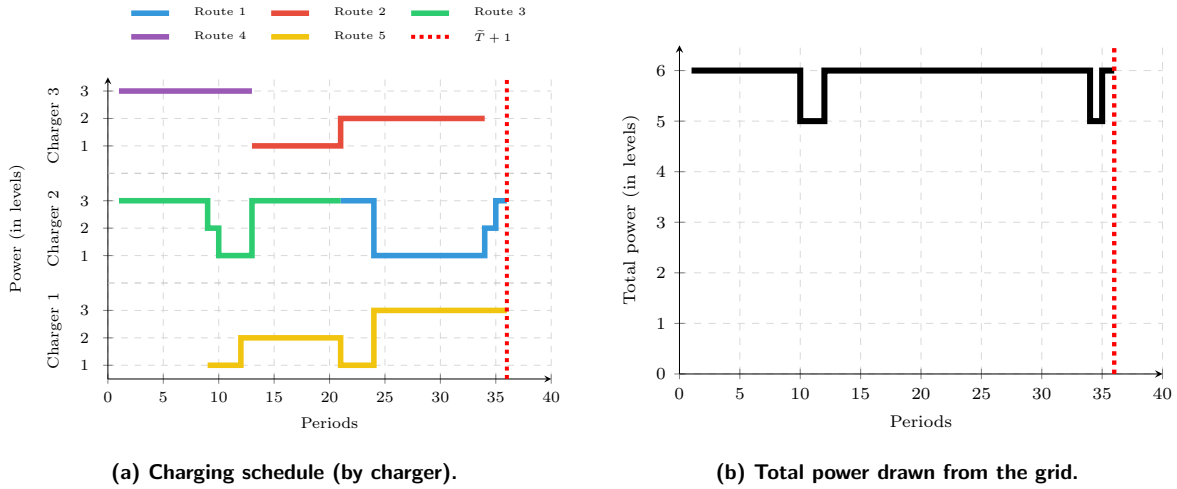
Figure 10: Optimal fixed-power C108-50 solution.

To contrast with the fixed-power solution, Table 4 and Figure 11 present the optimal adjustable-power solution. Table 4 reports the routing solution, with a total cost of 367.1. Figure 11(a) shows the corresponding adjustable-power charging schedule, in which $B = 3$ chargers are available, but at most two of them can operate simultaneously at full capacity due to the grid capacity constraints. Figure 11(b) shows the total power drawn from the grid in each period, measured in power levels.

Comparing this routing solution with that obtained under fixed power, we observe that the customer routes are identical. However, in the fixed-power setting, routes 2 and 3 have higher costs and lower energy consumption. In other words, to relieve the burden on the constrained charging schedule, these routes use arcs of the multigraph that consume less energy but require traveling a longer distance. Adjustable power introduces additional flexibility in the charging schedule, allowing these routes to instead select arcs that involve shorter travel distances even though they consume more energy. This flexibility can also be observed directly in the charging schedule. In some periods only two chargers operate at full capacity (e.g., period 1), while in others the full grid capacity is used but distributed across the three chargers. For example, in period 15 routes 2, 3, and 5 charge at power levels 1, 3, and 2, respectively. By exploiting this flexibility, the solution achieves a cost saving of 15.6 compared to the fixed-power case.

Table 4: Optimal adjustable-power C108-50 routing solution.

#	Cost	Energy	Customers	Charging Periods	Latest Departure	Value
1	50.7	699	[20, 24, 25, 27, 29, 30, 28, 26, 23, 22, 21]	[21, 35]	42	1.0
2	59.2	993	[5, 3, 7, 8, 10, 11, 9, 6, 4, 2, 1]	[13, 33]	42	1.0
3	101.7	1,481	[32, 33, 31, 35, 37, 38, 39, 36, 34]	[1, 20]	39	1.0
4	59.7	963	[43, 42, 41, 40, 44, 46, 45, 48, 50, 49, 47]	[1, 12]	46	1.0
5	95.8	1,539	[13, 17, 18, 19, 15, 16, 14, 12]	[9, 35]	42	1.0

**Figure 11: Optimal adjustable-power C108-50 charging scheduling solution.**

5.4 Sensitivity analysis on the charging scheduling time horizon

In this set of experiments, we evaluate the impact of extending the charging scheduling time horizon on solution cost and computational time. We restrict the analysis to the 25-customer instances, as looser charging schedules can create a prohibitively large state space (given the customer multigraph and adjustable power decisions), making a few 50-customer instances not solvable within the time limit. To this end, we create two additional copies of the 56 instances with 25 customers by increasing the multiplier μ^0 (defined in §5.1) by 0.2 and 0.4, generating instances with multipliers $\mu^1 = \mu^0 + 0.2$ and $\mu^2 = \mu^0 + 0.4$. Recall that increasing μ results in less constrained charging scenarios. Figure 12(a) presents box plots of the optimal costs for these three instance sets, while Figure 12(b) presents the box plots corresponding to computational times, with the y -axis shown on a logarithmic scale.

From Figure 12(a), we observe that the optimal cost decreases as the charging scheduling horizon becomes looser. The reduction is more pronounced from μ^0 to μ^1 than from μ^1 to μ^2 , as beyond a certain point enlarging the horizon no longer affects the optimal solution. Consistent with the previous discussion, in some cases (e.g., the C2 instances) increasing the charging horizon has little or no impact because the horizon with μ^0 is already too loose. In such cases, one may expect fixed and adjustable power to perform similarly, and the routing-only decomposition may also become effective, as routing solutions are more likely to admit feasible charging schedules.

From Figure 12(b), the algorithm may take over an order of magnitude longer to run when the charging schedule is more constrained. However, a looser charging horizon can also stall the algorithm due to the enlarged state space.

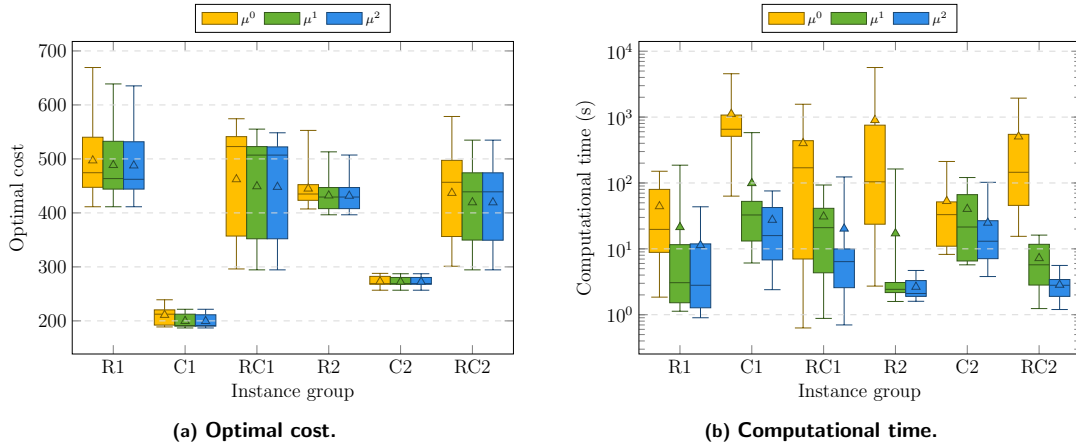


Figure 12: Sensitivity analysis on the charging scheduling time horizon (25-customer instances).

5.5 Benchmark against a routing-only decomposition and benefit of the integrated route-schedule representation

Our modeling approach inherits from Yamín et al. (2025) the pricing problem multigraph structure and the integrated route-and-schedule column representation. Because we develop a dedicated checking procedure for this new problem, one may wonder whether a routing-then-charging decomposition that leverages the checking problem and no-good cut mechanism yields a competitive alternative solution approach.

In this section, we benchmark our BPC-C algorithm against such a decomposition, in which only the routing component is considered (i.e., the charging scheduling component is not included in either the MP or the PP), and no-good cuts are added whenever the resulting routing solution does not admit a feasible charging schedule. The other components (e.g., SRC separation) are kept unchanged for a fair comparison. The potential benefit is that solving the routing component alone via CG can be much easier than solving the integrated routing and charging scheduling problem. The drawback is that the approach completely ignores the charging scheduling in the absence of no-good cuts. We limit this analysis to the Type 1, 25-customer instances, as they suffice to demonstrate that this alternative approach can be computationally intractable.

Table 5 reports the comparison between BPC-C with the integrated routing and scheduling representation and BPC-C with routing only. For the integrated routing and scheduling approach, Columns 2–7 report the lower bound at the root node, the BB nodes, the number of times the CP was invoked, the total computational time, the proportion of this time spent solving the CP, and the optimal objective value. Columns 8–14 report the same information for the routing-only approach and additionally include the number of no-good cuts in Column 11. The instances that are not listed correspond to cases where the routing-only approach exhausted the available memory (e.g., R101-25).

Both methods are exact and obtain the optimal objective value when the time limit is not exceeded. However, the integrated routing and scheduling approach yields a tighter average lower bound of 346.1 compared to 336.5 for the routing-only approach. This tighter bound leads to significantly smaller search trees (32 vs. 7,342 nodes, on average) and fewer calls to the CP (1 vs. 1,641, on average). As a result, the average running time for the integrated approach is roughly 480 seconds, whereas the routing-only approach requires 5,079 seconds on average and reaches the time limit in 9 out of the 22 instances reported in the table.

For the routing-only approach, we observe three cases. First, when the routing-only bound is sufficiently strong, the CP can be solved quickly, and only a few or no no-good cuts are required, then this approach can even be faster than the integrated one. This occurs, for example, in instance

Table 5: Comparison between BPC-C with integrated routing and scheduling and with routing only.

Instance	BPC-C Routing and Scheduling						BPC-C Routing Only						
	LB	Tree	CP Calls	Time		UB	LB	Tree	CP Calls	NG Cuts	Time		UB
				Total (s)	CP (%)						Total (s)	CP (%)	
R102-25	564.9	3	1	98.0	97.1	564.9	564.9	1	1	0	309.5	100.0	564.9
R103-25	493.0	13	1	10.1	53.3	493.8	485.4	5,789	862	861	194.0	67.6	493.8
R104-25	436.0	1	1	10.6	79.1	436.0	436.0	1	2	2	10,800.0	100.0	—
R106-25	518.8	25	1	18.2	25.4	531.9	515.7	2,219	348	346	244.7	90.8	531.9
R107-25	462.5	15	2	28.6	66.7	465.2	454.6	52	25	25	10,800.0	100.0	—
R108-25	417.2	1	1	1.9	13.0	417.2	417.2	1	1	0	0.8	43.2	417.2
R110-25	482.2	15	1	73.9	59.8	483.7	465.8	100,641	17,598	17,597	2,962.7	30.9	483.7
R111-25	462.5	15	2	21.3	52.1	465.2	454.6	636	319	319	10,800.0	99.8	—
R112-25	411.4	1	1	4.6	19.7	411.4	411.4	1	1	0	2.2	61.5	411.4
C101-25	224.5	347	1	4,552.1	0.0	239.1	191.7	9,903	4,908	4,908	10,800.0	1.6	—
C102-25	191.2	7	1	643.4	0.0	192.2	190.4	201	33	32	42.7	3.4	192.2
C103-25	190.3	1	1	63.1	0.2	190.3	190.3	1	1	0	3.6	2.8	190.3
C104-25	188.6	5	2	745.1	0.1	188.8	187.0	717	65	63	614.0	0.4	188.8
C105-25	212.2	17	1	254.8	0.0	212.4	191.5	4,087	2,038	2,038	10,800.0	0.6	—
C106-25	226.4	105	1	1,074.3	0.1	239.1	191.7	5,085	2,376	2,376	10,800.0	0.6	—
C107-25	219.5	23	1	653.7	0.0	220.4	191.5	2,425	1,204	1,204	10,800.0	0.4	—
C108-25	219.5	29	1	1,537.6	0.0	220.4	191.4	1,499	746	746	10,800.0	0.3	—
C109-25	193.4	11	1	511.1	0.0	195.1	191.4	3,901	557	555	9,562.1	0.3	195.1
RC104-25	317.3	1	0	0.7	0.0	317.3	313.1	125	22	21	6.3	10.1	317.3
RC106-25	516.6	55	2	232.7	56.2	518.1	510.7	6,018	2,995	2,995	10,800.0	97.3	—
RC107-25	369.2	11	1	9.1	12.9	370.4	363.1	18,177	2,000	1,998	587.3	10.6	370.4
RC108-25	296.2	1	0	0.6	0.0	296.2	294.6	47	12	9	3.2	11.6	296.2
Average	346.1	31.9	1.1	479.3	24.4	348.6	336.5	7,342.1	1,641.5	1,640.7	5,078.8	37.9	357.9

R108-25, where both approaches produce the same bound, which is also the optimal objective value. Second, the routing-only bound may still be strong, but the CP can take significant time to certify feasibility; for instance, for R102-25, both approaches attain the same bound, yet the routing-only approach takes almost three times longer due to the CP. Finally and most commonly (especially as instances grow larger and charging scheduling becomes tighter), the routing-only bound is too weak, making the approach computationally intractable. In this case, many CP calls are required, most of which prove infeasibility of the routing solution (since the charging schedule is ignored) and generate many no-good cuts; for example, for instance R110-25, the routing-only bound is 465.8 (compared to 482.2) and 17,597 no-good cuts are separated.

To fully evaluate the root-node bound improvement provided by the integrated route-and-schedule representation, Figure 13 displays box plots of these improvements across all 112 instances. Each box indicates the minimum, maximum, median, and quartiles of the bound improvement, which is measured as the gap with respect to the routing-only bound. The triangles represent the mean, and the points correspond to the individual instances.

The route-and-schedule representation yields significant bound improvements in most cases, reaching as high as 18% for the 25-customer instances and 12% for the 50-customer instances. The average improvement per group is also substantial, ranging from 0.18% to 8.7% for the 25-customer instances and from 0.6% to 5.4% for the 50-customer instances. Note that the benefit of this representation diminishes for the C2-25 and C2-50 instances. This occurs because, as discussed earlier, the charging scheduling problem in these instances is not sufficiently tight, making the routing-only bound relatively strong. Even in such cases, improving the bound through no-good cuts alone can be computationally prohibitive, especially for the 50-customer instances.

Overall, the routing-only approach may be useful in cases where the charging schedule is not actually binding. However, for larger instances with congested and tight charging schedules, the integrated route-and-schedule representation is clearly the preferable approach, as it ensures that at least a fractional charging schedule exists for the MP solutions. These results suggest that the integrated route-and-schedule representation may prove valuable for a broader class of problems involving tightly coupled, sequential operational decisions.

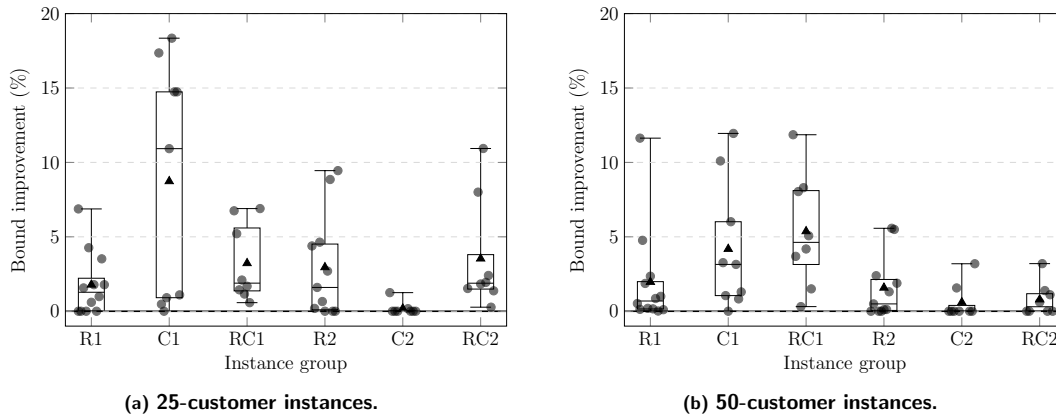


Figure 13: Root-node bound improvement of the integrated approach over routing only.

6 Conclusion

As electricity grid capacity is rapidly emerging as a major bottleneck in the large-scale integration of EVs into public transport and logistics systems, we address a new variant of the EV routing and charging scheduling problem that incorporates capacitated power grid constraints and adjustable charging decisions. The problem consists of determining a set of minimum-cost EV routes together with the preceding overnight on-site charging schedule. Routes must satisfy load, time window, and battery capacity constraints, while charging schedules must respect charger availability, grid capacity limits, and non-preemptive charging constraints. Piecewise-linear recharging functions and alternative arcs between locations are also considered.

To solve this problem, we developed an exact BPC-C algorithm that incorporates state-of-the-art techniques as well as problem-specific components. Our algorithm solves instances with up to 50 customers and three charging power levels to optimality, requiring about 8 minutes for the 25-customer instances and 25 minutes for the 50-customer instances, on average. Our computational experiments underscore the benefits of employing adjustable charging power, as the optimal solutions outperform, in terms of routing costs, those obtained on the same instances defined with fixed charging power. In fact, adjustable charging power can restore feasibility in instances that are infeasible under fixed power, at the expense of increased computational effort. Finally, our results suggest that both the CP and the integrated route-and-schedule master representation are key algorithmic components in making the EV routing and charging scheduling problem tractable in practice.

The modeling framework and solution approach presented here are sufficiently general to accommodate several relevant extensions. For instance, both the CG scheme and the network-based formulation of the CP can be adapted to incorporate TOU electricity tariffs or FRD charges. Furthermore, additional techniques can be adapted or developed to accelerate the solution method, especially in cases where the charging schedule is looser and the labeling algorithm or the CP may stall. These techniques include variable enumeration, variable fixing, branching rules used as an *escape* mechanism when the CP becomes too time-consuming, and *recombining* customer routes with previously generated charging schedules (or generating new charging schedules based on them). Finally, our results suggest that the integrated route-and-schedule representation, or a suitable abstraction thereof, may prove valuable for a broader class of problems involving tightly coupled, sequential operational decisions.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the authors used OpenAI's ChatGPT to assist with language editing and improving the clarity and presentation of selected text, mathematical explanations, and figures. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

- Abdelwahed A, van den Berg PL, Brandt T, Collins J, Ketter W (2020) Evaluating and optimizing opportunity fast-charging schedules in transit battery electric bus networks. *Transportation Science* 54(6):1601–1615.
- Baldacci R, Mingozzi A, Roberti R (2011) New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research* 59(5):1269–1283.
- Codato G, Fischetti M (2006) Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research* 54(4):756–766.
- Contardo C, Desaulniers G, Lessard F (2015) Reaching the elementary lower bound in the vehicle routing problem with time windows. *Networks* 65(1):88–99.
- Costa L, Contardo C, Desaulniers G (2019) Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science* 53(4):946–985.
- de Vos MH, van Lieshout RN, Dollevoet T (2024) Electric vehicle scheduling in public transit with capacitated charging stations. *Transportation Science* 58(2):279–294.
- Dolgui A, Kovalev S, Kovalyov MY (2025) Scheduling electric vehicle regular charging tasks: A review of deterministic models. *European Journal of Operational Research* 325(2):221–232.
- Froger A, Jabali O, Mendoza JE, Laporte G (2022) The electric vehicle routing problem with capacitated charging stations. *Transportation Science* 56(2):460–482.
- Hartvigsson E, Taljegard M, Odenberger M, Chen P (2022) A large-scale high-resolution geographic analysis of impacts of electric vehicle charging on low-voltage grids. *Energy* 261:125180.
- He Y, Liu Z, Song Z (2020) Optimal charging scheduling and management for a fast-charging battery electric bus system. *Transportation Research Part E: Logistics and Transportation Review* 142:102056.
- Jepsen M, Petersen B, Spoorendonk S, Pisinger D (2008) Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research* 56(2):497–511.
- Kullman ND, Goodson JC, Mendoza JE (2021) Electric vehicle routing with public charging stations. *Transportation Science* 55(3):637–659.
- Lam E, Desaulniers G, Stuckey PJ (2022) Branch-and-cut-and-price for the electric vehicle routing problem with time windows, piecewise-linear recharging and capacitated recharging stations. *Computers & Operations Research* 145:105870.
- Lera-Romero G, Bront JJM, Soullignac FJ (2024) A branch-cut-and-price algorithm for the time-dependent electric vehicle routing problem with time windows. *European Journal of Operational Research* 312(3):978–995.
- Löbel F, Borndörfer R, Weider S (2024) Electric bus scheduling with non-linear charging, power grid bottlenecks, and dynamic recharge rates. *arXiv preprint arXiv:2407.14446*.
- Mendoza JE, Guéret C, Hoskins M, Lobit H, Pillac V, Vidal T, Vigo D (2014) VRP-REP: A vehicle routing community repository. Third meeting of VeRoLog. Oslo, Norway.
- Michail D, Kinable J, Naveh B, Sichi JV (2020) JGraphT—A Java library for graph data structures and algorithms. *ACM Transactions on Mathematical Software* 46(2):1–29.
- Montoya A, Guéret C, Mendoza JE, Villegas JG (2016) A multi-space sampling heuristic for the green vehicle routing problem. *Transportation Research Part C: Emerging Technologies* 70:113–128.
- Montoya A, Guéret C, Mendoza JE, Villegas JG (2017) The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological* 103:87–110.
- Pelletier S, Jabali O, Laporte G (2018) Charge scheduling for electric freight vehicles. *Transportation Research Part B: Methodological* 115:246–269.

- Perumal SS, Lusby RM, Larsen J (2022) Electric bus planning & scheduling: A review of related problems and methodologies. *European Journal of Operational Research* 301(2):395–413.
- Wang W, Adulyasak Y, Cordeau JF (2026) Electric vehicle routing with heterogeneous charging stations. *Computers & Operations Research* 189:107374.
- Wang W, Adulyasak Y, Cordeau JF, He G (2025) The heterogeneous-fleet electric vehicle routing problem with nonlinear charging functions. *Transportation Research Part C: Emerging Technologies* 170:104932.
- Wu W, Lin Y, Liu R, Jin W (2022) The multi-depot electric vehicle scheduling problem with power grid characteristics. *Transportation Research Part B: Methodological* 155:322–347.
- Yamín D, Desaulniers G, Mendoza JE (2025) The electric vehicle routing and overnight charging scheduling problem on a multigraph. *INFORMS Journal on Computing* 37(4):808–830.