Chance-constrained battery management for electric bus scheduling

Léa Ricard[*,1], Guy Desaulniers[2], Andrea Lodi[3], and Louis-Martin Rousseau[2]

[1]School of Architecture, Civil and Environmental Engineering (ENAC), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[2]Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal, Canada
[3]Jacobs Technion-Cornell Institute, Cornell Tech and Technion - IIT, New York, USA

Abstract

The adoption of electric buses introduces challenges related to limited driving range, extended charging times, and strict battery management requirements. These requirements—often set by battery leasing companies or warranty providers—typically include recommended ranges for the battery's state-of-charge (SoC). This study addresses these challenges by proposing a novel chance-constrained model for the electric vehicle scheduling problem that accounts for stochastic energy consumption and battery management guidelines. The model ensures compliance with recommended SoC ranges while optimizing operational costs. A tailored branch-and-price heuristic with stochastic pricing problems is developed. Computational experiments on realistic instances show that the stochastic approach offers good trade-offs between operational costs and compliance with battery management guidelines, reducing costs by up to 10% compared to a deterministic worst-case benchmark. In contrast, we show that relying on expected energy consumption yields solutions with a near-certain probability of violating SoC guidelines, rendering such an approach unsuitable for practical use.

Keywords: Vehicle scheduling, Column generation, Chance-constrained optimization, Electric vehicles, Battery management

1 Introduction

The transition of public transport bus fleets towards battery electric buses (EBs) is gaining momentum worldwide as a solution to reduce local air and noise pollution. This global trend is reinforced by the COP26 declaration, which calls for all new vehicle sales—including buses—to be zero-emission by 2040 globally, and by 2035 in leading markets. However, the adoption of EBs brings its share of challenges

*Corresponding author: lea.ricard@epfl.ch

and additional constraints. Compared to traditional internal combustion engine buses, EBs require longer refueling times, have a shorter driving range, and rely on charging infrastructure that is often limited in capacity. Additionally, the batteries used in EBs, predominantly lithium-ion batteries (Zhang et al., 2021), are expensive and represent a significant portion of the cost of ownership. To manage this expense and reduce exposure to the high and unpredictable cost of premature battery failure, public transport agencies often lease the batteries from manufacturers or purchase them with long-term warranties (e.g., 10 years). These leasing and warranty contracts typically include specifications regarding battery storage conditions and recommended state-of-charge (SoC) operating ranges. Such guidelines aim to mitigate both loss of capacity due to charging and discharging cycles, known as cyclic aging, and degradation over time during storage, referred to as calendar aging (Pelletier et al., 2017). A common recommendation is to operate batteries within a range, typically 20–80% or 30–80%, during most operations (Kostopoulos et al., 2020; Jiang et al., 2013; Lu et al., 2013). The aim of this work is to develop a methodology that directly integrates these battery management guidelines into the planning of EB schedules.

At the operational planning stage, EB schedules are obtained by solving a vehicle scheduling problem (VSP), which assigns vehicles to timetabled trips so that each trip is covered exactly once. A trip is defined by a start time and location, an itinerary composed of a sequence of stops, and an end time and location. In practice, public transit operators update EB schedules on a monthly to seasonally basis. A common extension is the multi-depot vehicle scheduling problem (MDVSP), where vehicles can be stored overnight in several depots, and the vehicle capacity at each depot is restricted. The VSP and the MDVSP have been extensively studied in the past decades. Several exact and heuristic methods have been proposed, notably those of Ribeiro and Soumis (1994), Löbel (1998), Hadjar et al. (2006), Freling et al. (2001), and Kliewer et al. (2006).

When managing a fleet of EBs, the VSP must be extended to account for charging activities. This extension, referred to as the electric vehicle scheduling problem (E-VSP), incorporates constraints related to driving range and charger capacity. The E-VSP is proven to be NP-hard, even in the single-depot case (Sassi and Oulamara, 2017), and has been widely explored in the literature over the past decade. A detailed review on the (E)-VSP is provided in Desaulniers and Hickman (2007), Perumal et al. (2022), and Zhou et al. (2024b). Moreover, a detailed review of the charge scheduling problem, which constitutes a core subproblem of the E-VSP, is provided in (Dolgui et al., 2025).

Li (2014) is among the first to present a model for the single-depot E-VSP (SD-E-VSP). Their model is suitable for fast-charging or battery-swapping EB fleets, since it assumes that battery service time is fixed. Each charging station is time-expanded, and a time discretization technique is used, so that the capacity of each charging station can be constrained. A column generation-based algorithm is devised, and computational experiments on real-world and randomly generated instances are carried out. In van Kooten Niekerk et al. (2017), two alternative SD-E-VSP models are considered, one with continuous SoC values and the other with discrete SoC values. Additionally, the second model incorporates features such as time-of-day electricity pricing, flexible charging processes, and battery wear costs. A column generation-based algorithm is compared with a faster version with Lagrangian relaxation over four datasets. Wen et al. (2016) and Wang et al. (2021) developed an adaptive large neighborhood search (ALNS) heuristic and a genetic column generation-based algorithm, respectively, for solving the multi-depot electric vehicle scheduling problem (MD-E-VSP). Wu et al. (2022) introduced a bi-objective model for the MD-E-VSP

considering grid characteristics, namely time-of-use pricing and peak load risk. The bi-objective problem is reformulated using a lexicographic method and a branch-and-price heuristic featuring a trip chain pool strategy is devised. Zhang and Yang (2025) introduced a mixed-integer nonlinear programming formulation for the E-VSP with partial charging and battery degradation. A real-life case study is solved using a genetic algorithm. More recently, Gerbaux et al. (2025) proposed a column generation–based heuristic for the MD-E-VSP with piecewise linear charging functions and capacitated charging stations. To reduce the size of the underlying network, the authors restrict the arc set using either a greedy selection procedure or a supervised learning–based approach. Their method achieves average computational time reductions of up to 71.6% compared to the same column generation heuristic without network reduction.

Furthermore, several variants of the E-VSP have been investigated in the literature: E-VSP with crew scheduling (Perumal et al., 2021); E-VSP with timetabling (Xu et al., 2023); E-VSP with charging infrastructure design (Liu and Ceder, 2020; Li et al., 2021; Avishan et al., 2023; He et al., 2023; Najafi et al., 2025; Bazarnovi et al., 2025); E-VSP with a mixed fleet of electric and conventional buses (Olsen and Kliewer, 2020; Rinaldi et al., 2020; Alvo et al., 2021; Şule Yıldırım and Yıldız, 2021; Zhang et al., 2022; Zhou et al., 2024a); E-VSP with time windows (Gkiotsalitis et al., 2023; Jiang et al., 2023; Lu et al., 2025); and E-VSP with trip-shifting strategies (Liu et al., 2024; Duan et al., 2023).

Few studies have addressed the E-VSP under stochastic energy consumption. Bie et al. (2021) proposed a model incorporating stochastic travel time and energy consumption, aiming to minimize delays, energy consumption, and procurement costs. Chance-constraints and robust constraints are included for time and energy feasibility, respectively. They solved the E-VSP using a nondominated sorting genetic algorithm with the elitist strategy. Li et al. (2021) addressed the multi-depot vehicle location-routing-scheduling problem under range uncertainty. They formulated the problem as a two-stage stochastic program, where in the first stage routing, scheduling and location of charging infrastructure under different random range scenarios are generated and, in the second stage, ad hoc service to handle incomplete trips (due to energy shortage) are scheduled. A gradient descent algorithm is used to solve a so-called range reliability-based stochastic reformulation of the original problem. Avishan et al. (2023) proposed a robust optimization approach for the E-VSP under travel time and energy consumption uncertainty. A budgeted uncertainty set is introduced to control the robustness of the solutions, and a Monte Carlo simulation is used to assess the effectiveness of the approach. The problem is solved using a hybrid method combining a genetic algorithm with mathematical optimization techniques. In comparison with Bie et al. (2021), the proposed model additionally accounts for fleet sizing decisions, bus type selection, and charger capacity planning. The E-VSP with stochastic travel time—where energy consumption depends on travel time—is also studied in Tang et al. (2019), Jiang et al. (2021), and Yan et al. (2024). In particular, Yan et al. (2024) proposed an integrated learning and mixed-integer linear programming framework that adaptively reoptimizes EB schedules in response to disruptions such as travel time deviations or electric bus breakdowns.

None of the existing studies on the E-VSP explicitly enforce battery management guidelines using stochastic energy consumption information (i.e., the full distributional information). Compliance with SoC ranges recommended by third parties—such as battery leasing or warranty providers—can be addressed deterministically, either by (i) considering the expected energy consumption across all trips, or (ii) accounting for the worst-case energy consumption (or a robust uncertainty set) and ensuring that the resulting SoC always remains within the recommended ranges (see, e.g., Bie et al., 2021; Avishan et al., 2023). The former, while

simple, may result in frequent violations of the SoC limits due to daily variability in energy usage. The latter is overly conservative and could lead to solutions with high operational costs. Indeed, third-party guidelines usually emphasize staying within the recommended SoC range most of the times, rather than guaranteeing absolute compliance.

Importantly, the battery management system enforces hard constraints that prevent operation in critical SoC regimes—such as extremely low or high charge levels—that can cause irreversible battery damage. The remaining capacity is therefore technically available and safe to use, but frequent operations outside the recommended SoC range may lead to long-term battery degradation.

Our contribution is an approach for the E-VSP that incorporates both uncertain energy consumption and battery management strategies compliant with third-party battery policies. We propose a stochastic model that enforces typical recommended SoC ranges (e.g., 20–80% or 30–80%) by constraining the probability of operating an EB outside these bounds. This probabilistic control enables operators to comply with third-party requirements without excessive conservatism. To ensure a realistic representation of the system, we incorporate essential E-VSP characteristics, including partial en-route charging, nonlinear charging profiles, and limited-capacity charging stations. Furthermore, we develop a tailored branch-and-price heuristic that features stochastic pricing problems and an exact stochastic dominance rule. The results of our method are compared with those of two deterministic benchmarks to assess the value added by accounting for uncertainty in this problem.

The remainder of this paper is organized as follows. Section 2 presents a general definition of the E-VSP and introduces our chance-constrained formulation for the E-VSP with policy-guided battery management and stochastic energy consumption. Furthermore, a method to decompose the chance constraint by vehicle schedule is provided. Section 3 presents our branch-and-price heuristic. Then, in Section 4, we compare our approach to deterministic benchmarks and analyze the trade-off between operational costs and the probability of operating within the recommended SoC range. Section 5 summarizes our results.

2 The stochastic E-VSP with policy-guided battery management

We first define in Section 2.1 a model for the E-VSP in the general case, where battery management strategies are not considered. In Section 2.2, the implemented non-linear charging function approximation is presented. In Section 2.3, we introduce the chance-constrained formulation for the stochastic E-VSP with policy-guided battery management. A method to compute the probability of complying with the recommended SoC range is provided in Section 2.4.

2.1 General case

Let $\mathcal{V}$, $\mathcal{D}$, and $\mathcal{H}$ be a timetable of trips, a set of depots, and a set of charging stations, respectively. For each trip in $\mathcal{V}$, a departure time and location and an arrival time and location are given. Furthermore, the travel time and the energy consumption between any two locations in the EB network—each location being either a starting or ending location of a trip in $\mathcal{V}$, a depot in $\mathcal{D}$, or a charging station in $\mathcal{H}$—are known. The travel time of a trip $i \in \mathcal{V}$ is denoted t_i. Given this information, the E-VSP consists of finding a set of feasible vehicle schedules $\mathcal{S}^*$ that covers exactly once each timetabled trip while respecting the number of available EBs b_d at each depot $d \in \mathcal{D}$ and the number of available chargers g_h at each charging station $h \in \mathcal{H}$.

We model the capacity of each charging station with three additional sets. Let $\mathcal{R}$ be a set of $k+1$ time intervals, each of duration ρ, which partition the planning horizon. Each time interval $r \in \mathcal{R}$ is defined by its beginning and end times, denoted by B_r and E_r, respectively. Let also $\mathcal{H}^C$ and $\mathcal{H}^W$ be two sets of nodes, where each node is associated with a charging station in $\mathcal{H}$ and a time interval in $\mathcal{R}$. The nodes in $\mathcal{H}^C$, referred to as 'charging nodes', represent EBs occupying a charger, while the nodes in $\mathcal{H}^W$, referred to as 'waiting nodes', represent EBs waiting at a charging station. For example, we denote by $h^{c1}_{r5} \in \mathcal{H}^C$ and $h^{w1}_{r5} \in \mathcal{H}^W$ the nodes associated with charging station $h_1 \in \mathcal{H}$ and time interval $r_5 \in \mathcal{R}$, in the charging and the waiting states, respectively.

Our E-VSP model is defined on connection-based networks (Ribeiro and Soumis, 1994) with time-expanded charging and waiting nodes. With every depot $d \in \mathcal{D}$, we associate a network $G^d(V_d, A_d)$ with node set $V_d = \mathcal{V} \cup \{n^d_0, n^d_1\} \cup \mathcal{H}^C \cup \mathcal{H}^W$, where n^d_0 and n^d_1 represent depot d at the beginning and the end of the day, respectively, and arc set A_d. This network contains five types of arcs (i, j), namely *pull-out* arcs (i.e., (n^d_0, i) for $i \in \mathcal{V}$), *pull-in* arcs (i.e., (i, n^d_1) for $i \in \mathcal{V}$), *connection* arcs (i.e., (i, j), for $i, j \in \mathcal{V}$), *charging* arcs (i.e., (i, j), for $i \in \mathcal{H}^C \cup \mathcal{H}^W$, $j \in \mathcal{V} \cup \mathcal{H}^C \cup \mathcal{H}^W \cup \{n^d_1\}$), and *to charge* arcs (i.e., (i, j), for $i \in \mathcal{V}, j \in \mathcal{H}^W$). Figure 1 illustrates an example of such network with 6 trips v_1 to v_6, 1 charging station h_1, and 7 time intervals, r_0 to r_6.

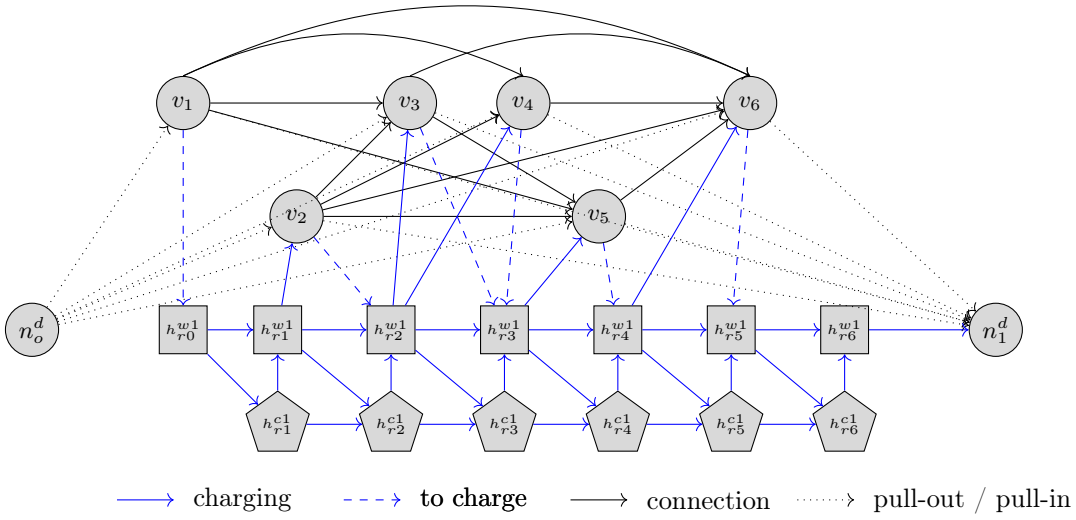


Figure 1: Connection-based network with time-expanded charging and waiting nodes

Two timetabled trips i and j are connected by an arc (i, j) in G^d, for all $d \in \mathcal{D}$, if and only if the departure time d_j of trip j is greater than or equal to the sum of the arrival time of trip i, the deadhead travel time t_{ij} between the locations of i and j, and a minimum layover time of τ minutes required for boarding passengers at the start of trip j. To minimize congestion at the terminals and reduce idle time for drivers, we implement a waiting time threshold of 45 minutes between consecutive trips i and $j \in \mathcal{V}$. If the waiting time exceeds 45 minutes, the vehicle must proceed to the nearest depot after completing trip i. At the depot, it will remain idle until it is time to depart for the starting location of trip j. Since no driver attendance is required at the depot, the connection to trip j remains feasible regardless of the waiting duration. We distinguish four types of charging arcs. First, for each charging station $h \in \mathcal{H}$, all consecutive nodes in $\mathcal{H}^C$ and $\mathcal{H}^W$

associated with h are connected, such that two disjoint chains are formed per charging station. The nodes in $\mathcal{H}^W$ associated with the latest time interval are connected to n_1^d. Second, we connect each node in $\mathcal{H}^W$, except the ones associated with the last time interval, to the node in $\mathcal{H}^C$ associated with the same charging station and the next time interval. Third, an arc connects each node in $\mathcal{H}^C$ to the node in $\mathcal{H}^W$ that belongs to the same time interval and charging station. Fourth, we add an arc between a waiting node $i \in \mathcal{H}^W$ with time interval $r+1$ and a trip $j \in V$ if the latest departure time such that the EB arrives on time for the departure time for the trip falls in r (i.e., $B_r \leq d_j - \tau - t_{ij} < E_r$). Finally, we create an arc 'to charge' between a trip $i \in \mathcal{V}$ and a node $j \in \mathcal{H}^W$ if the earliest EB arrival time at the charging station falls within the corresponding time interval r of the waiting node (i.e., $B_r \leq d_i + t_i + t_{ij} < E_r$).

By definition, a path in the graph G^d is a sequence of pairwise compatible timetabled trips and charging activities. This path is a feasible vehicle schedule if it starts at the source node n_0^d, ends at the sink node n_1^d, and respects energy-feasibility and charging-feasibility constraints. We assume that each vehicle begins its schedule with a SoC of σ^{init}. The energy-feasibility constraints ensure that the SoC of all EBs remains above the minimum allowed SoC, denoted as σ^{min}, throughout the day. Energy consumption values, SoC values, σ^{min}, and σ^{init} are expressed as percentages of the total battery capacity. All values are rounded to the nearest integer, resulting in admissible values within the set $\{0\%, 1\%, \ldots, 100\%\}$. Moreover, charging-feasibility constraints ensure that each time an EB visits a charging station, we schedule exactly one recharging activity that can be composed of one or more consecutive nodes in $\mathcal{H}^C$. These constraints are enforced by including two charging resources in the dynamic programming algorithm, as described in Section 3.1. The total duration of a recharging activity corresponds to the duration of a single time interval multiplied by the number of consecutive nodes in $\mathcal{H}^C$ included in the path. Charging may be partial—the SoC at the end of the activity is a function of the charging time, as detailed in Section 2.2—and the charging process is interrupted if the SoC reaches $\sigma^{\max}$, or a lower upper bound (see next section).

The cost of a vehicle schedule s passing through the set of arcs $A(s)$ is given by

$$c_s = \sum_{(i,j) \in A(s)} c_{ij}, \tag{1}$$

where c_{ij} is the cost of arc (i, j), consisting of a cost per vehicle used, a cost for each minute waiting outside the depot, a cost per unit of distance traveled, and a cost per charging activity. This latter cost is used, on the one hand, to discourage unnecessary visits to the charging stations when the SoC is still high and, on the other hand, to take into consideration operational costs, such as the salary of employees dedicated to each charging station.

2.2 Nonlinear charging profile

The potential risks associated with excessive voltage levels, which can permanently damage the battery, can be mitigated by charging the battery using the constant current and constant voltage (CC-CV) scheme (Pelletier et al., 2017). In the CC phase, the charging current is kept constant until the voltage reaches a threshold. Then, in the CV phase, the voltage is kept constant while the current decreases. These changes in voltage and current regimes make the charging functions nonlinear. In this work, we approximate these nonlinear charging functions with piecewise linear functions, as done in Montoya et al. (2017) and Olsen and Kliewer (2020).

Let ρ be the duration of a time interval and x be the SoC at the beginning of the interval. The SoC after charging for one interval is approximated using the piecewise linear function $\lambda(x, \rho)$. If a path in G^d includes m consecutive charging nodes, the resulting SoC is approximated by $\lambda(x, m\rho)$—charging over a single interval of duration $m\rho$—rather than applying $\lambda(x, \rho)$ iteratively m times. This approach avoids potential cumulative rounding errors. Let $\sigma^{up} \leq \sigma^{max}$ be a SoC bound. The output of $\lambda(x, \rho)$ is either less than σ^{up}—in the case of a partial charge—or equal to σ^{up} when the charging duration exceeds that required for a full charge.

2.3 Chance-constrained formulation

Consider a recommended SoC range, denoted as $[\sigma^{low}, \sigma^{up}]$, where $\sigma^{min} \leq \sigma^{low} \leq \sigma^{up} \leq \sigma^{max}$. To facilitate later references, we will say that a battery is *overused* when its SoC falls below σ^{low}. To manage adherence to this range, we apply a cut-off during charging when the SoC reaches the upper bound σ^{up}, and we impose a chance constraint at the lower bound to limit the probability of the SoC dropping below σ^{low}. Note that, if an EB is allowed to have a SoC below σ^{low}, we strictly enforce that the SoC remains above σ^{min} for energy feasibility.

The chance constraint is evaluated using the probability mass function (PMF) of the energy consumption for each trip $i \in \mathcal{V}$, denoted $e_i(\mu)$, and defined over finite supports. Let $\mathcal{K}$ be a finite set of exogenous scenarios (e.g., weather conditions, traffic congestion level, or temperature). Conditional on k, the random energy consumptions of the trips are assumed independent, with trip-specific PMFs $e_i(\mu \mid k)$. Deterministic energy consumption is assumed for pull-out, pull-in, and deadhead trips, as these trips are typically short and do not involve passengers, thereby eliminating many sources of uncertainty in energy usage. Factors such as passenger loading, heating, and air conditioning, which significantly impact energy consumption, are not present during these trips.

Our model uses the following additional notation, which is summarized in Table 1. Let $\mathcal{S}$ be the set of all feasible vehicle schedules and $\mathcal{S}^d \subset \mathcal{S}$ be the subset of feasible vehicle schedules for depot $d \in \mathcal{D}$. Let also a_{is} be a binary parameter equal to 1 if schedule $s \in \mathcal{S}$ covers trip i, y_s be a binary variable equal to 1 if schedule s is part of the solution, and w_s^{hr} be a binary parameter equal to 1 if schedule s includes a charging activity at station h in time interval r.

The stochastic E-VSP with policy-guided battery management can be formulated as the following chance-constrained integer program:

$$\min \quad \sum_{s \in \mathcal{S}} c_s y_s \tag{2}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} a_{is} y_s = 1, \qquad \forall i \in \mathcal{V} \tag{3}$$

$$\sum_{s \in \mathcal{S}^d} y_s \leq b_d, \qquad \forall d \in \mathcal{D} \tag{4}$$

$$\sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}^d} w_s^{rh} y_s \leq g_h, \qquad \forall h \in \mathcal{H}, r \in \mathcal{R} \tag{5}$$

$$\Pr\{\text{overusing the battery of one EB or more}\} \leq \epsilon \tag{6}$$

Notation	Definition
$\mathcal{V}$	Set of timetabled trips
$\mathcal{D}$	Set of depots
$\mathcal{H}$	Set of charging stations
$\mathcal{H}^C$	Set of charging nodes
$\mathcal{H}^W$	Set of waiting nodes
$\mathcal{R}$	Set of time intervals
$\mathcal{S}$	Set of all feasible vehicle schedules
$\mathcal{S}^d$	Set of all feasible vehicle schedules hosted in depot d
y_s	Binary variable equal to 1 if schedule s is selected
a_{is}	Binary parameter equal to 1 if schedule s covers trip i
w_s^{hr}	Binary parameter equal to 1 if schedule s includes a charging activity at station h in time interval r
τ	Minimum layover time
ρ	Duration of each time interval
σ^{low}	Lower bound on the recommended SoC range
σ^{up}	Upper bound on the recommended SoC range
σ^{init}	Initial SoC at the beginning of the planning period
σ^{min}	Minimum authorized SoC (e.g., 0%)
σ^{max}	Maximum authorized SoC (e.g., 100%)
b_d	Number of available EBs at depot d
g_h	Number of available chargers at charging station h
ϵ	Chance constraint threshold
$e_i(\mu)$	PMF with finite supports of the energy consumption of trip i (relative to the total battery capacity)

Table 1: Overview of the notation of the stochastic E-VSP with policy-guided battery management

$$y_s \in \{0,1\}, \qquad \forall s \in \mathcal{S}. \tag{7}$$

The objective function (2) minimizes the total operational costs, while constraints (3) ensure that each timetabled trip is covered exactly once by a schedule, constraints (4) impose vehicle availability at each depot, and constraints (5) guarantee that for each charging station $h \in \mathcal{H}$, a maximum of g_h chargers are used at the same time. Finally, the chance constraint (6) ensures that the probability of overusing the battery of one EB or more is less than or equal to a maximum threshold ϵ.

We assume conditional independence of battery overuse events across schedules, given common exogenous factors affecting all trips. For each schedule s and scenario k, define $P_s \mid k = \Pr\{\text{schedule } s \text{ remains within the recommended SoC range under scenario } k\}$, and let P_k be the probability of scenario k. Using this notation and the previous conditional independence assumption, we can rewrite constraint (6) as

$$1 - \sum_{k \in \mathcal{K}} \left(P_k \prod_{s \in \mathcal{S}: y_s = 1} (P_s|k) \right) \leq \epsilon, \tag{8}$$

which, by extending the summation to all vehicle schedules, is equivalent to

$$\sum_{k \in \mathcal{K}} \left(P_k \prod_{s \in \mathcal{S}} (P_s|k)^{y_s} \right) \geq 1 - \epsilon. \tag{9}$$

Due to the lack of reliable conditional data, we restrict attention to a single representative scenario k^*

(e.g., a typical winter weekday). Although the multi-scenario formulation in (9) can be extended with moderate algorithmic adaptations, the methodology developed in the remainder of this manuscript is presented exclusively for the single-scenario case. In the single-scenario case, (9) simplifies to

$$\prod_{s\in\mathcal{S}}(P_s|k^*)^{y_s} \geq 1-\epsilon, \tag{10}$$

which, by the properties of the logs, is equivalent to

$$\sum_{s\in\mathcal{S}} y_s ln(P_s|k^*) \geq ln(1-\epsilon). \tag{11}$$

Let $\beta_s = \ln(P_s|k^*)$ and $\beta = \ln(1-\epsilon)$. Then, we have

$$\sum_{s\in\mathcal{S}} y_s\beta_s \geq \beta. \tag{12}$$

Constraint (12) is a linear transformation of constraint (6), with the complexity transferred to the β_s parameters. These parameters are computed dynamically within the solution approach outlined in Section 3, using the recursive method detailed below. In the remainder of the manuscript, we assume that energy consumption distributions are conditioned on a single representative scenario k^* and, for notational simplicity, omit the explicit reference to k^*.

2.4 Computing the probability of overusing a battery

The charging activities are interrupted if the SoC reaches σ^{up} and EBs begin their schedules at σ^{init}. Therefore, for every EB schedule, the probability of observing a battery with a SoC greater than σ^{up} is null. In what follows, we explain how the probability of observing a SoC less than σ^{low} can be computed.

Consider a schedule $s = (0, 1, \ldots, n, n+1)$ where 0 and $n+1$ are the nodes n_0^d and n_1^d associated with a depot $d \in \mathcal{D}$, respectively, and $1, \ldots, n$ are other nodes in V_d. Let X_i^s be the SoC at the end of node i in schedule s and denote by $\mathcal{O}_i^s$ the event that no battery overuse has occurred in schedule s up to node i. We define $f_i^s(x) = \mathbb{P}(X_i^s = x \wedge \mathcal{O}_i^s)$ the joint probability that the SoC equals x at node i and that the schedule s has not incurred any battery overuse up to node i. Note that $f_i^s(x)$ is a joint, non-normalized PMF satisfying $\sum_x f_i^s(x) = \mathbb{P}(\mathcal{O}_i^s) \leq 1$, where the inequality is strict whenever battery overuse has positive probability prior to node i. We can compute $f_i^s(x)$ recursively with $f_0^s(\sigma^{init}) = 1$ as

$$f_i^s(x) = \begin{cases} \sum_{\mu=0}^{\sigma^{up}-x-\iota_{i-1,i}} e_i(\mu) f_{i-1}^s(x+\mu+\iota_{i-1,i}) & \text{if } i \in \mathcal{V} \text{ and } x \geq \sigma^{low} \\ f_{i-1}^s(\lambda^{-1}(x,\rho)) & \text{if } i \in \mathcal{H}^C \\ f_{i-1}^s(x+\iota_{i-1,i}) & \text{if } i \in \mathcal{H}^W \cup \{n_1^d\} \text{ and } x \geq \sigma^{low} \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

where $\lambda^{-1}(x,\rho)$ is the inverse of the charging function and outputs the initial SoC such that the final SoC after a charge of ρ minutes is x and $\iota_{i-1,i}$ is the deterministic energy consumption between nodes $i-1$ and i due to deadheading, pulling in, or pulling out. Observe that if $\sum_{\mu=0}^{\sigma^{up}} e_i(\mu) < 1$ (i.e., the energy consumption of trip i exceeds the allowed battery capacity with non-zero probability), the stochastic E-VSP with policy-

guided battery management becomes infeasible. Therefore, this case is excluded from consideration. The probability of staying within the recommended SoC range in schedule s is thus

$$P_s = \sum_{x=\sigma^{low}}^{\sigma^{up}} f_{n+1}^s(x) = F_{n+1}^s(\sigma^{up}), \tag{14}$$

where $F_i^s(x)$ is the cumulative distribution function (CDF) of $f_i^s(x)$, computed as

$$F_i^s(x) = \sum_{x'=\sigma^{low}}^{x} f_i^s(x'), \qquad x = \sigma^{low}, \sigma^{low}+1, \ldots, \sigma^{up}. \tag{15}$$

Although equation (14) only involves node $i = n+1$, we define F_i^s for an arbitrary node i as it is used later in the labeling algorithm (see Section 3.1).

3 Heuristic branch-and-price algorithm

The linear relaxation of the E-VSP formulated as a set partitioning problem provides a tight lower bound but involves a large number of decision variables, one for each EB schedule. Therefore, enumerating all possible schedules is neither practical nor efficient. To address this, a column generation algorithm (see, e.g., Desaulniers et al., 2005; Lübbecke and Desrosiers, 2005) is integrated into the branch-and-bound algorithm to dynamically generate variables as needed. This combined approach is known as branch-and-price (Barnhart et al., 1998; Costa et al., 2019).

The framework of our stochastic branch-and-price algorithm is illustrated in Figure 2. The branch-and-bound framework handles the initialization of the root node (i.e., the linear relaxation of (2)-(7), hereafter referred to as the *master problem* (MP)), the management of the list of active nodes, the update of the upper bound (UB), and the handling of the branching strategy and perturbation method. We adopt a heuristic branching strategy to obtain feasible solutions within a reasonable timeframe, which makes the overall framework heuristic. The UB is updated when an integer solution to the restricted MP (RMP)—the MP limited to a subset $\mathcal{S}' \subseteq \mathcal{S}$ of the schedule variables y_s—with a cost less than the current UB is found. Each node in the branch-and-bound tree is solved to optimality using a tailored column generation algorithm.

In each iteration of the column generation algorithm, a RMP is solved. Given the dual solution to the RMP, a pricing problem is then solved to generate new columns. For the stochastic E-VSP with policy-guided battery management, the pricing problem is separable by depot and consists in solving a shortest path problem with stochasticity (Boland et al., 2015; Wellman et al., 2013) in G^d for each $d \in \mathcal{D}$, where arc costs are adjusted based on the most recent dual solution of the RMP. If variables with negative reduced costs are identified, they are added to $\mathcal{S}'$, triggering a new iteration of the column generation algorithm. If no such variables are found, the column generation algorithm terminates, and the current RMP solution is guaranteed to be optimal for the MP of the corresponding branch-and-bound node.

Details of the stochastic dynamic programming algorithm used to solve the pricing problems, including label components, extension functions, and dominance rules, are provided in Section 3.1. The branching strategy and perturbation method employed to enhance computational efficiency are discussed in Sections 3.2 and 3.3, respectively.

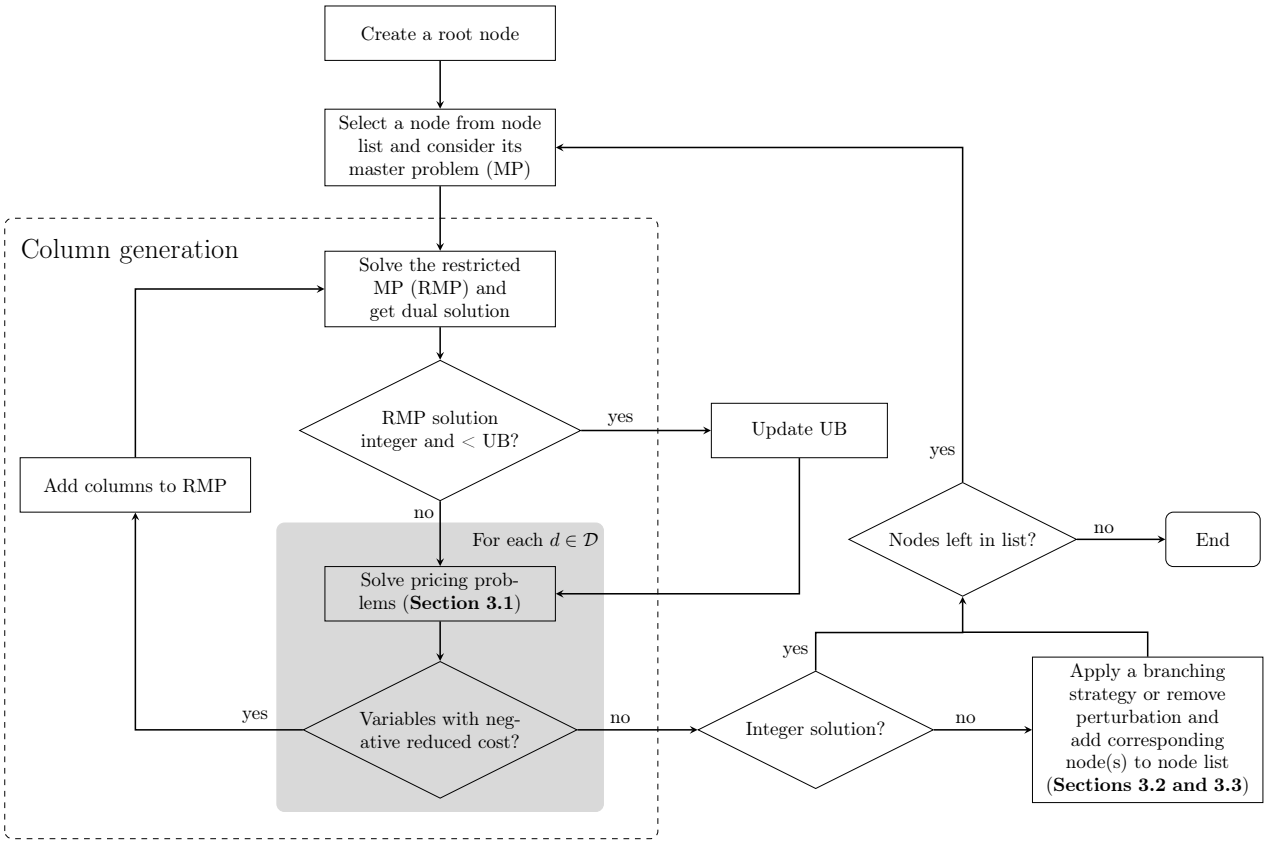


Figure 2: Flowchart of the branch-and-price algorithm (from Ricard et al., 2024)

3.1 Stochastic pricing problems

Let $(u_i)_{i\in\mathcal{V}}$, $(\pi_d)_{d\in\mathcal{D}}$, $(\alpha^{hr})_{h\in\mathcal{H},r\in\mathcal{R}}$, and θ be the dual variables associated with constraints (3), (4), (5), and (6), respectively, where the dual variables $(u_i)_{i\in\mathcal{V}}$ are unrestricted in sign and all the others are nonnegative.

A feasible schedule is a path in G^d starting at node n_0^d and ending at node n_1^d. The reduced cost $\tilde{c}_s$ of schedule s is therefore given by

$$\tilde{c}_s = c_s - \sum_{i\in\mathcal{V}} a_{is}u_i - \pi_d - \sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}} w_s^{hr}\alpha^{hr} - \theta \ln P_s. \tag{16}$$

The component of the reduced cost associated with the chance constraint (6), i.e., $\theta \ln P_s$, is not decomposable by arc. This is because the probability of overusing the battery in schedule s is only revealed once the entire schedule is constructed. Hence, we add this cost to the last arc of a schedule, i.e., the pull-in arc. As a result, the modified arc costs $\tilde{c}_{ij}^s$ are schedule-dependent. The definition of $\tilde{c}_{ij}^s$ for each arc $(i,j) \in V_d$ and vehicle schedule is reported in Table 2.

Each path is constructed using a dynamic programming approach, specifically a stochastic labeling algorithm (Wellman et al., 2013; Boland et al., 2015; Ricard et al., 2024). The algorithm operates as follows: starting from an initial label at the source node n_0^d, labels are propagated through the network using re-

Name	Start node i	End node j	$\tilde{c}^s_{ij}$
Pull-out	$i = n^d_0$	$j \in \mathcal{V}$	$c^s_{ij} - \pi_d$
Connection	$i \in \mathcal{V}$	$j \in \mathcal{V} \cup \mathcal{H}^W$	$c^s_{ij} - u_i$
Charging (from charg.)	$i = h^c_r \in \mathcal{H}^C$	$j \in \mathcal{H}^C \cup \mathcal{H}^W$	$c^s_{ij} - \alpha^{hr}$
Charging (from wait.)	$i \in \mathcal{H}^W$	$j \in V \cup \mathcal{H}^C \cup \mathcal{H}^W$	c^s_{ij}
To charge	$i \in \mathcal{V}$	$j \in \mathcal{H}^W$	c^s_{ij}
Pull-in (from terminal)	$i \in \mathcal{V}$	$j = n^d_1$	$c^s_{ij} - u_i - \theta \ \ln P_s$
Pull-in (from wait.)	$i \in \mathcal{H}^W$	$j = n^d_1$	$c^s_{ij} - \theta \ \ln P_s$

Table 2: Modified arc costs

source extension functions to construct partial paths. This process continues until the sink node is reached, where the paths represent complete schedules. Each label encodes the necessary information to evaluate the feasibility of any of its extension and to compute the contribution of the chance constraint to the reduced cost on the pull-in arc at the end of the schedule. Infeasible paths are discarded as soon as they are identified as such. To improve efficiency, a dominance procedure is applied to compare labels, eliminating non-Pareto-optimal labels. The main components of the stochastic labeling algorithm used in our tailored branch-and-price algorithm, namely the labeling procedure and the dominance rule, are detailed next.

3.1.1 Labeling procedure

Each label encodes a representation of the probability of overusing the battery of the EB, the SoC level, the worst-case SoC, the accumulated reduced cost, and two additional resources. To compute the probability of overusing the battery of an EB along a path p ending at node i and its extensions, it is necessary to store $F^p_i(x)$ for integer values of x between σ^{low} and σ^{up}. Simply relying on the CDF of the SoC is insufficient because it lacks memory. By definition, $F^p_i(x)$ incorporates this memory; it is the joint CDF of the SoC and the probability that the EB following path p has not yet experienced battery overuse. The two additional resources are for planning feasible charging activities. The first, R^p_i, represents the number of completed charging activities since the last timetabled trip in p. It ensures that at most one recharge occurs for each visit at a charging station. The second, E^p_i, tracks the number of times an EB has waited at a charging station without being recharged since the last timetabled trip in p. This resource enforces that a recharge occurs during such a visit. Both resources are initialized to 0.

A label is represented as $L^{p'}_i = (F^{p'}_i(\sigma^{low}), \ldots, F^{p'}_i(\sigma^{up}), \omega^{p'}_i, C^{p'}_i, R^{p'}_i, E^{p'}_i)$, where $\omega^{p'}_i$ is the worst-case SoC for path p', which starts at the origin node n^d_0 and ends at node i, and $C^{p'}_i$ is the accumulated reduced cost for path p'. The extension of $L^{p'}_i$ along the arc $(i,j) \in A_d$ is done by applying the following extension functions to create a new label L^p_j at node j.

Let σ^p_i be the SoC at node i of path p'. Assuming each EB starts the day at σ^{init} (i.e., $\sigma^p_i = \sigma^{init}$ for $i = n^d_0$), the extension function to compute $F^p_j(x)$ based on $F^{p'}_i(x)$, for $x = \sigma^{low}, \sigma^{low}+1, \ldots, \sigma^{up}$, is derived from Equations (13) and (15), and given by

$$F_j^p(x) = \begin{cases} \sum_{\mu=0}^{\sigma^{up}-x-\iota_{i,j}} e_j(\mu) F_i^{p'}(x+\mu+\iota_{i,j}) & \text{if } j \in \mathcal{V} \\ F_i^{p'}(\lambda^{-1}(x,\rho)) & \text{if } j \in \mathcal{H}^C \\ F_i^{p'}(x+\iota_{i,j}) & \text{if } j \in \mathcal{H}^W \cup \{n_1^d\} \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Furthermore, assuming $\omega_i^p = \sigma^{init}$ for $i = n_0^d$, ω_j^p can be computed using

$$\omega_j^p = \begin{cases} \omega_i^{p'} - \max\{\mu | e_j(\mu) > 0\} - \iota_{i,j} & \text{if } j \in \mathcal{V} \\ \lambda(\omega_i^{p'}, \rho) & \text{if } j \in \mathcal{H}^C \\ \omega_i^{p'} - \iota_{i,j} & \text{otherwise,} \end{cases} \tag{18}$$

where $\max\{\mu | e_j(\mu) > 0\}$ outputs the largest energy consumption value with positive probability. The accumulated reduced cost C_j^p is updated as

$$C_j^p = C_i^{p'} + \tilde{c}_{ij}^p, \tag{19}$$

where $\tilde{c}_{ij}^p$ is the modified cost of arc (i,j) in path p as defined in Table 2. Note that the subscripts s and p, referring to a vehicle schedule and a path, respectively, are used interchangeably. Lastly, the resources $R_i^{p'}$ and $E_i^{p'}$ are extended as

$$R_j^p = \begin{cases} R_i^{p'} + 1 & \text{if } i \in \mathcal{H}^W \text{ and } j \in \mathcal{H}^C \\ R_i^{p'} - 1 & \text{if } i \in \mathcal{H}^W \text{ and } j \in \mathcal{V} \\ R_i^{p'} & \text{otherwise,} \end{cases} \tag{20}$$

and

$$E_j^p = \begin{cases} E_i^{p'} + 1 & \text{if } i \in \mathcal{V} \text{ and } j \in \mathcal{H}^W \\ E_i^{p'} - 1 & \text{if } i \in \mathcal{H}^W \text{ and } j \in \mathcal{H}^C \\ E_i^{p'} & \text{otherwise.} \end{cases} \tag{21}$$

Figure 3 illustrates an example of the extension of the R_i^p and E_i^p resources along the partial path $(v_1, h_{r0}^{w1}, h_{r1}^{w1}, h_{r2}^{c1}, h_{r3}^{c1}, h_{r3}^{w1}, v_5)$. Updated values are reported in bold. Note that these two resources are set to zero outside charging activities.

To guarantee energy feasibility, a path p ending at node i is discarded if there is a positive probability of running out of energy, specifically if $\omega_i^p < \sigma^{min}$. Additionally, any path for which the chance constraint (6) would be automatically violated—whether for the path itself or any of its potential extensions—is excluded. This occurs when $\ln(F_i^p(\sigma^{up})) < \ln(1-\epsilon)$. Furthermore, a path p ending at node i is discarded if the resource values E_i^p or R_i^p exceed their respective resource windows. These windows require E_i^p and R_i^p to be less than or equal to 0 if $i \in \mathcal{V}$ and less than or equal to 1 otherwise.

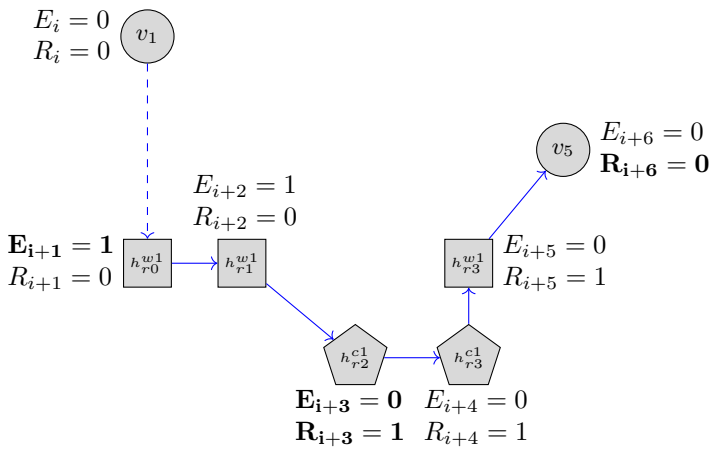


Figure 3: Example illustrating the evolution of two resource values for feasible charging activities along a partial path

3.1.2 Label dominance

Consider two paths p_1 and p_2, both ending at node i. Path p_1 dominates path p_2, and thus p_2 can be discarded, when the following conditions hold:

(i) $C_i^{p_1} \leq C_i^{p_2}$,

(ii) $R_i^{p_1} \leq R_i^{p_2}$,

(iii) $E_i^{p_1} \leq E_i^{p_2}$,

(iv) $\omega_i^{p_1} \geq \omega_i^{p_2}$, and

(v) $F_i^{p_1}(\sigma^{up}) - F_i^{p_1}(x-1) \geq F_i^{p_2}(\sigma^{up}) - F_i^{p_2}(x-1)$, for all $x \in \{\sigma^{low}, \sigma^{low}+1, \ldots, \sigma^{up}\}$,

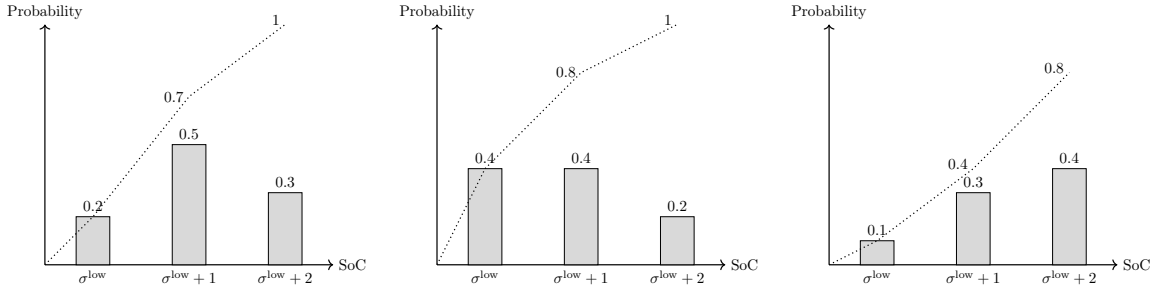
where $F_i^{p_1}(\sigma^{up}) = \mathbb{P}(\mathcal{O}_i^{p_1})$, the probability that no battery overuse has occurred along path p.

The first four conditions are standard in shortest path problems with resource constraints. Condition (v), however, is a dominance rule tailored to our problem. When $x = \sigma^{low}$, we have $F_i^{p_1}(\sigma^{low}-1) = F_i^{p_2}(\sigma^{low}-1) = 0$ by definition. Thus, condition (v) compares $\mathbb{P}(\mathcal{O}_i^{p_1})$ and $\mathbb{P}(\mathcal{O}_i^{p_2})$. It ensures that paths with a higher probability of maintaining the SoC within the recommended range are preferred. For $x > \sigma^{low}$, condition (v) compares the complementary CDFs of $F_i^{p_1}$ and $F_i^{p_2}$, thereby favoring paths with a higher probability of larger SoC values. Higher SoC levels are desirable because they reduce the risk of dropping below σ^{low} as the path is extended. Formally, these complementary CDFs represent the probability that the SoC at node i is at least x and that no battery overuse has occurred up to node i, that is, $F_i^{p_1}(\sigma^{up}) - F_i^{p_1}(x-1) = \mathbb{P}(X_i^{p_1} \geq x \wedge \mathcal{O}_i^{p_1})$. The following proposition formalizes the validity of condition (v).

Proposition 1. *(Stochastic dominance). Let $G_i^p(x) := F_i^p(\sigma^{up}) - F_i^p(x-1)$ for $x \in \{\sigma^{low}, \sigma^{low}+1, \ldots, \sigma^{up}\}$. If two labels satisfy $G_i^{p_1}(x) \geq G_i^{p_2}(x)$ for all $x \in \{\sigma^{low}, \sigma^{low}+1, \ldots, \sigma^{up}\}$, then for any common feasible extension ξ ending at node j, the extended labels satisfy $G_j^{p_1 \oplus \xi}(x) \geq G_j^{p_2 \oplus \xi}(x)$ for all $x \in \{\sigma^{low}, \sigma^{low}+1, \ldots, \sigma^{up}\}$.*

Proof. See Appendix A.

An illustration of dominance condition (v) is provided in Figure 4. We compare the PMFs f_i^p (bar charts) and CDFs F_i^p (dotted lines) for paths p_1, p_2, and p_3, considering a simplified setting in which each path has positive probability mass only for $x \in \{\sigma^{low}, \sigma^{low}+1, \sigma^{low}+2\}$. First, path p_1 dominates path p_2 with respect to condition (v) because $\mathbb{P}(X_i^{p_1} \geq x \wedge \mathcal{O}_i^{p_1}) \geq \mathbb{P}(X_i^{p_2} \geq x \wedge \mathcal{O}_i^{p_2})$ for all $x \in \{\sigma^{low}, \sigma^{low}+1, \ldots, \sigma^{up}\}$. In both cases, $F_i^{p_1}(\sigma^{up}) = F_i^{p_2}(\sigma^{up}) = 1$, which implies that the probability of battery overuse is zero. At $x = \sigma^{low}+1$, the probabilities $\mathbb{P}(X_i^{p_1} \geq x \wedge \mathcal{O}_i^{p_1})$ and $\mathbb{P}(X_i^{p_2} \geq x \wedge \mathcal{O}_i^{p_2})$ are 0.8 and 0.6, respectively; and at $x = \sigma^{low}+2$, they are 0.3 and 0.2, respectively. Hence, p_1 dominates p_2 because, for every SoC value, it yields a higher probability that the SoC is at least this value while avoiding battery overuse. Second, paths p_1 and p_3 are incomparable. Indeed, $F_i^{p_3}(\sigma^{up}) = 0.8$, which implies a 0.2 probability of battery overuse, whereas this probability is 0 for p_1. Moreover, at $x = \sigma^{low}+2$, $\mathbb{P}(X_i^{p_1} \geq x \wedge \mathcal{O}_i^{p_1}) = 0.3$ is not greater or equal to $\mathbb{P}(X_i^{p_3} \geq x \wedge \mathcal{O}_i^{p_3}) = 0.4$. Consequently, neither path dominates the other.



(a) $f_i^{p_1}$ and $F_i^{p_1}$ for path p_1 (b) $f_i^{p_2}$ and $F_i^{p_2}$ for path p_2 (c) $f_i^{p_3}$ and $F_i^{p_3}$ for path p_3

Figure 4: Example of dominance condition (v)

Finally, although condition (v) relies on probability distributions, it remains exact. This is because the algorithm propagates the full distributions and applies the dominance rules by explicitly accounting for all possible realizations.

3.2 Branching strategy

To limit the time spent exploring the branch-and-bound nodes and to identify integer solutions within a reasonable timeframe, we implement a heuristic branching strategy, namely a diving heuristic. First, we branch on the total number of vehicles used by creating one or two child nodes with upper and lower bounds on the depot capacity. Second, we apply three rounding strategies: (i) rounding schedule variables, (ii) rounding *connection* arcs, and (iii) rounding any arc (including *to charge* and *charging* arcs).

When applying strategy (i), a single branch-and-bound node is created by fixing one or more schedule variables to 1. Variables with the largest fractional values are prioritized for rounding. At each node, a maximum of three variables with fractional parts greater than or equal to 0.99 are selected. If fewer than three variables meet this threshold, all eligible variables are selected. If no variables exceed this threshold, the variable with the largest fractional part (below 0.99) is chosen. The threshold and the number of variables selected were determined empirically during a preliminary experimental campaign.

The same procedure is used for strategies (ii) and (iii), except that arcs are fixed instead of variables. Fixing an arc (i, j) to 1 is achieved by removing all arcs (i, k) and $(k, j) \in A_d$, $d \in \mathcal{D}$, such that $k \in V_d \setminus \{i, j\}$.

Once the branching on the total number of vehicles has been performed, subsequent branching decisions rely on the rounding strategies (i)–(iii). At each node, the strategy with the highest score is chosen. To prioritize strategies (i) and (ii), the score of strategy (iii) is multiplied by 0.7, while the scores of strategies (i) and (ii) are left unchanged (weight 1). The score is defined as the largest fractional value among the selected variables or arcs in strategies (i) and (ii). For strategy (iii), the score is defined as the largest fractional value among the selected arcs.

3.3 Constraint perturbation

E-VSPs formulated as a set partitioning problem are highly degenerate, and this degeneracy generally increases with the time horizon considered or, equivalently, the average number of trips per schedule (see Oukil et al., 2007; Benchimol et al., 2012). We use the simple constraint perturbation strategy of Charnes (1952) to reduce the degeneracy in our experiments. Let η_i^+ and η_i^- be perturbation variables that allow an under- and over-covering of trip $i \in \mathcal{V}$ of up to ξ_i^+ and ξ_i^-, respectively. The perturbed MP is given by

$$\min \quad \sum_{s \in \mathcal{S}} c_s y_s + \sum_{i \in \mathcal{V}} (\delta_i^+ \eta_i^+ + \delta_i^- \eta_i^-) \tag{22}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} a_{is} y_s + \eta_i^+ - \eta_i^- = 1, \qquad \forall i \in \mathcal{V} \tag{23}$$

$$(4), (5), (12),$$

$$0 \leq \eta_i^+ \leq \xi_i^+, \qquad \forall i \in \mathcal{V} \tag{24}$$

$$0 \leq \eta_i^- \leq \xi_i^-, \qquad \forall i \in \mathcal{V} \tag{25}$$

$$0 \leq y_s \leq 1, \qquad \forall s \in \mathcal{S}, \tag{26}$$

where δ_i^+ and δ_i^- are the penalties for under- and over-covering trip $i \in \mathcal{V}$, respectively.

4 Experimental results

We evaluate our approach using simulated instances of one EB line in the city of Montréal, which consists of 42 stops covering a distance of 8.5 kilometers and operating in two directions. Random instances with approximately 60, 150, and 250 trips are generated using the procedure described in Brasseur (2022). The original timetable is adjusted by randomly modifying the headway between trips and ensuring that the headway remains constant for trips within the same hour. Furthermore, the relative frequency of the EB line during each hour is preserved to ensure a realistic distribution of trips throughout the 19-hour operating period from 5:00 AM to midnight. Each simulated timetable contains slightly shifted departure times and a number of trips that may vary but remains close to a target value (e.g., 60, 150, or 250).[1]

We conduct five tests (each using a slightly different simulated instance) for each family of instances, denoted as I1, I2, and I3. Table 3 presents the average characteristics of the families of instances I1, I2, and

[1] Our simulated instances and detailed results are available at `https://github.com/learicard/evsp-policy-battery.git`

I3, namely the instance family name (Instance), the average number of timetabled trips ($|\mathcal{V}|$), the average number of arcs ($|A|$), and the number of chargers per charging station (Char. station capacity). In our tests, we address the single-depot and single-charging station case (i.e., $|\mathcal{D}| = 1$ and $|\mathcal{H}| = 1$), although the formulation and algorithm presented are designed for the multi-depot, multi-charging station cases.

Table 3: Average properties of the families of instances I1 - I3

Instance	$\lvert\mathcal{V}\rvert$	$\lvert A\rvert$	Char. station capacity
I1	60	2,190	1
I2	155	11,900	2
I3	248	29,499	3

We assume that the energy consumption rate (kWh/km) follows a truncated and discretized normal distribution, consistent with findings from previous studies (see Bie et al., 2021; Abdelaty et al., 2021). For each trip in $\mathcal{V}$, the mean of the energy consumption rate distribution is sampled from an exponential distribution $\text{Exp}(1.57, 0.26)$, where 1.57 and 0.26 represent the location and scale parameters, respectively. This distribution has a mean value of 1.83 kWh/km. The variance of the normal distribution is determined by sampling a value from a uniform distribution $\text{U}(0.35, 0.5)$. The parameters of the exponential and uniform distributions are calibrated using data from Basma et al. (2020). Then, the PMFs are constructed by sampling a large number of realizations (100,000 samples in our experiments) from the underlying normal distributions and forming a histogram over the admissible interval (i.e., $[0, \sigma^{up}]$), using bins defined at 1% SoC increments. All sampled values below 0 or above σ^{up} are discarded, and the resulting histogram is normalized so that the PMF sums to 1.

The following model parameters are used in our tests. The costs per vehicle used, per minute of travel (excluding the travel time of timetabled trips), per minute of waiting outside the depot, and per charging activity are set to 1,000, 0.4, 0.2, and 10, respectively. The penalties for under- and over-covering trip $i \in \mathcal{V}$, δ_i^+ and δ_i^-, are set to 1 for every trip $i \in \mathcal{V}$ and the upper bounds ξ_i^+ and ξ_i^- are randomly sampled from the uniform distribution $U(0, 0.1)$ for every trip. We consider 12-meter single-deck EBs with 300 kWh batteries, the same type of EBs as those studied in Basma et al. (2020) to ensure consistency, and time intervals of $\rho = 15$ minutes. Fast chargers with an approximate power output of 450 kW are used. These chargers deliver 7.5 kWh per minute for SoC levels between 0% and 80%, 6 kWh per minute for SoC levels between 80% and 90%, and 3.75 kWh per minute for SoC levels between 90% and 100%. Consequently, the chargers can fully charge a 300 kWh battery in 45 minutes. For each test, we set $\sigma^{init} = \sigma^{up}$.

We conduct our experiments on a Linux machine equipped with 16 Intel Xeon ES-2637 v4 processors running at 3.50 GHz and a RAM of 125 GB. The branch-and-price algorithm is implemented using the GENCOL library, version 4.5, and all linear programs are solved by the commercial solver CPLEX 22.1.

The remainder of this section is organized as follows. Section 4.1 describes how the two deterministic benchmark approaches are implemented for comparison. In Section 4.2, we analyze the algorithm behavior on I1, I2, and I3 for the deterministic and stochastic approaches. Finally, Section 4.3 presents and compares the results for the two approaches (deterministic and stochastic) in terms of operational costs and the probability

of daily battery overuse.

4.1 Benchmark approaches

We benchmark our stochastic approach against two alternative deterministic methods. The first, referred to as the *optimistic* approach, assumes that the energy consumption for each trip is fixed and equal to the expected value of its PMF. The second, called the *worst-case* approach, assumes that the energy consumption for each trip is fixed and equal to the worst-case value (i.e., the upper bound of the finite supports of its PMF).

In both benchmarks, energy consumption is treated deterministically, and battery overuse is controlled by enforcing SoC $\geq \sigma^{\text{low}}$ for all trips. Within our framework, this is equivalent to setting $\sigma^{\text{low}} = \sigma^{\min}$ or, equivalently, $\epsilon = 0\%$. As a result, the chance constraint (12) becomes redundant. The PMFs of the energy consumption that are used to compute the probability of battery overuse are replaced by single values: either the expected or the worst-case energy consumption for each trip.

The optimistic benchmark is expected to yield solutions with lower operational costs but may frequently violate the recommended SoC range when actual energy consumption exceeds expected values. In contrast, the worst-case benchmark, based on the most conservative assumptions, guarantees strict adherence to the recommended SoC range, albeit potentially at the expense of higher costs. These two benchmarks serve as baseline references for the comparative analysis that follows.

4.2 Heuristic performance

Table 4 summarizes the heuristic performance of our algorithm. The left-hand side ('Deterministic E-VSP') presents results for the deterministic benchmarks, while the right-hand side ('Stochastic E-VSP') displays results for the stochastic E-VSP with policy-guided battery management. The table reports the average relative difference in percentage between the UB and the lower bound (Gap), the number of branching nodes explored (BBn), and the computing times (CPU time) in seconds, including the total CPU time (Total), the time to solve the root node (Root), and the time to solve the pricing problems (Pricing). Note that the proportions of time spent solving the root node and the pricing problems do not add up to 100%, as part of the time spent on the root node includes the resolution of pricing problems. For the deterministic benchmarks, these metrics are averaged over two benchmark approaches, five tests per instance family, and two recommended SoC ranges, namely 20–80% and 30–80%. For the stochastic case, the metrics are averaged over five tests per instance family, 11 ϵ values, where $\epsilon \in \{0.001, 0.005, 0.01, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40, 0.50\}$, and the two same recommended SoC ranges. This amounts to a total of 110 tests per instance family for the stochastic case.

Our branch-and-price heuristic achieves near-optimal solutions for both the deterministic and stochastic approaches of the E-VSP, with average optimality gaps of 0.03% and 0.04%, respectively. Specifically, the gaps remain small, with the highest observed values being 0.05% for the deterministic case and 0.06% for the stochastic case. The majority of the CPU time is spent solving the pricing problems. In the deterministic E-VSP, between 74.7% and 87.0% of the total CPU time is spent on pricing on average, while in the stochastic E-VSP, this proportion increases to between 90.9% and 96.6%, depending on the instance family. This reflects the fact that the overall computational complexity of the proposed method is dominated by the

Table 4: Average heuristic performance of the deterministic E-VSP benchmarks and the stochastic E-VSP with policy-guided battery management

	Deterministic E-VSP					Stochastic E-VSP				
			CPU times (s)					CPU times (s)		
Inst.	Gap (%)	BBn	Total	Root (%)	Pric. (%)	Gap (%)	BBn	Total	Root (%)	Pric. (%)
I1	0.01	56	2.9	52.5	87.0	0.01	73	11.5	59.3	90.9
I2	0.03	149	71.1	44.6	78.1	0.05	443	1,703.8	50.0	96.4
I3	0.05	1,582	2,483.0	8.0	74.7	0.06	1,314	23,911.5	26.4	96.6
avg.	**0.03**	**69**	**852.3**	**35.0**	**79.9**	**0.04**	**172**	**8,542.3**	**45.2**	**94.6**

pricing problems, whose size grows with the number of trips, charging intervals, and discretized SoC levels. Compared to the deterministic E-VSP, the stochastic approach increases the dimensionality of the label representation. The time spent solving the root node varies significantly across instance families, ranging from 8.0% to 59.3% of the total CPU time on average. Instance I3 exhibits the lowest proportion of time allocated to solving the root node on average, likely because it also has the highest average number of nodes explored, at 1,582 and 1,314 in the deterministic and stochastic approaches, respectively. Additionally, we observe that the total CPU time increases significantly with the instance size. For instance, the time grows from 2.9 seconds (I1) to 2,483.0 seconds (I3) in the deterministic case and from 11.5 seconds (I1) to 23,911.5 seconds (I3) in the stochastic case. The growth rate is more pronounced for the stochastic E-VSP with policy-guided battery management compared to the deterministic benchmarks, highlighting the added complexity introduced by uncertainty in the stochastic model.

4.3 Comparison of the deterministic benchmarks and the stochastic E-VSP with policy-guided battery management

Table 5 reports the results of the two deterministic benchmarks—the worst-case and the optimistic—across the three instance families. Each entry represents the average of five tests per family. The columns include the instance family name (Inst.), the deterministic benchmark, and, for the recommended SoC ranges of 20–80% and 30–80%, the corresponding operational costs (Op. costs) and the average number of EBs used (# EBs).

Across all families of instances, the worst-case benchmark consistently yields larger operational costs and requires more EBs than the optimistic benchmark. For example, in I3 with the 20–80% SoC range, the worst-case benchmark generates solutions with average operational costs of 12,157.7 and average usage of 11.4 EBs, while the optimistic benchmark is at 11,609.8 with 11.0 EBs. Furthermore, the operational costs are higher for the 30–80% SoC range compared to the 20–80% range because the narrower operating range limits available energy resources. For instance, in I3 for the worst-case benchmark, the operational costs for the 20–80% range are 12,157.7, compared to 12,782.3 for the 30–80% range.

Our approach finds solutions that lie between these two extremes: the optimistic and worst-case benchmarks. These solutions should be better aligned with the requirements of battery leasing companies or

Table 5: Results of the deterministic benchmarks

		20–80% range		30–80% range	
Inst.	Benchmark	Op. costs	# EBs	Op. costs	# EBs
I1	Worst-case	4,102.2	3.8	4,312.7	4.0
	Optimistic	3,482.0	3.2	3,689.8	3.4
I2	Worst-case	8,313.6	7.8	8,540.2	8.0
	Optimistic	7,459.9	7.0	7,891.1	7.4
I3	Worst-case	12,157.7	11.4	12,782.3	12.0
	Optimistic	11,609.8	11.0	11,663.2	11.0

warranty providers than the optimistic benchmark, while remaining less conservative and less costly than the robust worst-case benchmark.

Table 6 presents the results of our approach, which balances operational costs with the threshold on the probability of not complying with the recommended SoC range in one or more vehicle schedules. Each value in the table represents the average over five tests conducted for each instance family. In addition to the columns also displayed in Table 5, the table includes the chance constraint threshold (ϵ), and, for the recommended SoC ranges of 20–80% and 30–80%, the improvement in operational costs relative to the worst-case benchmark (Impr. vs. w.-c.) and the deterioration in operational costs relative to the optimistic benchmark (Det. vs. opti.). For each instance family, the first row (with $\epsilon = 0$) corresponds to the worst-case deterministic benchmark.

For all instance families (I1, I2, I3), even a small increase in ϵ leads to important reductions in operational costs. This reduction is primarily attributed to the decrease in the number of EBs required, which is a desired outcome. For example, in I3 with the 30–80% SoC range, increasing ϵ from 0 to 0.005 reduces the average number of EBs from 12.0 to 11.2, yielding a 6.35% cost improvement relative to the worst-case benchmark and reducing the deterioration relative to the optimistic benchmark from 9.59% to 2.64%. Similar trends are observed in I1 and I2, where operational costs almost always decrease monotonically as the chance constraint is slightly relaxed.

Operational costs improvements generally stabilize after a certain threshold of ϵ. For example, in I1, improvements (relative to the worst-case benchmark) plateau at 5.07% for the 20–80% range as ϵ increases beyond 0.10. Similarly, for I2 and I3, cost improvements also stabilize, indicating diminishing returns from further relaxing the chance constraint. However, in I3, operational costs improve again for the 30–80% range at $\epsilon = 0.50$.

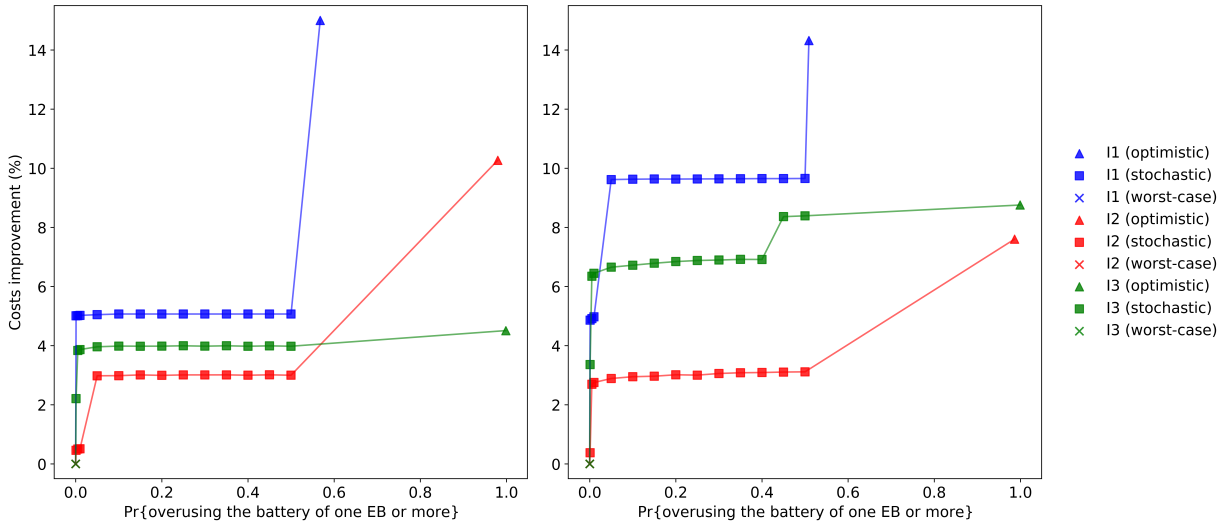
Figure 5 further compares the deterministic benchmarks and the stochastic approach with policy-guided battery management. Figures 5a and 5b correspond to the 20–80% and 30–80% recommended SoC ranges, respectively, with each line representing one instance (I1, I2, or I3). The plots show the relative improvement in operational costs relative to the worst-case benchmark, together with the approximate probability of not complying with the recommended SoC range in one vehicle schedule or more. This latter probability is computed offline using a Monte Carlo simulation with 1,000 iterations. At each iteration, random energy consumption values are sampled for all timetabled trips. The number of iterations in which the recommended SoC range is violated in at least one EB is then divided by the total number of iterations. In the worst-case benchmark, this value is by definition always 0, whereas in the stochastic E-VSP with policy-guided battery

Table 6: Results of the stochastic E-VSP with policy-guided battery management

		20–80% range				30–80% range			
Inst.	ϵ	Op. costs	Impr. vs. w.-c. (%)	Det. vs. opti. (%)	# EBs	Op. costs	Impr. vs. w.-c. (%)	Det. vs opti. (%)	# EBs
I1	0	4,102.2	0.00	17.81	3.8	4,312.7	0.00	16.88	4.0
	0.001	3,896.8	5.01	11.91	3.6	4,103.1	4.86	11.20	3.8
	0.005	3,896.5	5.01	11.90	3.6	4,100.8	4.91	11.14	3.8
	0.01	3,896.2	5.02	11.90	3.6	4,098.3	4.97	11.07	3.8
	0.05	3,895.2	5.05	11.87	3.6	3,898.0	9.62	5.64	3.6
	0.10	3,894.3	5.07	11.84	3.6	3,897.5	9.63	5.63	3.6
	0.15	3,894.3	5.07	11.84	3.6	3,897.2	9.63	5.62	3.6
	0.20	3,894.3	5.07	11.84	3.6	3,897.3	9.63	5.62	3.6
	0.25	3,894.3	5.07	11.84	3.6	3,897.1	9.64	5.62	3.6
	0.30	3,894.3	5.07	11.84	3.6	3,897.0	9.64	5.61	3.6
	0.40	3,894.3	5.07	11.84	3.6	3,896.6	9.65	5.61	3.6
	0.50	3,894.3	5.07	11.84	3.6	3,896.4	9.65	5.60	3.6
I2	0	8,313.6	0.00	11.44	7.8	8,540.2	0.00	8.23	8.0
	0.001	8,275.6	0.46	10.94	7.8	8,508.0	0.38	7.82	8.0
	0.005	8,271.9	0.50	10.89	7.8	8,310.2	2.69	5.31	7.8
	0.01	8,271.4	0.51	10.88	7.8	8,305.3	2.75	5.25	7.8
	0.05	8,066.0	2.98	8.13	7.6	8,293.6	2.89	5.10	7.8
	0.10	8,065.6	2.98	8.12	7.6	8,288.5	2.95	5.04	7.8
	0.15	8,063.5	3.01	8.09	7.6	8,287.0	2.97	5.02	7.8
	0.20	8,064.8	2.99	8.11	7.6	8,282.9	3.01	4.96	7.8
	0.25	8,063.5	3.01	8.09	7.6	8,283.9	3.00	4.98	7.8
	0.30	8,063.5	3.01	8.09	7.6	8,279.2	3.06	4.92	7.8
	0.40	8,064.4	3.00	8.10	7.6	8,276.4	3.09	4.88	7.8
	0.50	8,064.4	3.00	8.10	7.6	8,274.4	3.11	4.86	7.8
I3	0	12,157.7	0.00	4.72	11.4	12,782.3	0.00	9.59	12.0
	0.001	11,889.6	2.21	2.41	11.2	12,352.8	3.36	5.91	11.6
	0.005	11,691.6	3.83	0.70	11.0	11,971.1	6.35	2.64	11.2
	0.01	11,687.7	3.87	0.67	11.0	11,958.4	6.45	2.53	11.2
	0.05	11,676.3	3.96	0.57	11.0	11,932.1	6.65	2.31	11.2
	0.10	11,673.4	3.98	0.55	11.0	11,923.4	6.72	2.23	11.2
	0.15	11,673.9	3.98	0.55	11.0	11,915.0	6.79	2.16	11.2
	0.20	11,673.7	3.98	0.55	11.0	11,907.8	6.84	2.10	11.2
	0.25	11,672.2	3.99	0.54	11.0	11,903.1	6.88	2.06	11.2
	0.30	11,673.8	3.98	0.55	11.0	11,901.1	6.89	2.04	11.2
	0.40	11,674.0	3.98	0.55	11.0	11,898.6	6.91	2.02	11.2
	0.50	11,674.0	3.98	0.55	11.0	11,710.0	8.39	0.40	11.0

management, it equals ϵ whenever constraint 12 is active.

The optimistic benchmark achieves the lowest costs (see Tables 5 and 6) but leads to high probabilities of battery overuse: above 50% for I1 and close to 100% for I2 and I3, regardless of the SoC range. Such high probabilities render this approach unsuitable for practical use, as it is poorly aligned with the battery



(a) 20–80% range

(b) 30–80% range

Figure 5: Improvement in operational costs (relative to worst-case benchmark) vs. probability of not complying with the recommended SoC range in one vehicle schedule or more—deterministic and stochastic approaches

management guidelines set by battery leasing companies or warranty providers. Conversely, the worst-case benchmark guarantees strict adherence with the recommended SoC range, but results in solutions with substantially higher operational costs. The stochastic approach, by adjusting the chance constraint threshold ϵ, offers good trade-offs between operational costs and battery management guidelines compliance.

5 Conclusions

In this study, we presented a chance-constrained integer linear program for the E-VSP with policy-guided battery management and stochastic energy consumption. The chance constraint ensures compliance with recommended SoC ranges, such as 20–80% or 30–80%, as specified by battery leasing companies or warranty providers, by limiting the probability of battery overuse. Unlike deterministic approaches based on optimistic or worst-case assumptions, our method explicitly accounts for uncertainty in energy consumption. The model also accounts for partial en-route charging, nonlinear charging profiles, and limited-capacity charging stations. Nonlinear charging functions are approximated using piecewise linear functions.

To solve the model, we proposed a heuristic branch-and-price algorithm featuring stochastic pricing problems, specifically stochastic shortest path problems with resource constraints. A labeling algorithm with stochastic dominance rules based on the joint CDFs of the SoC and the probability of not having experienced battery overuse is developed. Computing time is reduced by implementing a heuristic branching strategy, namely a diving heuristic, and a constraint perturbation method.

Computational experiments conducted on realistic synthetic instances validated the performance of our heuristic, demonstrating its ability to derive near-optimal solutions. The results showed that even small values of the chance constraint probability threshold ϵ can yield important cost savings compared to the

deterministic worst-case benchmark, primarily by reducing the number of required EBs. Moreover, while the optimistic benchmark achieved the lowest operational costs, it did so at the expense of frequent violations of the battery management recommendations—with daily probabilities of one or more EBs operating outside the recommended SoC range ranging from 50% to nearly 100%. By delivering more cost-effective solutions than the worst-case benchmark while keeping battery overuse probabilities low—and thus aligned with battery management guidelines—the stochastic approach is likely to be the preferred choice for decision-makers.

While our work demonstrated the added value of our stochastic model for the E-VSP with policy-guided battery management, it remains limited to relatively small instances. Future research could focus on reducing computational times to enable the solution of larger instances.

6 Acknowledgments

We thank the Optimization Algorithm Development team at GIRO Inc. for the insightful discussions that helped shape the problem definition. This work was partially supported by GIRO Inc. and the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant RDCPJ 520349-17. Léa Ricard gratefully acknowledges the financial support of NSERC through the BESC D3-558645-2021 scholarship. The work of the first and third authors was supported by the Canada Excellence Research Chair in 'Data Science for Real-Time Decision-Making'. We also thank the three anonymous referees for their thorough reviews and insightful comments.

References

Abdelaty, H., Al-Obaidi, A., Mohamed, M., Farag, H.E., 2021. Machine learning prediction models for battery-electric bus energy consumption in transit. Transportation Research Part D: Transport and Environment 96, 102868. doi:https://doi.org/10.1016/j.trd.2021.102868.

Alvo, M., Angulo, G., Klapp, M.A., 2021. An exact solution approach for an electric bus dispatch problem. Transportation Research Part E: Logistics and Transportation Review 156, 102528. doi:https://doi.org/10.1016/j.tre.2021.102528.

Avishan, F., İhsan Yanıkoğlu, Alwesabi, Y., 2023. Electric bus fleet scheduling under travel time and energy consumption uncertainty. Transportation Research Part C: Emerging Technologies 156, 104357.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H., 1998. Branch-and-price: Column generation for solving huge integer programs. Operations Research 46, 316–329. doi:https://doi.org/10.1287/opre.46.3.316.

Basma, H., Mansour, C., Haddad, M., Nemer, M., Stabat, P., 2020. Comprehensive energy modeling methodology for battery electric buses. Energy 207, 118241. doi:https://doi.org/10.1016/j.energy.2020.118241.

Bazarnovi, S., Cokyasar, T., Verbas, O., Mohammadian, A.K., 2025. Problem of locating and allocating charging equipment for battery electric buses under stochastic charging demand. European Journal of Operational Research doi:https://doi.org/10.1016/j.ejor.2025.07.064.

Benchimol, P., Desaulniers, G., Desrosiers, J., 2012. Stabilized dynamic constraint aggregation for solving set partitioning problems. European Journal of Operational Research 223, 360–371. URL: https://www.sciencedirect.com/science/article/pii/S0377221712005255, doi:https://doi.org/10.1016/j.ejor.2012.07.004.

Bie, Y., Ji, J., Wang, X., Qu, X., 2021. Optimization of electric bus scheduling considering stochastic volatilities in trip travel time and energy consumption. Computer-Aided Civil and Infrastructure Engineering 36, 1530–1548. doi:https://doi.org/10.1111/mice.12684.

Boland, N., Dickson, S., Savelsbergh, M., Smilowitz, K., 2015. Dominance in pricing problems with stochasticity. http://www.optimization-online.org/DB_HTML/2015/08/5048.html. Accessed: 2025-03-21.

Brasseur, J., 2022. Accélération d'une méthode d'agrégation dynamique de contraintes par apprentissage automatique pour le problème de construction d'horaires de conducteurs d'autobus. Master's thesis. Polytechnique Montréal.

Charnes, A., 1952. Optimality and degeneracy in linear programming. The Econometric Society 20, 160–170.

Costa, L., Contardo, C., Desaulniers, G., 2019. Exact branch-price-and-cut algorithms for vehicle routing. Transportation Science 53, 946–985. doi:10.1287/trsc.2018.0878.

Desaulniers, G., Desrosiers, J., Solomon, M., 2005. Column Generation. doi:10.1007/b135457.

Desaulniers, G., Hickman, M.D., 2007. Chapter 2 public transit, in: Barnhart, C., Laporte, G. (Eds.), Transportation. Elsevier. volume 14 of *Handbooks in Operations Research and Management Science*, pp. 69–127. doi:https://doi.org/10.1016/S0927-0507(06)14002-5.

Dolgui, A., Kovalev, S., Kovalyov, M.Y., 2025. Scheduling electric vehicle regular charging tasks: A review of deterministic models. European Journal of Operational Research 325, 221–232. doi:https://doi.org/10.1016/j.ejor.2024.11.044.

Duan, M., Liao, F., Qi, G., Guan, W., 2023. Integrated optimization of electric bus scheduling and charging planning incorporating flexible charging and timetable shifting strategies. Transportation Research Part C: Emerging Technologies 152, 104175.

Freling, R., Wagelmans, A.P.M., Paixão, J.M.P., 2001. Models and algorithms for single-depot vehicle scheduling. Transportation Science 35, 165–180. doi:10.1287/trsc.35.2.165.10135.

Gerbaux, J., Desaulniers, G., Cappart, Q., 2025. A machine-learning-based column generation heuristic for electric bus scheduling. Computers Operations Research 173, 106848.

Gkiotsalitis, K., Iliopoulou, C., Kepaptsoglou, K., 2023. An exact approach for the multi-depot electric bus scheduling problem with time windows. European Journal of Operational Research 306, 189–206. doi:https://doi.org/10.1016/j.ejor.2022.07.017.

Hadjar, A., Marcotte, O., Soumis, F., 2006. A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. Operations Research 54, 130–149. doi:https://doi.org/10.1287/opre.1050.0240.

He, Y., Liu, Z., Song, Z., 2023. Joint optimization of electric bus charging infrastructure, vehicle scheduling, and charging management. Transportation Research Part D: Transport and Environment 117, 103653.

Jiang, J., Shi, W., Zheng, J., Zuo, P., Xiao, J., Chen, X., Xu, W., Zhang, J.G., 2013. Optimized operating range for large-format lifepo4/graphite batteries. Journal of the Electrochemical Society 161, 336–341. doi:10.1149/2.052403jes.

Jiang, M., Zhang, Y., Zhang, Y., 2021. Optimal electric bus scheduling under travel time uncertainty: A robust model and solution method. Journal of Advanced Transportation 2021, 1191443. doi:https://doi.org/10.1155/2021/1191443.

Jiang, M., Zhang, Y., Zhang, Y., 2023. A branch-and-price algorithm for large-scale multidepot electric bus scheduling. IEEE Transactions on Intelligent Transportation Systems 24, 15355–15368.

Kliewer, N., Mellouli, T., Suhl, L., 2006. A time–space network based exact optimization model for multi-depot bus scheduling. European Journal of Operational Research 175, 1616–1627. doi:https://doi.org/10.1016/j.ejor.2005.02.030.

van Kooten Niekerk, M.E., Van den Akker, J., Hoogeveen, J., 2017. Scheduling electric vehicles. Public Transport 9, 155–176.

Kostopoulos, E.D., Spyropoulos, G.C., Kaldellis, J.K., 2020. Real-world study for the optimal charging of electric vehicles. Energy Reports 6, 418–426. doi:https://doi.org/10.1016/j.egyr.2019.12.008.

Li, J.Q., 2014. Transit bus scheduling with limited energy. Transportation Science 48, 521–539.

Li, L., Lo, H.K., Huang, W., Xiao, F., 2021. Mixed bus fleet location-routing-scheduling under range uncertainty. Transportation Research Part B: Methodological 146, 155–179. doi:https://doi.org/10.1016/j.trb.2021.02.005.

Liu, T., Ceder, A., 2020. Battery-electric transit vehicle scheduling with optimal number of stationary chargers. Transportation Research Part C: Emerging Technologies 114, 118–139. doi:https://doi.org/10.1016/j.trc.2020.02.009.

Liu, Y., Zuo, X., Li, X., Nie, S., 2024. A genetic algorithm with trip-adjustment strategy for multi-depot electric bus scheduling problems. Engineering Optimization 56, 1200–1219.

Lu, L., Han, X., Li, J., Hua, J., Ouyang, M., 2013. A review on the key issues for lithium-ion battery management in electric vehicles. Journal of Power Sources 226, 272–288. doi:https://doi.org/10.1016/j.jpowsour.2012.10.060.

Lu, Z., Xing, T., Li, Y., 2025. Optimization of electric bus vehicle scheduling and charging strategies under time-of-use electricity price. Transportation Research Part E: Logistics and Transportation Review 196, 104021.

Lübbecke, M.E., Desrosiers, J., 2005. Selected topics in column generation. Operations Research 53, 1007–1023. doi:https://doi.org/10.1287/opre.1050.0234.

Löbel, A., 1998. Vehicle scheduling in public transit and lagrangean pricing. Management Science 44, 1637–1649.

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. doi:https://doi.org/10.1016/j.trb.2017.02.004.

Najafi, A., Gao, K., Parishwad, O., Tsaousoglou, G., Jin, S., Yi, W., 2025. Integrated optimization of charging infrastructure, electric bus scheduling and energy systems. Transportation Research Part D: Transport and Environment 141, 104664.

Olsen, N., Kliewer, N., 2020. Scheduling electric buses in public transport: Modeling of the charging process and analysis of assumptions. Logistics Research 13, 18. doi:https://doi.org/10.23773/2020_4.

Oukil, A., Amor, H.B., Desrosiers, J., El Gueddari, H., 2007. Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems. Computers & Operations Research 34, 817–834. doi:https://doi.org/10.1016/j.cor.2005.05.011.

Pelletier, S., Jabali, O., Laporte, G., Veneroni, M., 2017. Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models. Transportation Research Part B: Methodological 103, 158–187. doi:10.1016/j.trb.2017.01.020.

Perumal, S.S., Dollevoet, T., Huisman, D., Lusby, R.M., Larsen, J., Riis, M., 2021. Solution approaches for integrated vehicle and crew scheduling with electric buses. Computers & Operations Research 132, 105268. doi:https://doi.org/10.1016/j.cor.2021.105268.

Perumal, S.S., Lusby, R.M., Larsen, J., 2022. Electric bus planning & scheduling: A review of related problems and methodologies. European Journal of Operational Research 301, 395–413. doi:https://doi.org/10.1016/j.ejor.2021.10.058.

Ribeiro, C.C., Soumis, F., 1994. A column generation approach to the multiple-depot vehicle scheduling problem. Operations Research 42, 41–52.

Ricard, L., Desaulniers, G., Lodi, A., Rousseau, L.M., 2024. Increasing schedule reliability in the multiple depot vehicle scheduling problem with stochastic travel time. Omega 127, 103100. doi:https://doi.org/10.1016/j.omega.2024.103100.

Rinaldi, M., Picarelli, E., D'Ariano, A., Viti, F., 2020. Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications. Omega 96, 102070. doi:https://doi.org/10.1016/j.omega.2019.05.006.

Sassi, O., Oulamara, A., 2017. Electric vehicle scheduling and optimal charging problem: complexity, exact and heuristic approaches. International Journal of Production Research 55, 519–535. doi:10.1080/00207543.2016.1192695.

Tang, X., Lin, X., He, F., 2019. Robust scheduling strategies of electric buses under stochastic traffic conditions. Transportation Research Part C: Emerging Technologies 105, 163–182. doi:https://doi.org/10.1016/j.trc.2019.05.032.

Wang, C., Guo, C., Zuo, X., 2021. Solving multi-depot electric vehicle scheduling problem by column generation and genetic algorithm. Applied Soft Computing 112, 107774. doi:https://doi.org/10.1016/j.asoc.2021.107774.

Wellman, M.P., Ford, M., Larson, K., 2013. Path planning under time-dependent uncertainty. arXiv:1302.4987.

Wen, M., Linde, E., Ropke, S., Mirchandani, P., Larsen, A., 2016. An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. Computers & Operations Research 76, 73–83. doi:https://doi.org/10.1016/j.cor.2016.06.013.

Wu, W., Lin, Y., Liu, R., Jin, W., 2022. The multi-depot electric vehicle scheduling problem with power grid characteristics. Transportation Research Part B: Methodological 155, 322–347. URL: https://www.sciencedirect.com/science/article/pii/S0191261521002125, doi:https://doi.org/10.1016/j.trb.2021.11.007.

Xu, X., Yu, Y., Long, J., 2023. Integrated electric bus timetabling and scheduling problem. Transportation Research Part C: Emerging Technologies 149, 104057. doi:https://doi.org/10.1016/j.trc.2023.104057.

Yan, Y., Wen, H., Deng, Y., Chow, A.H., Wu, Q., Kuo, Y.H., 2024. A mixed-integer programming-based q-learning approach for electric bus scheduling with multiple termini and service routes. Transportation Research Part C: Emerging Technologies 162, 104570.

Şule Yıldırım, Yıldız, B., 2021. Electric bus fleet composition and scheduling. Transportation Research Part C: Emerging Technologies 129, 103197.

Zhang, A., Li, T., Zheng, Y., Li, X., Abdullah, M.G., Dong, C., 2022. Mixed electric bus fleet scheduling problem with partial mixed-route and partial recharging. International Journal of Sustainable Transportation 16, 73–83.

Zhang, L., Wang, S., Qu, X., 2021. Optimal electric bus fleet scheduling considering battery degradation and non-linear charging profile. Transportation Research Part E: Logistics and Transportation Review 154, 102445. doi:https://doi.org/10.1016/j.tre.2021.102445.

Zhang, M., Yang, M., 2025. Optimal electric bus scheduling considering battery degradation effect and charging facility capacity. Transportation Letters 17, 491–501.

Zhou, Y., Meng, Q., Ong, G.P., Wang, H., 2024a. Electric bus charging scheduling on a bus network. Transportation Research Part C: Emerging Technologies 161, 104553.

Zhou, Y., Wang, H., Wang, Y., Yu, B., Tang, T., 2024b. Charging facility planning and scheduling problems for battery electric bus systems: A comprehensive review. Transportation Research Part E: Logistics and Transportation Review 183, 103463.

A Proof of Proposition 1

Consider first a single-arc extension along arc (i,j). From Equation (13), the function $f_j^{p\oplus(i,j)}$ can be written as a nonnegative linear combination of f_i^p, namely

$$f_j^{p\oplus(i,j)}(x) = \sum_{t=x+\iota_{i,j}}^{\sigma^{up}} a_j(x,t)\, f_i^p(t), \qquad a_j(x,t) \geq 0, \tag{27}$$

all $x \in \{\sigma^{low}, \ldots, \sigma^{up}\}$, where

$$(t, a_j(x,t)) = \begin{cases} \big(x+\mu+\iota_{i,j},\ e_j(t-x-\iota_{i,j})\big) & \text{if } j \in \mathcal{V}, \\ \big(\lambda^{-1}(x,\rho),\ \mathbb{1}\{t=\lambda^{-1}(x,\rho)\}\big) & \text{if } j \in \mathcal{H}^C, \\ \big(x+\iota_{i,j},\ \mathbb{1}\{t=x+\iota_{i,j}\}\big) & \text{if } j \in \mathcal{H}^W \cup \{n_1^d\}. \end{cases}$$

By definition of G,

$$G_j^{p\oplus(i,j)}(x) = \sum_{x'=x}^{\sigma^{up}} f_j^{p\oplus(i,j)}(x'). \tag{28}$$

Substituting (27), exchanging the order of summation, and using the hypotheses of the proposition, we can write

$$G_j^{p_1\oplus(i,j)}(x) = \sum_{t=x+\iota_{i,j}}^{\sigma^{up}} \left(\sum_{x'=x}^{\sigma^{up}} a_j(x',t) \right) f_i^{p_1}(t) \tag{29}$$

$$\geq \sum_{t=x+\iota_{i,j}}^{\sigma^{up}} \left(\sum_{x'=x}^{\sigma^{up}} a_j(x',t) \right) f_i^{p_2}(t) = G_j^{p_2\oplus(i,j)}(x) \tag{30}$$

for all $x \in \{\sigma^{low}, \ldots, \sigma^{up}\}$, which proves the proposition in the single-arc extension case. The result follows for any extension ξ by repeating the same argument for each arc of ξ.