

A Constraint Programming Approach to Real-Time Electric Vehicle Itinerary Planning with Simultaneous Activities

Alexandra Wins^{a,d,*}, Benedikt Heidrich^a, Felix Haug^a, Sebastian Dorn^b, Daniel Bruckner^a, Sven Stahlmann^a, Jens Nahm^a, Sascha Alpers^c and Andreas Oberweis^d

^aMercedes-Benz Tech Innovation GmbH, Wilhelm-Runge-Str. 11, Ulm, 89081, Germany

^bMercedes-Benz AG, Mercedesstraße 120, Stuttgart, 70372, Germany

^cHeilbronn University of Applied Sciences, Max-Planck-Str. 39, Heilbronn, 74081, Germany

^dKarlsruhe Institute of Technology, Kaiserstr. 12, Karlsruhe, 76131, Germany

ARTICLE INFO

Keywords:

Electric Vehicle Routing
Itinerary Planning
Charging-Curve Linearization
Activity Scheduling
Constrained Shortest Path Problem
Constraint Programming

ABSTRACT

This article introduces the Electric Vehicle Itinerary Planning with Activities (EV-IPA) problem, which jointly optimizes vehicle routing, parallel activity scheduling, non-linear charging decisions, and time-window constraints. This tight coupling is increasingly relevant as conversational in-car assistants let drivers negotiate itineraries in natural language. However, directly embedding continuous, non-linear state-of-charge dynamics into a discrete decision space yields non-convex models incompatible with the sub-ten-second latencies demanded by interactive systems. We propose a Constraint Programming (CP) framework that decouples offline physics evaluation from online combinatorial search. Its core is a Dynamic Linear Energy Approximation that abstracts battery consumption into segment-wise bounds. Enforced via a structural Segment-Length Condition, this natively guarantees state-of-charge feasibility at every continuous point along the route. Furthermore, a hierarchical colocation graph captures parallel activities (e.g., charging while dining), and a precomputed convex envelope linearizes non-linear, temperature-dependent charging curves. On EV-IPA-1000, a new open benchmark of 1,000 European trip-planning instances, the framework produces itineraries with a median end-to-end latency of 1.58 seconds on production-grade data. By actively synchronizing charging events with user activities, the colocation architecture reduces average trip times by 7.9% on activity-bearing instances. Validation against a production-grade powertrain model confirms the fidelity of the linear abstractions: a 100% energy feasibility rate with mean state-of-charge and charge-gain deviations of 1.60 and 0.81 percentage points. The same rate is obtained on the open benchmark under its published surrogate model, and 99.9% is retained under a stress configuration that removes all soft safety margins.

1. Introduction

The transition to Battery Electric Vehicles (BEVs) makes mid-route charging mandatory for long trips. Because most mass-market EVs still use 400 V architectures, the allowable cable current and associated thermal constraints limit the charging power these vehicles can draw (Dutta and Bauman, 2024), so a typical 10–80 % DC fast-charging session usually takes roughly half an hour. Fast-charging coverage is also uneven: outside the trans-European core corridors, where regulation mandates high-power pools at regular intervals (European Parliament and Council of the European Union, 2023), most public charging points across Europe are urban rather than highway-adjacent (International Energy Agency, 2025). Given these constraints, prolonged vehicle dwell times become an unavoidable physical outcome. In response, drivers tend to reduce this perceived non-productive interval

by temporally aligning charging events with routine activities, such as eating or resting. However, conventional routing engines typically treat this as a post-processing filter, routing for energy first, then passively querying nearby points of interest. True efficiency requires a joint optimization paradigm: the vehicle's itinerary must explicitly balance charger power against human activity duration by actively routing to chargers spatially co-located with target amenities. We formalize this tightly coupled vehicle routing, facility selection, and activity scheduling problem as Electric Vehicle Itinerary Planning with Activities (EV-IPA), as illustrated in Figure 1.

Historically, capturing the complex, disjunctive preferences required to solve the EV-IPA problem (e.g., flexible time windows, precedence relationships, and conditional constraints) was prohibitive via traditional, touch-based vehicle interfaces. However, the recent integration of Large Language Models (LLMs) into automotive infotainment systems enables the extraction of highly expressive constraint profiles directly from natural speech. In the future, drivers could negotiate and iteratively critique their journeys on the fly. A user might naturally dictate: “*Stop for coffee around 1 pm, ideally somewhere we can charge at the same time. We need to be at the hotel for check-in by 9 pm, so plan dinner on the way around 7 pm, and if there's still time, fit in some sightseeing in Denmark.*”. While LLMs

*Corresponding author

✉ alexandra.wins@mercedes-benz.com (A. Wins);
benedikt.heidrich@mercedes-benz.com (B. Heidrich);
felix.haug@mercedes-benz.com (F. Haug);
sebastian.s.dorn@mercedes-benz.com (S. Dorn);
daniel.bruckner@mercedes-benz.com (D. Bruckner);
sven.stahlmann@mercedes-benz.com (S. Stahlmann);
jens.nahm@mercedes-benz.com (J. Nahm); sascha.alpers@hs-heilbronn.de
(S. Alpers); andreas.oberweis@kit.edu (A. Oberweis)
ORCID(s):

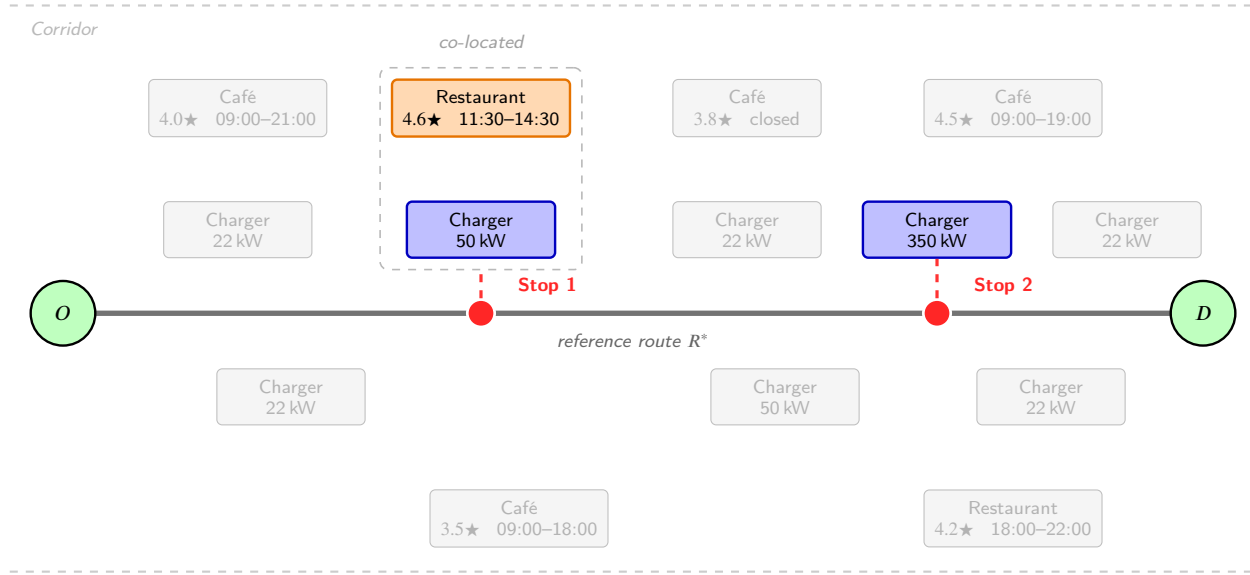


Figure 1: Illustrative EV-IPA instance. A reference route R^* runs from origin O to destination D . The dashed band depicts the candidate corridor, the geographic region around R^* , containing cafés/restaurants (with rating and opening hours) and chargers. The CP solver jointly selects (colored) a single stop colocating a 4.6★ restaurant with a 50 kW charger, letting the vehicle charge in parallel to the lunch window, and a second stop with a 350 kW charger before the destination. Non-selected candidates (gray; low rating, closed during the arrival window, or dominated on power/detour) remain in the candidate pool for re-optimization if the driver revises the request mid-conversation.

excel at parsing these constraints, they lack the deterministic reasoning capabilities required to natively solve the underlying mathematical scheduling problem (Xie et al., 2024). To sustain a fluid conversational exchange, the vehicle requires a dedicated combinatorial backend that accepts each turn’s constraint instantiation and returns a physically feasible itinerary within a sub-ten-second latency budget. This bound is not arbitrary: ten seconds is the established usability limit for keeping a user’s attention focused on an interactive dialog, beyond which users disengage and turn to other tasks while waiting (Nielsen et al., 1993).

Constraint Programming (CP) provides a declarative paradigm for modeling this conversational EV-IPA domain. The spatio-temporal requirements, conditional scheduling intervals, and logical disjunctions elicited from user interactions map directly onto native CP primitives and global constraints. Embedding the continuous, non-linear powertrain dynamics and the temperature- and state-dependent, non-convex charging curves directly in the discrete search space, however, yields models that cannot be solved within the interactive latency budget.

To resolve this, we propose a multi-stage CP framework that completely decouples offline physical evaluation from the online combinatorial search. Once an upstream routing component fixes a baseline reference path, a precomputation stage abstracts the continuous physical dynamics into conservative, route-dependent discrete artifacts. Operating directly on these inputs, the core methodological contribution of this work is a *Dynamic Linear Energy Approximation* (DLEA): it bounds continuous battery consumption

into segment-wise limits and, through a structural *Segment-Length Condition* (SLC), reduces the route’s non-linear energy dynamics to constraints the CP solver can enforce directly. Enforced natively within the solver as a minimum entry-energy constraint, the SLC mathematically guarantees physical energy feasibility at every continuous coordinate along the trajectory, given the conservative consumption and detour bounds and the realized charging gains (Assumptions 1, 3, validated in Section 7.4). Coupled with a hierarchical colocation graph formulation for parallel activity synchronization and a precomputed piecewise-linear convex envelope for non-linear charging curves, the framework compiles the full EV-IPA instance into a high-speed CP-SAT formulation. The pre-cached physical bounds let the model execute re-optimization queries across consecutive dialog turns within few seconds, as the user refines constraints.

The performance of the proposed framework is evaluated on a new open-source routing benchmark. We compare against an exact FRVCP solver (frvcpy (Froger et al., 2019; Kullman et al., 2021)) to quantify how closely our framework approaches the provably optimal charging strategy, against a sequential greedy heuristic to isolate the benefit of jointly optimizing routing, activities, and charging, and against a MILP formulation of the same model solved with HiGHS to isolate the choice of the search technology. The resulting itineraries are validated against a production-grade EV powertrain simulation. The primary contributions of this article are summarized as follows:

- (i) A **Dynamic Linear Energy Approximation** governed by a structural Segment-Length Condition that provides a provable mid-segment continuous-route

energy feasibility guarantee within a discrete search space.

- (ii) A PAVA-based construction of **context-conditioned charging envelopes**, turning production-grade, temperature-dependent tapering curves into linear constraints.
- (iii) A **hierarchical colocation graph** architecture that solves the parallel-execution problem of charging while performing an activity, by mapping spatially proximate services into synchronized scheduling intervals so that charging and human activities overlap rather than run sequentially.
- (iv) A **unified CP formulation** utilizing CP-SAT global constraints to solve highly constrained EV-IPA instances within interactive latency limits to support real-time vehicle critique loops.
- (v) The open-source, semi-synthetic **EV-IPA-1000 dataset**, establishing a reproducible public benchmark for charging-aware itinerary optimization.

The remainder of this article is organized as follows. Section 2 reviews related work in electric vehicle routing and activity-based scheduling. Section 3 formally defines the EV-IPA problem. Section 4 details the multi-stage framework. Section 5 derives the dynamic linear energy bounds and the associated continuous-route safety guarantees. Section 6 presents the complete CP-SAT mathematical model. Section 7 reports the empirical evaluation and validation results against the powertrain simulator, specifically addressing the following research questions:

- **RQ1 (Latency Consistency):** Can the CP framework consistently generate feasible itineraries within the strict interactive latency thresholds required by conversational in-vehicle systems?
- **RQ2 (Solution Quality):** How does the globally optimized CP approach perform relative to a chronological, look-ahead sequential heuristic baseline in minimizing total journey time?
- **RQ3 (Approximation Cost):** What price in solution optimality do the conservative linear energy abstractions and the candidate filtering incur, and what reduction in search effort do they buy?
- **RQ4 (Physical Fidelity):** Are the linear consumption bounds and charging envelopes faithful to high-fidelity powertrain physics: do planned itineraries remain energy-feasible in validation, and is the dynamic enforcement of the Segment-Length Condition necessary for this?

2. Related Work

Research on EV routing broadly divides into multi-vehicle fleet logistics and individual single-user navigation. The fleet side has been surveyed extensively (Erdelić

and Carić, 2019; Pelletier et al., 2016), with single-vehicle energy-optimal path problems treated there as special cases of the fleet problem rather than as a strand in their own right (Erdelić and Carić, 2019). The fleet-oriented literature (Erdelić and Carić, 2019; Kucukoglu et al., 2021; Pelletier et al., 2016; Xiao et al., 2021), with Xiao et al. (2021) additionally contributing a comprehensive MILP with nonlinear energy recharging and consumption, has evolved from the canonical Electric Vehicle Routing Problem with Time Windows (E-VRPTW) (Schneider et al., 2014) toward increasing physical realism, in particular regarding the charging process. Partial recharging under a linear charging model was solved to optimality early, by branch-price-and-cut over four E-VRPTW variants (Desaulniers et al., 2016) and alongside heuristic treatments (Keskin and Çatay, 2016). Non-linear charging functions were introduced to the E-VRP by Montoya et al. (2017) via piecewise-linear approximation and subsequently strengthened by Froger et al. (2019), who replace node-based time and state-of-charge tracking with arc-based tracking. Recent exact methods build on this: a branch-and-price algorithm that prices over charging-free route segments (Lee, 2021), a branch-and-cut-and-price algorithm for piecewise-linear recharging at capacitated stations (Lam et al., 2022), and a time-dependent branch-cut-and-price algorithm whose labels carry piecewise-linear state-of-charge functions (Lera-Romero et al., 2024). Where these exact methods certify optimality, matheuristics prioritize scale: Keskin and Çatay (2018) embed an exact charging-decision subproblem, a MILP over a fixed route, within an ALNS search. Exact and heuristic alike, these approaches minimize fleet-level objectives (vehicle count and travel distance, duration, or recharging cost) under offline compute budgets, where solving a single instance may take tens of minutes. Scheduling the activities of a human user and the interactive, corridor-based itinerary planning of an individual driver both lie outside their scope.

A second strand adds station congestion and operational uncertainty: time-dependent queueing at recharging stations (Keskin et al., 2019), explicit charger capacities around which visits must be scheduled (Froger et al., 2022), robust formulations against consumption uncertainty (Jeong et al., 2024; Pelletier et al., 2019), and link-level energy predictions whose variance informs chance-constrained charging decisions (Basso et al., 2021). EV-IPA models neither explicitly. Consumption enters as a point estimate from a production powertrain model, over which we impose deterministic conservative margins: a high-percentile rate over the links of a segment and a reserved energy surplus. Both are calibrated offline and enter the solver as constants, so the feasibility guarantee (Theorem 1) is deterministic under those bounds rather than probabilistic.

For individual EV navigation, finding an energy-optimal route between an origin and a destination is formulated as a constrained shortest path problem over the road graph, with battery bounds enforced at every prefix of the path (Artmeier et al., 2010). Merting et al. show that admitting recharging along the route changes the structure of the solution: because

resources can be regained along the way, an optimal route is in general not a shortest path but a shortest *walk*, which may revisit nodes arbitrarily often and may return to the same charging station more than once (Merting et al., 2015). Exact label-setting algorithms that propagate arc-wise consumption profiles together with a per-label state-of-charge function (Baum et al., 2015) guarantee battery feasibility over the entire trajectory, at the price of resolving energy inside the search on the road graph itself, arc by arc in the basic variant and, once contraction hierarchies are applied, over shortcut arcs each carrying a precomputed consumption profile for a fixed underlying path. For a fixed visit sequence, the residual charging subproblem (FRVCP) (Montoya et al., 2017), inserting charging stops and choosing charge quantities, admits an exact labeling solution (Froger et al., 2019), which we use as an optimality reference in Section 7. Closest to our setting in re-solving during the trip rather than once before departure, Dastpak et al. re-plan a single EV's charging stops and charging amounts whenever real-time information about station availability changes (Dastpak et al., 2024). The re-solve responds to the environment rather than to the driver, and each one addresses the same fixed set of decisions.

The integration of user activities into EV itineraries connects closely to Orienteering and Generalized Traveling Salesman Problem (GTSP) structures (Karbowska-Chilinska and Chociej, 2020; Schiffer et al., 2019; Wang et al., 2018), where an activity type is bound to a candidate set of admissible locations of which exactly one is selected. EV-IPA shares this activity resolution with the closely related joint planners WDMEV (Cuchy et al., 2018) and DACO-EV (Rizopoulos and Esztergár-Kiss, 2020, 2023), but targets a fundamentally different operational regime: the repeated, per-request re-solving of a single vehicle's itinerary in a production cloud, under an interactive latency budget and against that vehicle's live context. WDMEV achieves highly efficient query times by precomputing regional, all-pairs Pareto route sets offline (Cuchy et al., 2018), which suits static regional planning. In a multi-vehicle production cloud environment, however, energy consumption is highly sensitive to dynamic weather, live traffic, and vehicle-specific characteristics, and offline precomputation of all environmental permutations is intractable. EV-IPA therefore replaces region-wide static graphs with an on-demand, context-specific physics evaluation: consumption and charging curves are computed for the reference route and the candidate charging stations across discrete context buckets, while the energy costs of localized detours are conservatively approximated. This targeted setup requires only a few additional seconds on the first request for a route and establishes a dynamic cache over which subsequent conversational requests re-solve the joint decision space in milliseconds to few seconds (230–501 ms median; Section 7.8).

The conversational latency budget likewise forces a departure from how these planners model state-of-charge safety and activity synchronization. WDMEV derives its exact energy feasibility directly from the offline resolution

on granular road edges (Cuchy et al., 2018), whereas DACO-EV relies on evaluating the energy feasibility of complete tours within its genetic operators (Rizopoulos and Esztergár-Kiss, 2020). Because interactive response times preclude both exhaustive offline graph generation and granular online integration, EV-IPA instead evaluates conservative linear energy approximations over coarsened route segments. The Segment-Length Condition (SLC) then recovers a continuous feasibility guarantee within that abstraction rather than by resolving the energy exactly. Colocation of activities with charging stations, finally, is neither a fixed node property nor a post-processing step (Cuchy et al., 2018; Rizopoulos and Esztergár-Kiss, 2020) but a concurrent solver decision, with the shared dwell priced against high-fidelity, temperature-dependent charging envelopes.

Constraint Programming (CP) has received comparatively little attention in EV routing, it does not appear as a solution category in the classifications of the major surveys (Erdelić and Carić, 2019; Kucukoglu et al., 2021; Xiao et al., 2021). Booth and Beck (2019) introduced the first CP formulations for the E-VRPTW. Their single-resource transformation represents the problem as one task sequence over an extended horizon instead of modelling the vehicles explicitly, and it outperforms a general-purpose MILP solve of the Schneider et al. (2014) formulation on most medium-to-large problem classes, though not on distance minimization over long horizons. CP also appears as an embedded component in exact fleet optimization: Lam et al. delegate the scheduling of vehicles at capacitated chargers to a CP subproblem within branch-and-cut-and-price (Lam et al., 2022), evidence of CP's strength on disjunctive scheduling structure, though deployed inside an offline exact method.

Our use of CP is driven by the dynamic nature of conversational interfaces. The upstream dialog layer maps each natural language request onto a predefined vocabulary of requirements-spanning both hard constraints (like time windows, activity precedence, and appointments) and soft preferences (like charging behavior). Across different dialog turns, the active combination and parameterization of these constraints and preferences change constantly under a strict latency budget. CP is well-suited to this setting because any combination of these requirements can be handled by the same, unchanged search procedure. By contrast, adding a new requirement type to existing EV routing algorithms demands redesigns. In exact labeling methods, each new feature requires a custom label component and dominance rule (cf. Baum et al., 2015; Froger et al., 2019; Lera-Romero et al., 2024). In evolutionary approaches like DACO-EV, it requires modifying the genetic operators and fitness evaluation. In our CP model, introducing a new constraint or preference is simply a declarative modeling update, leaving the core search architecture completely intact. Ultimately, no existing formulation combines corridor-based itinerary planning, parallel activity execution, and non-linear charging with a continuous safety guarantee at sub-ten-second re-solving latencies.

3. Problem Definition: EV Itinerary Planning with Activities (EV-IPA)

We consider an electric vehicle (EV) driver travelling from an origin O to a destination D who, along the way, wants to complete a set of required *activity types* such as lunch, groceries, or buy present. Each activity type is an abstract requirement met by a *service* at some location near the route. A non-charging service, such as a restaurant or shop, is an *amenity*. Charging is a service too, but unlike the requested activity types it is discretionary, performed only where needed to keep the route energy-feasible rather than at the user's request. We assume that an upstream routing component has already fixed a reference road route R^* from O to D . The task is then to choose where to stop and which activities and charging to perform along and near this route.

The **EV-IPA** instance is defined by:

- (1) a fixed reference route R^* on a road network, with known travel times and energy consumption along R^* and metrics (e.g., distance) to estimate travel time and energy consumption for paths within a geographic corridor around it;
- (2) an effective battery interval $[E_{\min}, E_{\max_eff}]$, an initial energy E_{initial} at O , and a target energy E_{target} at D ;
- (3) a finite set of required activity types $\mathcal{A}_{\text{req}}$ (e.g., lunch, groceries, ...). Each requested type $a \in \mathcal{A}_{\text{req}}$ carries duration bounds $[d_a, \bar{d}_a]$ specifying how long the activity is to be performed, taken from the user request where stated (“a quick half-hour lunch”) and otherwise defaulted per activity category. A requested type may additionally arrive pinned to a user-supplied location and/or clock window (an appointment), in which case its candidate set reduces to the designated site;
- (4) a finite set of candidate locations $C = \{c_1, \dots, c_M\}$ in a corridor around R^* . Each candidate c comprises a physical stop anchor and one or more available services (e.g., dining, shopping, and/or vehicle charging). Each service is defined by its type, opening hours, and rating, its service time is that of the activity type it fulfils. For services offering charging, the maximum charging power is also provided. We write $\mathcal{A}(c) \subseteq \mathcal{A}_{\text{req}}$ for the set of required activity types that candidate c can support, and $C_{\text{charge}} \subseteq C$ for the subset of candidates that offer vehicle charging; and
- (5) vehicle-specific, non-linear charging curves mapping the required energy gain to a physical charging duration. These curves depend on the station's hardware capabilities (e.g., maximum power limit, AC/DC) and the vehicle's dynamic physical context upon arrival (e.g., ambient temperature along the route and prior driving time).

Solution: A solution is an itinerary consisting of: an ordered sequence of distinct candidates $(c_1, \dots, c_m)$ to be

visited, a vehicle path following R^* (with detours), for each visited candidate c_j , an arrival/departure time and choice of local activity instances, and for each charging location, a charging duration.

Feasibility: An itinerary is *feasible* if: every required activity type $a \in \mathcal{A}_{\text{req}}$ is performed at least once, travel times and service durations yield consistent schedules respecting time windows, and the energy model confirms the SoC stays above $E_{\min}$ and within $[E_{\min}, E_{\max_eff}]$ at all stops.

Objective: Among all feasible itineraries, the goal is to find one that provides an acceptable trade-off between total trip time, detours/number of stops, quality of selected POIs, and adherence to user preferences.

4. The Multi-Stage Segment-Based Heuristic Framework

We approximate EV-IPA via a multi-stage heuristic framework (Algorithm 1) that partitions the problem into offline preprocessing and online combinatorial search.

Stage 1 partitions the reference route into segments short enough that respecting an energy budget at their boundaries guarantees the battery stays within bounds everywhere inside. These segments then act as discrete accounting intervals. Stage 2 constructs a pruned candidate graph over those segments. Stage 3 solves a unified CP model on the graph. Figure 2 illustrates the pipeline on an Alpine example. This separation lets us replace continuous physics with linear energy approximations (Section 5.1) evaluated only at segment boundaries, while co-located services are grouped into sets with integer-encoded timing and charging decisions.

Alongside the EV-IPA instance introduced in Section 3, the framework consumes three precomputed artifacts: per-link energy rates along R^* (a link being a single edge of the underlying road network, so that each segment spans multiple links), candidate detour penalties to R^* , and piecewise-linear charging-time envelopes for each charger (Section 6).

4.1. Stage 1: Energy-Based Segmentation

Given R^* and the energy context $\Theta_{\text{ctx}} = (E_{\text{initial}}, E_{\min}, E_{\max_eff}, E_{\text{target}})$, we partition R^* into K segments by a single scan with a fixed per-segment energy budget $E_{\text{seg}} = \rho_{\text{seg}} E_{\max_eff}$ (Algorithm 1, Stage 1; ρ_{seg} in Table 3, swept in Section 7.7): the cut caps each segment's recuperation-clamped reference consumption and depends only on the per-link energies of R^* , so the partition is fixed before any charging decision. The Segment-Length Condition (Lemma 2) is enforced online as the entry-energy constraint (11) on $E_{\text{dep}, k-1}$.

4.2. Stage 2.1: Candidate Filtering and Proximity Clustering

To maintain CP solution times in the sub-second to few-seconds range, we apply a multi-stage filtering pipeline over C . After a time-window feasibility check (discarding candidates whose opening hours cannot intersect the vehicle's feasible arrival window, `TIMEWINDOWFILTER` in

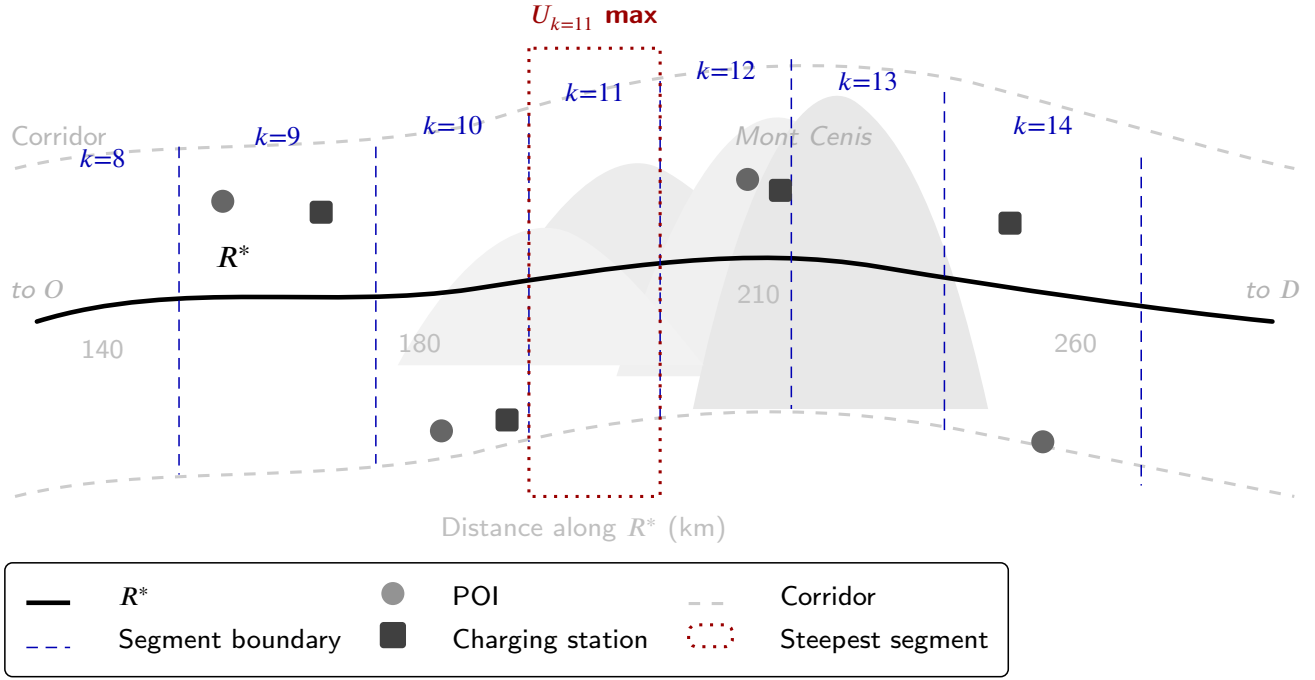


Figure 2: A mid-size EV with 60% initial SoC plans a 490 km Alpine route at 50% target SoC, requesting lunch and groceries. Stage 1 partitions the route into $K = 20$ segments. During the steep ascent ($k = 8-14$), the high consumption rate forces shorter segments. Stage 2.1 prunes the 287 POI and 94 charger candidates to 16 amenities and 22 chargers (one per bin at $B_{\text{chg}}=1$, plus colocation-partner retention), a 90% reduction. Stages 2.2 and 3 build the candidate graph and solve the CP model in 2.8 s, yielding a 2-stop itinerary that colocates lunch with a charger at a service area during the descent.

Algorithm 1), candidates are grouped by physical location into proximity clusters $\{P_\ell\}$. Within each cluster we locally deduplicate services, retaining the Pareto-optimal charging stations (along charging power p_c and detour time τ_c) and the highest-rated amenity per requested activity type.

Selection logic then splits by candidate type. Each energy segment is subdivided by energy into B_{chg} equal bins along R^* , since battery depletion ties charging opportunities to specific positions along the route. Within each bin, strict dominance pruning on (charging power, detour penalty) permanently discards inferior options. The single best remaining charger per bin is retained by a weighted utility

$$U_{\text{chg}}(c) = w_{\text{pow}} \hat{p}_c - w_{\text{det}} \hat{\tau}_c + w_{\text{rat}} \hat{\sigma}_c^{\text{am}}, \quad (1)$$

which rewards charging power, penalizes detour, and adds a bonus for colocation potential: p_c denotes the charger's maximum charging power, τ_c its estimated round-trip detour time, and $\sigma_c^{\text{am}} \in [0, 5]$ the five-star rating of the highest-rated amenity supporting a requested activity type within walking distance of c (radius ρ_{walk} , Section 6.2.1), with $\sigma_c^{\text{am}} = 0$ if no such amenity exists. Chargers with attractive co-located amenities are thus preferentially retained. The weights $w_{\text{pow}}, w_{\text{det}}, w_{\text{rat}} \in [0, 1]$ balance the three criteria. Each feature is scaled to $[0, 1]$ against a fixed reference maximum: $\hat{p}_c = p_c/p_{\text{max}}$, $\hat{\tau}_c = \tau_c/\tau_{\text{max}}$, and $\hat{\sigma}_c^{\text{am}} = \sigma_c^{\text{am}}/5$, with p_{max} and τ_{max} fixed network-wide constants.

Amenity candidates are evaluated globally rather than per segment, ranked by the utility

$$U_{\text{poi}}(c) = w_{\text{rat}} \hat{\sigma}_c - w_{\text{det}} \hat{\tau}_c + w_{\text{pow}} \hat{p}_c^{\text{chg}}, \quad (2)$$

the mirror image of Eq. (1): $\hat{p}_c^{\text{chg}}$ denotes the normalized maximum charging power among chargers within walking distance ρ_{walk} of c , with $\hat{p}_c^{\text{chg}} = 0$ if none exists. Just as chargers with attractive co-located amenities are preferentially retained, POIs with colocation potential are preferentially ranked. The top B_{poi} by this utility are kept. Two retention rules additionally exceed the budgets, completing this colocation bias. First, pairs are preserved in both directions: any POI sharing a proximity cluster with a selected charger is retained beyond the POI budget, and, conversely, for every retained amenity without a walkable retained charger, the best charger within ρ_{walk} is retained beyond the per-segment budget. Every amenity enters the graph together with its potential colocation partner, even where that partner lost its own selection round.

4.3. Stage 2.2: Graph Construction and Stage 3: Online CP Solve

The filtered candidates are assembled into the hierarchical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ formally introduced in Section 6.2.1: nodes represent services at specific locations, colocation sets Q group candidates sharing a physical stop (to model parallel execution such as charging while dining), and clusters $\{C_r\}$ group mutually exclusive candidates fulfilling a user

Algorithm 1: EV-IPA Multi-Stage Framework

Input : Reference route R^* with link rates $L_i \in \mathbb{R}$ and travel times $\tau_i \in \mathbb{R}_{\geq 0}$;
Energy context $\Theta_{\text{ctx}} = (E_{\text{initial}}, E_{\text{min}}, E_{\text{max_eff}}, E_{\text{target}}) \in (\mathbb{R}_{\geq 0})^4$;
Required activity types $\mathcal{A}_{\text{req}}$; each $a \in \mathcal{A}_{\text{req}}$ carries duration bounds $[\underline{d}_a, \bar{d}_a]$;
Raw candidate pool C ; each $c \in C$ has rating σ_c , window $[e_c, l_c]$,
activity types $\mathcal{A}(c)$, power p_c (chargers), detour estimate τ_c ;
Piecewise-linear charging envelopes $\{\Phi_c\}_{c \in C_{\text{charge}}}$ (Section 6)

Param : Segment energy budget $E_{\text{seg}} = \rho_{\text{seg}} E_{\text{max_eff}}$ (Table 3); filter budgets $B_{\text{chg}}, B_{\text{poi}} \in \mathbb{N}$ (retention may exceed them); walking radius ρ_{walk} ; weights of Eqs. 1–2

Output: Feasible itinerary $\pi = (c_1, \dots, c_m)$ minimizing the weighted trip-cost objective Z (Eq. 28)

```

// Stage 1: Energy-based segmentation (Section 4.1)
1  $k \leftarrow 1$ ;
2  $b \leftarrow 0$ ;
3 foreach link  $i$  along  $R^*$  do
4    $b \leftarrow b + \max(0, e_i)$ ;
5   if  $b > E_{\text{seg}}$  then
6     | split link  $i$  at the boundary; close  $s_k$ ;  $k \leftarrow k+1$ ;  $b \leftarrow b - E_{\text{seg}}$ ;
7   end
8 end
9  $s_k \leftarrow (L_{\text{ref},k}, U_k, L_{\text{det},k}, E_k^*)$  for all  $k$ ; // Section 5.1
10 Assign each candidate  $c \in C$  a segment index  $k(c)$  via spatial projection onto  $R^*$ ;

// Stage 2.1: Candidate filtering and pruning (Section 4.2)
11  $C' \leftarrow \text{TIMEWINDOWFILTER}(C, \{[\underline{d}_a, \bar{d}_a]\}_{a \in \mathcal{A}_{\text{req}}})$ ;
12  $\{P_\ell\} \leftarrow \text{PROXIMITYCLUSTER}(C')$ ;
13  $C_{\text{chg}} \leftarrow \emptyset$ ;
14 Deduplicate each  $P_\ell$ : keep the Pareto-optimal chargers w.r.t.  $(p_c, -\tau_c)$  and the highest-rated amenity supporting each  $a \in \mathcal{A}_{\text{req}}$ ;
15 foreach segment  $s_k$  and equal-energy bin  $b \in \{1, \dots, B_{\text{chg}}\}$  of  $s_k$  do
16   | add  $\arg \max_{c \in D_b} U_{\text{chg}}(c)$  to  $C_{\text{chg}}$ , with  $D_b$  the strictly non-dominated chargers of  $b$  w.r.t.  $(p_c, -\tau_c)$ ; // Eq. 1
17 end
18  $C_{\text{poi}} \leftarrow \text{top } B_{\text{poi}} \text{ amenity candidates by } U_{\text{poi}}$  (Eq. 2);
// Colocation-partner retention, both directions (may exceed the budgets)
19 Add the proximity-cluster amenities of every  $c \in C_{\text{chg}}$  to  $C_{\text{poi}}$ ;
20 For every  $c \in C_{\text{poi}}$  without a charger of  $C_{\text{chg}}$  within  $\rho_{\text{walk}}$ , add its best charger within  $\rho_{\text{walk}}$  (by  $U_{\text{chg}}$ ) to  $C_{\text{chg}}$ , if one exists;

// Stage 2.2: Graph construction (Section 4.3)
21  $C_{\text{sel}} \leftarrow C_{\text{chg}} \cup C_{\text{poi}}$ ;
22 foreach  $c \in C_{\text{sel}}$  do
23   | Refine detour penalties  $(\Delta E_c, \tau_c)$  via routing API;
24 end
// Merge walkable charger/amenity pairs into joint colocation sets
25 foreach walkable pair  $(c_{\text{chg}}, c_{\text{am}})$ ,  $c_{\text{chg}} \in C_{\text{chg}}$ ,  $c_{\text{am}} \in C_{\text{poi}}$  do
26   | Create colocation set  $Q \in \mathcal{Q}$  containing  $c_{\text{chg}}$  and  $c_{\text{am}}$ ;
27 end
28  $\mathcal{G} \leftarrow \text{BUILDGRAPH}(C_{\text{sel}}, \{s_k\})$ ; //  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{Q}, \{C_r\})$ 
29 // edges:  $\tau_{uv} \approx \tau_u + \tau_{\text{on-route}} + \tau_v$ 

// Stage 3: Online CP solve (Section 4.3)
30  $\pi \leftarrow \text{CPSATSOLVE}(\mathcal{G}, \{s_k\}, \{\Phi_c\}, \Theta_{\text{ctx}}, T_{\text{limit}})$ ;
31 return  $\pi$ ;

```

intent. Given $\mathcal{G}$, the segmentation $\{s_k\}$, and the precomputed charging envelopes $\{\Phi_c\}$, Stage 3 constructs the CP model of Section 6 and solves it under a 5-second search limit, returning an itinerary π consistent with all routing, temporal, and energy constraints.

5. Dynamic Energy Bounds and Linear Outer Approximations

The CP solver requires linear, integer-valued energy constraints, but battery dynamics along the route are continuous and non-linear. To bridge this gap, we introduce the *Dynamic Linear Energy Approximation* (DLEA), a linear

outer approximation of the continuous dynamics that decomposes R^* into segments acting as discrete energy accounting intervals.

The following subsections develop the DLEA in sequence: the adaptive segment-wise bounds (Section 5.1) and the structural assumptions they rely on (Section 5.2), the discrete energy flow at segment boundaries (Section 5.3), the dynamic energy budget for detours (Section 5.4), and the online enforcement of a minimum entry energy (Section 5.6). None of these alone guarantees continuous SoC feasibility: the endpoint analysis (Section 5.5, Lemma 1) first establishes feasibility at the discrete segment boundaries, and Theorem 1 extends the guarantee to every continuous point of the route.

5.1. Adaptive Segment-Specific Bounds

To ensure strictly conservative energy planning, we compute dynamic upper bounds based on the worst-case physical consumption of the road network. For each route segment, we aggregate high-resolution link-level data and then convert this data into a linear outer approximation that explicitly accounts for terrain-induced consumption peaks along the segment. Taking a high percentile of the per-link rates, rather than their mean, conservatively absorbs the segment's most demanding sub-sections. Combined with a gross-consumption floor (item 3 below), this yields a linear approximation that over-estimates the interior on-route consumption of the intended path by construction, and conservatively bounds localized detours through a distance-based rate inflated by a safety margin.

1. **Segment Reference Rate ($L_{\text{ref},k}$):** For each segment k , we compute a localized consumption baseline $L_{\text{ref},k}$ (in J/m). This is derived as the q -th percentile (e.g., $q = 90$) of the energy consumption rates of all individual road links contained within segment k .
2. **Recuperation Clamping:** Negative link consumption (recuperation) is clamped to zero per link during rate estimation, giving the gross total $E_k^+ := \sum_{i \in s_k} \max(e_i, 0)$, and the net segment total is floored likewise,

$$E_k^* := \max\left(\sum_{i \in s_k} e_i, 0\right) \leq E_k^+, \quad (3)$$

so that the linear lower bound remains monotonically non-increasing and never overestimates the SoC available after the segment, guarding against expected downhill gains that a detour fails to realize.

3. **Interior Upper Bound (U_k):** To bound consumption variance on the known route path, we define the segment's absolute energy limit U_k using an empirical safety multiplier $\gamma_{\text{safe}} > 1$ (e.g., 1.08):

$$U_k = \max(L_{\text{ref},k} \cdot |s_k|, E_k^+) \cdot \gamma_{\text{safe}} \quad (4)$$

where $|s_k|$ is the total geometric length of segment k . Figure 3 illustrates the resulting per-segment limit.

Since any interior prefix consumes, neglecting recuperation, at most $E_k^+ \leq U_k$, the interior component of Assumption 1 is satisfied by construction. In net-descending segments $E_k^* = 0$, while any interior ascent is still captured by $E_k^+ \leq U_k$. The flow terms (6)–(7) use the floored E_k^* .

4. **Localized Detour Rate ($L_{\text{det},k}$):** Detour costs are modeled with higher uncertainty. To prevent underestimation in downhill segments (where $L_{\text{ref},k} \rightarrow 0$), we introduce a minimum physical consumption floor L_{min} . The detour rate is defined as:

$$L_{\text{det},k} = \max(L_{\text{min}}, L_{\text{ref},k} \cdot \gamma_{\text{detour}}) \quad (5)$$

where $\gamma_{\text{detour}} \geq 1$ scales off-route uncertainty. In topographically uncertain regions it can be set above γ_{safe} (e.g., 1.25).

5.2. Foundational Assumptions

To establish formal feasibility guarantees for our linear approximation, we adopt the following structural assumptions for the energy model:

1. **Linear Outer Approximations:** Energy consumption within the interior of segment k (ignoring recuperation) is upper-bounded by U_k , off-route detour consumption is bounded by $L_{\text{det},k} \cdot d(\pi)$ for any off-route path π , and for every visited candidate v the priced detour energy ΔE_v upper-bounds the gross energy consumed on the driven detour plus any net recuperation of the bypassed stretch of R^* between its departure and rejoin points.
2. **Reference Feasibility:** The reference route R^* inherently respects battery bounds without any detours: $E_{\text{min}} \leq E_{\text{dep},k-1} - E_k^* \leq E_{\text{max,eff}}$ for all k .
3. **Charging Realizability:** On leaving each charging node v after the planned dwell dur_v , the physical state of charge is at least the planned post-charge level $y_{\text{post},v}$.

5.3. Segment Energy Flow and Linear Approximations

Rather than integrating SoC continuously, we track energy only at segment boundaries, distinguishing $E_{\text{arr},k}$ (energy on reaching the end of segment k , before charging) from $E_{\text{dep},k}$ (energy after charging).

Definition 1 (Discrete Energy Flow). For segment k , let ΔE_k^{det} be the aggregated decision-dependent detour consumption, and $G_k \geq 0$ be the aggregated charging gain assigned to the segment. The discrete linear dynamics exposed to the solver are:

$$E_{\text{arr},k} = E_{\text{dep},k-1} - E_k^* - \Delta E_k^{\text{det}} \quad (6)$$

$$E_{\text{dep},k} = E_{\text{arr},k} + G_k \quad (7)$$

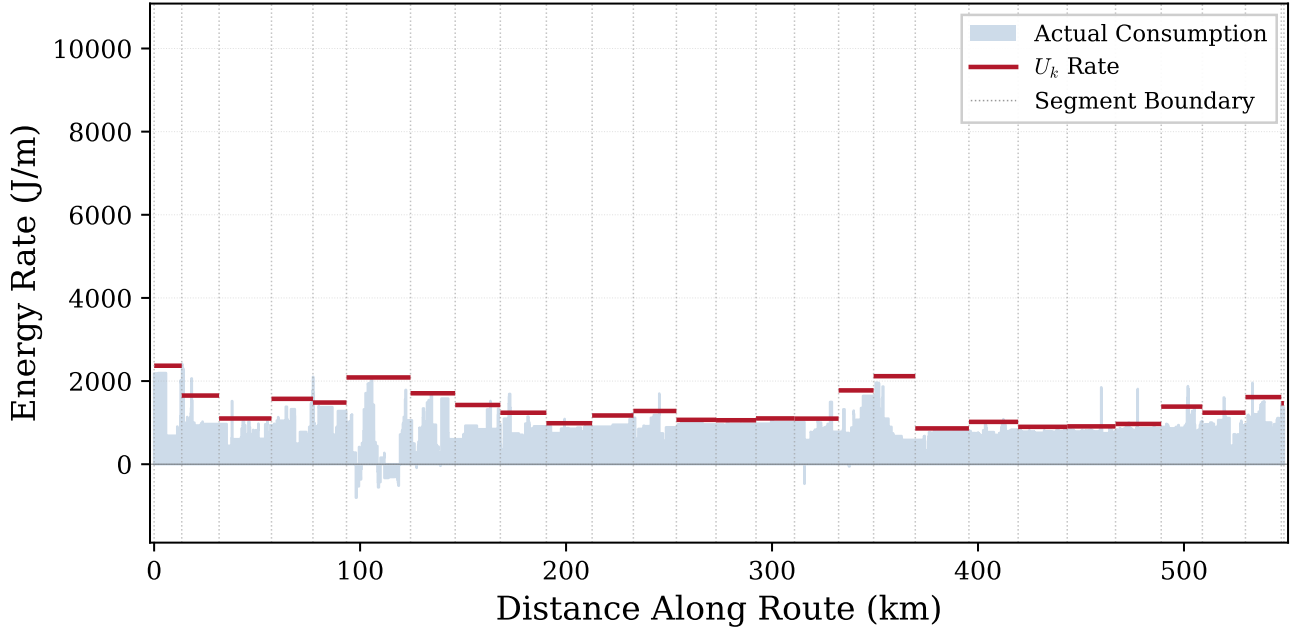


Figure 3: Adaptive segment-specific bounds applied to a sample route. The actual energy consumption profile (blue area) is estimated using a production-grade powertrain model. The red lines show the per-segment U_k rate, a few links exceed it, but applying this high-percentile rate over the full segment length (with the γ_{safe} margin) keeps U_k above the segment's total consumption.

subject to the battery limits $E_{\min} \leq E_{\text{arr},k} \leq E_{\text{max_eff}}$ and $E_{\min} \leq E_{\text{dep},k} \leq E_{\text{max_eff}}$. By construction (Section 5.1), $E_k^* \geq 0$ and $\Delta E_k^{\text{det}} \geq 0$.

Equation (6) deducts the entire segment's consumption before applying any charging gain G_k . This ordering enforces a mathematically strict lower bound, underestimating the intermediate SoC to guarantee protection against battery depletion ($E_{\min}$).

5.4. Dynamic Budget and Mid-Segment Feasibility

To prevent mid-segment depletion, we expose a decision-dependent energy budget to the solver.

Definition 2 (Dynamic Energy Budget). We define the post-boundary surplus B_k as the energy remaining at the start of segment k after deducting the unavoidable reference consumption and the minimum battery threshold, and the entry surplus $\tilde{B}_k$ as the energy above the threshold before any consumption is deducted:

$$\begin{aligned} B_k &:= E_{\text{dep},k-1} - E_k^* - E_{\min}, \\ \tilde{B}_k &:= E_{\text{dep},k-1} - E_{\min} = B_k + E_k^*. \end{aligned} \quad (8)$$

To absorb unpredicted spatial deviations, the total detour energy ΔE_k^{det} is constrained to a fraction $\alpha \in (0, 1)$ of the post-boundary surplus:

$$\Delta E_k^{\text{det}} \leq \alpha \cdot B_k. \quad (9)$$

Only this fraction of the post-boundary surplus may be spent on local detours. The Segment-Length Condition

below reserves $(1 - \alpha) \tilde{B}_k$ of the entry surplus to absorb the worst-case consumption of the linear approximation within the continuous segment interior.

The fractional form is deliberate: it bounds the least certified model input, the priced detour energies (Assumption 1), by a share of the available surplus rather than by a fixed reserve. The physical detour consumption may thus exceed its estimate by up to a factor of $1/\alpha$ (ten at $\alpha = 0.1$) before $E_{\min}$ is breached at the segment boundary (Lemma 1), whereas an additive reserve such as $E_{\text{dep},k-1} \geq E_{\min} + U_k + \Delta E_k^{\text{det}} + \epsilon_E(k)$ tolerates only a fixed absolute, rather than multiplicative, error. The trade-off is that the permissible detour energy shrinks with the available surplus (Table 9).

5.5. Theoretical Guarantees (Endpoints)

Lemma 1 (Arrival Feasibility). Assume the initial energy satisfies $E_{\min} \leq E_{\text{initial}} \leq E_{\text{max_eff}}$. If the Dynamic Energy Budget (Definition 2) is enforced alongside (6), and $B_k \geq 0$, then each segment arrival level inherently satisfies the minimum bound:

$$E_{\text{arr},k} \geq E_{\min} \quad \forall k.$$

Proof. Substituting (9) into (6) yields $E_{\text{arr},k} \geq E_{\text{dep},k-1} - E_k^* - \alpha B_k$. Expanding B_k via (8) gives $E_{\text{arr},k} \geq E_{\text{dep},k-1} - E_k^* - \alpha(E_{\text{dep},k-1} - E_k^* - E_{\min}) = E_{\min} + (1 - \alpha)B_k$. Since we constrain $B_k \geq 0$ and $\alpha \in (0, 1)$, it follows that $E_{\text{arr},k} \geq E_{\min}$. The upper bound trivially holds since $E_k^* \geq 0$ and $\Delta E_k^{\text{det}} \geq 0$, meaning energy strictly decreases prior to the charging event. $\square$

5.6. Dynamic Safety Buffer and Continuous Discretization Guarantee

While Lemma 1 guarantees feasibility at the discrete boundaries, we must also ensure that the vehicle does not deplete its battery at any continuous coordinate along the interior of segment k . Let $\varepsilon_E(k)$ denote the bound on the fixed-point quantization error incurred by the solver within segment k . With the fixed-point scaling of Section 7.2, each quantized energy term carries a rounding error of at most one scaling unit (10 kJ). We reserve $\varepsilon_E(k) = 30$ kJ per segment, one unit each for the rounded reference consumption, interior bound, and detour sum, negligible against $E_{\min}$ ($\approx 0.3\%$ of it) but retained so that the guarantee holds exactly over the solver's integer arithmetic.

A detour leaving R^* in segment k may rejoin it downstream, even in a later segment. This does not affect the per-segment guarantee below, which rests on energy attribution rather than geometric containment: the detour's full priced energy ΔE_v is deducted in segment k via (6), while the bypassed stretch of R^* stays priced through the reference terms. Under Assumption 1, the ledger therefore never overstates the energy at the rejoin point, and the downstream segments inherit the guarantee unchanged.

Lemma 2 (Prefix Feasibility). *Under Assumption 1 (Linear Outer Approximations) and the budget constraint (9), the continuous prefix energy satisfies $E_{\text{prefix}} \geq E_{\min} + \alpha E_k^* \geq E_{\min}$ at any physical point within segment k if the following Segment-Length Condition (SLC) holds:*

$$U_k \leq (1 - \alpha) \tilde{B}_k - \varepsilon_E(k). \quad (10)$$

Proof. In the worst case (a heavily front-loaded detour executed immediately at the interior boundary combined with maximum reference consumption), the continuous energy level drops by the entire maximum interior consumption U_k , the entire allowed detour energy, and the quantization error, so $E_{\text{prefix}} \geq E_{\text{dep},k-1} - \Delta E_k^{\text{det}} - U_k - \varepsilon_E(k)$. Substituting (9) for ΔE_k^{det} and (10) for U_k yields, with $\tilde{B}_k = B_k + E_k^*$,

$$\begin{aligned} E_{\text{prefix}} &\geq E_{\text{dep},k-1} - \alpha B_k \\ &\quad - ((1 - \alpha)(B_k + E_k^*) - \varepsilon_E(k)) - \varepsilon_E(k) \\ &= E_{\text{dep},k-1} - B_k - (1 - \alpha) E_k^*. \end{aligned}$$

Using $B_k = E_{\text{dep},k-1} - E_k^* - E_{\min}$, this equals $E_{\min} + E_k^* - (1 - \alpha) E_k^* = E_{\min} + \alpha E_k^*$. $\square$

Dynamic SLC Enforcement in CP Our model explicitly enforces the required safety buffer dynamically as a structural constraint on the entry energy for each segment. By substituting the definition of the entry surplus $\tilde{B}_k$ into the Segment-Length Condition (10) and rearranging terms, we derive the Minimum Entry Energy constraint:

Corollary 1 (Online SLC Enforcement). *Let Assumption 1 hold and let the budget constraint (9) be enforced. If the CP solver bounds the entry energy by*

$$E_{\text{dep},k-1} \geq E_{\min} + \left\lceil \frac{U_k + \varepsilon_E(k)}{1 - \alpha} \right\rceil, \quad (11)$$

then, together with the invariant $U_k \geq E_k^*$, which the precomputation guarantees by construction ($E_k^+ \geq E_k^*$ and $\gamma_{\text{safe}} \geq 1$, Section 5.1), $B_k > 0$ holds and the Segment-Length Condition (10) is satisfied. Consequently $E_{\text{prefix}} \geq E_{\min} + \alpha E_k^* \geq E_{\min}$ at every point of segment k by Lemma 2, and $E_{\text{arr},k} \geq E_{\min}$ by Lemma 1.

Proof. Rearranging (11) gives $\tilde{B}_k = E_{\text{dep},k-1} - E_{\min} \geq (U_k + \varepsilon_E(k))/(1 - \alpha)$. Multiplying by the positive denominator $(1 - \alpha)$ recovers (10). Moreover, since $\alpha \in (0, 1)$, $\tilde{B}_k \geq U_k + \varepsilon_E(k) \geq E_k^*$, hence $B_k = \tilde{B}_k - E_k^* \geq \varepsilon_E(k) > 0$. $\square$

Theorem 1. Continuous Route Safety

Let Assumptions 1 and 3 hold. Given the adaptive bound definitions (Section 5.1), if the solver satisfies the discrete flow conservation (6)–(7), the budget constraint (9), and the minimum entry-energy condition (11), then the physical State of Charge satisfies $\text{SoC} \geq E_{\min}$ at every continuous point along the trajectory, and the planned energy profile satisfies $E_{\text{arr},k}, E_{\text{dep},k} \in [E_{\min}, E_{\text{max_eff}}]$ at every segment boundary.

Proof. The result follows directly from the preceding lemmas. At the discrete segment boundaries, Lemma 1 and the flow equations (6)–(7) guarantee that the planned profile satisfies $E_{\text{arr},k}, E_{\text{dep},k} \in [E_{\min}, E_{\text{max_eff}}]$. Within the segment interiors, Corollary 1 enforces the Segment-Length Condition, which by Lemma 2 ensures that the physical state of charge satisfies $\text{SoC} \geq E_{\min}$ at all continuous coordinates. Since the boundaries and interiors collectively partition the entire trajectory, $\text{SoC} \geq E_{\min}$ holds everywhere. $\square$

The interior clause of Assumption 1 holds by construction (Section 5.1). Its detour clauses and Assumption 3 are not enforced within the model: $L_{\text{det},k}$, ΔE_v and the realized charging gain are calibrated conservatively rather than certified. All three are assessed in Section 7.4.

6. Implementation

This section details how the framework introduced in Section 4 is realized end-to-end. Section 6.1 describes the service-oriented preprocessing pipeline, including the charging-envelope construction based on the Pool Adjacent Violators Algorithm (PAVA). Section 6.2 then specifies how Stage 2.2 assembles the candidate graph $\mathcal{G}$ from the filtered candidates. Finally, Section 6.3 formulates Stage 3, the CP model solved with Google OR-Tools CP-SAT (version 9.15.6755) (Perron and Didier, 2024). The model combines fixed-point integer decision variables and linear inequality constraints with native CP global constraints, in particular linear flow-conservation constraints on the candidate graph for routing, AddMaxEquality for parallel dwell synchronization, and OnlyEnforceIf for conditional charging activation. An equivalent MILP encoding (e.g., via the open-source HIGHS solver (Huangfu and Hall, 2018)) would

require Big-M reformulations of these structural constraints together with subtour-elimination formulations such as MTZ or DFJ.

CP-SAT was chosen as the underlying solver because the requirement templates instantiated by the dialog layer map onto native constructs of the same model: time windows and precedence onto linear temporal constraints, coverage onto the cluster disjunctions (Eq. 22), appointments onto restricted candidate sets, and charging preferences onto objective terms. Activating or re-parameterizing a template across turns therefore leaves the search procedure unchanged. Specialized labeling algorithms remain faster on the simpler resource-constrained shortest-path formulation (Section 7.5.1). Their per-template extension cost is what the declarative formulation avoids.

6.1. Preprocessing Pipeline

The framework operates on top of a service-oriented preprocessing pipeline (cf. Algorithm 1). Once a reference route R^* is requested, the system fetches per-link weather data, then concurrently performs three vehicle-specific computations: (i) powertrain-based per-segment energy estimates E_k^* , (ii) the corridor candidate search along R^* , and (iii) piecewise-linear charging envelopes $\{\Phi_c\}$.

Charging envelope preprocessing

Vehicle-specific charging curves are obtained from the powertrain service across discrete buckets of ambient temperature and prior driving time, the latter capturing the battery preconditioning state.

Each raw profile is then processed via the Pool Adjacent Violators Algorithm (PAVA) (Ayer et al., 1955; Barlow, 1972), which enforces monotonicity in charging power by iteratively pooling adjacent SoC buckets that violate the monotonicity constraint into their time-preserving harmonic mean. Because the monotonized power curve is non-increasing, the corresponding minimum-charging-time function is convex and piecewise linear, and its segments directly provide the supporting lines $\{(r_j, c_j)\}_{j=1}^J$ used in the CP model. Equivalently, the monotonization replaces the bucketed minimum-charging-time function by its greatest convex minorant.

Figure 4 illustrates the construction at 10 °C from a cold start: the power of the still-cool pack rises mildly with SoC and PAVA pools the affected buckets. On a warm pack the envelope reproduces the powertrain-service curve exactly. On colder batteries the warm-up region widens and the convex model class becomes optimistic for short charging sessions. We outline this regime, and a two-piece piecewise-linear solver extension that removes it, in Appendix A.

This representation captures the non-linear charging dynamics without invoking the powertrain model during search. Its application within the CP model is described in Section 6.5.

Caching and conversational re-optimization

The CP framework treats these artifacts as fixed inputs, thereby rendering conversational re-optimization computationally inexpensive. When the user modifies only preference parameters or weights, without altering the underlying route or the set of requested activities, the graph structure remains unchanged, and re-optimization reduces to a single CP solve on the pre-existing model instance.

When the user introduces an additional activity, the route-dependent artifacts, the weather-conditioned energy consumption estimates, the reference segmentation, the previously constructed graph, and the vehicle-specific charging envelopes all remain valid. In this scenario, only the candidate locations for the newly requested activity type must be retrieved, evaluated (e.g., with respect to detour costs and time-window feasibility), and integrated into the graph, after which the model is re-solved. In neither case is the computationally expensive physical precomputation phase re-executed.

6.2. Stages 1, 2.1, and 2.2: Segmentation, Filtering, and Graph Construction

Stage 1 (energy-based segmentation) and Stage 2.1 (candidate filtering) are implemented as linear scans following the specifications in Sections 4.1 and 4.2 respectively, with Algorithm 1 detailing the filtering pipeline. The remainder of this subsection focuses on Stage 2.2, which assembles the surviving candidates into the candidate graph $\mathcal{G}$ that serves as input to the CP-SAT model in Section 6.3.

6.2.1. Graph Formalization

The candidate graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed graph with a problem-specific structure designed to control the dominant scaling bottleneck of CP-SAT on EV-IPA: the size of the arc set, which determines the number of Boolean and precedence variables in the model. Mandatory waypoint nodes $b_1, \dots, b_{K-1}$ inserted at the segment boundaries of R^* restrict every routing arc to a single segment span, reducing the arc count from $\mathcal{O}(|\mathcal{V}|^2)$ for a fully-connected colocation graph to $\mathcal{O}(K \cdot \bar{n})$, where $\bar{n}$ is the average number of content nodes per segment. With flow conservation on the arcs (Section 6.4), Stage 3 then solves a constrained shortest path from v_O to v_D subject to cluster coverage, energy feasibility, and time windows, rather than a closed-tour TSP or GTSP. The GTSP-style “exactly one node per cluster” constraint nevertheless is kept as a side condition on activity selection.

Nodes. Each candidate of $\mathcal{C}$ contributes one or more nodes. Each $v \in \mathcal{V}$ is a tuple

$$v = (\lambda_v, k(v), [\underline{d}_v, \bar{d}_v], [e_v, l_v], a(v), \Delta E_v, \tau_v),$$

denoting a service at physical location λ_v , segment index $k(v)$, service-time bounds $[\underline{d}_v, \bar{d}_v]$, time window $[e_v, l_v]$, the activity type $a(v) \in \mathcal{A}_{\text{req}} \cup \{\text{charge}\}$ that the node fulfils, and off-route detour cost $(\Delta E_v, \tau_v)$ in energy and time. We further distinguish $\mathcal{V}_{\text{charge}} = \{v \in \mathcal{V} : a(v) = \text{charge}\}$

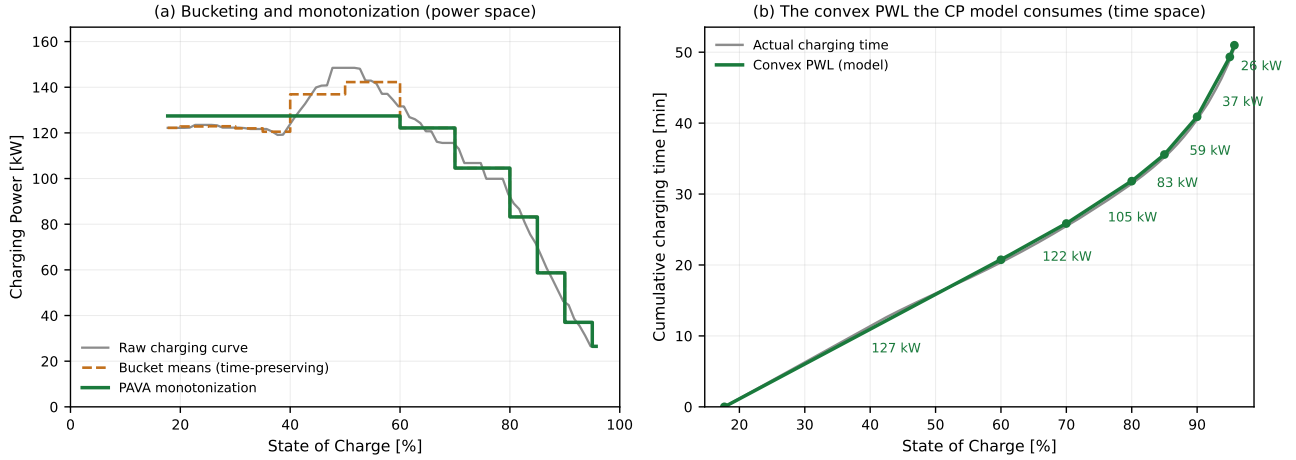


Figure 4: Charging-envelope construction for a DC 400 kW station at 10 °C, cold start: (a) power space, (b) charging-time space.

as nodes offering charging. For amenity nodes the service-time bounds are inherited from the requested activity type, $[\underline{d}_v, \bar{d}_v] = [\underline{d}_{a(v)}, \bar{d}_{a(v)}]$, for charging nodes they equal the operational charging-duration limits (Section 7.2).

Colocation sets. Real-world stops often combine multiple services within walking distance of each other, a charging station next to a restaurant, or a service area with a café and a supermarket. To exploit this, candidates whose physical locations lie within a walking radius ρ_{walk} (e.g. 400 m) are grouped into a colocation set $Q \in \mathcal{Q}$:

$$Q = \{v \in \mathcal{V} : \text{dist}(\lambda_v, \lambda_Q) \leq \rho_{\text{walk}}\},$$

where λ_Q is the geographic anchor of the set. The sets Q partition $\mathcal{V}$ at the node level: each node belongs to exactly one set. Standalone services not within walking distance of any other candidate form singleton sets. A colocation set comprises at least one node and includes no more than one charging node, as well as no more than one amenity node for each requested activity type. Consequently, a single set may associate a charging facility with multiple distinct amenities (e.g., a café and a supermarket within the same service area).

Within a set, nodes are jointly executed: when the stop is visited, all nodes in the set are activated, with the vehicle charging while the driver performs the amenity activity. The dwell time at a colocation set Q is defined as the maximum of the charger duration and the sum of amenity durations:

$$D_Q = \max \left(d_{v_Q^{\text{chg}}}, \sum_{v \in Q \setminus \{v_Q^{\text{chg}}\}} d_v \right),$$

where $v_Q^{\text{chg}} \in Q \cap \mathcal{V}_{\text{charge}}$ denotes the charging node within Q (if present) and $d_v \in [\underline{d}_v, \bar{d}_v]$ is the realized service time at node v . Charging thus runs in parallel to the sequence of amenity activities, capturing the realistic behavior of a driver completing multiple errands during a single charging stop. The corresponding synchronization and routing constraints

are detailed in Section 6.3. Figure 5 illustrates this for a charger co-located with a restaurant.

To preserve feasibility when a co-located amenity is unavailable (e.g., the restaurant is closed when the vehicle arrives), each charger participating in a multi-node colocation set is additionally instantiated as a distinct node forming its own standalone singleton set. Both instances denote the same physical charger: one as a serving term of the activity's coverage disjunction (Eq. 22), one as an optional charging node that remains usable when the amenity is closed. No constraint links them, and none is needed: with the state of charge tracked continuously, a second visit adds no energy but another stop cost, setup time and detour penalty, and is thus dominated. None of our experimental solutions visits both.

Clusters. Beyond geographic grouping, nodes are organized into clusters by user intent. Each required activity type $r \in \mathcal{A}_{\text{req}}$ defines a cluster $C_r = \{v \in \mathcal{V} : a(v) = r\}$ collecting all candidates that fulfill r . The clusters are pairwise disjoint and mirror the cluster structure of the Generalized Traveling Salesman Problem (GTSP): a feasible itinerary visits exactly one node from each cluster C_r . If no candidate fulfilling r remains in the graph after Stage 2.1 filtering (e.g., due to restrictive opening hours), the corresponding cluster is dropped from the requirement set, and the activity type is omitted from the itinerary rather than rendering the problem infeasible.

Segment-boundary nodes. To exploit the geometric ordering imposed by the corridor R^* , we augment $\mathcal{V}$ with $K - 1$ mandatory *boundary nodes* $b_1, \dots, b_{K-1}$, where b_k marks the end of segment k . These are dummy nodes: they carry no service, no detour, and no cost, and exist only to structure the routing arcs. Adopting the convention $b_0 \equiv v_O$ and $b_K \equiv v_D$, the boundaries partition the routing arcs by segment, one arc group per segment of R^* . Each

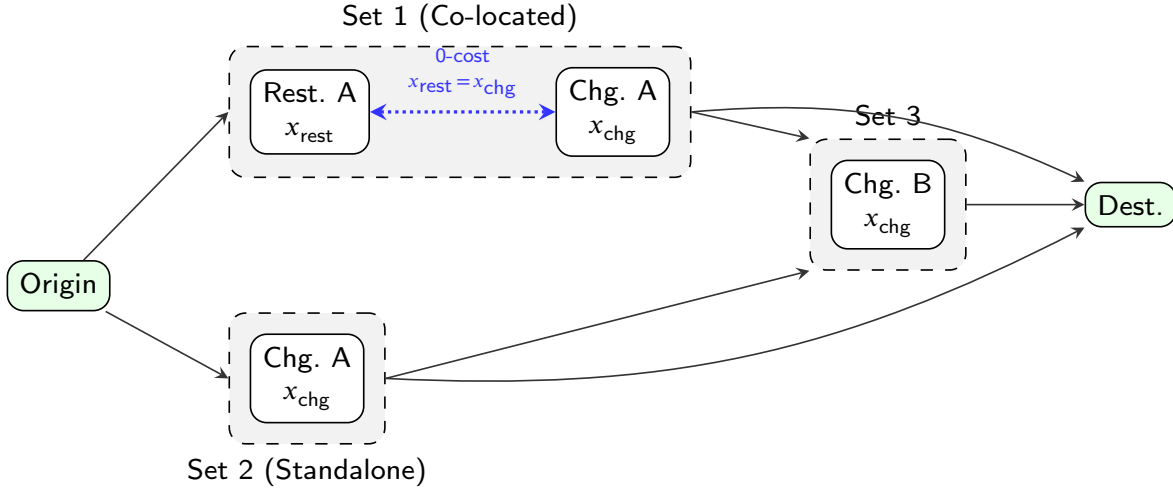


Figure 5: Graph structure. Co-located activities (Set 1) are linked by zero-cost edges and synchronized via a shared visit boolean to model concurrent dwell time. The standalone counterpart (Set 2) preserves charging flexibility if the restaurant is closed.

boundary node $b_1, \dots, b_{K-1}$ is forced to be visited unconditionally via a singleton mandatory cluster. Together with the source/sink semantics of v_O and v_D under flow conservation (Section 6.4), the boundaries ensure that every itinerary traverses each segment exactly once. Figure 6 illustrates the resulting structure and the per-edge cost composition for $K=3$ segments. The figure depicts the full edge set, of which any single itinerary selects a mutually exclusive subset.

Edges. The edge set $\mathcal{E}$ is organized by segment. For each segment k , $\mathcal{E}$ contains a directed *transit edge* $b_{k-1} \rightarrow b_k$ traversing the segment along R^* without stopping, and a pair of *detour edges* $b_{k-1} \rightarrow v$ and $v \rightarrow b_k$ for every content node v mapped to segment k , encoding a single service visit during the segment. Directed *intra-segment edges* $u \rightarrow v$ between non-co-located content nodes of the same segment, ordered along R^* , permit multiple services within one segment. Bidirectional zero-cost *intra-set edges* link every pair of nodes within the same colocation set, synchronizing their visit booleans and arrival times.

Edge cost computation. Every edge cost is assembled from two components precomputed once during graph construction: the on-route travel time/energy ($\tau_{p,q}^*$, $E_{p,q}^*$) between any two positions $p, q \in [0, |R^*|]$ along the route, where a position is an arc-length coordinate on R^* (boundary nodes and projections $\pi(v)$ are positions, content nodes are not), obtained from a single forward sweep over the route polyline, and the asymmetric off-route detour ($\tau_v^{\text{in}}, \tau_v^{\text{out}}$) of each content node v : the road-network legs from v 's projection $\pi(v)$ to v and back. Concretely, for segment k :

$$\tau_{b_{k-1}, b_k} = \tau_{b_{k-1}, b_k}^* \quad (\text{transit}) \quad (12)$$

$$\tau_{b_{k-1}, v} = \tau_{b_{k-1}, \pi(v)}^* + \tau_v^{\text{in}} \quad (\text{in-detour}) \quad (13)$$

$$\tau_{v, b_k} = \tau_{\pi(v), b_k}^* + \tau_v^{\text{out}} \quad (\text{out-detour}) \quad (14)$$

Energy costs are composed analogously, with $E_{p,q}^*$ in place of $\tau_{p,q}^*$ and the detour legs priced into ΔE_v (Section 5.1). Boundary nodes are on-route ($\pi(b_k) = b_k$, $\tau_{b_k} = 0$), so transit edges contain no detour term.

Cost composition across segments. Because the boundaries are mandatory, every feasible itinerary traverses the full corridor and pays each on-route slice τ_{b_{k-1}, b_k}^* exactly once: visiting a content node v in segment k replaces the transit edge with the in-detour and out-detour pair, which together still cover the same on-route span (since $\tau_{b_{k-1}, \pi(v)}^* + \tau_{\pi(v), b_k}^* = \tau_{b_{k-1}, b_k}^*$) but additionally charge $\tau_v^{\text{in}} + \tau_v^{\text{out}}$. The total time of an itinerary visiting content nodes $v_1, \dots, v_m$ across distinct segments therefore reduces to

$$\tau_{v_O, v_D}^* + \sum_{i=1}^m (\tau_{v_i}^{\text{in}} + \tau_{v_i}^{\text{out}}),$$

i.e. a fixed corridor traversal plus the sum of round-trip detours of the visited candidates.

Two content nodes in non-adjacent segments, say u in segment k_u and v in segment $k_v > k_u$, are not directly connected. The solver evaluates their connection along the path $u \rightarrow b_{k_u} \rightarrow b_{k_u+1} \rightarrow \dots \rightarrow b_{k_v-1} \rightarrow v$. The on-route slices are additive along R^* :

$$\tau_{\pi(u), b_{k_u}}^* + \sum_{j=k_u}^{k_v-2} \tau_{b_j, b_{j+1}}^* + \tau_{b_{k_v-1}, \pi(v)}^* = \tau_{\pi(u), \pi(v)}^*, \quad (15)$$

so the path cost collapses to $\tau_u^{\text{out}} + \tau_{\pi(u), \pi(v)}^* + \tau_v^{\text{in}}$, recovering the same A-to- R^* -to-B decomposition that an all-pairs colocation graph would use, without enumerating quadratic edges.

6.3. CP-SAT Formulation

The Stage 3 model is formulated over the candidate graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ defined in Section 6.2.1. Continuous

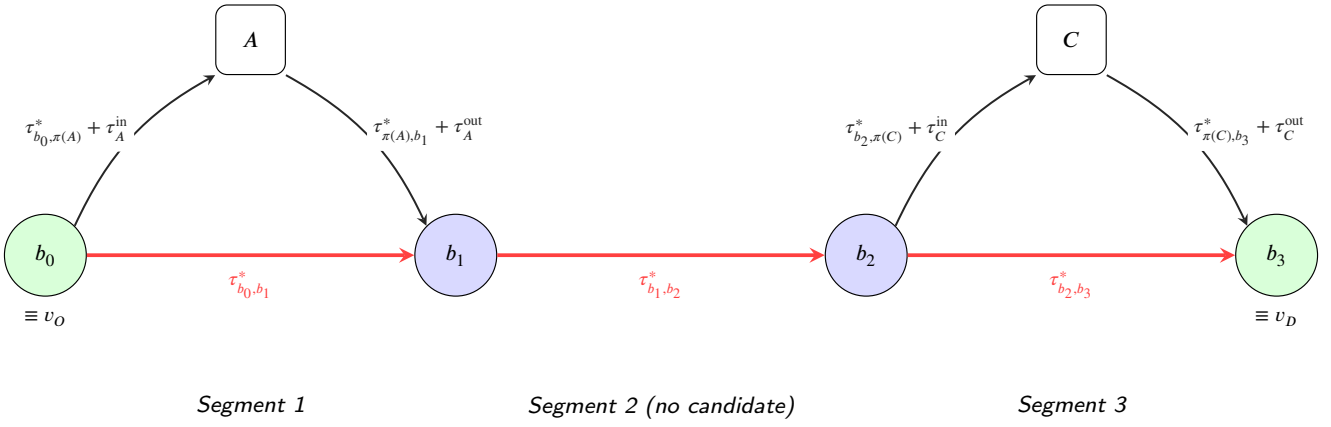


Figure 6: Edge construction in the candidate graph for $K=3$ segments, with one content node in each non-empty segment (A in segment 1, C in segment 3; segment 2 has no candidate). Red arcs are the on-route transit edges between the mandatory boundaries $b_0 \equiv v_O, \dots, b_3 \equiv v_D$; black arcs are the detour pairs $b_{k-1} \rightarrow v \rightarrow b_k$ through each content node. Within a segment the transit edge and its detour pair are mutually exclusive under flow conservation, and the A -to- C connection runs via b_1, b_2 rather than a direct edge. See Section 6.4 for the cost composition and the telescoping identity (Eq. 15).

Table 1

Primary integer decision variables of the CP-SAT model.

Symbol	Domain	Meaning
x_v, a_{uv}	$\{0, 1\}$	Node visited / edge used
u_v	$\{0, 1\}$	Charging explicitly utilized at node v ($u_v \leq x_v$)
t_v, dur_v	$\mathbb{N}_0$	Arrival time / active duration at v
$D_Q, wait_{uv}$	$\mathbb{N}_0$	Dwell time at Q / slack wait on (u, v)
β_v	$\mathbb{N}_0$	Charging blocks (size Δ_B) at v
$E_{arr,k}, E_{dep,k}$	$\mathbb{N}_0$	Arrival / departure energy for segment k
$G_k, \Delta E_k^{\det}$	$\mathbb{N}_0$	Charging gain / detour loss for segment k
$y_{pre,v}, y_{post,v}, g_v$	$\mathbb{N}_0$	SoC before/after the charging event at v ; gain $g_v = y_{post,v} - y_{pre,v}$

physical values (energy, time) are scaled to fixed-point integer arithmetic. Table 1 lists the primary integer decision variables.

6.4. Routing, Structural, and Temporal Constraints

The constraints below operate over the graph notation $(\mathcal{V}, \mathcal{Q}, \mathcal{C}, \mathcal{E})$ and the membership function $Q(v) \in \mathcal{Q}$ from Section 6.2.1.

Routing is enforced via flow conservation on the directed arcs a_{uv} . The origin emits one unit of flow, the visited destination absorbs it, and every visited intermediate node $v \in \mathcal{V}_{\text{int}} := \mathcal{V} \setminus (\{v_O\} \cup \mathcal{C}_D)$ has equal in- and out-flow:

$$\sum_{v \in \mathcal{V}} a_{v_O, v} = 1, \quad \sum_{u \in \mathcal{V}} a_{u, v_O} = 0, \quad (\text{source}) \quad (16)$$

$$\sum_{u \in \mathcal{V}} a_{u, v} = x_v, \quad \sum_{w \in \mathcal{V}} a_{v, w} = x_v, \quad \forall v \in \mathcal{V}_{\text{int}}, \quad (17)$$

$$\sum_{u \in \mathcal{V}} a_{u, v} = x_v, \quad \sum_{w \in \mathcal{V}} a_{v, w} = 0, \quad \forall v \in \mathcal{C}_D. \quad (18)$$

Together with the cluster-coverage constraint (21), $\sum_{v \in \mathcal{C}_D} x_v = 1$, this selects a unique source-to-sink path. Cycles are excluded on both levels: intra-set arcs receive the acyclic

orientation induced by a fixed total order $<$,

$$a_{vu} = 0 \quad \forall u < v \in \mathcal{Q}; \quad a_{uv} = x_u \quad \text{for } \mathcal{Q} = \{u, v\}, \quad (19)$$

so a visited set is internally acyclic and can only be entered over an inter-set arc, and inter-set arcs carry strictly positive travel time, which no cycle can accommodate under monotone t . Colocation sets are integrated into the routing without sequential travel accumulation by sharing visit decisions and arrival times across all nodes within a set:

$$x_u = x_v, \quad t_u = t_v \quad \forall u, v \in \mathcal{Q}, \quad \mathcal{Q} \in \mathcal{Q}. \quad (20)$$

Cluster-level coverage enforces that for each required activity type $r \in \mathcal{A}_{\text{req}}$, exactly one node fulfilling r is visited:

$$\sum_{v \in \mathcal{C}_r} x_v = 1 \quad \forall r \in \mathcal{A}_{\text{req}}. \quad (21)$$

Equation 21 generalizes to a disjunction when a required activity type admits alternative realizations: a standalone cluster and one or more merged colocation sets can each satisfy the same activity r . The model selects exactly one such alternative,

$$\sum_{Q \in \mathcal{Q}(r)} x_Q = 1 \quad \forall r \in \mathcal{A}_{\text{req}}, \quad (22)$$

where $Q(r)$ collects the colocation sets (standalone or merged) that fulfill r and x_Q is the shared visit variable of set Q .

Standalone charging nodes are optional, activated solely for energy feasibility. To decouple physical presence at a co-located location from charging execution, the charge-utilization boolean is conditionally bounded by the visit boolean: $u_v \leq x_v$ for nodes co-located with non-charging activities, tightening to $u_v = x_v$ for pure standalone chargers where a visit without charging would be meaningless.

The dwell time D_Q defined in Section 6.2.1 is realized in the model via CP-SAT's AddMaxEquality global constraint. Temporal precedence restricts travel time and slack waits to inter-set edges, with operational time-window bounds:

$$\begin{aligned} a_{uv} = 1 \wedge Q(u) \neq Q(v) &\implies t_v = t_u + D_{Q(u)} \\ &\quad + \tau_{uv} + wait_{uv}, \quad (23) \\ x_v = 1 &\implies e_v \leq t_v \leq l_v - dur_v, \\ \underline{d}_v &\leq dur_v \leq \bar{d}_v. \quad (24) \end{aligned}$$

For charging nodes, $dur_v = \beta_v \cdot \Delta_B$, where Δ_B is the charging block size (120 s in our implementation). If charging is not utilized ($u_v = 0$), then $\beta_v = g_v = 0$.

6.5. Energy Integration and Non-linear Charging

The discrete energy flow established in Section 5.3 (Eqs. 6–7) is realized in the CP model via per-segment integer variables. Let $Q_k = \{Q \in \mathcal{Q} : k(v) = k \ \forall v \in Q\}$ denote the colocation sets mapped to segment k , and $\Delta E_Q = \max_{v \in Q} \Delta E_v$ the worst-case detour penalty per set. We write $x_Q := x_v$ for the shared visit variable of set Q , which is well-defined for any $v \in Q$ by Eq. 20. Aggregate detour consumption is computed once per visited set:

$$\Delta E_k^{\det} = \sum_{Q \in Q_k} \Delta E_Q \cdot x_Q, \quad G_k = \sum_{v: k(v)=k} g_v. \quad (25)$$

The minimum entry-energy condition (Corollary 1) is enforced as a linear inequality on $E_{\text{dep},k-1}$.

Non-linear charging via the precomputed envelope. Battery tapering is modelled by exposing the convex charging envelope $\{(r_j, c_j)\}_{j=1}^J$ from Section 6.1 as linear constraints. For each active charging node v with $u_v = 1$, the state-of-charge variable $y_{\text{pre},v}$ is coupled to the segment entry energy by proportionally deducting the precomputed on-route reference consumption up to the candidate's projection on R^* . This deterministic coupling prevents the solver from artificially lowering entry energy to exploit steeper charging rates.

The envelope's cumulative charging time up to $y_{\text{pre},v}$, a reference point on the curve rather than time actually spent, is computed via AddMaxEquality:

$$T_{\text{pre},v} = \max_{j=1,\dots,J} (r_j \cdot y_{\text{pre},v} - c_j). \quad (26)$$

The physical charging duration for gain $g_v = y_{\text{post},v} - y_{\text{pre},v}$ is then fixed on the departure side by the same construction:

$$dur_v = \max_{j=1,\dots,J} (r_j \cdot y_{\text{post},v} - c_j) - T_{\text{pre},v} + \tau_{\text{setup}}, \quad (27)$$

where τ_{setup} accounts for fixed connector handling. The charging duration is thus fully determined by the entry and exit state of charge. For unvisited nodes it is zero.

6.6. Multi-Objective Scalarization

The competing planning objectives (total trip time, number of stops, candidate quality, charging-strategy preferences, and soft time-window adherence) are aggregated into a single integer objective Z , measured in equivalent travel-seconds ($c_{\text{drive}} = 1$), via weighted-sum scalarization. Let $N_{\text{stops}} = \sum_{Q \in \mathcal{Q}} \mathbf{1}_{[Q \text{ visited}]}$ denote the number of physical stops. The objective is

$$\begin{aligned} Z = & c_{\text{drive}} \cdot T_{\text{travel}} + (c_{\text{charge}} - c_{\text{drive}}) \cdot T_{\text{chg}}^{\text{xs}} \\ & + c_{\text{stop}} \cdot N_{\text{stops}} + \sum_{v \in \mathcal{V}} x_v \cdot P_{\text{rating}}(v) + P_{\text{SoC}} + P_{\text{time}}, \end{aligned} \quad (28)$$

where T_{travel} is the total elapsed time (driving, dwell, and slack waits) and $T_{\text{chg}}^{\text{xs}} = \sum_{Q \in \mathcal{Q}} (D_Q - \sum_{v \in Q \setminus \{v_Q^{\text{chg}}\}} dur_v)$, with $dur_v = 0$ at unvisited nodes, the charging time not overlapped by an activity: a charging second that prolongs a stop costs c_{charge} rather than c_{drive} , whereas charging that finishes within an activity adds nothing beyond the activity's dwell. c_{stop} is a fixed price per physical stop that biases the solver toward fewer stops. $P_{\text{rating}}(v) = \mu_{\text{rating}}(5 - \sigma_v)$ prices the rating shortfall of amenity nodes (zero for all other nodes). P_{SoC} is a piecewise-linear penalty on the shortfall below the destination target E_{target} and the preferred SoC E_{pref} at intermediate charging stops, with tier rates $\mu_1 < \mu_2 < \mu_3$ escalating with the depth of the shortfall, so that a deep shortfall outweighs the price of an additional stop. P_{time} prices deviations from soft time windows linearly, at μ_{late} per second of lateness and μ_{early} per second of earliness. Table 3 lists the values.

7. Evaluation

This section evaluates the four research questions posed in Section 1. We assess **RQ1** (latency) through the end-to-end computational performance on both the semi-synthetic benchmark and production data (Section 7.3) and through the per-turn cost of conversational re-optimization (Section 7.8), **RQ2** (solution quality) through the comparison against the sequential greedy heuristic (Section 7.5.2), the MILP reference solved with HiGHS (Section 7.5.3), and the colocation ablation (Section 7.6.1), **RQ3** (approximation cost) through the FRVCP optimality gap, measured both with and without the conservative safety margins so as to separate the price of the SLC reserve from that of the formulation itself (Section 7.5.1), the candidate-filtering ablation (Section 7.6.3), and the sensitivity analysis (Section 7.7), and **RQ4** (physical fidelity) through the validation against the production-grade powertrain model (Section 7.4) and the SLC ablation (Section 7.6.2).

Table 2

Composition of the EV-IPA-1000 benchmark.

Route length	Activities per instance						Total
	0	1	2	3	4	5	
200–400 km	34	45	36	52	41	42	250
400–600 km	39	41	39	44	45	42	250
600–800 km	49	40	30	44	43	44	250
800–1,000 km	41	44	35	40	44	46	250
Total	163	170	140	180	173	174	1 000

Corridor candidates	Mean	Median	Min	Max
Charging stations	194.2	154.5	1	1 086
POIs	1531.1	1 287	86	5 755

7.1. Benchmark

To enable reproducible evaluation, we release **EV-IPA-1000** (Wins, 2026), a semi-synthetic benchmark of 1,000 instances on the European road network with route lengths of 200–1,000 km. The range follows the routing requests issued to the in-car navigation of a European EV fleet: since short routine trips are mostly driven without navigation, only about 15 % of requests are below 100 km, about 60 % below 500 km, and trips beyond 1,000 km are rare. The 200–1,000 km band is thus where jointly planning charging and intermediate activities matters most.

7.1.1. Instances.

Each scenario instance integrates empirically observed infrastructure with synthetically generated attributes. Route computation is performed using the Valhalla routing engine¹ on top of OpenStreetMap data². From each resulting route, charging stations and points of interest (POIs) located within a 1 km corridor are identified and extracted. Elevation profiles along the route, as well as elevation at charger locations, are obtained from Valhalla's elevation service based on the AWS terrain tiles³. Europe terrain data produced using Copernicus data and information funded by the European Union - EU-DEM layers. Ratings, opening hours and untagged charging powers are synthetic. Table 2 summarises the composition: origin–destination pairs are stratified into four route-length bins of 250 instances each. Independently of length, the number of requested activities is uniform on $\{0, \dots, 5\}$, drawn from five POI categories with durations uniform in $[15, 90]$ min (the 163 instances without activities reduce to pure charging problems). Charging powers are taken from OpenStreetMap where tagged and otherwise drawn from $\{22, 50, 150, 300\}$ kW.

7.1.2. Charging surrogate.

Instead of proprietary charging profiles, a charging session is evaluated in four steps: the pack's warmth on arrival, the state-of-charge (SoC)-dependent taper, the delivered power, and the resulting charging duration. The result

is a piecewise-linear charging function: the surrogate adopts the abstraction introduced to the E-VRP by Montoya et al. (2017) and used in the formulations building on it (Froger et al., 2019). Each connector carries a rated power P_{stat} and a vehicle its battery capacity C_{batt} (Section 7.2). The remaining parameters below are identical across vehicles and deliberately coarse. Delivered power is capped by the station rating alone, the surrogate models no connector current, pack voltage or vehicle-side acceptance limit.

(i) A pack warms under load, and low cell temperature reduces its power (Keyser et al., 2017). The surrogate tracks this with a warm-up proxy driven by driving time rather than a temperature. With t (s) the driving time accumulated since the start of the route, reset only at a stop where the vehicle does not charge, the pack's warmth on arrival is

$$w = \min \left\{ 1, \max \left\{ 0, \frac{t - t_{\text{cold}}}{t_{\text{warm}} - t_{\text{cold}}} \right\} \right\}, \quad (29)$$

with $t_{\text{cold}} = 300$ s and $t_{\text{warm}} = 3600$ s, so the pack is fully cold ($w = 0$) at stops reached within 5 minutes of departure or of the previous activity (not charging) stop, fully warm ($w = 1$) once at least 60 minutes have been driven, and interpolates linearly in between. Dwell time at an activity stop neither counts towards t nor resets it, so cold derating affects only stops reached early in a driving leg.

(ii) SoC is split into five bands of 20 %. A warm pack follows $\tau^{\text{warm}} = [1.0, 1.0, 0.8, 0.5, 0.3]$ over the bands and a cold pack $\tau^{\text{cold}} = [0.5, 0.5, 0.4, 0.3, 0.2]$, with intermediate warmth interpolating per band b :

$$\tau_b = \tau_b^{\text{cold}} + w(\tau_b^{\text{warm}} - \tau_b^{\text{cold}}), \quad (30)$$

so charge acceptance scales down as the pack cools. Both vectors are non-increasing, so the charging function is concave at every warmth level, matching the form of Montoya et al. (2017). The taper falls in the two directions in which charge acceptance is known to fall: towards high SoC, where cell voltage approaches its limit, and at low pack temperature (Keyser et al., 2017).

(iii) Charge acceptance scales the station's rated power, $P_b = \tau_b P_{\text{stat}}$ in band b with no vehicle- or connector-side cap, and constant power per band makes charging time linear in SoC within a band:

$$t_{\text{ch}} = \sum_{b=1}^5 \frac{\Delta s_b C_{\text{batt}}}{\tau_b P_{\text{stat}}}, \quad (31)$$

where Δs_b is the fraction of pack capacity charged within band b and $\sum_b \Delta s_b$ the session's total SoC gain.

The EV-IPA-1000 repository (Wins, 2026) releases the instances, per-file schemas, the generation methodology and the licence attributions of the underlying sources. Evaluating a plan requires the instance files, the vehicle parameters of Section 7.2, and Eqs. (29)–(31).

7.2. Experimental Setup

All experiments are executed on an Apple M3 Max with 64 GB of memory, using Python and Google OR-Tools CP-SAT (v9.15.6755). We simulate a mid-size battery-electric

¹<https://github.com/valhalla/valhalla>

²<https://www.openstreetmap.org/copyright>

³<https://registry.opendata.aws/terrain-tiles/>

SUV with 400 V powertrain architecture and $C_{\text{batt}} = 104.1$ kWh. The minimum safety buffer is set to $E_{\text{min}} = 3\%$ state of charge (SoC); the target arrival energy at the destination is $E_{\text{target}} = 50\%$ of effective capacity, and the preferred minimum SoC upon arrival at intermediate charging stops is configured at $E_{\text{pref}} = 10\%$. Initial SoC is sampled uniformly at random from $[20\%, 100\%]$ across instances.

Table 3 lists the configuration used in all experiments, grouped by the component it controls. The objective and filter weights are given at the granularity that governs the reported trade-offs. The implementation additionally carries second-order refinements (small tie-breaking terms) that are dominated by the listed terms and do not qualitatively alter the comparisons in this section. All experiments below (unless specified otherwise) use the production-grade data cohort (Section 7.3.2): charger inventories, opening hours and ratings, real vehicle- and weather-specific charging curves, and the production-grade powertrain model. Since these are live services, reported latencies carry minor run-to-run variation, all routes, in planning and validation alike, are requested under no-traffic conditions. All production-grade queries were executed in mid-August 2026, meaning the weather-conditioned energy and charging evaluations reflect a warm-weather regime. By contrast, the synthetic EV-IPA-1000 benchmark does not model weather, its surrogate charging curves alternate solely based on prior driving time rather than ambient temperature (Section 7.1).

With $\gamma_{\text{detour}} = 1.0$, detour and on-route consumption are bounded equally conservatively. Section 7.7 validates the chosen values.

7.3. Computational Performance and Scalability

We evaluate the framework on both cohorts described in Section 7.2: the semi-synthetic EV-IPA-1000 benchmark (Section 7.3.1) and the production-grade data cohort over the same 1,000 O-D pairs and energy contexts (Section 7.3.2), which additionally differs in the routing provider and hence in the resulting routes and candidate sets. Tables 4 and 5 report the resulting statistics.

7.3.1. Evaluation on Semi-Synthetic Data

On the semi-synthetic benchmark, filtering reduces the candidate set from **877.8** raw candidates to **110.6** graph nodes and **417.8** edges per instance, a reduction of **87.4%** in aggregate and **76.1%** on average per instance ($p_{50} = 82.7\%$). CP-SAT explores a mean of **529.9** branches (median **163**).

The median end-to-end latency is **1.21 s**, with $p_{90} = 3.14$ s and a maximum of **7.09 s**, inside the 10 s interactive budget. Stage 3 completes within the 5 s CP-SAT limit on **987** of the 999 solved instances (**98.8%**), the remaining twelve reach the cap.

Exactly one of the 1,000 instances fails to admit a feasible plan, and this failure is attributable not to a modelling artifact but to an inconsistency in the routing engine. Specifically, for this origin–destination pair, the corridor query returns a path traversing the Kaliningrad exclave, whereas the stop-to-stop queries along the same corridor yield routes

Table 3

Default configuration. Objective terms in equivalent driving-seconds ($c_{\text{drive}} = 1$). SoC-shortfall tiers in seconds per percentage point (pp).

Symbol	Value	Description
<i>Route segmentation</i> (Sec. 4.1)		
ρ_{seg}	0.05	Segment energy budget, share of $E_{\text{max_eff}}$ ($E_{\text{seg}} = \rho_{\text{seg}} E_{\text{max_eff}}$)
<i>Candidate filtering</i> (Sec. 4.2)		
B_{chg}	1	Charging candidates per energy segment
B_{poi}	10	Global POI candidates
ρ_{walk}	400 m	Walking radius for colocation sets
<i>Discretization</i>		
Ω_E	10^4	Fixed-point scaling factor, energy (J)
Δ_B	120 s	Charging block, clamped to 10–90 min
<i>Adaptive energy bounds</i> (Sec. 5.1)		
q	90	Consumption percentile
γ_{safe}	1.08	Safety multiplier
γ_{detour}	1.0	Localized detour multiplier
α	0.1	Detour-budget surplus fraction (Cor. 1)
<i>Objective scalarization</i> (Sec. 6.6)		
c_{drive}	1	Travel-second (normalization)
c_{charge}	1.5	Charging-second weight
c_{stop}	1800 s	Penalty per physical stop
μ_{rating}	600 s	Per star below maximum rating
μ_1, μ_2, μ_3	20/100/300 s	SoC-shortfall tiers, per pp (Sec. 6.6)
$\mu_{\text{late}}, \mu_{\text{early}}$	5/0.5	Soft time-window deviation, per s
<i>Filter utilities</i> (Eqs. (1)–(2))		
w_{pow}	0.5	Charging-power weight
w_{det}	0.07 / 0.05	Detour weight (charger / POI)
w_{rat}	0.1	Rating weight

that bypass it. The remaining 999 instances are solved with all requested activities scheduled.

7.3.2. Evaluation using Production-Grade Data

On production-grade data, filtering reduces the candidate set from **1,502.8** raw candidates to **120.0** graph nodes and **467.8** edges per instance, a reduction of **92.0%** in aggregate and **87.5%** on average per instance ($p_{50} = 90.1\%$). The post-filter graphs are almost identical in size to those of the semi-synthetic benchmark (110.6 nodes, 417.8 edges; Table 4), although built from a larger raw pool (1,502.8 vs. 877.8 candidates). Search effort, by contrast, differs: CP-SAT explores **5.2×** as many branches on average (**2,765.3** vs. 529.9) and **2.7×** as many at the median (447 vs. 163), with a strongly right-skewed distribution ($p_{90} = 9,465$, maximum 33,872). We attribute this to the charging model. The intentionally coarse surrogate introduced in Section 7.1 is constructed by scaling a single five-band, constant-power SoC curve by the rated power, such that all stations share one common piecewise-linear charging-time function, differing only by a rated-power scale factor. By contrast, the production-grade model employs a library of 48–144 distinct charging profiles that are conditioned on contextual factors. After enforcing monotonicity, each profile is reduced to a

Table 4

Performance with the semi-synthetic data on the 1,000 EV-IPA-1000 instances.

Metric	Mean	Median (p_{50})	p_{90}	Max
<i>Instance</i>				
Distance (km)	601.1	597.9	917.2	1 001.7
Charging stops	2.5	2.0	4.0	7.0
<i>Graph construction</i>				
Raw candidates	877.8	606	2 033	5 590
Nodes	110.6	106	179	238
Edges	417.8	372	859	1 073
Reduction (%)	76.1	82.7	92.2	96.3
<i>CP-SAT model</i>				
Variables	3 075.7	2 974	4 847	7 019
Constraints	5 204.7	5 031	8 166	11 898
Branches	529.9	163	963	10 246
Conflicts	15.5	3	50	550
<i>Latency (s)</i>				
Stage 1+2 (preproc.)	0.64	0.57	1.04	3.07
Stage 3 (CP-SAT)	0.94	0.54	2.27	5.09
Total	1.57	1.21	3.14	7.09

Table 5

Performance with production-grade data on the 1,000 EV-IPA-1000 origin-destination pairs.

Metric	Mean	Median (p_{50})	p_{90}	Max
<i>Instance</i>				
Distance (km)	601.3	600.2	916.3	999.8
Charging stops	2.4	2.0	4.0	6.0
<i>Graph construction</i>				
Raw candidates	1502.8	1166.5	2818.0	12 977
Nodes	120.0	116	194	250
Edges	467.8	415	929	1 337
Reduction (%)	87.5	90.1	94.9	99.6
<i>CP-SAT model</i>				
Variables	4280.5	4 119	6 579	9 177
Constraints	7369.7	7 147	11 286	15 787
Branches	2765.3	447	9 465	33 872
Conflicts	72.2	19	95	3 320
<i>Latency (s)</i>				
Stage 1+2 (preproc.)	0.61	0.56	1.06	1.88
Stage 3 (CP-SAT)	1.62	0.98	4.70	5.43
Total	2.25	1.58	5.72	6.72
<i>Solution quality</i>				
Optimality gap (%)	0.21	0.0	0.0	7.59

similarly compact set of supporting line segments (seven in Figure 4). However, each node preserves its own envelope with non-uniform charging rates. This induced heterogeneity in node-specific charging-time functions substantially increases the complexity of the resulting objective landscape.

The median total latency is **1.58 s**, with $p_{90} = 5.72$ s and a maximum of 6.72 s, well within the 10 s interactive budget. Of the 1,000 instances solved on production-grade data, CP-SAT closes the dual gap to zero on **93.9 %** within the 5 s limit. On the remaining 61 the optimality gap is small (mean 3.5 %, median 3.3 %, $p_{90} = 5.3$ %, max 7.6 %). Figure 7 breaks this latency down by route distance and requested activity count,

showing that solve time scales primarily with distance, or rather with the number of required charging stops.

7.4. Validation against the Powertrain Model

To ensure that our computational simplifications do not degrade physical fidelity, each generated itinerary is validated using a production-grade powertrain simulation. This validation step independently re-assesses the plan: it first recalculates the route between each pair of consecutive stops as it would actually be driven, and then re-evaluates the cost of each resulting leg with the powertrain model. For both the planning and validation stages, routes are requested under no-traffic conditions. Each charging stop is then re-priced with the full charging model, queried with the connector's power, voltage and current limits, the battery temperature and preconditioning state, and the driving time accumulated since the previous stop, which determines how warm the battery is on arrival and hence the shape of the charging curve. Both models read the same per-position weather and temperature that the planning request itself resolves for the route. Crucially, the validation carries its own state: each stop is priced from the state of charge the replay has itself reached, never from the value the plan predicted. A plan counts as energy-feasible if the validated state of charge stays above the minimum on every driving link.

Under this procedure, the validation confirms that our dynamic linear approximations under the released configuration of Table 3, provide a highly robust planning baseline, achieving an energy feasibility rate of **100 %** across all produced plans. The average state-of-charge (SoC) estimation deviation is **1.60 pp** at the destination (median 1.35, $p_{90} = 2.86$, max 10.54), and the average charge-gain deviation is **0.81 pp** of SoC per charging stop (median 0.70, $p_{90} = 1.46$, max 4.82). Figures 8a and 8b show the full distribution of both deviations, each concentrated near zero.

The same procedure applied to the semi-synthetic cohort, with the surrogate charging model of Section 7.1 in place of the production charging model, yields a 100 % energy feasibility rate.

Under a stress configuration that removes the soft SoC margins of the default settings (Section 7.6.2), the energy feasibility rate remains at **99.9 %**.

7.5. Baseline Comparison

To evaluate solution quality, we benchmark our framework against three baselines. The first is an exact solver for the Fixed-Route Vehicle Charging Problem (FRVCP) (Froger et al., 2019; Montoya et al., 2017), the charging-only special case of EV-IPA in which the route is fixed, no activities are requested, and only the charging decisions, i.e., which stations to insert along the route and how much to charge at each, are optimized. This baseline provides an upper bound on how closely our pipeline approaches the provably optimal charging strategy. The second baseline is a sequential greedy heuristic, also implemented using *frvcpy*, that addresses the full integrated problem. This heuristic enables us to quantify the benefit of global search and jointly optimized planning of activities and charging, relative to a sequential approach

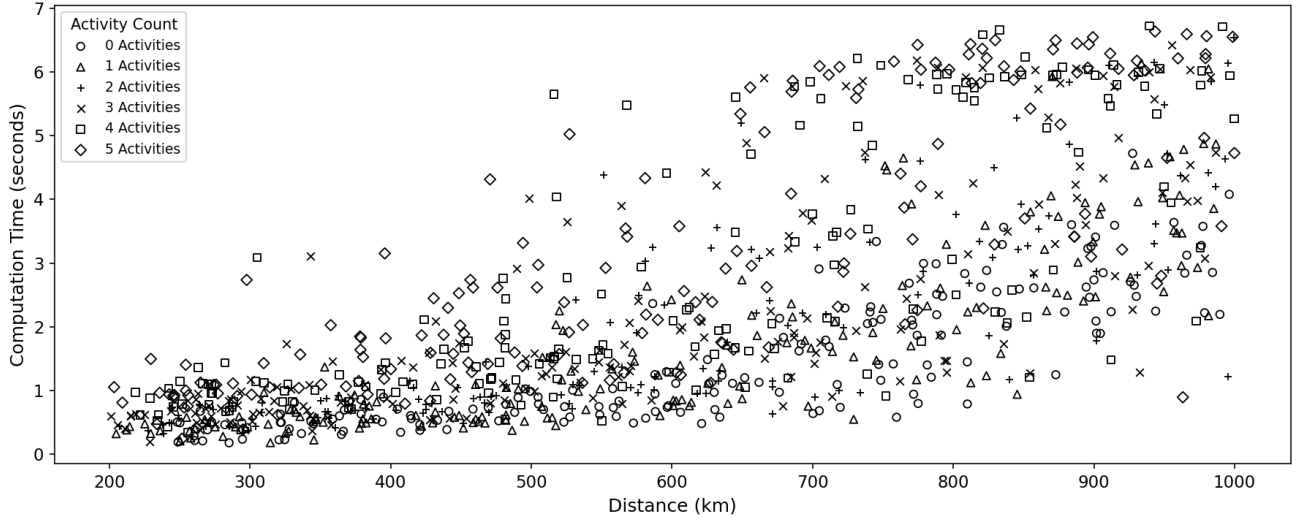


Figure 7: End-to-end computation time (Stages 1–3) versus route distance, stratified by the number of requested activities (0–5). Latency grows with distance, and the banding just above five seconds reflects the fixed CP-SAT search limit on long routes. The activity-count strata are visually intermixed across the distance range, suggesting route length rather than activity count is the primary driver of solve time.

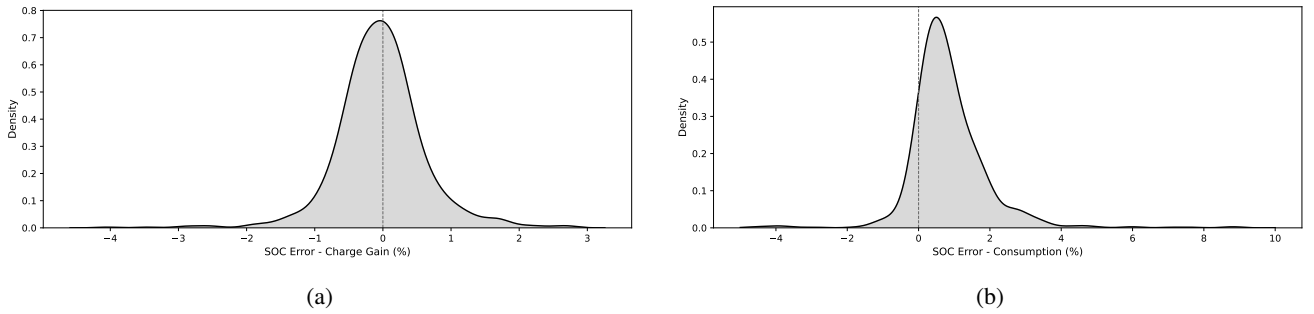


Figure 8: Distribution of per-instance deviations between the linear energy abstractions and the production-grade powertrain model over the 1,000 routes. (a) Charge-gain error and (b) consumption error.

in which these components are planned separately. The third is a mixed-integer linear programming formulation of the identical Stage-3 model, solved with HiGHS, which isolates the choice of the combinatorial search technology.

7.5.1. FRVCP Optimality Bound

We employ *frvcpy* (Kullman et al., 2021)⁴, an open-source implementation of the exact labeling algorithm of Froger et al. (2019), to compute the exact FRVCP optimum and thereby quantify the optimality gap of our framework. The benchmark uses the semi-synthetic EV-IPA dataset without activities, with both approaches on the same fixed, per-station synthetic charging curve (Section 7.1). For each instance we fix the reference route and compare our framework (Stage 2.1 filtering followed by CP-SAT on the filtered graph) against *frvcpy* on the same graph. CP-SAT minimizes the sum of traversal and charging times, with both the minimum and preferred state-of-charge (SoC) set to zero.

⁴<https://github.com/e-VRO/frvcpy>

Table 6

Optimality gap against the *frvcpy* FRVCP optimum, with and without the safety-margin constraints.

	Mean	Median (p_{50})	p_{90}	Max
Optimality gap (%)				
Safety margin on	0.89	0.67	1.65	9.22
Safety margin off	0.13	0.01	0.42	3.88

Table 6 reports the comparison. Across 1,000 instances our framework attains a median gap of **0.67%** and a mean of **0.89%**. The deviation is the conservative energy margin we reserve, which induces earlier charging and higher SoC than the theoretical optimum. *frvcpy* exhausts the battery, arriving at 0% SoC on 884 instances, whereas our plans retain 5.5% on average. Disabling the safety-margin constraints collapses the median gap to **0.01%** and the mean to **0.13%**, isolating the margin, rather than the solver formulation, as the source. We accept this cost deliberately. On this sub-problem *frvcpy* is faster and exact, but it addresses neither

activities, time windows nor preferences. The gap thus prices the generality of our formulation where both apply.

7.5.2. Sequential Greedy Heuristic

The second baseline mirrors the sequential practice of deployed in-car navigation systems (Section 1): charging is planned first, exactly on the fixed route, and the requested activities are committed one at a time, with *frvcpy* re-invoked as an exact oracle after each insertion. Both methods share the per-station charging curve (Section 7.1), the preprocessed candidate graph, and the energy configuration of Section 7.5.1: $E_{\min}$ and E_{pref} are set to zero, and only the destination target $E_{\text{target}} = 50\%$ is retained. Since *frvcpy* natively requires only a non-negative state of charge, the target is imposed through the instance: every arc into the destination node is inflated by E_{target} in the energy matrix, so that non-negative arrival in *frvcpy* corresponds to physical arrival with at least E_{target} . We do not optimize the heuristic for computational performance, since its purpose is to isolate the benefit of joint optimization.

Algorithm 2 outlines the heuristic. Because the charging curve tapers at higher SoC, colocation is restricted to chargers reached at a high charging rate: a charger c is eligible for activity a if it lies within ρ_{walk} , is open on arrival, and is reached within a band above a planned stop s (Algorithm 2, $y_{\text{pre},\cdot}$ denotes the SoC on reaching a node). Eligible chargers are scored by their net time saving, the detour and dwell time saved minus the time-window wait and the slower charging at $y_{\text{pre},c}$:

$$g_{\text{coloc}}(c, a, s) = \underbrace{(\tau_s + \tau_a) - \tau_c}_{\text{detour saved}} + \underbrace{\min(t_c, d_a)}_{\text{dwell overlap}} - \underbrace{w_c}_{\text{TW wait}} - \underbrace{(t_c - t_s)}_{\text{taper penalty}} \quad (32)$$

If no score is positive, a is inserted as a standalone stop at its minimum-detour candidate.

Table 7 reports the comparison, paired over the 783 of the 837 activity-bearing instances on which both arms return a plan (the hard 50 % target is unreachable for the greedy arm on 54).

CP-SAT reduces the total trip duration by 15.15 min on average (median 13.35, shorter trips on 609 instances, 77.8%) and decreases non-charging dwell time, i.e. stationary time not overlapped by a charging session, by 33.58 min (median 26.51, Figure 9). A one-sided Wilcoxon signed-rank test confirms the reduction in non-charging dwell ($W = 299,613$, $p = 2.8 \times 10^{-119}$; 782 of the 783 pairs non-zero) and in total trip duration ($W = 270,508$, $p = 4.5 \times 10^{-83}$; 775 non-zero). While CP-SAT spends 2.59 min less on driving, it charges 21.03 min longer. Two effects account for much of this: charging in parallel to an activity adds no trip time, so the solver lets such sessions run for the activity’s duration, which inflates the charging time without lengthening the trip; and CP-SAT holds a larger buffer above $E_{\min}$ en route, both arms depositing the same energy at the destination.

The heuristic also illustrates the rigidity argued in Section 2: colocation alone required a hand-crafted eligibility band and scoring rule (Eq. 32), and each further requirement type would need logic of its own, whereas the CP model absorbs it as a declarative addition.

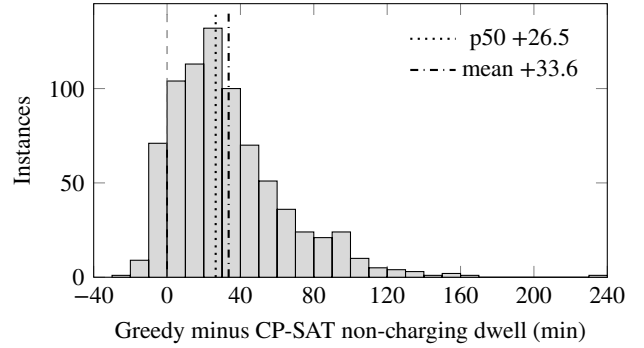


Figure 9: Paired reduction of non-charging dwell (greedy minus CP-SAT), all 783 paired instances.

7.5.3. MILP Reference (HiGHS)

To situate the CP-SAT choice against the classical exact alternative, we formulate the identical Stage-3 model (same candidate graph, discretization, and objective) as a mixed-integer linear program and solve it with HiGHS (version 1.15.1) (Huangfu and Hall, 2018)⁵ under a 180 s time limit. The deployment stack is restricted to open-source solvers, making open-source MILP performance the relevant reference. We assess 100 randomly selected instances from EV-IPA-1000. HiGHS returns a feasible solution on 95 of them within the 180 s limit and certifies the optimality of its own formulation on 91 of those 95 (95.8 %). For those instances on which both solvers prove optimality, the corresponding optimal solutions coincide exactly: the relative objective difference $(J_{\text{CP}} - J_{\text{MILP}}) / J_{\text{MILP}}$ has mean, median, 90th percentile, and maximum all equal to 0.00%. The two formulations are therefore empirically equivalent on this sample. The cost is latency: the MILP requires a median of 4.3 s per solve (mean 8.3 s, p_{90} 21.7 s, max 51.8 s), against a median of 0.7 s (mean 1.0 s, p_{90} 1.8 s, max 3.8 s) for CP-SAT on the same instances. Identical solution quality at roughly an order of magnitude higher latency empirically supports constraint programming as the backend for the interactive setting.

7.6. Ablation Studies

To validate the individual contributions of our core algorithmic components, we evaluate three restricted variants of our framework: one disabling graph-level stop merging, one removing the localized energy bounds, and one disabling candidate filtering.

7.6.1. Opportunistic Colocation (Stop Merging)

To evaluate our graph preprocessing, we conducted an ablation study disabling opportunistic colocation on the

⁵<https://highs.dev>

Algorithm 2: Sequential Greedy Baseline

Input : Route R^* (origin–destination); activity set $\mathcal{A}_{\text{req}}$; candidate graph $\mathcal{G}$; floors $E_{\min}$ (route-wide) and E_{target} (destination arrival)

Output: Itinerary π , or INFEASIBLE

// FRVCP(R): exact optimal charging on a fixed route (Froger et al., 2019), subject to $E_{\min}$ and E_{target} , or infeasible

// $Z(\pi)$: objective value of itinerary π (total trip duration)

// $\omega = (\omega_1, \dots, \omega_{|\mathcal{A}_{\text{req}}|})$: an ordering of $\mathcal{A}_{\text{req}}$. GreedyPlan() operates on sequences: it consumes the head and recurses on the remainder

// $y_{\text{pre},x}$: SoC at which the vehicle reaches x . The admissible band above a planned stop s is $[y_{\text{pre},s}, \bar{y}(s)]$ with $\bar{y}(s) = \min\{y_{\text{cap}}, y_{\text{pre},s} + \delta\}$, ceiling $y_{\text{cap}} = 50\%$, tolerance $\delta = 10\text{pp}$

// **Entry point** – every one of the $|\mathcal{A}_{\text{req}}|!$ orderings is planned independently; the best feasible itinerary is returned

```

1 Function GreedyBaseline( $R^*, \mathcal{A}_{\text{req}}$ ):
2    $\Pi \leftarrow \emptyset$ ;
3   foreach ordering  $\omega$  of  $\mathcal{A}_{\text{req}}$  do
4      $\pi_\omega \leftarrow \text{GreedyPlan}(R^*, \omega, \text{null})$ ; // independent; may run in parallel
5     if  $\pi_\omega \neq \text{INFEASIBLE}$  then  $\Pi \leftarrow \Pi \cup \{\pi_\omega\}$ ;
6   end
7   if  $\Pi = \emptyset$  then return INFEASIBLE;
8   return  $\arg \min_{\pi \in \Pi} Z(\pi)$ ; // best over all orderings
9 end

// Inner routine – fixes exactly one activity per call, then recurses; a pass terminates in  $|\mathcal{A}|$  steps
10 Function GreedyPlan( $R, \mathcal{A}, S$ ):
11   if  $S = \text{null}$  then  $S \leftarrow \text{FRVCP}(R)$ ;
12   if  $S = \text{INFEASIBLE}$  then return INFEASIBLE;
13   if  $|\mathcal{A}| = 0$  then return  $\pi = (R, S)$ ;
14    $a \leftarrow \mathcal{A}_1$ ; // head of the sequence
15   // eligible pairs: charger  $c$  walk-reachable from  $a$ , open on arrival, passed inside the admissible band of the planned stop  $s$  it displaces
16    $\mathcal{W} \leftarrow \{(c, s) \in \mathcal{G} \times S : \text{dist}(c, a) \leq \rho_{\text{walk}}, c \text{ open on arrival}, y_{\text{pre},s} \leq y_{\text{pre},c} \leq \bar{y}(s)\}$ ;
17   if  $\mathcal{W} \neq \emptyset$  and  $\max_{(c,s) \in \mathcal{W}} g_{\text{coloc}}(c, a, s) > 0$  then
18     //  $g_{\text{coloc}}$ : Eq. (32)
19      $(c^*, s^*) \leftarrow \arg \max_{(c,s) \in \mathcal{W}} g_{\text{coloc}}(c, a, s)$ ;
20      $R \leftarrow R$  with  $a$  co-located at  $c^*$ ;
21   else
22      $R \leftarrow R$  with  $a$  at its minimum-detour candidate (standalone);
23   end
24    $S \leftarrow \text{FRVCP}(R)$ ; // re-solve charging on the augmented route
25   return GreedyPlan( $R, (\mathcal{A}_2, \dots, \mathcal{A}_{|\mathcal{A}|}), S$ ); //  $a$  now a fixed stopover
26 end

```

production-grade data cohort over all 1,000 instances: each instance is re-solved on the same reference route, candidate pool and charging curves, so that only the merge step differs. Trip-time effects are reported on the activity-bearing instances, on which the merge step can act.

Trip duration on these instances falls from a mean of 582.9 to **537.0** minutes (-7.9%) and from a median of 576.2 to **531.1** minutes (-7.8%), with the 90th percentile dropping from 824.7 to 750.4 (-9.0%) and the maximum from 1,059.9 to 955.0 minutes (-9.9%). Per instance, merging saves a mean of **45.9** minutes (median 42.7, p90 80.8, maximum 152.3). Merging also makes the model easier to solve. Relative to the merged arm, which is the main-run configuration of Table 5, disabling merging nearly doubles the mean

Stage 3 time (3.09 vs. 1.62 s) and drives the median to the search limit (5.04 vs. 0.98 s).

7.6.2. Dynamic Energy Budget and SLC

To test the necessity of the continuous mid-segment guarantee, we ablate the Segment-Length Condition (SLC) together with the dynamic energy budget: the solver may spend the full surplus on detours ($\alpha = 1$, relaxing Eq. (9)), and the minimum entry-energy constraint of Corollary 1, which reserves the interior worst-case consumption U_k , is dropped, so that energy feasibility is checked only at the segment boundaries through Eqs. (6)–(7). Both arms are evaluated with the validation procedure of Section 7.4. The share of routes whose validated SoC stays above $E_{\min}$ on every driving link is reported as the energy feasibility rate (EFR). Under the standard configuration of Section 7.2 the

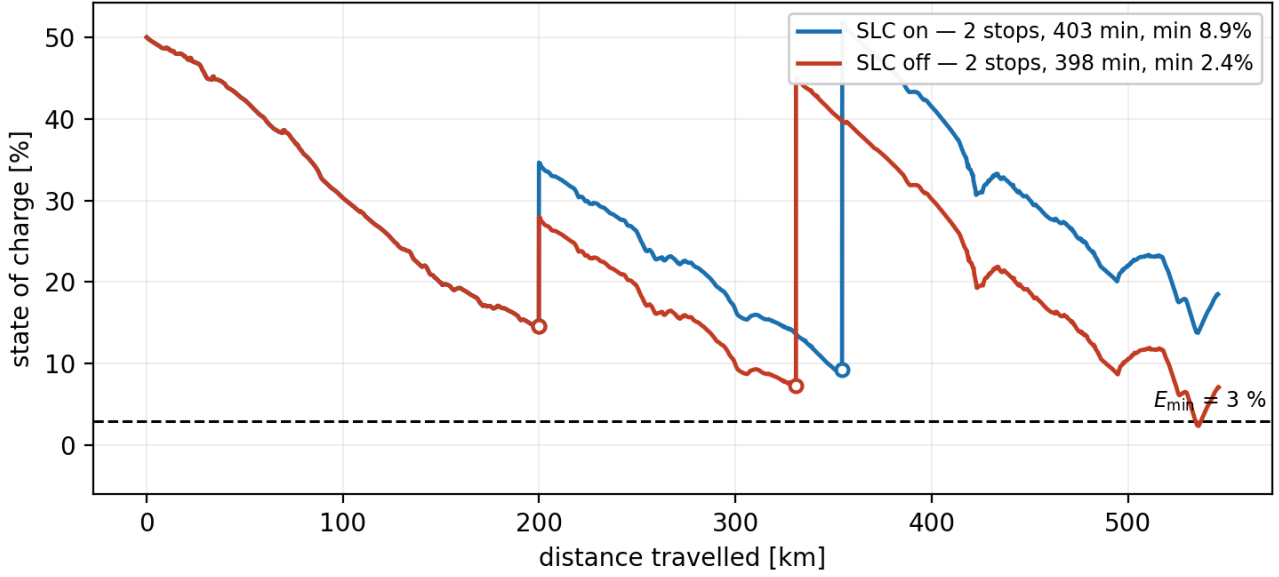


Figure 10: Validated minimum on-route SoC with (SLC on) and without (SLC off) the Segment-Length Condition under the stress configuration. The dashed line marks $E_{\min} = 3\%$.

Table 7

Paired differences (greedy – CP-SAT) over the 783 instances on which both arms return a plan. Means are additive across components, percentiles are computed per component and therefore do not sum to the total-trip row.

	Mean	p50	p90	Max
<i>Paired difference, greedy – CP-SAT (min)</i>				
Non-charging dwell	+33.58	+26.51	+77.93	+238.35
Travel	+2.59	+1.85	+12.11	+48.34
Charging	–21.03	–12.83	–0.18	+27.43
Total trip	+15.15	+13.35	+39.11	+197.18
Charging stops (count)	+0.70	+1.00	+2.00	+8.00
<i>Solve time (s)</i>				
Greedy	1.94	0.21	6.03	37.43
CP-SAT	2.16	1.42	5.00	5.44

ablation shows no effect: the soft SoC preferences (10% at intermediate stops, 50% target at the destination) keep the profile clear of $E_{\min}$, and both arms are energy-feasible on every replayed route. We therefore repeat the experiment under a stress configuration that eliminates this margin: $E_{\min}$, the preferred intermediate SoC E_{pref} , and the destination target E_{target} are all fixed at 3%, while all other settings remain the same, so that the objective drives the plan down to $E_{\min}$ at every stop and at the destination.

Under this parameterization the SLC matters (Table 8, Figure 10). To isolate the impact of our consumption-side guarantee from charging approximation errors, we evaluate the plans under two validation regimes: a pure-consumption regime (where planned charging gains are assumed to be

realized) and a combined regime (where both consumption and powertrain-simulated charging dynamics are evaluated).

With the SLC active, the EFR remains high. Under pure-consumption validation, exactly 2 routes fail (EFR **99.8%**). These two failures stem directly from an underestimation of the detour consumption by the distance-based rate (Assumption 1 breached). When full powertrain charging dynamics are reintroduced (combined regime), one of these two failures is artificially padded by a slight charging overestimation, leaving 1 violation (EFR **99.9%**). While the model does not actively include a safety margin for charging variations (Assumption 3), the structural reserve enforced by the SLC successfully absorbed all other charging deviations across the remaining 999 instances.

Without the SLC, violations become frequent. Under the pure-consumption validation regime, 26 of the replayed routes fall below $E_{\min}$ (yielding an EFR of **97.4%**). As illustrated in Figure 10, these failures occur because mid-segment SoC troughs can breach the minimum threshold completely undetected between checked boundary nodes. When the full powertrain charging dynamics are reinstated, the number of failures increases to 66 (EFR: **93.4%**). Without the structural reserve, any overestimation of a charging gain propagates downstream and compounds the mid-segment dips.

While the SLC neutralizes on-route variance, the two remaining pure-consumption violations occur because the current detour estimation relies solely on distance, occasionally underestimating steep or slow off-route excursions (Assumption 1). To resolve this, future work will refine detour bounds by incorporating travel time, which is already available without additional latency, alongside parallel elevation queries for candidate nodes.

Table 8

SLC ablation under the stress configuration; Δ is on minus off, SoC in %. The energy feasibility rate (EFR) is reported for both the pure-consumption (charging validation disabled) and combined validation regimes.

Metric	SLC off		SLC on		Δ	
	mean	med.	mean	med.	mean	med.
EFR, pure-cons. (%)	97.4		99.8		+2.4	
EFR, combined (%)	93.4		99.9		+6.5	
Min. SoC on route	7.61	5.27	11.96	10.04	+4.35	+4.04
Trip time (min)	439.0	440.2	444.5	444.8	+5.5	+2.3
Charging time (min)	49.4	48.5	53.7	50.8	+4.4	+2.9
Charging stops	1.80	2.00	1.92	2.00	+0.12	0.00
Stage 3 solve (s)	1.53	0.98	1.45	0.90	-0.08	-0.04

7.6.3. Candidate Filtering (Stage 2.1)

We evaluate the candidate filtering from two complementary perspectives: (i) its impact on the full EV-IPA objective when solved using our own CP-SAT solver, and (ii) its impact on charging-time optimality in isolation, as assessed by an independent exact solver for the charging subproblem.

Full-model filtering (CP-SAT). To evaluate the impact of the dominance-based filtering pipeline, we re-solve all 1,000 instances of the production-grade data cohort with Stage 2.1 pruning disabled. The ablation removes the selection logic of Section 4.2. The time-window feasibility check remains active in both arms, as a candidate whose opening hours cannot intersect the feasible arrival window instantiates no node. The unfiltered arm receives an extended 900 s search limit in place of the 5 s default budget.

Disabling filtering increases the median graph size from **116** to **385** nodes and from **415** to **7,256** edges (Table 5), the edge count growing quadratically in the per-segment node count through the intra-segment edges. The median Stage 3 solve time rises from **0.98** s to **5.40** s and p_{90} from **4.70** s to **106** s, with the hardest unfiltered instances exhausting the 900 s limit.

Within its extended budget the unfiltered arm proves optimality on **97.5%** of its solved instances, the filtered arm on **93.9%** within the default budget. On the routes where both arms prove optimality, comparing the two exact optima isolates the price of filtering on the full EV-IPA objective: the filtered optimum exceeds the unfiltered one by **+0.14%** at the median (mean **+0.52%**, p_{90} **+2.20%**, max **+8.79%**). As the filtered graph is a subgraph of the unfiltered one, this gap is nonnegative by construction.

Charging-only filtering (exact FRVCP solver). To validate the quality of the filter using a method independent of our own solver, we restrict the problem to the charging subproblem and solve it exactly with `frvcpy` (Kullman et al., 2021). For each of the 1,000 benchmark instances, each with its own reference route and corridor station inventory, we solve the charging subproblem twice: once on the unfiltered candidate graph containing all stations of the corridor, and

Table 9

Solvability and EFR under the stress configuration (250 instances, 50% initial SoC). *Viol.* counts replayed routes dropping below $E_{\min}$ on some driving link.

Configuration	Solved	EFR (%)	Viol.
Baseline ($\alpha=0.1$, $\gamma_{\text{safe}}=1.08$)	250/250	100.0	0
<i>Surplus fraction α</i>			
0	167/250	100.0	0
0.5	250/250	100.0	0
1.0	250/250	97.6	6
<i>Safety multiplier γ_{safe}</i>			
1.00	250/250	100.0	0
1.05	250/250	100.0	0
1.25	250/250	100.0	0
1.75	250/250	100.0	0

once on the Stage 2.1-filtered graph. We report the resulting optimality gap as the relative increase in total traversal-plus-charging time of the filtered-graph optimum over the unfiltered-graph optimum. Since both optima are exact, the gap isolates the loss attributable to filtering.

Over the 1,000 instances, the gap is **0.00%** at the median (mean **+0.26%**, p_{90} **+1.00%**, max **+2.89%**). The zero median reflects exact ties: on **593** routes (59.3%), the filtered graph still contains an unfiltered optimum and the two objectives coincide.

7.7. Sensitivity Analysis

We independently varied the surplus fraction α (Definition 2), the safety multiplier γ_{safe} (Section 5.1), the retention budgets B_{chg} and B_{poi} (Stage 2.1), and the segment energy budget ρ_{seg} (Section 4.1), on 250 randomly sampled instances with a fixed 50% initial SoC. The energy parameters α and γ_{safe} are swept under the stress configuration of Section 7.6.2 ($E_{\min}$, preferred intermediate SoC, and destination target all at 3%). Since both parameters act on the consumption side, the sweep is validated in the pure-consumption regime of Section 7.6.2, with planned charging gains assumed realized. Table 9 reports solvability and Energy Feasibility Rate (EFR). Figure 11 reports the cost of B_{chg} .

Table 9 presents the impact of α and γ_{safe} . The surplus fraction binds at both extremes. At $\alpha = 0$ no detour energy may be spent, no off-route candidate is reachable, and solvability drops to **66.8%**. At $\alpha = 1.0$ the entire surplus is released to detours and the minimum entry-energy constraint of Corollary 1 degenerates. All instances remain solvable, but 6 of these 250 validated routes fall below $E_{\min}$ (EFR **97.6%**). Over $\alpha \in [0.05, 0.5]$ both metrics saturate at 100%. The safety multiplier binds on neither axis over $[1.00, 1.75]$, including at $\gamma_{\text{safe}} = 1.00$, where the reference bound carries no margin of its own and the guarantee rests on the SLC reserve and the high-percentile rate alone.

The B_{chg} and B_{poi} budgets are not equally expensive. Raising B_{poi} from 10 to 50 leaves median solve time (1.94, 1.96, 1.87 s), proof rate (87.0, 85.8, 85.1 %) and mean residual gap (0.61, 0.62, 0.66 %) unchanged, in contrast to the

Table 10Segment energy budget ρ_{seg} under the stress configuration (250 instances, 50% initial SoC.)

ρ_{seg}	Mean	Median (p_{50})	p_{90}	Max
<i>Segments K</i>				
0.025	65.0	68	99	109
0.05	32.7	34	50	55
0.10	16.6	17	25	28
0.20	8.6	9	13	14
<i>Graph edges</i>				
0.025	309.8	299	532	627
0.05	267.1	240	493	597
0.10	249.2	207	498	594
0.20	244.1	187	521	640
<i>Charger candidates</i>				
0.025	58.4	57	91	111
0.05	42.4	41	65	84
0.10	30.9	30	47	60
0.20	23.6	23	38	49
<i>Trip time (min)</i>				
0.025	449.1	450.4	660.7	795.7
0.05	450.8	451.8	660.8	806.9
0.10	454.5	454.0	666.8	813.0
0.20	466.6	470.9	686.9	865.7
<i>Stage-3 solve (ms)</i>				
0.025	2 274	1 182	5 116	5 140
0.05	1 896	836	5 087	6 659
0.10	1 202	547	3 713	5 079
0.20	639	337	1 568	5 063

degradation under B_{chg} (Figure 11). The asymmetry follows from scope: B_{chg} is enforced per energy segment, whereas B_{poi} is a global ranking whose candidates distribute across segments.

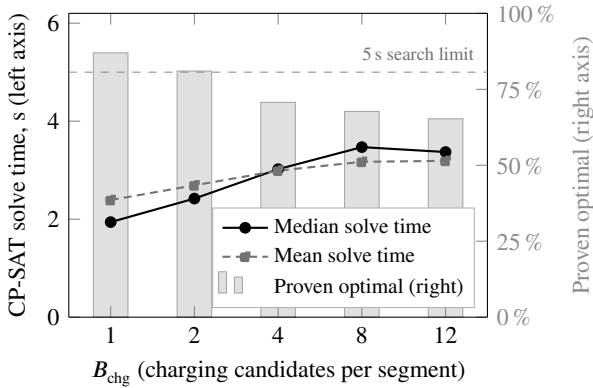


Figure 11: Cost of the per-segment charging-candidate budget B_{chg} (250 instances per configuration).

Table 10 shows the effect of the segment budget ρ_{seg} . Increasing it from 0.025 to 0.20 reduces the median segment count from 68 to 9 and the Stage-3 solve time by 3.5 $\times$. Trip time is flat up to $\rho_{\text{seg}} = 0.10$ (+0.8%) and rises at 0.20 (+4.6% over 0.025), which is the cost of having a larger reserve: longer segments demand entering each segment with higher energy, which in turn requires charging sooner. Charger and edge counts fall more slowly than K (2.5 $\times$ and

Table 11

Per-turn incremental re-optimization latency in milliseconds over the production-grade data cohort.

Turn	n	Mean	Median	p_{90}	Max
P0: + activity 1	658	421	230	1,044	3,258
P1: + appointment	619	452	223	1,312	3,312
P2: + activity 2	580	464	261	1,130	3,500
P3: + order reversal	511	998	501	2,545	5,071

1.6 $\times$ against 7.6 $\times$), because the Stage 2.1 retention rules (Section 4.2) exceed the allocated budget for each segment.

7.8. Incremental Re-optimization

To quantify the cost of conversational re-optimization (Section 6.1), we simulate a four-turn refinement dialog over each solved itinerary of the production-grade data cohort, applying the modifications cumulatively so that every turn inherits the constraints of the previous one. From a charging-only base plan, the driver (i) *adds an activity* of a category absent from the original request; (ii) imposes an *appointment*, a hard arrival-time window of $[-30, +60]$ minutes for the arrival at that activity; (iii) *adds a second activity*; and (iv) *reverses the order* of the two activities through a hard sequence constraint while retaining the appointment. Every turn reuses the cached route-dependent artifacts of the original solve (energy estimates, reference segmentation, candidate graph, charging envelopes) and recomputes only the lightweight combinatorial model. The two activity-addition turns additionally retrieve and assess the newly generated candidates. Each result is validated structurally with soft-constraint relaxation disabled, so a requested modification cannot be silently dropped.

Table 11 reports the per-turn latency distributions across the cohort ($n=658$ itineraries with at least two available activity categories). The first three turns stay near a quarter of a second at the median (230, 223, and 261 ms). The order reversal is the most expensive turn (median 501 ms, p_{90} 2.5 s). The n column traces feasibility: each turn is executed only on the itineraries for which all preceding modifications remain satisfiable.

8. Limitations and Future Work

The framework plans along a fixed reference route and does not revisit the underlying road routing. This is a property of the setting rather than a restriction of the evaluation: precomputing the route-dependent physics once is what enables the interactive latency, and the deployment context supplies the reference route from the navigation system. Two aspects of the operating environment are abstracted away in the present evaluation. Routes are computed under no-traffic conditions, and charging stations are presumed to be available upon arrival. Neither is intrinsic to the proposed formulation: predicted station occupancy levels and connector availability can be incorporated into the filtering as an additional layer. Similarly, congestion-sensitive travel

times can be modeled with the same edge-cost structure. Furthermore, the deployment context mitigates these limitations, as modern EVs continuously receive real-time traffic, infrastructure availability, and telemetry data throughout the trip. Consequently, any plan whose underlying assumptions are invalidated during execution can be re-optimized and presented to the driver in a form analogous to an alternative-route suggestion.

The guarantee itself is conditional. While the DLEA and SLC make evaluating energy solely at segment boundaries mathematically safe (Theorem 1), that theorem rests on two approximations the model does not certify: that off-route detour penalties bound physical consumption (Assumption 1), and that the piecewise-linear envelopes do not overestimate the realized charging gain (Assumption 3). The stress configuration of Section 7.6.2 exposes both. Optimality is similarly exchanged for reduced latency: candidate filtering incurs an average degradation of 0.52% in the value of the full objective, and the adoption of conservative margins results in an additional 0.89% loss relative to the FRVCP optimum (see Sections 7.6.3 and 7.5.1).

Three immediate refinements suggest themselves. First, the detour bound can be generalized from a purely distance-based constraint to one that also incorporates travel time and localized elevation information, thereby targeting the residual violations observed in Section 7.6.2. Second, the two-piece formulation introduced in Appendix A can be employed to reduce the remaining optimism in the charging-gain estimation. Third, the interior bound can be tightened to the exact per-segment prefix drawdown available from the cached link-level profile. Since the percentile construction deliberately embeds a uniform safety margin, this tightening is best paired with a margin calibrated explicitly from observed link-level prediction-error statistics. Independently of that recalibration, the surplus-fraction budget (9) could be replaced by a single decision-dependent entry condition, which certifies the same interior guarantee with less reserve but forgoes the multiplicative tolerance to detour misestimation discussed in Section 5.4. We leave these refinements to future work.

Two additional avenues for future work pertain to the scope of the evaluation. The present benchmark is confined to a single vehicle configuration (104.1 kWh, 400 V) operated under mostly warm-weather conditions. Although preliminary experiments conducted on additional operating profiles indicate similar feasibility and performance characteristics, a comprehensive, systematic exploration across a broader range of hardware architectures and meteorological conditions remains outstanding. Similarly, the conversational setting is currently instantiated via a scripted four-turn dialogue (Section 7.8). Future work will therefore include an end-to-end study with a complete LLM dialog layer and real drivers, under representative usage conditions.

9. Conclusion

This article introduces EV-IPA, a framework for the joint optimization of vehicle routing, facility selection within a corridor around the route, parallel activity scheduling, and non-linear charging under battery constraints. It demonstrates that Constraint Programming (CP) can solve this integrated planning problem within an interactive latency budget by decoupling computationally expensive, offline evaluations of battery physics from the online combinatorial search.

The primary methodological challenge is the integration of continuous, non-linear battery dynamics into a discrete CP search space. We address this challenge through two complementary linearization strategies. On the energy consumption side, the Dynamic Linear Energy Approximation reduces route-level energy dynamics to segment-wise linear constraints and simultaneously assigns a cost to each off-route detour by charging it against a decision-dependent energy budget derived from the surplus available on the same route segment. The Segment-Length Condition subsequently allocates the segment's worst-case internal energy consumption in advance of that budget, thereby ensuring that state-of-charge feasibility is maintained not only at the discrete segment boundaries but also at every continuous point along the trajectory. On the charging side, a precomputed convex envelope captures non-linear, temperature-dependent battery tapering as a small set of linear facets. On EV-IPA-1000, the open benchmark of 1,000 European instances released with this article, the framework returns itineraries at a median end-to-end latency of 1.58 s on production-grade data, within 0.67% of the exact FRVCP optimum on the charging subproblem, and reduces trip time by 7.9% on activity-bearing instances by synchronizing charging with the driver's activities rather than sequencing them. When validated against a production-grade powertrain model, the itineraries remain energy-feasible on 99.9% of routes even after the complete removal of all soft safety margins, thereby demonstrating that these abstractions are not only computationally tractable but also operationally safe.

A central advantage of the CP formulation is that it is declarative. Expressive user requirements drawn from the supported template vocabulary map directly onto native constraints without algorithmic reformulation: ordering relations (dinner before the hotel), anchored temporal windows (check-in by 9 pm), and periodic rest breaks can be simply activated and parameterized in the model. The two-phase architecture makes this flexibility interactive: the expensive route-dependent physics, the energy estimation, segmentation, and charging envelopes, is computed once and reused across the dialog. A change in preferences or weights then requires the lightweight combinatorial solve, while a request for a new activity additionally triggers a fresh corridor search for the relevant points of interest, but still reuses the cached physical artifacts rather than recomputing them. In either case, the vehicle performs rapid re-optimization across successive dialog turns within a timescale of seconds,

thereby transforming trip planning into a conversation with an optimization engine rather than a single, static query.

We see in-dialog personalization as the most compelling direction this work opens. Preferences such as favoring more frequent but shorter stops, prioritizing minimal total travel time, or restricting routing to well-lit charging stations translate naturally into reweighted objective terms or re-parameterized constraint templates, after which the cached precomputation returns a revised plan within seconds. This responsiveness matters beyond convenience: decision psychology has long held that human preferences are often constructed in the course of making a decision rather than known in advance, and depend systematically on how they are elicited (Slovic, 1995). A backend capable of re-optimizing this quickly is what makes dialog preference elicitation possible, allowing drivers to refine an itinerary through dialog and arrive at a plan that reflects their true preferences.

CRediT authorship contribution statement

Alexandra Wins: Conceptualization, Methodology, Formal analysis, Software, Data curation, Investigation, Writing – original draft. **Benedikt Heidrich:** Formal analysis, Software, Writing – review & editing. **Felix Haug:** Software, Data curation. **Sebastian Dorn:** Formal analysis, Writing – review & editing. **Daniel Bruckner:** Software. **Sven Stahlmann:** Software. **Jens Nahm:** Conceptualization, Resources. **Sascha Alpers:** Supervision, Writing – review & editing. **Andreas Oberweis:** Formal analysis, Supervision, Writing – review & editing.

Declarations

Declaration of competing interest The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Alexandra Wins, Benedikt Heidrich, Felix Haug, Daniel Bruckner, Sven Stahlmann and Jens Nahm are employed by Mercedes-Benz Tech Innovation GmbH. Sebastian Dorn is employed by Mercedes-Benz AG. Mercedes-Benz AG has filed a patent application covering the algorithmic framework described in this article. Sascha Alpers and Andreas Oberweis declare no competing interests.

Data availability The EV-IPA-1000 dataset generated and analysed in this study is publicly available on Zenodo (Wins, 2026).

Declaration of generative AI and AI-assisted technologies in the writing process During the preparation of this work the authors used Claude (Anthropic) in order to improve the language and readability of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

- Artmeier, A., Haselmayr, J., Leucker, M., Sachenbacher, M., 2010. The shortest path problem revisited: Optimal routing for electric vehicles, in: Annual conference on artificial intelligence, Springer. pp. 309–316.
- Ayer, M., Brunk, H.D., Ewing, G.M., Reid, W.T., Silverman, E., 1955. An empirical distribution function for sampling with incomplete information. The annals of mathematical statistics , 641–647.
- Barlow, R.E., 1972. Statistical inference under order restrictions: The theory and application of isotonic regression .
- Basso, R., Kulcsár, B., Sanchez-Diaz, I., 2021. Electric vehicle routing problem with machine learning for energy prediction. Transportation Research Part B: Methodological 145, 24–55.
- Baum, M., Dibbelt, J., Gamsa, A., Wagner, D., Zündorf, T., 2015. Shortest feasible paths with charging stops for battery electric vehicles, in: Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems, pp. 1–10.
- Booth, K.E., Beck, J.C., 2019. A constraint programming approach to electric vehicle routing with time windows, in: International Conference on integration of constraint programming, artificial intelligence, and operations research, Springer. pp. 129–145.
- Cuchý, M., Štolba, M., Jakob, M., 2018. Integrated route, charging and activity planning for whole day mobility with electric vehicles, in: International Conference on Agents and Artificial Intelligence, Springer. pp. 274–289.
- Dastpak, M., Errico, F., Jabali, O., Malucelli, F., 2024. Dynamic routing for the electric vehicle shortest path problem with charging station occupancy information. Transportation Research Part C: Emerging Technologies 158, 104411.
- Desaulniers, G., Errico, F., Irnich, S., Schneider, M., 2016. Exact algorithms for electric vehicle-routing problems with time windows. Operations Research 64, 1388–1405.
- Dutta, S., Bauman, J., 2024. An overview of 800 v passenger electric vehicle onboard chargers: challenges, topologies, and control. IEEE Access 12, 105850–105864.
- Erdelić, T., Carić, T., 2019. A survey on the electric vehicle routing problem: Variants and solution approaches. Journal of Advanced Transportation 2019, 5075671.
- European Parliament and Council of the European Union, 2023. Regulation (EU) 2023/1804 on the deployment of alternative fuels infrastructure (afir). Official Journal of the European Union, L 234. URL: <https://eur-lex.europa.eu/eli/reg/2023/1804/oj>. art. 3(4)(a)(i); Accessed: 2026-05-26.
- Froger, A., Jabali, O., Mendoza, J.E., Laporte, G., 2022. The electric vehicle routing problem with capacitated charging stations. Transportation Science 56, 460–482.
- Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2019. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. Computers & Operations Research 104, 256–294.
- Huangfu, Q., Hall, J.J., 2018. Parallelizing the dual revised simplex method. Mathematical programming computation 10, 119–142.
- International Energy Agency, 2025. Global EV Outlook 2025. Technical Report. IEA. Paris. URL: <https://www.iea.org/reports/global-ev-outlook-2025/electric-vehicle-charging>. accessed: 2026-05-26.
- Jeong, J., Ghaddar, B., Zufferey, N., Nathwani, J., 2024. Adaptive robust electric vehicle routing under energy consumption uncertainty. Transportation Research Part C: Emerging Technologies 160, 104529.
- Karbowska-Chilinska, J., Chociej, K., 2020. Genetic algorithm for generation multistage tourist route of electrical vehicle, in: International conference on computer information systems and industrial management, Springer. pp. 366–376.
- Keskin, M., Çatay, B., 2016. Partial recharge strategies for the electric vehicle routing problem with time windows. Transportation research part C: emerging technologies 65, 111–127.
- Keskin, M., Çatay, B., 2018. A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. Computers & operations research 100, 172–188.

- Keskin, M., Laporte, G., Çatay, B., 2019. Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Computers & Operations Research* 107, 77–94.
- Keyser, M., Pesaran, A., Li, Q., Santhanagopalan, S., Smith, K., Wood, E., Ahmed, S., Bloom, I., Dufek, E., Shirk, M., 2017. Enabling fast charging—battery thermal considerations. *Journal of Power Sources* 367, 228–236.
- Kucukoglu, I., Dewil, R., Cattrysse, D., 2021. The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering* 161, 107650.
- Kullman, N.D., Froger, A., Mendoza, J.E., Goodson, J.C., 2021. frvcpy: An open-source solver for the fixed route vehicle charging problem. *INFORMS Journal on Computing* 33, 1277–1283.
- Lam, E., Desaulniers, G., Stuckey, P.J., 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.
- Lee, C., 2021. An exact algorithm for the electric-vehicle routing problem with nonlinear charging time. *Journal of the Operational Research Society* 72, 1461–1485.
- Lera-Romero, G., Bront, J.J.M., Soullignac, F.J., 2024. A branch-cut-and-price algorithm for the time-dependent electric vehicle routing problem with time windows. *European Journal of Operational Research* 312, 978–995.
- Merting, S., Schwan, C., Strehler, M., 2015. Routing of electric vehicles: Constrained shortest path problems with resource recovering nodes, in: 15th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2015), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017. The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological* 103, 87–110.
- Nielsen, J., et al., 1993. Usability engineering. volume 1. Morgan Kaufmann San Francisco.
- Pelletier, S., Jabali, O., Laporte, G., 2016. 50th anniversary invited article—goods distribution with electric vehicles: review and research perspectives. *Transportation science* 50, 3–22.
- Pelletier, S., Jabali, O., Laporte, G., 2019. The electric vehicle routing problem with energy consumption uncertainty. *Transportation Research Part B: Methodological* 126, 225–255.
- Perron, L., Didier, F., 2024. Cp-sat. Google OR-Tools, version 9.10. URL: https://developers.google.com/optimization/cp/cp_solver/. accessed: 2024-05-07.
- Rizopoulos, D., Esztergár-Kiss, D., 2020. A method for the optimization of daily activity chains including electric vehicles. *Energies* 13, 906.
- Rizopoulos, D., Esztergár-Kiss, D., 2023. Heuristic time-dependent personal scheduling problem with electric vehicles. *Transportation* 50, 2009–2048.
- Schiffer, M., Schneider, M., Walther, G., Laporte, G., 2019. Vehicle routing and location routing with intermediate stops: A review. *Transportation Science* 53, 319–343.
- Schneider, M., Stenger, A., Goeke, D., 2014. The electric vehicle-routing problem with time windows and recharging stations. *Transportation science* 48, 500–520.
- Slovic, P., 1995. The construction of preference. *American psychologist* 50, 364.
- Wang, Y.W., Lin, C.C., Lee, T.J., 2018. Electric vehicle tour planning. *Transportation Research Part D: Transport and Environment* 63, 121–136.
- Wins, A., 2026. EV-IPA-1000: A benchmark for electric vehicle itinerary planning with activities. URL: <https://doi.org/10.5281/zenodo.22015711>.
- Xiao, Y., Zhang, Y., Kaku, I., Kang, R., Pan, X., 2021. Electric vehicle routing problem: A systematic review and a new comprehensive model with nonlinear energy recharging and consumption. *Renewable and Sustainable Energy Reviews* 151, 111567.
- Xie, J., Zhang, K., Chen, J., Zhu, T., Lou, R., Tian, Y., Xiao, Y., Su, Y., 2024. Travelpplanner: A benchmark for real-world planning with language agents, in: International Conference on Machine Learning (ICML).

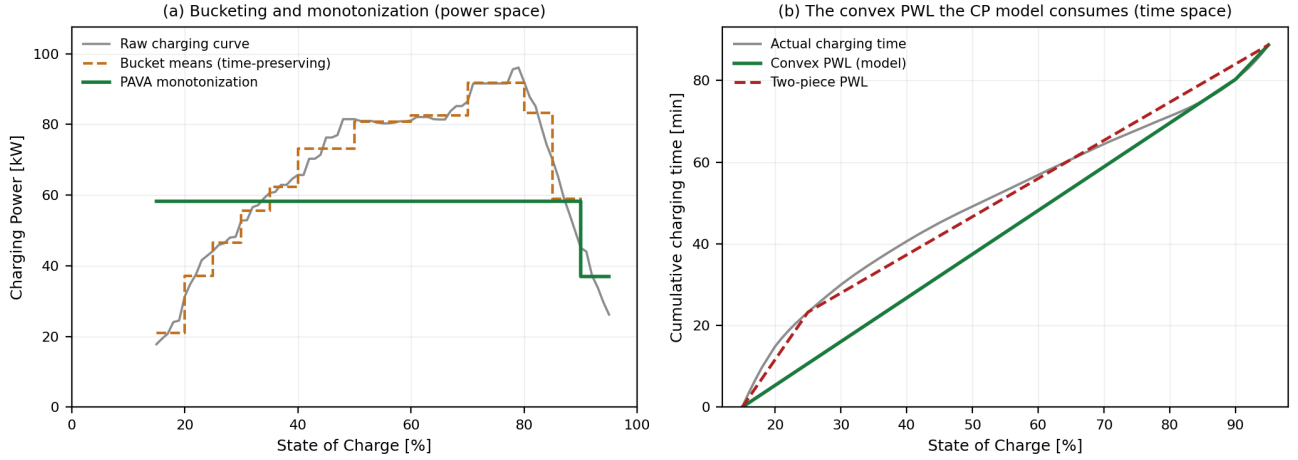


Figure 12: Charging-envelope construction at -20°C , cold start: (a) bucketing and monotization in power space; (b) actual charging time, convex envelope, and the two-piece PWL in charging-time space.

A. Outlook: Cold Charging Curves

At low battery temperatures the charging power initially rises while the battery warms during the session, so the minimum-charging-time function is no longer convex in SoC. The convex envelope then prices sessions spanning the pooled warm-up region, but is optimistic for sessions ending inside it (Figure 12).

Because a charging session occupies a contiguous SoC interval, a non-convex piecewise-linear charging-time function with two segments (slow while cold, fast once warm) can be encoded exactly with two interval-overlap variables per stop, without additional Boolean structure. We implemented this as an optional solver mode: curves that are already non-increasing keep the convex supporting-line formulation, and only non-monotone (cold) curves receive the two-piece model.

In a preliminary A/B on long routes at -20°C ambient, enabling the two-piece model changed neither solve time nor search effort within the solver's run-to-run variance, and all plans remained feasible under independent validation. Its practical reach is limited to sessions beginning on a cold battery, i.e., at departure before preconditioning can act, or after a prolonged non-charging stop in which the pack cools down again. A systematic evaluation of this regime, across vehicle platforms and their charging-curve families, is left to future work.