Transportation Research Part C 183 (2026) 105480

Contents lists available at ScienceDirect

Transportation Research Part C

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

A stochastic electric vehicle routing problem under uncertain energy consumption

Andrea Spinelli [a], Dario Bezzi [b], Ola Jabali [c,d], Francesca Maggioni [a,*]

[a] Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione, Università di Bergamo, Viale G. Marconi 5, 24044, Dalmine, Italy
[b] Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione G. Marconi, Università di Bologna, Viale del Risorgimento 2, Bologna, 40136, Italy
[c] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via G. Ponzio 34, 20133, Milano, Italy
[d] HEC Montréal, Montréal, Canada

ARTICLE INFO

Keywords:
Routing
Electric vehicles
Uncertain energy consumption
Stochastic programming
Iterated local search
Scenario reduction

ABSTRACT

The increasing adoption of Electric Vehicles (EVs) for service and goods distribution operations has led to the emergence of Electric Vehicle Routing Problems (EVRPs), a class of vehicle routing problems addressing the unique challenges posed by the limited driving range and recharging needs of EVs. While the majority of EVRP variants have considered deterministic energy consumption, this paper focuses on the Stochastic Electric Vehicle Routing Problem with a Threshold recourse policy (SEVRP-T), where the uncertainty in energy consumption is considered, and a recourse policy is employed to ensure that EVs recharge at Charging Stations (CSs) whenever their State of Charge (SoC) falls below a specified threshold. We formulate the SEVRP-T as a two-stage stochastic mixed-integer second-order cone model, where the first stage determines the sequences of customers to be visited, and the second stage incorporates charging activities. The objective is to minimize the expected total duration of the routes, composed by travel times and recharging operations. To cope with the computational complexity of the model, we propose a heuristic based on an Iterated Local Search (ILS) procedure coupled with a Set Partitioning problem. To further speed up the heuristic, we develop two lower bounds on the corresponding first-stage customer sequences. Furthermore, to handle a large number of energy consumption scenarios, we employ a scenario reduction technique. Extensive computational experiments are conducted to validate the effectiveness of the proposed solution strategy and to assess the importance of considering the stochastic nature of the energy consumption. The research presented in this paper contributes to the growing body of literature on EVRP and provides insights into managing the operational deployment of EVs in logistics activities under uncertainty.

1. Introduction

Several nations and local governments have established measures aimed at increasing the commercial use of Electric Vehicles (EVs) and progressively phasing out conventional vehicles. Indeed, sales of electric light commercial vehicles worldwide have almost doubled in 2022 relative to 2021 to more than 310,000 vehicles, making their market share around 3.6 %. Moreover, electric medium- and heavy-duty truck sales totalled nearly 60,000 in 2022, which constitutes 1.2 % of the total number of registrations for that category worldwide (see IEA, 2023).

* Corresponding author.
E-mail address: francesca.maggioni@unibg.it (F. Maggioni).

https://doi.org/10.1016/j.trc.2025.105480
Received 17 February 2025; Received in revised form 17 October 2025; Accepted 21 November 2025
Available online 3 December 2025

Logistics companies using EVs must plan the operation of such vehicles considering limited driving ranges and slow charging times. Specifically, planning recharging operations along the EV routes is particularly relevant in the context of the mid- and long-haul logistics, since the driving range of medium-duty EVs remains limited (see Schiffer et al., 2018, 2021). The reader is referred to Juan et al. (2016) for a survey on various research challenges related to the introduction of EVs in logistics and transportation.

A number of scientific contributions have focused on the operational challenges of using EVs in a logistics context. In this framework, a commonly studied problem is the Electric Vehicle Routing Problem (EVRP, see Kucukoglu et al., 2021 for a survey), where a set of customers must be visited by a homogeneous fleet of EVs that start and end their routes at a given depot. In EVRPs, the limited autonomy of EVs is explicitly modeled and Charging Stations (CSs) are often assumed to be privately owned by the EV operator (see Froger et al., 2019). This latter assumption entails that the charging demand from EVs not owned by the operator may be ignored. Each CS may host a different charging technology (e.g., fast, medium or slow). In the most general EVRP case, charging times are assumed to follow nonlinear charging functions, and partial charging is allowed. This entails that establishing which CS to visit and how much to charge at it are part of the decision variables.

The vast majority of the literature on EVRPs assumes the EV energy consumption to be deterministic and proportional to the traveled distance (see Lam et al., 2022). In reality, energy consumption is not fully predictable, as it may heavily depend on uncertain factors like terrain type, travel speed, driving style, traffic conditions, temperature and wind speed (see Asamer et al., 2016). As ignoring such uncertainties may lead an EV to get stranded without battery, it is pivotal to account for them when planning EV routes. This is particularly critical in operational contexts such as urban distribution under uncertain traffic conditions, where frequent acceleration and braking increase energy use (see Zhao et al., 2025), or in long-haul logistics operations, where unpredictable consumption can prevent reaching the next planned CS unless costly recourse actions are taken. To this end, in this paper we consider an EVRP with stochastic energy consumption. In order to cope with the uncertainty, we adopt a *threshold recourse policy*. Given a fixed sequence of customers to visit, this policy entails that when the State of Charge (SoC) of the EV falls below a given threshold, the vehicle will immediately detour towards a CS and charge a sufficient amount of energy to bring its SoC at the subsequent customer to a given fixed level. The resulting problem is called the Stochastic Electric Vehicle Routing Problem with a Threshold recourse policy (SEVRP-T), and it has been introduced by Bezzi et al. (2024). In this problem, CSs with multiple technologies are considered, and nonlinear charging functions are approximated via piecewise linear concave functions. Notably, while detour decisions in the classical EVRP are taken at the nodes, in the SEVRP-T such decisions may be taken at any point while traversing an arc. Indeed, an EV may reach the threshold SoC level while traversing an arc, at which point it will immediately head towards a suitable CS, rather than fully traversing the arc. As the energy consumption is stochastic, the departing SoC from a node is uncertain as well. Thus, the length of a detour cannot be calculated a priori as part of the input data. Allowing detours to be initiated from any point along an arc is particularly relevant in operational contexts that may include, among others, situations where node distances are long relative to battery capacity, where energy consumption is highly uncertain due to external factors, and where charging infrastructure is limited. In such settings, waiting until the next node to recharge may be infeasible or lead to significantly higher costs. While node-based recharging is often sufficient for short-haul or well-equipped urban networks, the arc-level flexibility introduced in the SEVRP-T captures important practical cases within mid- and long-haul logistics or contexts with limited infrastructure. Furthermore, node-based detours are a special case of arc-based detours. Indeed, the former assumes that a CS may only be reached from other nodes, whereas the latter enables reaching a CS from the entire network. As this entails working with a continuous domain of CS detours, the difficulty of our problem far-exceeds that with node-based detours. However, this difficulty allows us to model a much more realistic problem, where one observes energy consumption as arcs are being traversed, and is not restricted to detour to CSs from customer nodes or the depot.

In this paper, we formulate SEVRP-T as a two-stage stochastic mixed-integer second-order cone model, where the first-stage decisions determine the sequence of customers to visit by each EV, while the second-stage ones incorporate charging activities when needed. To take into account uncertain energy consumption, we consider a discrete set of scenarios. Our formulation is an improved version of the one presented in Bezzi et al. (2024). The objective is to minimize the expected total duration of the routes, composed by travel times and recharging times. The resulting model is very challenging as it combines stochasticity and discrete decisions. Therefore, to cope with realistic instances, we propose a heuristic for the SEVRP-T. The heuristic is made up of two components: a *route generator* and a *solution assembler*. The route generator builds a pool of high-quality routes, which are sent to a Set Partitioning formulation at the end. This formulation selects (i.e., assembles) the most promising combination of routes from the pool, such that each customer is served by a single EV. The route generator relies on an Iterated Local Search procedure (ILS), which uses a Variable Neighborhood Descent metaheuristic (VND). Additionally, we apply a scenario reduction technique, which allows us to handle instances with large sets of energy consumption scenarios. To evaluate route durations within the VND, we formulate and exactly solve a Stochastic Fixed Route Vehicle Charging Problem with a Threshold policy (SFRVCP-T). Furthermore, we derive two lower bounds on the duration of fixed routes. We use these bounds to filter unpromising moves. To validate the performance of our proposed heuristic, we have carried out extensive computational experiments on sets of instances adapted from available benchmarks. For small-sized instances which can be solved by an off-the-shelf solver, we further benchmark its performance.

Summarizing, our contributions are as follows:

- Propose a two-stage stochastic programming formulation for the Stochastic Electric Vehicle Routing Problem with a Threshold recourse policy and analyze its key characteristics;
- Design a heuristic algorithm based on an ILS with Set Partitioning to handle large problem instances;
- Derive lower bounds on the duration of fixed routes to filter unpromising moves during the VND phase of the ILS heuristic;

- Apply a scenario reduction technique (see Heitsch and Römisch, 2003) to maintain a low number of considered energy consumption scenarios;
- Provide extensive numerical experiments with the aim of testing the proposed heuristic on instances derived from the EVRP literature and compare its performance with state-of-the-art solvers.

The remainder of the paper is organized as follows. In Section 2 we present an overview of the EVRP literature. In Section 3 we formally introduce the SEVRP-T and present its mathematical formulation. We describe our heuristic in Section 4, and discuss the scenario reduction technique in Section 5. In Section 6 we present our computational results. Finally, we summarize our conclusions and outline future research perspectives in Section 7.

2. Literature review

The EVRP is an extension of the traditional Vehicle Routing Problem (VRP, see Toth and Vigo, 2014) whose objective is to serve a given set of customers by establishing least-cost routes for a fleet of homogeneous vehicles. These are EVs in the case of the EVRP, and thus battery capacity and charging facilities are considered.

When dealing with EVRPs, various modeling elements have to be taken into account, including the charging process of battery and the energy consumption between customers' visits. For recent surveys on the EVRP and its variants, the reader is referred to Macrina et al. (2020), Kucukoglu et al. (2021) and Xiao et al. (2021).

In the extant literature on EVRPs, different charging policies are explored, including battery swapping, full recharge and partial recharge (see Liu et al., 2023). In the case of battery swapping, the EVs visit specific stations to replace the existing batteries by fully charged ones (e.g., Jie et al., 2019). This practice remains relatively uncommon (see Dastpak et al., 2024) and most EVRP studies focus instead on charging-based routing approaches (see Han et al., 2025). In the Green Vehicle Routing Problem (GVRP, see Erdoğan and Miller-Hooks, 2012) EVs fully recharge their batteries upon visiting a CS. The corresponding charging time is either assumed to be constant (e.g., Andelmin and Bartolini, 2017) or linearly dependent on the SoC of the vehicle when arriving at the CS (e.g., Schneider et al., 2014). Whenever a partial charging policy is adopted, the charging time is included as a decision variable of the model. In contrast to a full charging policy, a partial one results in savings in both energy consumption and time (see Froger et al., 2019). Different works assume that the charging process is linear (see Felipe et al., 2014; Desaulniers et al., 2016 and Wang and Zhao, 2023). However, in practice the charge retrieved at a CS is a nonlinear concave function of the charging time (see Pelletier et al., 2017). To account for this, several authors modeled the charging process using piecewise linear functions (e.g., Montoya et al., 2017 and Klein and Schiffer, 2023). With this choice, the computational complexity is increased by the effect of breakpoints introduced to model the varying charging rates of the battery (see Dönmez et al., 2022). Alternative approximations of the charging function are proposed in Lee (2021) and Schulz (2024).

With respect to the charging configuration, different technologies exist, i.e., normal, fast, and super-fast chargers (see Keskin and Çatay, 2018). This aspect, in conjunction with a partial recharging policy, has been explored in Felipe et al. (2014). In Froger et al. (2019) two compact mixed-integer linear programming formulations have been designed for the EVRP with nonlinear charging functions. The authors propose an arc-based tracking of the time and the SoC rather than the traditional node-based tracking. In addition, they develop an exact labeling algorithm to find the optimal charging decisions on the given routes. Bezzi et al. (2023) have presented a branch-and-price algorithm for EVRP with partial recharges assuming linear charging processes pertaining to multiple technologies. Recently, in Wang et al. (2025) an EVRP with heterogeneous fleet and nonlinear charging functions has been introduced. The problem involves utilizing multiple charging modes and accounts for time-dependent waiting time functions at CSs.

The overwhelming majority of EVRP studies assume that all parameters are known with certainty when taking decisions. In such a context, the EV energy consumption is one of the most relevant parameters. Typically, it is assumed to be a linear function of traveled distance (see Felipe et al., 2014; Breunig et al., 2019 and Almouhanna et al., 2020). However, to achieve more accurate and realistic results, its computation should account for multiple factors, both exogenous and endogenous, including road quality, vehicle characteristics, cargo weight, and environmental conditions (see Kucukoglu et al., 2021). In Goeke and Schneider (2015) a nonlinear energy consumption model incorporating vehicle speed, road gradient and cargo load is proposed. In Bruglieri et al. (2023) an EVRP with time windows is addressed considering different factors for the energy consumption, i.e., payload and vehicle speed. Possible stops at the CSs are permitted en route, allowing for partial recharge. Recently, Xiong et al. (2024) have introduced drivetrain losses and traffic congestion as a nonlinear function of vehicle speed, incorporating them into a comprehensive energy consumption model. Nonetheless, each additional factor complicates the computation of the energy consumption rate. Thus, there is a crucial trade-off between solution times and the number, as well as the nature, of the considered factors in modeling energy consumption (see Kucukoglu et al., 2021). Asamer et al. (2016) have conducted a sensitivity analysis assessing the impact of various factors on energy consumption, such as total mass, efficiency of driving and rolling friction coefficient. The authors show that several factors, even if difficult to be precisely measured, significantly affect the estimation. Therefore, it is reasonable to treat such factors as uncertain, opening the door to stochastic and robust optimization approaches (see, e.g., Filippi et al., 2025; Spinelli et al., 2025).

Bruni et al. (2020) have introduced an EVRP with stochastic energy consumption for a single vehicle and formulated it as a two-stage stochastic programming model. Tactical routes are operated on a daily basis, being energy feasible in expectation. It is assumed that the energy consumption over the entire network is revealed at the beginning of each operational period, i.e., the energy consumption of each arc in the network is known at the beginning of the day (prior to starting the routes), and the first-stage routes are executed accordingly. At which point, en route recharging activities are possibly added, allowing the EV to travel from a customer node (or from the depot) to a suitable CS. Conversely, in our problem, we assume that the uncertain energy consumption is only

revealed when an arc starts being traversed by an EV, which more closely reflects real-world operational settings. A probabilistic Bayesian machine learning approach is explored by Basso et al. (2021) to predict the expected energy consumption and its variance on a road network. Specifically, an EV routing model composed of two stages is presented. In the first stage, minimum energy paths connecting the nodes are devised, while in the second stage the best order to visit the customers is determined. If necessary, partial charging activities are planned. The information provided by the machine learning techniques are cast into the model through a set of chance-constraints, assuring that the SoC of the vehicles will never go below a fixed threshold within a confidence interval. Basso et al. (2022) have considered stochasticity in terms of both energy consumption and customer demands. The resulting stochastic dynamic EVRP is modeled through a Markov decision process and solved with reinforcement learning techniques.

To the best of our knowledge, the first robust EVRP optimization model considering energy consumption uncertainty has been proposed by Pelletier et al. (2019). The aim is to plan minimum cost routes with pre-specified guarantees that they will not fail (i.e., run out of battery). Randomness is directly integrated in the model with respect to various uncertainty sets (e.g., boxed, budgeted and ellipsoidal uncertainty sets). Jeong et al. (2024) have proposed an adaptive robust optimization model for EVs with time windows and uncertain energy consumption rate. A partial recharging policy is adopted with a linear recharging function. Adaptive decisions are made after the uncertainty realization on each arc and are related to the SoC, the service time of the EV at each node, and the battery recharging amount. The aim is to minimize the worst-case energy consumption while ensuring deliveries at the assigned time windows without running out of charge. The model is solved through a column-and-constraint generation based heuristic, which is coupled with a variable neighborhood search-tabu search metaheuristic and alternating direction algorithm. Han et al. (2025) have recently applied adaptive robust optimization techniques to an EVRP with travel time and energy consumption uncertainty. A bi-level search algorithm has been devised to identify robustly feasible routes and provide conservative cost estimates. The algorithm integrates multiple components, including adaptive large neighborhood search, local search, set partitioning, and bidirectional labeling.

Beyond energy consumption, EVRPs may be subject to several other sources of uncertainty, e.g., customer demands, waiting times at CSs and their availability. Schiffer and Walther (2018) have proposed the first robust location-routing problem under uncertain spatial customer distribution, demand, and service time windows. The aim is to optimally route EVs and locate CSs while minimizing the total cost. The resulting problem is solved using a combination between adaptive large neighborhood search and dynamic programming. Uncertain waiting time at CSs have been explored in Keskin et al. (2021), where the EVs may wait in a queue before charging starts. The problem is formulated as a two-stage stochastic program with recourse, allowing some customers to be skipped and served by an extra EV, incurring a penalty cost. A simulation-based adaptive large neighborhood search with destroy-repair operators is applied to solve real case instances. An EVRP with time windows under time uncertainty has been studied in Wang and Zhou (2021). The problem, formulated through robust optimization techniques and based on route-dependent uncertainty sets, is solved through an adaptive large neighborhood search heuristic. Liu et al. (2023) have considered a pickup and delivery problem using EVs under demand uncertainty. The model is formulated as a two-stage adaptive robust model where the uncertain demand falls within a budget-type uncertainty set. The accumulated load variables are treated as adaptive with respect to the demand scenario, while the routing, the service start times and remaining battery capacities along a route are fixed before the realization of uncertain demands. Dastpak et al. (2024) have optimized an EV route from an origin to a destination considering that public CSs occupancy indicators are known in real-time. Whenever new information on the status of a CS is received, the route is dynamically reoptimized.

All the approaches discussed so far and related to EVRPs under uncertainty are reported in Table 1.

In this paper, we present a two-stage stochastic programming model for an EVRP under uncertain energy consumption. In the first stage we determine customer sequences, each of which is to be served by a single EV that starts and ends its route at the depot. In the second stage we include partial recharging operations following the threshold recourse policy. Furthermore, we account for different charging technologies, each modeled by a piecewise linear concave function. We note that most existing studies on the EVRP rely on node-level recharging decisions. However, in real-world operations, recharging decisions often need to be made at any point along a route, depending on the vehicle's state of charge. To the best of our knowledge, the present work and Bezzi et al. (2024) represent the only contributions in the literature in which detours to CSs may be performed from any arbitrary intermediate point of an arc, and not only from customers or depot locations. By allowing arc-level detours, our approach captures this practical aspect of electric vehicle routing and provides greater modeling flexibility. However, with respect to Bezzi et al. (2024) where only small instances with 10 customers were considered and no solution algorithm was provided, in this paper we propose an enhanced model and a heuristic algorithm. As a result, we are able to tackle instances with up to 80 customers within reasonable computational times.

3. Problem description and formulation

Let $\mathcal{I}$ be the set of customers and C be the set of CSs at which the EVs may fully or partially recharge their batteries. The customers are served by a homogeneous fleet of EVs located at a single depot denoted by node 0. To allow multiple visits to a CS, as common in flow-based EVRP formulations, we define a set $\mathcal{K}$ which includes n copies of each CS in C, i.e., $|\mathcal{K}| = n|C|$. Each CS $k \in \mathcal{K}$ is equipped with a charging technology described by a piecewise linear concave charging function with set of breakpoints $\mathcal{B}_k = \{0, \dots, b_k\}$. We denote by c_{kb} and a_{kb} the charging time and the SoC of breakpoint $b \in \mathcal{B}_k$ at CS $k \in \mathcal{K}$, respectively. Let $\mathcal{I}^+ := \mathcal{I} \cup \{0\}$ and $\mathcal{V} := \mathcal{I}^+ \cup \mathcal{K}$. Our problem is defined on the complete graph $\mathcal{G} := (\mathcal{V}, \mathcal{A})$. Let $(X_i, Y_i) \in \mathbb{R}^+ \times \mathbb{R}^+$ be the Cartesian coordinates of node $i \in \mathcal{V}$. Traveling from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$ incurs a deterministic driving time $t_{ij} \in \mathbb{R}^+$, assumed to be linearly correlated with Euclidean distance $d_{ij} \in \mathbb{R}^+$ between nodes i and j. The EVs leave the depot fully charged, with a SoC equal to Q^{max}.

To account for EV energy consumption uncertainty, we consider a set $\mathcal{S}$ of possible energy consumption scenarios. For every scenario $s \in \mathcal{S}$, we define $e_{ijs} \in \mathbb{R}^+$ as the energy consumption along arc $(i, j) \in \mathcal{A}$ under scenario s (i.e., in a given scenario we assume that the energy consumption remains constant along the arc). We denote by $p_s \in [0, 1]$ the probability of scenario $s \in \mathcal{S}$, such

Table 1
A selected literature review on EVRPs under uncertainty.
Charging policy: F: Full; P: Partial; B: Battery swapping.
Solution strategy: AD: Alternating Direction; ALNS: Adaptive Large Neighborhood Search; B&P&C: Branch-and-Price-and-Cut; BL: Bidirectional Labeling; CCG: Column-and-Constraint Generation; CRG: Column-and-Row Generation; DP: Dynamic Programming; LNS: Large Neighborhood Search; LS: Local Search; ML: Machine Learning; ROI: Reoptimization method with Occupancy Indicator information; RL: Reinforcement Learning; SP: Set Partitioning; TS: Tabu Search; VND: Variable Neighborhood Descent; VNS: Variable Neighborhood Search.

Reference	Minimization objective	Modeling approach	Charging policy	Charging function	Source of uncertainty	Recourse policy	Solution strategy
Schiffer and Walther (2018)	Cost	Two-stage adaptive robust optimization	P	Linear function	Customer spatial distribution and demand, service time windows	-	ALNS with DP
Pelletier et al. (2019)	Cost	Static robust optimization	-	-	Energy consumption	-	LNS with DP
Bruni et al. (2020)	Time	Two-stage stochastic optimization	P	Piecewise linear concave function	Energy consumption	Node-level detour	Integer L-shaped method
Basso et al. (2021)	Consumed energy	Two-stage chance constrained optimization	P	Linear function	Energy consumption	-	Bayesian ML
Keskin et al. (2021)	Cost	Two-stage stochastic optimization	F	Linear function	Waiting time at charging stations	Serving skipped customers	ALNS
Wang and Zhou (2021)	Cost	Static robust optimization	F	Linear function	Travel time	-	ALNS
Basso et al. (2022)	Cost	Markov decision process formulation	F	Linear function	Customer demand and energy consumption	-	RL, approximate DP
Liu et al. (2023)	Cost	Two-stage adaptive robust optimization	B	-	Customer demand	-	CRG with B&P&C, DP and BL algorithm, with filtering through lower bounds
Bezzi et al. (2024)	Time	Two-stage stochastic optimization	P	Piecewise linear concave function	Energy consumption	Arc-level detour	Commercial solver
Dastpak et al. (2024)	Time	Markov decision process formulation	P	Piecewise linear concave function	Waiting time at charging stations	-	ROI
Jeong et al. (2024)	Consumed energy	Two-stage adaptive robust optimization	P	Linear function	Energy consumption	-	CCG with VNS-TS and AD algorithm
Han et al. (2025)	Cost	Two-stage adaptive robust optimization	P	Linear function	Energy consumption and travel time	-	ALNS, LS, SP and BL algorithm
Current work	Time	Two-stage stochastic optimization	P	Piecewise linear concave function	Energy consumption	Arc-level detour	ILS with VND, SP and filtering through lower bounds

that $\sum_{s\in S} p_s = 1$. Details on the generation of the energy consumption scenarios adopted in our numerical study are provided in Section 6.

We model the SEVRP-T as a two-stage stochastic program with the objective of minimizing the expected total EV travel times and charging times. The first-stage decisions consist of routes satisfying the following conditions: 1) each customer is served exactly once by a single vehicle; 2) each route starts and ends at the depot. First-stage routes are described by binary decision variables x_{ij}, taking value 1 if arc (i, j) with $i, j \in \mathcal{I}^+, i \neq j$ is traversed and 0 otherwise. No en route charging operations are considered in the first stage. Furthermore, we assume that the visiting sequence determined in the first stage must be respected in the second stage. This assumption aligns with the notion of *a priori* routes, commonly considered in the stochastic VRP literature (e.g., Oyola et al., 2018). Indeed, a priori routes are useful when delivery times are communicated to customers prior to starting the route. Maintaining the sequence indirectly helps in preserving such communicated arrival times. The actual energy consumption e_{ijs} in scenario $s \in S$ of arc $(i, j) \in \mathcal{A}$ is revealed when it starts being traversed by an EV at the second stage, and the energy consumption per unit of distance is presumed to be equal along the arc in scenario s (i.e., $\frac{e_{ijs}}{d_{ij}}$). It is worth noting that, while energy consumption can vary due to several factors, e.g., differences in terrain, road type, driving behaviour, external traffic dynamics, and particularly acceleration variance (see Montanino et al., 2025), the assumption of uniform energy consumption per unit of distance is widely adopted in the EVRP literature (see, e.g., Erdoğan and Miller-Hooks, 2012; Desaulniers et al., 2016; Pelletier et al., 2019; Jeong et al., 2024). This simplification echos the general assumption, adopted in the vast majority of the routing literature, that an arc's travel time is linearly proportional to its distance. Furthermore, such assumptions of proportionality are often made due to lack of actual data. This is further compounded by the fact that an arc in a routing problem is an abstraction of the shortest path on a road network between two nodes. However, after the presentation of the mathematical model, we discuss how non-uniform energy consumption may be incorporated in our model.

As the realized energy consumption may lead the EV to run out of battery, we propose the following threshold recourse policy to govern the second stage. Whenever the SoC of an EV drops to $Q^T < Q^{max}$, while driving towards a node or at a node, the EV determines a CS to perform en route charging. Intuitively, Q^T may be viewed as the tolerance up to which an EV driver feels the necessity to recharge. This mimics the behavior of conventional vehicle drivers who tend to look for refueling stations after a certain level of fuel is reached. The EV will head to its chosen CS, recharge and visit the subsequent customer on its a priori route. We assume that, when performing a detour along arc (i, j) with $i \in \mathcal{I}^+, j \in \mathcal{I}, i \neq j$, the energy consumption from the detour point (i.e., once the SoC of an EV drops to Q^T) to the chosen CS and from the CS to customer $j \in \mathcal{I}$ follows the same energy consumption of arc (i, j) under scenario $s \in S$, which equals $\frac{e_{ijs}}{d_{ij}}$. The amount of energy to charge is such that the EV's SoC upon arriving at the subsequent customer is exactly $Q^G > Q^T$. Additionally, we allow the SoC of each EV to be lower than Q^T when returning to the depot in the last arc of the planned route. Lastly, we impose that at most a single visit to a CS between two consecutive nodes should be performed.

In order to model the threshold recourse policy, we introduce binary variables ϕ_{ijks} and continuous variables z_{ijs} for each arc (i, j) with $i, j \in \mathcal{I}^+, i \neq j$ in each scenario $s \in S$: variables ϕ_{ijks} take value 1 if the vehicle traversing arc (i, j) detours towards CS $k \in \mathcal{K}$ under scenario $s \in S$ and 0 otherwise, while $z_{ijs} \in [0, 1]$ represent the portion of arc (i, j) actually traversed from node i to the detour point i.e., the point when the SoC reaches Q^T. Note that variables z_{ijs} do not depend on the CS. Auxiliary variables $(X_{ijs}, Y_{ijs}) \in \mathbb{R}^+ \times \mathbb{R}^+$ keep track of the Euclidean coordinates of the detour point in arc (i, j) with $i, j \in \mathcal{I}^+, i \neq j$ under scenario $s \in S$, while ℓ_{ijks} represent the distance between the detour point and CS $k \in \mathcal{K}$. Finally, we make use of classical node-based tracking variables for piecewise charging function (see Froger et al., 2019): $y_{is} \in \mathbb{R}^+$ represent the SoC of a vehicle when it departs from node $i \in \mathcal{V}$ under scenario $s \in S$; $\underline{q}_{ks}, \overline{q}_{ks} \in \mathbb{R}^+$ represent the SoC of a vehicle when it enters/leaves CS $k \in \mathcal{K}$ under scenario $s \in S$, respectively; $\underline{c}_{ks}, \overline{c}_{ks} \in \mathbb{R}^+$ represent scaled start/end time for charging a vehicle at CS $k \in \mathcal{K}$ under scenario $s \in S$, according to the charging function of CS k, respectively; $\underline{w}_{kbs}, \overline{w}_{kbs} \in \{0, 1\}$ take value 1 if the SoC of a vehicle is between $a_{k(b-1)}$ and a_{kb}, with $b \in \mathcal{B}_k \setminus \{0\}$, when entering/leaving CS $k \in \mathcal{K}$ under scenario $s \in S$, respectively, and 0 otherwise; $\underline{\lambda}_{kbs}, \overline{\lambda}_{kbs} \in \mathbb{R}^+$ are continuous coefficients associated with breakpoint $b \in \mathcal{B}_k$ under scenario $s \in S$ when the EV enters/leaves CS $k \in \mathcal{K}$, respectively.

To demonstrate the benefits of the proposed arc-based threshold recourse policy, we consider an example involving three customers and two CSs (see Figs. 1 and 2). For the sake of comparability, we define a node-based threshold recourse policy by restricting the continuous variable z_{ijs} to be binary. This entails that CSs are visited strictly from the nodes of the network. Specifically, upon arrival to a node $i \in \mathcal{I}^+$ the policy verifies if arriving directly to $j \in \mathcal{I}$ under scenario $s \in S$ would yield a SoC that is less than Q^T. In which case, a detour is performed and the amount of energy charged is such that the SoC at j is equal to Q^G. Notably, this change entails that from node i decisions are made based on the knowledge of the energy consumption along arc (i, j) under scenario s. As such, the node-based threshold policy uses anticipatory information when compared to our arc-based policy. In Fig. 1 we consider the node-based threshold recourse policy, while in Fig. 2 the arc-based alternative. In both examples, CS k_1 provides one unit of charge every two units of time, whereas CS k_2 provides one unit of charge per unit of time, i.e., k_2 has a faster technology than k_1. In both cases, the first-stage route is $\{0, 1, 2, 3, 0\}$, with energy parameters $Q^{max} = 10, Q^T = 2$ and $Q^G = 7$. Under the node-based threshold recourse policy (see Fig. 1), upon arrival at customer node 2, the SoC is 5. According to the node-based policy, this value is insufficient to reach either customer node 3 or CS k_2, requiring a detour to CS k_1 to recharge the battery up to a SoC of 10, ensuring a SoC of $Q^G = 7$ upon reaching node 3. Conversely, under the arc-based threshold policy (see Fig. 2), a detour to CS k_2 is sufficient to complete the planned route. Note that, although CS k_1 is closer to the detour point than k_2, its slower charging technology makes k_2 the preferred choice when minimizing total time. Since both CSs are reachable from the detour point, the model opts for the faster one. Additionally, the detour starts from the point of arc $(2, 3)$ where the SoC reaches the threshold level Q^T, allowing recharging decisions to be made adaptively along the route. This additional flexibility leads to routes of shorter total duration: in this example, assuming that one unit of energy is consumed per unit of time, the total time decreases from 31 under the node-based policy to 23 under the arc-based policy.

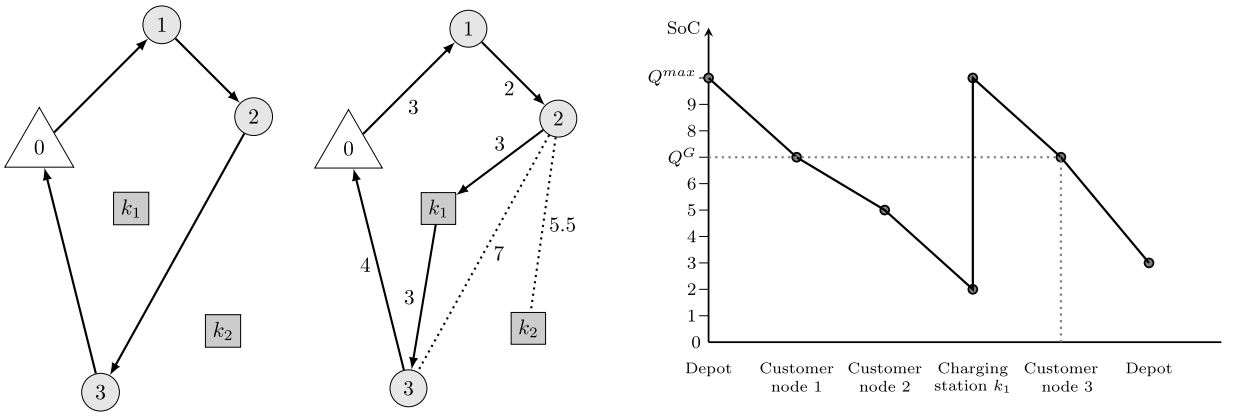


Fig. 1. An example with three customers $(1, 2, 3)$ and two CSs (k_1, k_2) under the node-based threshold recourse policy. Left panel: the first-stage route $\{0, 1, 2, 3, 0\}$ is determined. Middle panel: after the realization of the uncertain energy consumption (values shown on the arcs), a stop at CS k_1 is prescribed to recharge the battery. Right panel: variations in the SoC of the EV in the second stage.

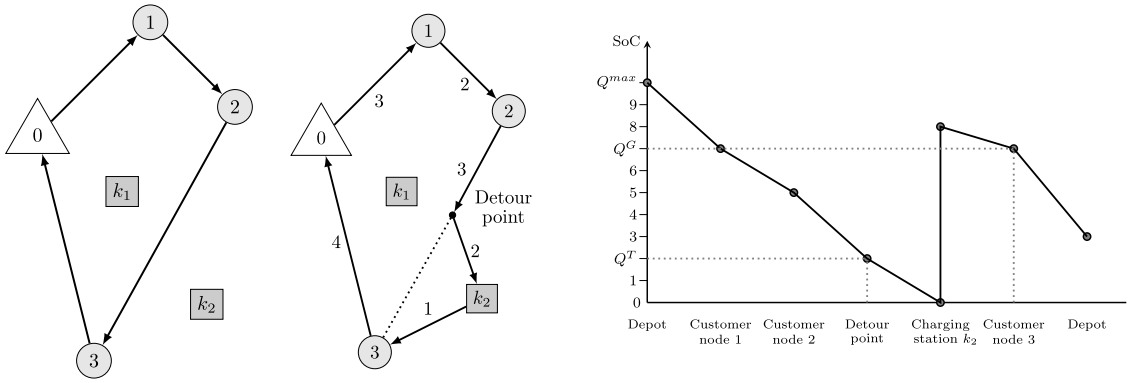


Fig. 2. An example with three customers $(1, 2, 3)$ and two CSs (k_1, k_2) under the arc-based threshold recourse policy. Left panel: the first-stage route $\{0, 1, 2, 3, 0\}$ is determined. Middle panel: after the realization of the uncertain energy consumption (values shown on the arcs), a stop at CS k_2 between customer nodes 2 and 3 is prescribed to recharge the battery. Right panel: variations in the SoC of the EV in the second stage.

As discussed, we represent the charging process at CS $k \in \mathcal{K}$ through a piecewise linear concave function with set of breakpoints $\mathcal{B}_k$. This approach is widely adopted in the EVRP literature (see, e.g., Montoya et al., 2017; Klein and Schiffer, 2023 and Dastpak et al., 2024) as it allows the integration of a more realistic charging process, when compared to strictly linear charging functions. The number and placement of breakpoints are selected to keep the maximum deviation from the true nonlinear charging curve within a small error (see, e.g., Montoya et al., 2017), thereby limiting any bias in the estimation of SoC levels. At the same time, we treat energy consumption as stochastic and handle it independently in the second stage. Therefore, approximation errors from the charging function do not directly amplify the variability introduced by stochastic consumption. When higher accuracy is required, the approximation can be refined by increasing the number of breakpoints without modifying the operational solution framework. The implementation of the proposed threshold recourse policy contributes to incorporating stochastic energy consumption within the charging process. In particular, charging decisions are scenario-dependent in the second stage and are determined based on the SoC of the EV.

The notation adopted in this work is summarized in Table 2.

In the following, we present the extensive formulation of the SEVRP-T, modeled as a two-stage stochastic mixed-integer second-order cone program. We first describe the objective function, then we present the families of constraints.

Objective function

$$\min \sum_{i,j \in \mathcal{I}^+, i \neq j} t_{ij} x_{ij} + \sum_{s \in S} p_s \left(\sum_{k \in \mathcal{K}} (\bar{c}_{ks} - \underline{c}_{ks}) + \sum_{i,j \in \mathcal{I}^+, i \neq j} t_{ij} z_{ijs} + \sum_{i,j \in \mathcal{I}^+, i \neq j} \sum_{k \in \mathcal{K}} \left(\frac{t_{ij}}{d_{ij}} \ell_{ijks} + (t_{kj} - t_{ij}) \phi_{ijks} \right) \right) \tag{1}$$

The objective function Eq. (1) minimizes the expected total duration of the route, defined as the sum of travel times along arcs included in the first-stage route, plus any additional expected time due to the recourse actions taken in the second stage (i.e., the charging time, the time to reach the detour point, the time to arrive at the chosen CS, and the time to reach the next customer on the route).

Table 2
Sets, parameters and decision variables.

Sets	
$\mathcal{I}$	Set of customers.
$\mathcal{I}^+ := \mathcal{I} \cup \{0\}$	Set of customers and depot.
$\mathcal{C}$	Set of Charging Stations (CS).
$\mathcal{K}$	Set of duplicated charging stations.
$\mathcal{B}_k$	Set of breakpoints for the piecewise linear charging function of CS $k \in \mathcal{K}$.
$\mathcal{V} := \mathcal{I}^+ \cup \mathcal{K}$	Set of all nodes.
$\mathcal{A}$	Set of all arcs.
$\mathcal{S}$	Set of energy consumption scenarios.
Deterministic Parameters	
$(X_i, Y_i) \in \mathbb{R}^+ \times \mathbb{R}^+$	Euclidean coordinates of node $i \in \mathcal{V}$.
$d_{ij} \in \mathbb{R}^+$	Distance between nodes $i \in \mathcal{V}$ and $j \in \mathcal{V}$.
$t_{ij} \in \mathbb{R}^+$	Time consumption between nodes $i \in \mathcal{V}$ and $j \in \mathcal{V}$.
$Q^{max} \in \mathbb{R}^+$	Maximum battery charge level.
$Q^T \in \mathbb{R}^+$	Minimum threshold battery charge level, with $Q^T < Q^{max}$.
$Q^G \in \mathbb{R}^+$	Goal level of energy after each detour, with $Q^T < Q^G < Q^{max}$.
$c_{kb} \in \mathbb{R}^+$	Charging time of breakpoint $b \in \mathcal{B}_k$ of CS $k \in \mathcal{K}$.
$a_{kb} \in \mathbb{R}^+$	State of Charge (SoC) of breakpoint $b \in \mathcal{B}_k$ of CS $k \in \mathcal{K}$.
$M \in \mathbb{R}^+$	A Big-M number.
Stochastic Parameters	
$p_s \in [0,1]$	Probability of scenario $s \in \mathcal{S}$.
$e_{ijs} \in \mathbb{R}^+$	Energy consumption between nodes $i \in \mathcal{V}$ and $j \in \mathcal{V}$ under scenario $s \in \mathcal{S}$.
Decision Variables	
$x_{ij} \in \{0,1\}$	1 if the EV traverses arc (i,j) with $i,j \in \mathcal{I}^+, i \neq j$, 0 otherwise.
$u_i \in \mathbb{R}^+$	Dummy variable for subtour elimination constraints involving customer $i \in \mathcal{I}^+$.
$\phi_{ijks} \in \{0,1\}$	1 if the EV detours along arc (i,j) with $i,j \in \mathcal{I}^+, i \neq j$ towards CS $k \in \mathcal{K}$ under scenario $s \in \mathcal{S}$, 0 otherwise.
$z_{ijs} \in [0,1]$	Percentage of arc (i,j) with $i,j \in \mathcal{I}^+, i \neq j$ covered from node i to detour point under scenario $s \in \mathcal{S}$.
$y_{is} \in \mathbb{R}^+$	SoC of the EV when it departs from node $i \in \mathcal{V}$ under scenario $s \in \mathcal{S}$.
$\ell'_{ijks} \in \mathbb{R}^+$	Distance between the detour point along arc (i,j) with $i,j \in \mathcal{I}^+, i \neq j$ to CS $k \in \mathcal{K}$ under scenario $s \in \mathcal{S}$.
$(X_{ijs}, Y_{ijs}) \in \mathbb{R}^+ \times \mathbb{R}^+$	Euclidean coordinates of the detour point in arc (i,j) with $i,j \in \mathcal{I}^+, i \neq j$ under scenario $s \in \mathcal{S}$.
$\underline{q}_{ks} \in \mathbb{R}^+$	SoC of the EV when it enters CS $k \in \mathcal{K}$ under scenario $s \in \mathcal{S}$.
$\overline{q}_{ks} \in \mathbb{R}^+$	SoC of the EV when it leaves CS $k \in \mathcal{K}$ under scenario $s \in \mathcal{S}$.
$\underline{c}_{ks} \in \mathbb{R}^+$	Scaled start time for charging the EV at CS $k \in \mathcal{K}$ under scenario $s \in \mathcal{S}$.
$\overline{c}_{ks} \in \mathbb{R}^+$	Scaled end time for charging the EV at CS $k \in \mathcal{K}$ under scenario $s \in \mathcal{S}$.
$\underline{w}_{kbs} \in \{0,1\}$	1 if the SoC of the EV is between $a_{k(b-1)}$ and a_{kb}, with $b \in \mathcal{B}_k \setminus \{0\}$, when entering CS $k \in \mathcal{K}$ under scenario $s \in \mathcal{S}$, 0 otherwise.
$\overline{w}_{kbs} \in \{0,1\}$	1 if the SoC of the EV is between $a_{k(b-1)}$ and a_{kb}, with $b \in \mathcal{B}_k \setminus \{0\}$, when leaving CS $k \in \mathcal{K}$ under scenario $s \in \mathcal{S}$, 0 otherwise.
$\underline{\lambda}_{kbs} \in \mathbb{R}^+$	Continuous coefficient associated with breakpoint $b \in \mathcal{B}_k$ under scenario $s \in \mathcal{S}$ when the EV enters CS $k \in \mathcal{K}$.
$\overline{\lambda}_{kbs} \in \mathbb{R}^+$	Continuous coefficient associated with breakpoint $b \in \mathcal{B}_k$ under scenario $s \in \mathcal{S}$ when the EV leaves CS $k \in \mathcal{K}$.

Routing constraints

$$\sum_{j \in \mathcal{I}^+, j \neq i} x_{ij} = 1 \qquad \forall i \in \mathcal{I} \tag{2}$$

$$\sum_{j \in \mathcal{I}^+, j \neq i} x_{ij} - \sum_{j \in \mathcal{I}^+, j \neq i} x_{ji} = 0 \qquad \forall i \in \mathcal{I}^+ \tag{3}$$

$$u_i - u_j + |\mathcal{I}| x_{ij} \leq |\mathcal{I}| - 1 \qquad \forall i,j \in \mathcal{I}^+, i \neq j \tag{4}$$

Constraints (2)–(4) involve only first-stage decision variables. In particular, constraints (2) ensure that each customer is visited once, constraints (3) ensure route connectivity, and constraints (4) eliminate subtours.

Detour and station visit constraints

$$\sum_{i,j \in \mathcal{I}^+, i \neq j} \phi_{ijks} \leq 1 \qquad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{5}$$

$$\sum_{k \in \mathcal{K}} \phi_{ijks} \leq x_{ij} \qquad \forall i,j \in \mathcal{I}^+, i \neq j, s \in \mathcal{S} \tag{6}$$

$$z_{ijs} \leq \sum_{k \in \mathcal{K}} \phi_{ijks} \qquad \forall i, j \in \mathcal{I}^+, i \neq j, s \in \mathcal{S} \tag{7}$$

$$\ell_{ijks} \leq M \phi_{ijks} \qquad \forall i, j \in \mathcal{I}^+, i \neq j, k \in \mathcal{K}, s \in \mathcal{S} \tag{8}$$

Constraints (5) ensure that every CS is visited at most once. Constraints (6) forbid to detour along arcs not included in the first-stage route and ensure that every first-stage arc can be subject to at most one detour. Constraints (7) and (8) ensure that, when there is no detour along arc (i, j), the associated detour variables z_{ijs} and ℓ_{ijks} are set to zero. Parameter M is a sufficiently large constant.

Energy tracking without detours constraints

$$y_{js} \geq Q^T \left(x_{ij} - \sum_{k \in \mathcal{K}} \phi_{ijks}\right) \qquad \forall i \in \mathcal{I}^+, j \in \mathcal{I}, i \neq j, s \in \mathcal{S} \tag{9}$$

$$y_{ks} \leq Q^{max} \sum_{i,j \in \mathcal{I}^+, i \neq j} \phi_{ijks} \qquad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{10}$$

$$y_{is} - y_{js} \geq e_{ijs}\left(x_{ij} - \sum_{k \in \mathcal{K}} \phi_{ijks}\right) - Q^{max}\left(1 - (x_{ij} - \sum_{k \in \mathcal{K}} \phi_{ijks})\right) \qquad \forall i \in \mathcal{I}^+, j \in \mathcal{I}, i \neq j, s \in \mathcal{S} \tag{11}$$

$$y_{is} - y_{js} \leq e_{ijs}\left(x_{ij} - \sum_{k \in \mathcal{K}} \phi_{ijks}\right) + Q^{max}\left(1 - (x_{ij} - \sum_{k \in \mathcal{K}} \phi_{ijks})\right) \qquad \forall i \in \mathcal{I}^+, j \in \mathcal{I}, i \neq j, s \in \mathcal{S} \tag{12}$$

$$y_{is} \geq e_{i0s}\left(x_{i0} - \sum_{k \in \mathcal{K}} \phi_{i0ks}\right) - Q^{max}\left(1 - (x_{i0} - \sum_{k \in \mathcal{K}} \phi_{i0ks})\right) \qquad \forall i \in \mathcal{I}, s \in \mathcal{S} \tag{13}$$

Constraints (9) ensure that when the EV traverses arc (i, j) with $i \in \mathcal{I}^+, j \in \mathcal{I}, i \neq j$ its SoC at node j is always greater than Q^T if no detour operations are performed. Constraints (10) set Q^{max} as upper bound on the maximum battery level for each visited station and 0 for each non-visited station. Constraints (11) and (12) ensure that, for every first-stage arc (i, j) with $i \in \mathcal{I}^+, j \in \mathcal{I}, i \neq j$, the difference between the SoCs at nodes i and j is equal to the energy consumption along (i, j), if there is no detour along the arc. Constraints (13) are the equivalent for the last arc of every route where we allow the EV to return to the depot with an empty battery.

Energy tracking with detours constraints

$$y_{ks} - y_{js} \geq e_{kjs} \sum_{i \in \mathcal{I}^+, i \neq j} \phi_{ijks} - Q^{max}\left(1 - \sum_{i \in \mathcal{I}^+, i \neq j} \phi_{ijks}\right) \qquad \forall j \in \mathcal{I}, k \in \mathcal{K}, s \in \mathcal{S} \tag{14}$$

$$y_{ks} - y_{js} \leq e_{kjs} \sum_{i \in \mathcal{I}^+, i \neq j} \phi_{ijks} + Q^{max}\left(1 - \sum_{i \in \mathcal{I}^+, i \neq j} \phi_{ijks}\right) \qquad \forall j \in \mathcal{I}, k \in \mathcal{K}, s \in \mathcal{S} \tag{15}$$

$$y_{ks} \geq e_{k0s} \sum_{i \in \mathcal{I}} \phi_{i0ks} - Q^{max}\left(1 - \sum_{i \in \mathcal{I}} \phi_{i0ks}\right) \qquad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{16}$$

$$y_{js} \geq Q^G - Q^{max}\left(1 - \sum_{i \in \mathcal{I}^+, k \in \mathcal{K}, i \neq j} \phi_{ijks}\right) \qquad \forall j \in \mathcal{I}, s \in \mathcal{S} \tag{17}$$

$$y_{js} \leq Q^G + Q^{max}\left(1 - \sum_{i \in \mathcal{I}^+, k \in \mathcal{K}, i \neq j} \phi_{ijks}\right) \qquad \forall j \in \mathcal{I}, s \in \mathcal{S} \tag{18}$$

$$e_{ijs} z_{ijs} \geq y_{is} - Q^T - Q^{max}(1 - \phi_{ijks}) \qquad \forall i, j \in \mathcal{I}^+, i \neq j, k \in \mathcal{K}, s \in \mathcal{S} \tag{19}$$

$$e_{ijs} z_{ijs} \leq y_{is} - Q^T + Q^{max}(1 - \phi_{ijks}) \qquad \forall i, j \in \mathcal{I}^+, i \neq j, k \in \mathcal{K}, s \in \mathcal{S} \tag{20}$$

$$\underline{q}_{ks} = \sum_{i,j \in \mathcal{I}^+, i \neq j} \left(Q^T \phi_{ijks} - \frac{e_{ijs}}{d_{ij}} \ell_{ijks}\right) \qquad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{21}$$

$$y_{ks} = \overline{q}_{ks} \qquad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{22}$$

$$y_{0s} = Q^{max} \qquad \forall s \in \mathcal{S} \tag{23}$$

Constraints (14) and (15) determine the SoC of the vehicle when returning to customer $j \in \mathcal{I}$ after a detour. Constraints (16) are equivalent to the previous ones but hold only for the last arc of each route. Constraints (14)–(16) use Q^{max} as an upper bound for the SoC. Constraints (17) and (18) enforce the recharge policy for every arc except for the last one of each route. In particular, if a detour occurs when traversing arc (i, j), with $i \in \mathcal{I}^+$ and $j \in \mathcal{I}$, then the SoC at node j must be equal to Q^G. Constraints (19) and (20) fix the detour point, ensuring that every detour starts only when the vehicle reaches battery level Q^T. Constraints (21) establish the SoC of a vehicle when it enters station $k \in \mathcal{K}$. The SoC is equal to 0 if the station is never visited. Constraints (22) update the SoC of the vehicle when leaving a CS. Constraints (23) set the battery charge level to Q^{max} at the depot.

Charging process constraints

- Arriving at a CS

$$\underline{q}_{ks} = \sum_{b \in \mathcal{B}_k} \underline{\lambda}_{kbs} a_{kb} \qquad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{24}$$

$$\underline{c}_{ks} = \sum_{b \in \mathcal{B}_k} \underline{\lambda}_{kbs} c_{kb} \qquad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{25}$$

$$\sum_{b\in\mathcal{B}_k} \underline{\lambda}_{kbs} = \sum_{b\in\mathcal{B}_k\setminus\{0\}} \underline{w}_{kbs} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{26}$$

$$\sum_{b\in\mathcal{B}_k\setminus\{0\}} \underline{w}_{kbs} = \sum_{i,j\in\mathcal{I}^+, i\neq j} \phi_{ijks} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{27}$$

$$\underline{\lambda}_{k0s} \leq \underline{w}_{k1s} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{28}$$

$$\underline{\lambda}_{kbs} \leq \underline{w}_{kbs} + \underline{w}_{k(b+1)s} \qquad \forall k\in\mathcal{K}, b\in\mathcal{B}_k\setminus\{0,b_k\}, s\in\mathcal{S} \tag{29}$$

$$\underline{\lambda}_{kb_ks} \leq \underline{w}_{kb_ks} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{30}$$

- Leaving from a CS

$$\overline{q}_{ks} = \sum_{b\in\mathcal{B}_k} \overline{\lambda}_{kbs} a_{kb} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{31}$$

$$\overline{c}_{ks} = \sum_{b\in\mathcal{B}_k} \overline{\lambda}_{kbs} c_{kb} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{32}$$

$$\sum_{b\in\mathcal{B}_k} \overline{\lambda}_{kbs} = \sum_{b\in\mathcal{B}_k\setminus\{0\}} \overline{w}_{kbs} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{33}$$

$$\sum_{b\in\mathcal{B}_k\setminus\{0\}} \overline{w}_{kbs} = \sum_{i,j\in\mathcal{I}^+, i\neq j} \phi_{ijks} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{34}$$

$$\overline{\lambda}_{k0s} \leq \overline{w}_{k1s} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{35}$$

$$\overline{\lambda}_{kbs} \leq \overline{w}_{kbs} + \overline{w}_{k(b+1)s} \qquad \forall k\in\mathcal{K}, b\in\mathcal{B}_k\setminus\{0,b_k\}, s\in\mathcal{S} \tag{36}$$

$$\overline{\lambda}_{kb_ks} \leq \overline{w}_{kb_ks} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{37}$$

$$\underline{q}_{ks} \leq \overline{q}_{ks} \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{38}$$

Following the approach in Froger et al. (2019), constraints (24)–(30) establish the SoC and the charging time of the EV when arriving at CS $k\in\mathcal{K}$, based on the piecewise linear approximation of its charging function. Similarly, constraints (31)–(38) define the SoC and charging time upon departure from the same CS $k\in\mathcal{K}$.

Detour point coordinates and length constraints

$$X_{ijs} = X_j z_{ijs} + X_i(1-z_{ijs}) \qquad \forall i,j\in\mathcal{I}^+, i\neq j, s\in\mathcal{S} \tag{39}$$

$$Y_{ijs} = Y_j z_{ijs} + Y_i(1-z_{ijs}) \qquad \forall i,j\in\mathcal{I}^+, i\neq j, s\in\mathcal{S} \tag{40}$$

$$\ell_{ijks} + M(1-\phi_{ijks}) \geq \sqrt{(X_{ijs}-X_k)^2 + (Y_{ijs}-Y_k)^2} \qquad \forall i,j\in\mathcal{I}^+, k\in\mathcal{K}, s\in\mathcal{S} \tag{41}$$

Constraints (39) and (40) determine the Euclidean coordinates (X_{ijs}, Y_{ijs}) of the detour point along arc (i,j) with $i,j\in\mathcal{I}^+, i\neq j$ under scenario $s\in\mathcal{S}$. Such coordinates are derived from the Internal Section Formula based on the variable z_{ijs}. Constraints (41) are second-order cone constraints that define the distance ℓ_{ijks} between the detour point along arc (i,j) and the selected charging station $k\in\mathcal{K}$ under scenario $s\in\mathcal{S}$. Despite being nonlinear, these constraints can be handled efficiently by commercial solvers (see Maggioni et al., 2009).

Domain of decision variables

$$x_{ij} \in \{0,1\} \qquad \forall i,j\in\mathcal{I}^+, i\neq j \tag{42}$$

$$u_i \geq 0 \qquad \forall i\in\mathcal{I}^+ \tag{43}$$

$$\phi_{ijks} \in \{0,1\} \qquad \forall i,j\in\mathcal{I}^+, i\neq j, k\in\mathcal{K}, s\in\mathcal{S} \tag{44}$$

$$0 \leq z_{ijs} \leq 1 \qquad \forall i,j\in\mathcal{I}^+, i\neq j, s\in\mathcal{S} \tag{45}$$

$$\ell_{ijks} \geq 0 \qquad \forall i,j\in\mathcal{I}^+, i\neq j, k\in\mathcal{K}, s\in\mathcal{S} \tag{46}$$

$$X_{ijs}, Y_{ijs} \geq 0 \qquad \forall i,j\in\mathcal{I}^+, i\neq j, s\in\mathcal{S} \tag{47}$$

$$y_{is} \geq 0 \qquad \forall i\in\mathcal{V}, s\in\mathcal{S} \tag{48}$$

$$\underline{q}_{ks}, \overline{q}_{ks}, \underline{c}_{ks}, \overline{c}_{ks} \geq 0 \qquad \forall k\in\mathcal{K}, s\in\mathcal{S} \tag{49}$$

$$\underline{\lambda}_{kbs}, \overline{\lambda}_{kbs} \geq 0 \qquad \forall k\in\mathcal{K}, b\in\mathcal{B}_k, s\in\mathcal{S} \tag{50}$$

$$\underline{w}_{kbs}, \overline{w}_{kbs} \in \{0,1\} \qquad \forall k\in\mathcal{K}, b\in\mathcal{B}_k\setminus\{0\}, s\in\mathcal{S} \tag{51}$$

Constraints (42)–(51) define the domains of all decision variables. In particular, constraints (42) concern the first-stage routing variables x_{ij}, with $i,j\in\mathcal{I}^+, i\neq j$, while constraints (43) set the domain of the dummy variables u_i, with $i\in\mathcal{I}^+$, used in subtour elimination constraints. The remaining constraints (44)–(51) specify the domain of all second-stage decision variables.

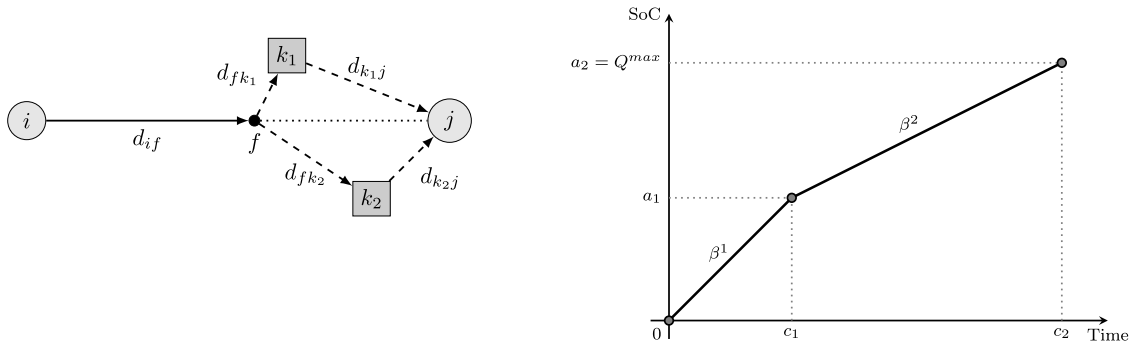


Fig. 3. Example in which the chosen CS is not the closest one to the detour point f. Left panel: graph with relevant nodes and distances. Customers are represented as circles, while CSs are squares. Right panel: charging function of the CSs k_1 and k_2.

Note that the energy consumption along arc (i, j) under scenario s is represented by the parameter e_{ijs}. The value of e_{ijs} can, in principle, be derived from any continuous energy consumption function that specifies the total energy consumed by traversing arc $(i, j) \in \mathcal{I}^+ \times \mathcal{I}^+$ under scenario $s \in \mathcal{S}$. In our model, however, we assume that the total consumed energy (i.e., e_{ijs}) is uniformly distributed per unit of distance traversed along that arc. As for detour path along (i, j) toward a CS k under scenario s, our model may accommodate any energy consumption rate from the point of failure to k, and from k to j. Specifically, this can be achieved by replacing $\frac{e_{ijs}}{d_{ij}}$ in constraint (21) with an alternative rate. However, since CS detours are likely to be in the vicinity of arc (i, j), we assume this rate to be equal to $\frac{e_{ijs}}{d_{ij}}$.

With respect to the formulation proposed in Bezzi et al. (2024), we relax the assumption that the nominal energy consumption of the first-stage routes should not exceed Q^{max}. Additionally, we allow the SoC of each EV to fall below the threshold Q^T upon returning to the depot in the final arc of the route. This choice enhances the model's flexibility, providing a more realistic representation of EV operations.

Observation. In the SEVRP-T, the objective is the minimization of the expected total duration of the route, composed of travel times and charging times. Since we are considering a piecewise linear concave charging function, when the energy threshold is attained by the EV, the CSs chosen in the second stage are not necessarily the closest to the detour point. We formalize this observation in the following proposition.

Proposition 1. *Let $i \in \mathcal{I}^+$, $j \in \mathcal{I}$ and $k_1, k_2 \in \mathcal{K}$. Suppose that CSs k_1, k_2 have the same technology, i.e. $\mathcal{B}_{k_1} = \mathcal{B}_{k_2} = \mathcal{B} = \{0, 1, 2\}$ and $c_{k_1b} = c_{k_2b} = c_b$, $a_{k_1b} = a_{k_2b} = a_b$ for all $b \in \mathcal{B}$. The corresponding piecewise linear charging function is defined as follows (see Fig. 3):*

$$SoC(t) = \begin{cases} \beta^1 t & if \quad 0 \le t \le c_1 \\ \beta^2 (t - c_2) + Q^{max} & if \quad c_1 < t \le c_2, \end{cases} \tag{52}$$

with $\beta^1 = a_1/c_1 \ge 1$ and $\beta^2 = (Q^{max} - a_1)/(c_2 - c_1) < \beta^1$. Assume that when traversing arc (i, j) under scenario $s \in \mathcal{S}$, the EV reaches a SoC equal to Q^T at point f, such that the total distance to complete the path $f - k_1 - j$ is shorter than that of path $f - k_2 - j$, i.e., $d_{fk_1} + d_{k_1j} < d_{fk_2} + d_{k_2j}$. If the following conditions hold:

$$0 \le Q^T - e_{fks} < a_1 \qquad \forall k \in \{k_1, k_2\} \tag{53}$$

$$Q^{max} - e_{kjs} \ge Q^G \qquad \forall k \in \{k_1, k_2\} \tag{54}$$

$$(t_{fk_1} - t_{fk_2}) + \frac{e_{ijs}}{\beta^1 d_{ij}}(d_{fk_1} - d_{fk_2}) + \frac{1}{\beta^2}(e_{k_1js} - e_{k_2js}) + (t_{k_1j} - t_{k_2j}) > 0, \tag{55}$$

then recharging at CS k_1 takes more time than recharging at CS k_2.

Proof. Let T^k be the total detour time, i.e.:

$$T^k = t_{if} + t_{fk} + t_k + t_{kj}, \tag{56}$$

where t_k is the total recharging time at CS k. By hypothesis (53), t_k can be expressed as the sum of two components: t_k^1, representing the time required to reach a SoC of a_1, and t_k^2, representing the time required to reach a SoC of $Q^G + e_{kjs}$. Note that the charging operations are always feasible under hypotheses (53) and (54). Therefore, using the charging function given in Eq. (52), we obtain:

$$t_k^1 = c_1 - \frac{1}{\beta^1}(Q^T - e_{fks})$$

$$t_k^2 = \frac{1}{\beta^2}(Q^G + e_{kjs} - a_1),$$

with $k \in \{k_1, k_2\}$. Thus, the total time Eq. (56) for each CS corresponds to:

$$T^{k_1} = t_{if} + t_{fk_1} + c_1 - \frac{1}{\beta^1}Q^T + \frac{1}{\beta^1}e_{fk_1s} + \frac{1}{\beta^2}Q^G + \frac{1}{\beta^2}e_{k_1js} - \frac{1}{\beta^2}a_1 + t_{k_1j}$$

$$T^{k_2} = t_{if} + t_{fk_2} + c_1 - \frac{1}{\beta^1}Q^T + \frac{1}{\beta^1}e_{fk_2s} + \frac{1}{\beta^2}Q^G + \frac{1}{\beta^2}e_{k_2js} - \frac{1}{\beta^2}a_1 + t_{k_2j}.$$

Subtracting the two previous expressions, yields:

$$T^{k_1} - T^{k_2} = (t_{fk_2} - t_{fk_1}) + \frac{1}{\beta^1}(e_{fk_1s} - e_{fk_2s}) + \frac{1}{\beta^2}(e_{k_1js} - e_{k_2js}) + (t_{k_1j} - t_{k_2j}). \tag{57}$$

According to the model's assumption of uniform energy consumption from the detour point f to each CS and from the CS to the subsequent customer, we have that:

$$e_{fks} = \frac{e_{ijs}}{d_{ij}}d_{fk} \qquad \forall k \in \{k_1, k_2\}.$$

Hence, Eq. (57) can be reformulated as:

$$T^{k_1} - T^{k_2} = (t_{fk_1} - t_{fk_2}) + \frac{e_{ijs}}{\beta^1 d_{ij}}(d_{fk_1} - d_{fk_2}) + \frac{1}{\beta^2}(e_{k_1js} - e_{k_2js}) + (t_{k_1j} - t_{k_2j}),$$

which is strictly positive for hypothesis (55). □

Proposition 1 can be naturally extended to cases where the CSs have different technologies. Furthermore, Proposition 1 may be generalized to charging functions with more than three breaking points.

4. Solution strategy

The SEVRP-T is NP-hard since it is an extension of the well-known capacitated vehicle routing problem (see Toth and Vigo, 2014). Given that significant computational effort is required to find optimal solutions for medium- and large-scale EVRP instances (see Afroditi et al., 2014), much of the EVRP literature proposes approximate solution techniques. In this paper, we propose a matheuristic for the SEVRP-T, which is based on an Iterated Local Search procedure with a Set Partitioning phase (ILS-SP). This technique builds upon the work of Froger et al. (2022), where a similar algorithmic structure showed good performance for deterministic versions of the EVRP.

Our heuristic is composed of two phases: a *route generator* phase and a *route assembly* phase (see Section 4.1). The route generator phase builds a pool of high-quality routes via an ILS procedure. This phase is performed via two steps: *diversification* and *intensification*. Diversification is implemented as a perturbation procedure (see Section 4.1.1), while intensification is carried out through a Variable Neighborhood Descent search strategy (VND) (see Section 4.1.2). To speed up the evaluation of a given first-stage solution, we derive two lower bounds on the expected duration of fixed routes (see Section 4.1.3). Once the pool of routes has been populated, a solution to the SEVRP-T is determined by solving a Set Partitioning problem (SP) over those routes (see Section 4.1.4).

The structure of our heuristic is outlined in Algorithm 1. Lines 2–12 are related to the route generator phase, while line 13 is the route assembly phase. Firstly, the pool Ω of routes is empty, and the current best solution S^* is set equal to an input initial solution S_0 (line 2). A solution S is comprised of a number of first-stage routes (i.e., sequences of customer visits). At the same time the number I of iterations is initialized to zero. Next, the algorithm enters its main iterative process (line 3). Excluding the first iteration (lines 4-5), the current best solution S^* is perturbed to move from the local optimum, investigating a different part of the search space (line 7). After this procedure, a new solution S' is obtained and used as input for the intensification phase through the VND algorithm (line 8). A possibly new solution S is thus identified, and all its new routes are inserted in the pool Ω (line 9). If the objective function value of S is less than that of the incumbent solution S^*, then the incumbent solution is updated to be S (line 11). The main iterative process is repeated until the maximum number of iterations I_{max} is reached. Lastly, a SP problem is solved considering the pool Ω of generated routes (line 13). In what follows, we detail all algorithmic components of our heuristic.

4.1. Route generator and route assembler

We use an ILS algorithm (see Lourenço et al., 2003) to populate the pool of routes used to assemble a solution for the SEVRP-T. An ILS algorithm escapes from local optima by perturbing the incumbent solution. We find the local optima via a VND procedure. Further details are provided in the following sections (see Sections 4.1.1 and 4.1.2). Finally, in Section 4.1.3, we introduce an exact algorithm for the fixed route vehicle charging problem, which is a subproblem in the VND phase.

4.1.1. Perturbation phase

In Algorithm 1 at line 7, whenever a local optimal solution S^* is reached, it is perturbed. The perturbation phase, also known as *shaking*, is similar to the one used by Froger et al. (2022), which is performed as follows. Firstly, a random customer $i \in \mathcal{I}$ is selected. Then, the κ geographically closest customers to i are removed from their respective routes, with κ a random integer in the interval $[\min\{|\mathcal{I}|, 5\}, \max\{\min\{|\mathcal{I}|, 5\}, \lceil\sqrt{|\mathcal{I}|}\rceil\}]$. Finally, the removed customers are reinserted at different positions in the solution, one at a time and in a random order, according to the following rules:

Algorithm 1 ILS-SP.

Input: An instance of the SEVRP-T with feasible solution S_0, a maximum number of iterations I_{max}
Output: A solution for the SEVRP-T
Let $g(S)$ be the value of the objective function for a solution S and assume $g(\emptyset) = +\infty$.

1: **function** ILS-SP
2: $\Omega \leftarrow \emptyset$, $S^* \leftarrow S_0$, $I \leftarrow 0$
3: **while** $I < I_{max}$ **do**
4: **if** $I = 0$ **then**
5: $S' \leftarrow S_0$
6: **else**
7: $S' \leftarrow$ PERTURB(S^*) ▷ See Section 4.1.1
8: $S \leftarrow$ VND(S') ▷ See Section 4.1.2
9: Insert all the new routes of S into Ω
10: **if** $g(S) < g(S^*)$ **then**
11: $S^* \leftarrow S$
12: $I \leftarrow I + 1$
13: Solve a Set Partitioning problem on Ω ▷ See Section 4.1.4

- A customer cannot be placed back into the same route from which it was removed;
- A customer is reinserted in the position yielding the minimum increase in the expected time with respect to the base solution, i.e., the one without the removed customers. This is carried out by evaluating the increase in the expected time of every feasible insertion of the customer in every other route, reoptimizing the charging decisions through Algorithm 3 (see Section 4.1.3);
- If no feasible insertion is possible, a new route with the removed customer is set up.

4.1.2. Variable Neighborhood Descent search phase

In the intensification phase of the ILS-SP algorithm (see line 8 in Algorithm 1) we apply a VND procedure (see Mladenović and Hansen, 1997 and Golden et al., 2008). This technique explores an ordered set of neighborhoods $\mathcal{D}$, with respect to the current solution S. As soon as an improved solution is found, the algorithm stops searching in the current neighborhood and restarts the search with the first operator of $\mathcal{D}$. Otherwise, the VND moves to the next neighborhood in $\mathcal{D}$. A local optimum is reached when the last operator fails to improve the solution S. Let r_1 and r_2 represent any two routes in the solution S. The neighborhoods in $\mathcal{D}$ are generated based on the following operators:

1. *1-0 vertex exchange*: one customer is moved from route r_1 to route r_2;
2. *1-1 vertex exchange*: interchange between customer i from route r_1 and customer j from route r_2;
3. *2-0 vertex exchange*: two consecutive customers i and j are moved from route r_1 to route r_2. The opposite arc (j, i) is also considered;
4. *2-1 vertex exchange*: interchange between consecutive customers i and j from route r_1 and customer $\tilde{i}$ from route r_2. The opposite arc (j, i) is also considered as well;
5. *2-2 vertex exchange*: interchange between consecutive customers i and j from route r_1 and two other adjacent customers $\tilde{i}$ and $\tilde{j}$ from route r_2. The opposite arcs (j, i) and $(\tilde{j}, \tilde{i})$ are also considered, yielding four possible combinations;
6. *2-opt*: two nonconsecutive edges from routes r_1 and r_2 are removed, and two new edges are formed to make a new route;
7. *Separate:* (introduced by Froger et al., 2022) this operator splits a single route into two routes by adding a return to the depot after a customer visit.

In our implementation we use intra-route and inter-route versions of the first five operators of the list, resulting in a total of 12 different moves. The structure of the VND procedure is outlined in Algorithm 2. For every neighborhood and for every move, we apply a couple of filtering strategies to prune unpromising moves (lines 8-9). Details of these strategies are discussed in the next section.

4.1.3. The stochastic fixed route vehicle charging problem with a threshold policy

In the execution of Algorithm 2, the evaluation of the expected duration of a first-stage route r is a subproblem arising several times throughout our algorithm. We call this problem the Stochastic Fixed Route Vehicle Charging Problem with a Threshold policy (SFRVCP-T). The objective of a fixed route vehicle charging problem (see Froger et al., 2019) is to identify optimal charging operations, related to which CSs to insert in the given route (comprised of a sequence of customer visits) and the amount of energy to recharge at them, while minimizing the total duration of the route. In the special case of the SFRVCP-T, the CSs to visit for each scenario $s \in S$ for each route must be identified.

One possibility to solve the SFRVCP-T is to derive a second-order cone mixed-integer programming model from the one presented in Section 3. Specifically, the first-stage variables x_{ij} are set to 1 if arc (i, j) belongs to r and 0 otherwise. As a consequence, only the second-stage variables should be determined. However, this approach is predictably slow from a computational perspective, as demonstrated empirically in Section 6.3. For this reason, we propose an exact algorithm called E-SFRVCP-T(r) (see Algorithm 3).

Algorithm 3 operates as follows. First, we initialize to zero the duration t_r of the route r (line 2). Then, the algorithm enters in two iterative processes, one over the set of scenarios and one over the arcs of the route. If no detour is needed under scenario

Algorithm 2 VND(S).

Input: An instance of the SEVRP-T with a solution S
Output: A new solution S^* of the SEVRP-T
For $d \in \mathcal{D}$, let R_d be the set of first-stage routes r impacted by move d, with corresponding total first-stage travel time t_{R_d} and R'_d the newly created routes after applying move d to R_d.

1: **function** VND
2: $S^* \leftarrow S, d \leftarrow 0$
3: $\mathcal{D} \leftarrow$ list of neighborhoods
4: $\delta^* \leftarrow +\infty$ ▷ Best improvement so far
5: **while** $d < |\mathcal{D}|$ **do**
6: $\delta \leftarrow 0$
7: **for** every move in neighborhood d **do**
8: **if** $\sum_{r \in R'_d} \sum_{(i,j) \in r} t_{ij} < \gamma t_{R_d}$ **then** ▷ If first-stage travel times are promising. See Section 4.1.3
9: **if** $\sum_{r \in R'_d}$ LB-SFRVCP-T$(r, \bar{s}) < t_{R_d}$ **then** ▷ See Section 4.1.3
10: $\delta \leftarrow t_{R_d} - \sum_{r \in R'_d}$ E-SFRVCP-T(r) ▷ See Section 4.1.3
11: $R^* \leftarrow R'_d$
12: **if** $\delta < \delta^*$ **then**
13: $\delta^* \leftarrow \delta$
14: **if** $\delta^* < 0$ **then**
15: $S^* \leftarrow S^* \cup R^* \setminus R_d$
16: **else**
17: $d \leftarrow d + 1$
18: **return** S^*

$s \in S$ when traversing arc $(i, j) \in r$ (line 6), then no recharging operations are needed. This happens when the EV has enough energy to completely traverse the arc with a SoC that is greater than Q^T. In this case, the EV's SoC is updated, as well as the duration of the route (lines 7-8). Otherwise, the best CS yielding the lowest travel detour and recharging time is chosen (lines 10-19). The selection is performed on a precomputed set C_{ij} of non-dominated CSs. The construction of set C_{ij} is outlined in Algorithm 5 in Appendix A. We denote by $\psi_k^{-1}(q_i, q_f)$ the inverse charging function of CS $k \in C_{ij}$, where q_i and q_f are the initial and final SoC of the vehicle, respectively. In line 16, if no feasible CSs exist, i.e., the EV runs out of battery before reaching any CSs from the detour point or the amount to charge is greater than Q^{max}, the route r is infeasible (lines 20-21). The total duration $\tilde{t}_{rs}$ under scenario s is updated (line 23), as well as the expected recourse duration t_r of route r, weighted by the probability p_s of scenario $s \in S$ (line 25).

For a given route composed by l arcs from an instance with $|C|$ charging stations, Algorithm 3 has a worst-case complexity equal to $\mathcal{O}(l|C|)$. Indeed, the travelling time of the arcs without detours is evaluated in constant time, but every detour requires a loop over all the possible CSs.

Given the complexity of E-SFRVCP-T(r), coupled with the with need to frequently call this algorithm within the VND, we propose two lower bounds that are used within the heuristic to accelerate it. The two lower bounds operate on a given first-stage route r and are efficiently computed. We use these bounds in sequence to filter unpromising solutions. Let R_d be the set of first-stage routes impacted by move $d \in \mathcal{D}$ with total duration t_{R_d}, and R'_d the newly created routes after applying move d to R_d.

The first lower bound is related only to the duration of the first-stage route, without considering any recharging operations (see line 8 in Algorithm 2). Specifically, if the total first-stage travelled time of the routes in R'_d exceeds the total duration t_{R_d} of the current set of routes R_d multiplied by a factor $\gamma \geq 1$, then the corresponding move is discarded. Parameter γ avoids evaluating excessively lengthy routes. If the condition holds, the algorithm proceeds to the second bound.

The second lower bound accounts for both the technology of the CSs and the duration of the detour. We refer to it as the Lower Bound for SFRVCP-T (LB-SFRVCP-T(r, S)). Its computation is performed in Algorithm 4, relying on a simplified version of Algorithm 3. Let r be a given first-stage route. If no detour is needed under scenario $s \in S$ while traversing arc $(i, j) \in r$ (line 6), recharging operations are unnecessary, and the SoC and the route duration are simply updated (lines 7-8). Conversely, when a detour is required, the total duration of the route is updated to include both travelling and recharging times (line 10). For the total duration of the detour, we use the lower bound Eq. (58) derived in Proposition 2. Further details on its computation are provided below. Regarding recharging times, the SoC of the EV is assumed to be Q^T upon arrival at the CS and sufficient to reach customer j with a SoC of Q^G upon departure. The total recharging time is then calculated by converting the required charge amount into time using a_j^{fast}, i.e., the fastest charge rate of the CS with the fastest technology close to node j. All these assumptions ensure that the computation remains consistent with the goal of determining a lower bound.

Algorithm 4 is designed to compute the lower bound for all possible scenarios in the scenario tree $\mathcal{S}$. However, to further speed up the procedure, in line 9 of Algorithm 2 we compute such lower bound only for the average scenario $\bar{s}$.

The correctness of the lower bound in line 10 of Algorithm 4 is proven in Proposition 2.

Algorithm 3 E-SFRVCP-T(r).

Input: A fixed route r
Output: Expected recourse duration t_r for route r

function E-SFRVCP-T
  $t_r \leftarrow 0$
  for every scenario $s \in \mathcal{S}$ **do**
    $\tilde{t}_{rs} \leftarrow 0, SoC \leftarrow Q^{max}$
    for every arc $(i,j) \in r$ **do**
      if $SoC - e_{ijs} > Q^T$ **then** ▷ No detour in arc (i,j)
        $SoC \leftarrow SoC - e_{ijs}$
        $\tilde{t}_{rs} \leftarrow \tilde{t}_{rs} + t_{ij}$
      else
        $t_r^* \leftarrow \infty$
        for every CS $k \in C_{ij}$ **do**
          $z_{ijs} \leftarrow (SoC - Q^T)/e_{ijs}$ ▷ Percentage of arc (i,j) covered until detour
          $X_{ijs} \leftarrow X_j z_{ijs} + X_i(1 - z_{ijs})$ ▷ X-coordinate of the detour point
          $Y_{ijs} \leftarrow Y_j z_{ijs} + Y_i(1 - z_{ijs})$ ▷ Y-coordinate of the detour point
          $\ell_{ijks} \leftarrow \sqrt{(X_{ijs} - X_k)^2 + (Y_{ijs} - Y_k)^2}$ ▷ Distance detour point–CS
          if $Q^T - \frac{e_{ijs}}{d_{ij}}\ell_{ijks} \geq 0$ and $Q^G + e_{kjs} \leq Q^{max}$ **then**
            $t_r \leftarrow t_{ij} z_{ijs} + \frac{t_{ij}}{d_{ij}}\ell_{ijks} + \psi_k^{-1}(Q^T - \frac{e_{ijs}}{d_{ij}}\ell_{ijks}, Q^G + e_{kjs}) + t_{kj} - t_{ij}$
            if $t_r < t_r^*$ **then**
              $t_r^* \leftarrow t_r$
        if $t_r^* = \infty$ **then**
          return ∞ ▷ Route r is infeasible
        else
          $\tilde{t}_{rs} \leftarrow \tilde{t}_{rs} + t_r^*$
        $SoC \leftarrow Q^G$
    $t_r \leftarrow t_r + p_s \tilde{t}_{rs}$
  return t_r

Algorithm 4 LB-SFRVCP-T($r, \mathcal{S}$).

Input: A fixed route r, a set of scenarios $\mathcal{S}$
Output: A lower bound on the expected recourse duration t_r for route r

function LB-SFRVCP-T
  $t_r \leftarrow 0$
  for every scenario $s \in \mathcal{S}$ **do**
    $\tilde{t}_{rs} \leftarrow 0, SoC \leftarrow Q^{max}$
    for every arc $(i,j) \in r$ **do**
      if $SoC - e_{ijs} > Q^T$ **then** ▷ No detour in arc (i,j)
        $SoC \leftarrow SoC - e_{ijs}$
        $\tilde{t}_{rs} \leftarrow \tilde{t}_{rs} + t_{ij}$
      else
        $\tilde{t}_{rs} \leftarrow \tilde{t}_{rs} + \frac{t_{ij}}{d_{ij}} \max\{d_{ij}, \underline{d}_i + \underline{d}_j\} + a_j^{fast}(Q^G + \frac{e_{ijs}}{d_{ij}}\underline{d}_j - Q^T)$
        $SoC \leftarrow Q^G$
    $t_r \leftarrow t_r + p_s \tilde{t}_{rs}$
  return t_r

Proposition 2. *Let (i,j) be an arc with $i,j \in \mathcal{I}$ and $\underline{k}^i := \arg\min_{k \in \mathcal{K}} d_{ik}$ be the CS yielding the minimum distance $\underline{d}_i$ from customer $i \in \mathcal{I}$. If the EV reaches the threshold Q^T under scenario $s \in \mathcal{S}$ while traversing arc (i,j), then the quantity:*

$$\max\{d_{ij}, \underline{d}_i + \underline{d}_j\} \tag{58}$$

is a lower bound on the total length of any detour along arc (i,j).

Proof. As distances are Euclidean, d_{ij} is by definition the length of the shortest path from i to j. It trivially follows that d_{ij} is a lower bound on the total length of any detour along (i,j); it may be a tight bound if the selected CS is on the arc (i,j).

We first prove that $\underline{d}_i + \underline{d}_j$ is also a valid lower bound. Recall that any detour from customer i to customer j must follow this order:

1. travel from node i to the detour point;

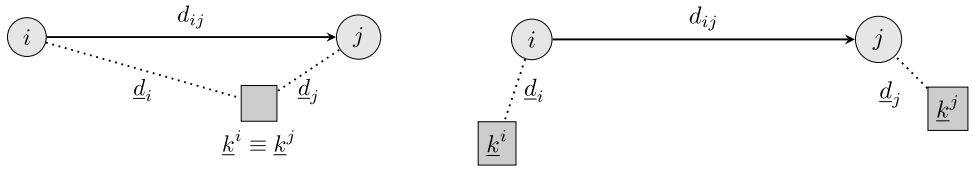


Fig. 4. Relevant distances between nodes and CSs in the computation of the lower bound Eq. (58). Left panel: the case with $\underline{k}^i \equiv \underline{k}^j$. Right panel: the case with $\underline{k}^i \not\equiv \underline{k}^j$.

2. travel from the detour point to a CS k;
3. travel from CS k to node j.

Since the detour must end in j and we assume $\underline{k}^j$ to be the nearest CS to j, $\underline{d}_j$ is a lower bound on the length of the path from the chosen CS k to j (point 3). The same is not true for node i, since the vehicle will not travel directly from i to the CS k, but the triangle inequality implies that $\underline{d}_i$ is a valid lower bound on the total distance covered from i to the CS k (points 1 and 2). This lower bound may be tight if $\underline{k}^i \equiv \underline{k}^j$ (see Fig. 4 left panel) and the detour happens at node i. Note that $\underline{d}_i + \underline{d}_j$ could be less than d_{ij} if $\underline{k}^i$ and $\underline{k}^j$ are different (see Fig. 4 right panel), hence we use $\max\{d_{ij}, \underline{d}_i + \underline{d}_j\}$ as a stronger lower bound. □

4.1.4. Set partitioning

Once the pool of feasible routes Ω has been populated by the ILS procedure, the final solution of the SEVRP-T is obtained through the second component of the heuristic. As in Montoya et al. (2017) and Andelmin and Bartolini (2019), this second component consists of solving a Set Partitioning problem considering the set of routes in Ω.

Let x_r be a binary variable equal to 1 if route $r \in \Omega$ is selected, 0 otherwise. The expected recourse duration of route $r \in \Omega$ is denoted as t_r, while α_{ir} is a binary parameter equal to 1 if route $r \in \Omega$ visits customer $i \in \mathcal{I}$, 0 otherwise. The corresponding Set Partitioning problem on the set Ω is formulated as follows:

$$\text{minimize} \sum_{r \in \Omega} t_r x_r \tag{59}$$

subject to:

$$\sum_{r \in \Omega} \alpha_{ir} x_r = 1 \quad \forall i \in \mathcal{I} \tag{60}$$

$$x_r \in \{0,1\} \quad \forall r \in \Omega. \tag{61}$$

Objective Eq. (59) selects a subset of routes from Ω that minimizes the total duration. Constraints (60) ensure that each customer $i \in \mathcal{I}$ is visited exactly once. Finally, constraints (61) set the domains of the decision variable x_r, with $r \in \Omega$.

5. A scenario reduction technique for the SEVRP-T

The application of the ILS-SP heuristic outlined in Section 4 heavily depends on the size of $\mathcal{S}$ (see line 3 in Algorithm 3). The larger $|\mathcal{S}|$ is, the greater the informative power of the scenario tree is, enabling to capture a broader range of energy consumption outcomes. However, this comes at the cost of increased computational effort. For this reason, in this section, we present the technique we have adopted for reducing the number of scenarios in the case of medium- and large-sized SEVRP-T instances. The scenario reduction technique is mainly adapted from Dupačová et al. (2003) and Heitsch and Römisch (2003). Let P be the original distribution defined over a finite set $\mathcal{S}$ of energy consumption scenarios. The technique determines a subset of scenarios $\mathcal{S}^{red} \subset \mathcal{S}$ and the corresponding probability distribution P^{red} with $|\mathcal{S}^{red}|$ as an input. The set $\mathcal{S}^{red}$ and the probability P^{red} are determined with the objective of minimizing the probabilistic distance between P and P^{red}, which is established by solving an appropriately defined optimal transportation problem. In what follows we detail the technique.

Let $e_s := [e_{ijs}]_{(i,j)\in\mathcal{A}}$ represent the vector of all realizations of the energy consumption across the entire network in scenario $s \in \mathcal{S}$. The initial probability distribution P on $\mathcal{S}$ is written as $P := \sum_{s\in\mathcal{S}} p_s \delta_{e_s}$, where p_s is the probability of scenario $s \in \mathcal{S}$ and δ_{e_s} denotes the Dirac distribution at e_s. Given a subset $\mathcal{S}^{red} \subset \mathcal{S}$ of fixed cardinality $|\mathcal{S}^{red}| < |\mathcal{S}|$, we define $\mathcal{S}^{del} := \mathcal{S} \setminus \mathcal{S}^{red}$. Specifically, $\mathcal{S}^{red}$ and $\mathcal{S}^{del}$ are the sets of the energy consumption scenarios which are preserved and deleted, respectively, after the application of the scenario reduction technique. In this regard, we construct the probability distribution P^{red} based on scenarios e_s with probabilities $p_s^{red} \in [0,1]$ for $s \in \mathcal{S}^{red}$, i.e., $P^{red} := \sum_{s\in\mathcal{S}^{red}} p_s^{red} \delta_{e_s}$, with $\sum_{s\in\mathcal{S}^{red}} p_s^{red} = 1$. This is achieved by deleting all scenarios $s \in \mathcal{S}^{del}$ and by assigning new probabilities p_s^{red} to each preserved scenario with $s \in \mathcal{S}^{red}$. For simplicity, let $p^{red} := [p_s^{red}]_{s\in\mathcal{S}^{red}}$ be the vector of reassigned probabilities. Therefore, the objective of the scenario reduction technique is to optimally choose a subset $\mathcal{S}^{red}$ of energy consumption scenarios and a suitable probability p^{red} from the feasible set $\{\mathcal{S}^{red} \subset \mathcal{S}$ with $|\mathcal{S}^{red}|$ fixed, $p^{red} \geq 0, \sum_{s\in\mathcal{S}^{red}} p_s^{red} = 1\}$ so that the distance $D(P, P^{red})$ between the probability distributions P and P^{red} is minimized. This is performed by solving the following

discrete version of the *optimal transportation problem* (see Dupačová et al., 2003):

$$\text{minimize } D(P, P^{red}) = \sum_{s\in S}\sum_{\tilde{s}\in S^{red}} \|e_s - e_{\tilde{s}}\|_2 \eta_{s\tilde{s}}$$

subject to:

$$\sum_{s\in S} \eta_{s\tilde{s}} = p^{red}_{\tilde{s}} \qquad \forall \tilde{s} \in S^{red}$$
$$\sum_{\tilde{s}\in S^{red}} \eta_{s\tilde{s}} = p_s \qquad \forall s \in S$$
$$\eta_{s\tilde{s}} \geq 0 \qquad \forall s \in S, \tilde{s} \in S^{red},$$

where $\eta_{s\tilde{s}}$ is the probability of assigning scenario $s \in S$ to scenario $\tilde{s} \in S^{red}$. If S^{red} is predefined, the problem of determining optimal probabilities p^{red}_s can be easily handled, as established in the following theorem (see Theorem 2.1 in Heitsch and Römisch, 2003 or Theorem 2 in Dupačová et al., 2003):

Theorem 1 (Heitsch and Römisch, 2003 and Dupačová et al., 2003). *Let S^{red} be given. The minimum distance $D(P, P^{red})$ over the set $\{p^{red} \geq 0, \sum_{s\in S^{red}} p^{red}_s = 1\}$ is equal to:*

$$D(P, P^{red}) = \sum_{\tilde{s}\notin S^{red}} p_{\tilde{s}} \cdot \min_{s\in S^{red}} \|e_s - e_{\tilde{s}}\|_2. \tag{62}$$

Additionally, the minimum is achieved at:

$$p^{red}_s = p_s + \sum_{\tilde{s}\in S^{del}_s} p_{\tilde{s}} \qquad \forall s \in S^{red}, \tag{63}$$

where $S^{del}_s := \{\tilde{s} \in S^{del} : s = s(\tilde{s})\}$ and $s(\tilde{s}) \in \arg\min_{s\in S^{red}} \|e_s - e_{\tilde{s}}\|_2$ for all $\tilde{s} \in S^{del}$.

Eq. (63) is known as *optimal redistribution rule*. It states that the new probability of each retained scenario s in S^{red} corresponds to the sum of its original probability p_s and the probabilities of all deleted scenarios closest to it in terms of the ℓ_2-distance.

On the other hand, the choice of optimally determining S^{red} with fixed cardinality $|S^{red}|$ may be established by solving a *set-covering* problem over S, which is known to be NP-hard. However, efficient solution algorithms are available for the cases of $|S^{red}| = |S| - 1$ and $|S^{red}| = 1$.

When $|S^{red}| = 1$, the set covering problem simplifies to:

$$\min_{\bar{s}\in S} \sum_{s\in S} p_s \|e_s - e_{\bar{s}}\|_2. \tag{64}$$

If the minimum occurs at $\bar{s}$, only $e_{\bar{s}}$ is retained and its probability will be $p^{red}_{\bar{s}} = 1$, consistently with the redistribution rule Eq. (63). We recursively solve problem Eq. (64), thus allowing us to consider cases in which $|S^{red}| > 1$, selecting more scenarios that will not be deleted. This procedure, called *forward selection algorithm*, determines an index set $\{\bar{s}_1, \dots, \bar{s}_{|S^{red}|}\}$ such that:

$$\bar{s}_m \in \arg\min_{\tilde{s}\in S^{del}_{[m-1]}} \sum_{o\in S^{del}_{[m-1]}\setminus\{\tilde{s}\}} p_o \cdot \min_{s\notin S^{del}_{[m-1]}\setminus\{\tilde{s}\}} \|e_o - e_s\|_2 \qquad \forall m = 1, \dots, |S^{red}|, \tag{65}$$

where $S^{del}_{[0]} := S$ and $S^{del}_{[m-1]} := S \setminus \{\bar{s}_1, \dots, \bar{s}_{m-1}\}$ for $m = 2, \dots, |S^{red}|$. The procedure reflects the structure of $D(P, P^{red})$ in Eq. (62) and can be executed using the *Fast Forward Selection* (FFS) algorithm (see Algorithm 6 in Appendix B).

For the sake of illustration, Fig. 5 presents an example where the FFS algorithm is applied to a one-dimensional scenario tree with $|S| = 50$ scenarios, each with the same probability $p_s = \frac{1}{50}$. The goal is to reduce the tree to a subset of $|S^{red}| = 10$ scenarios. The FFS algorithm selects the scenarios shown in black in Fig. 5a and assigns to them new probabilities (following Eq. (63)), which are depicted in Fig. 5b.

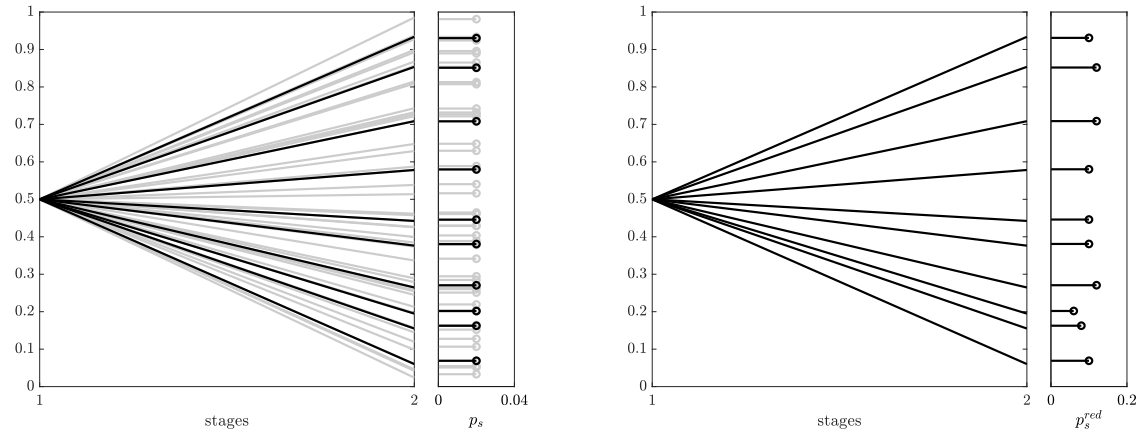
6. Computational results

In this section, we present and discuss the results of the numerical experiments with the aim of:

- Evaluating the performance of the proposed heuristic in both deterministic and stochastic settings;
- Validating the stochastic model in terms of in-sample stability (see Kaut and Wallace, 2007);
- Measuring the impact of uncertainty and the quality of the deterministic solution in a stochastic setting (see Birge and Louveaux, 2011);
- Performing a sensitivity analysis by examining the model's behaviour under varying conditions.

The model and the heuristic were implemented in Python (v. 3.12.2) and solved using Gurobi (v. 11.0). All computational experiments were run on a MacBookPro17.1 with an Apple M1 processor of 8 cores and 16 GB of RAM memory. Unless otherwise specified, a runtime limit of three hours (10800 seconds) is imposed on the solver and for the heuristic.

In the following, we start describing the instances and the parameters, together with the scenario generation procedure we opted for (see Section 6.1). Then, we assess the performance of the heuristic algorithm both in deterministic and stochastic settings (see Section 6.2 and Section 6.3, respectively). Finally, we discuss managerial insights and the results of a sensitivity analysis (see Section 6.4).



(a) Original scenario tree (50 scenarios). (b) Reduced scenario tree (10 scenarios).

Fig. 5. Example of the FFS algorithm application. Left panel: the original scenario tree with 50 scenarios and their corresponding probabilities. The 10 selected scenarios are highlighted in black. Right panel: the reduced scenario tree, consisting of the 10 scenarios selected by the FFS algorithm, along with their reassigned probabilities.

6.1. Instances and parameters

We considered a selection of benchmark instances taken from Montoya et al. (2017) (available from http://www.vrp-rep.org/). We used instances composed by 10, 20, 40 and 80 customers. Additionally, we created a new set of twenty instances, each having 15 customers, by removing the last five customers from the 20-customer instances. Instances are named using the following coding scheme: tcAcBsCcDE. Specifically, A is the method used to locate customers with 0: random uniform distribution, 1: random clustered distribution, 2: mixture of both; B is the number of customers; C is the number of CSs; D is equal to "t" if the CSs are located using a *p-median* heuristic and "f" if the CSs are randomly located; E is the number of the instance for each combination of parameters. Three types of CSs are considered (slow, moderate, and fast).

As in Montoya et al. (2017) and Froger et al. (2022), we considered an EV with a consumption rate of 0.125 kWh per unit of distance. However, we assumed $Q^{max} = 24$ kWh, as opposed to a battery capacity of 16 kWh considered in Montoya et al. (2017) and Froger et al. (2022). This choice was made to avoid infeasibility issues due to uncertain nature of the energy consumption. Other relevant parameters are fixed as follows: n copies of each CS equal to 3; $Q^T = 30\,\%\ Q^{max} = 7.2$ kWh; $Q^G = 80\,\%\ Q^{max} = 19.2$ kWh; M is set to the greatest distance between two nodes. In Algorithm 1 we impose a maximum number I_{max} of ILS iterations equal to 2000, and in Algorithm 2 parameter $\gamma = 1$. Finally, we have removed instances tc1c10s2cf3 and tc1c10s3cf3 due to infeasibility issues arising with a large number of scenarios.

6.1.1. Scenario generation

In this section, we discuss the procedure we adopted to generate scenario trees for the uncertain energy consumption. Considering the energy consumption $\hat{e}_{ij} \in \mathbb{R}^+$ from Montoya et al. (2017), we have created $|S|$ energy consumption scenarios for each arc. We recall that we denote by e_{ijs} the energy consumption along arc $(i,j) \in \mathcal{A}$ under scenario s and p_s its probability of occurrence. If $|S| = 1$, then $e_{ij1} = \hat{e}_{ij}$ for all $i,j \in \mathcal{V}$. On the other hand, if $|S| > 1$, scenarios were sampled according to a Monte Carlo procedure (see Shapiro, 2003) from three possible probability distributions (see Maggioni et al., 2019): Uniform (U), truncated Normal (N) and truncated Exponential (E). The energy consumption on each arc $(i,j) \in \mathcal{A}$ was sampled independently. To ensure comparability across experiments, we considered the same initial random seed. We assumed that the three distributions have the same mean equal to the deterministic energy consumption, i.e., $\mu_{ij}^U = \mu_{ij}^N = \mu_{ij}^E = \hat{e}_{ij}$. For the Uniform distribution we randomly sampled from the interval $[0.75\hat{e}_{ij}, 1.25\hat{e}_{ij}]$, such that a maximum deviation of 25 % from the average consumption $\hat{e}_{ij}$ is allowed. For the Uniform distribution, this corresponds to a variance equal to $\sigma_{ij}^U = \frac{(0.5\hat{e}_{ij})^2}{12}$. For the Normal and Exponential distributions, we considered the same variance of the Uniform case, i.e., $\sigma_{ij}^N = \sigma_{ij}^E = \sigma_{ij}^U$. Lastly, we set $p_s = \frac{1}{|S|}$. Similar scenario generation assumptions have been adopted in Pelletier et al. (2019) and Bruni et al. (2020).

6.2. Performance of the ILS-SP heuristic in the deterministic setting

In this section, we assess the performance of our heuristic on a selection of benchmark instances, assuming $|S| = 1$ with $e_{ij1} = \hat{e}_{ij}$. We compare the performance of the mathematical model solved by Gurobi and the ILS-SP heuristic through different indicators in Table 3: the number of instances solved to optimality out of total number of instances (# Opt / # Inst), CPU time, number of routes,

Table 3
Comparison between Algorithm 1 and Gurobi solver for instances with $|\mathcal{I}| = \{10, 15, 20\}$ customers in the deterministic setting ($|S| = 1$). Average results are reported.

$\|\mathcal{I}\|$	# Opt / # Inst	Time (s)		Number of routes		Gap	
		Algorithm	Model	Algorithm	Model	Gurobi	Algorithm-Model
10	18 / 18	11.40	71.59	1.56	1.56	0.00 %	0.00 %
15	15 / 20	39.06	4552.97	1.40	1.40	6.95 %	0.00 %
20	2 / 20	91.49	10709.27	1.65	1.65	12.14 %	−0.29 %

Table 4
CPU time (in milliseconds) comparison between the exact Algorithm 3 and Gurobi when solving the SFRVCP-T.

	SFRVCP-T solved by Gurobi (ms)					SFRVCP-T solved by E-SFRVCP-T (ms)				
	$\|S\|$					$\|S\|$				
l	1	5	20	50	100	1	5	20	50	100
3	84.8	195.7	613.2	1343.7	2690.5	0.0	0.3	1.0	2.4	4.9
4	52.2	151.3	547.2	1232.0	2445.2	0.0	0.1	0.5	1.4	2.7
5	39.1	121.2	389.5	897.6	1722.8	0.0	0.1	0.5	1.1	2.3
6	68.7	230.6	918.5	2061.5	4068.5	0.1	0.2	1.0	2.7	5.5
7	101.1	312.8	1219.2	2639.2	5331.2	0.1	0.5	1.8	4.5	9.0
8	98.2	342.6	1295.0	2911.1	5906.4	0.1	0.5	1.8	4.2	8.7
9	116.5	456.6	1728.7	3912.7	7843.8	0.2	0.7	2.7	6.7	13.0
10	123.2	466.1	1724.5	3925.2	7962.7	0.1	0.6	2.1	5.2	10.3
11	153.7	552.6	2004.6	4676.7	9340.6	0.2	0.7	2.7	6.8	14.0
12	169.2	646.1	2599.4	5957.0	12160.2	0.2	1.0	3.7	9.2	18.4
13	151.5	575.9	2242.4	5115.9	10368.8	0.2	0.8	3.1	7.0	14.0
14	319.3	978.8	3447.1	8501.7	17336.0	0.3	1.6	6.1	14.9	31.5
15	166.7	768.1	2916.4	6879.5	14122.0	0.3	1.1	4.0	9.9	19.5
avg	126.4	446.0	1665.0	3850.2	7792.2	0.1	0.6	2.3	5.8	11.8

optimality gap provided by Gurobi and gap between the objective function values z_{algo} and z_{model} computed as follows:

$$Gap := \frac{z_{algo} - z_{model}}{z_{model}}. \tag{66}$$

If Gurobi is not able to solve the model within the runtime limit, we set z_{model} to the objective function value achieved by the incumbent solution found by the solver. Otherwise, z_{model} corresponds to the optimal value.

In Table 3 we outline the average results of instances with 10, 15 and 20 customers. Detailed results are reported in Tables C.11–C.13 in Appendix C. We note that all the 10-customer instances are solved to optimality by the model as well as the heuristic. Conversely, in five out of twenty and eighteen out of twenty instances with 15 and 20 customers, respectively, the model is not able to find an optimal solution within the runtime limit, resulting in an average Gurobi gap of 6.95 % and 12.14 %, respectively. However, in all 15-customer and 20-customer instances solved to optimality by the solver, the heuristic found the optimal solutions as well. Moreover, the solutions provided by the heuristic are all either equal or better than the best feasible solution found by Gurobi.

In the three sets of instances, solving the model with the heuristic compared to Gurobi provides an average CPU time saving of 84.08 %, 99.14 % and 99.15 %, respectively. From this analysis, we conclude that the proposed ILS-SP heuristic provides high-quality solutions for small-sized instances for the deterministic version of SEVRP-T, within reasonable computational times.

6.3. Performance of the ILS-SP heuristic in the stochastic setting

In this section, we discuss the performance of the ILS-SP heuristic when solving the SEVRP-T under uncertain energy consumption. We start by comparing the two different approaches in solving the fixed route vehicle charging problem SFRVCP-T outlined in Section 4.1.3. Then, we assess the performance of our heuristic over different energy consumption probability distributions. Finally, we provide the results of an in-sample stability analysis (see Section 6.3.1) and measure the impact of uncertainty on the problem under investigation (see Section 6.3.2).

In Table 4 we report the computational time of the fixed route model outlined in Section 4.1.3 solved by Gurobi versus Algorithm 3 (E-SFRVCP-T(r)), over a set of 12 fixed routes with a number of arcs $l = \{3, 4, \dots, 15\}$. These fixed routes are obtained by randomly sampling sequences of $l - 1$ customers from SEVRP-T instances. We consider $|S| = \{1, 5, 20, 50, 100\}$ for each fixed route.

We observe that Algorithm 3 is several orders of magnitude faster than the corresponding SFRVCP-T model solved by Gurobi, consistently across all sampled routes. Indeed, the average saving in terms of CPU time is of 99.87 %.

An analysis of the quality of the solutions in the stochastic setting is presented in Table 5. We consider the 10-customer instances and provide a comparison between the ILS-SP heuristic (Algorithm 1) and the solver, in terms of the number of instances solved to optimality out of total number of instances (# Opt / # Inst), CPU time, number of routes and Gap. We consider an increasing number

Table 5
Aggregate results for ILS-SP heuristic (Algorithm 1) and Gurobi for 10-customer instances and $|S| = \{5, 10, 20, 50\}$.

Distribution	$\|S\|$	# Opt / # Inst	Time (s)		Number of routes		Gap	
			Algorithm	Model	Algorithm	Model	Gurobi	Algorithm-Model
Uniform	5	16 / 18	29.49	1936.69	1.78	1.78	0.68 %	0.00 %
	10	8 / 18	52.46	7168.62	1.89	1.89	4.12 %	−0.17 %
	20	- / 18	90.38	10800.00	1.94	1.94	12.18 %	−0.59 %
	50	- / 18	208.43	10800.00	1.94	2.06	24.17 %	−5.47 %
Normal	5	17 / 18	29.63	1443.24	1.72	1.72	0.45 %	0.00 %
	10	7 / 18	51.22	6934.15	1.72	1.72	4.53 %	−0.24 %
	20	- / 18	96.53	10800.00	2.00	2.00	11.69 %	−0.67 %
	50	- / 18	211.86	10800.00	2.00	1.94	23.06 %	−5.27 %
Exponential	5	17 / 18	29.09	1574.14	1.61	1.61	1.18 %	0.00 %
	10	8 / 18	49.51	6941.21	1.67	1.67	4.45 %	−0.25 %
	20	- / 18	90.18	10800.00	1.78	1.78	11.88 %	−0.79 %
	50	- / 18	199.29	10800.00	2.00	1.94	24.02 %	−6.07 %

of scenarios $|S| = \{5, 10, 20, 50\}$, and the three probability distributions of the energy consumption defined in Section 6.1.1. Detailed results are reported in Tables D.14–D.25 in Appendix D.

For simplicity, in the following we describe the results obtained in the case of uniform distribution. Similar findings can be drawn for the normal and exponential probability distributions. First, we notice that when $|S| = \{5, 10\}$, on average more than 60 % of the instances are solved to optimality by Gurobi. In the cases not solved to optimality, the Gurobi gap is relatively low, i.e., 0.68 % and 4.12 %, respectively. Conversely, for larger values of $|S|$, Gurobi is not able to guarantee optimality within the runtime limit for all instances. On the other hand, the ILS-SP heuristic provides better solutions in less than four minutes.

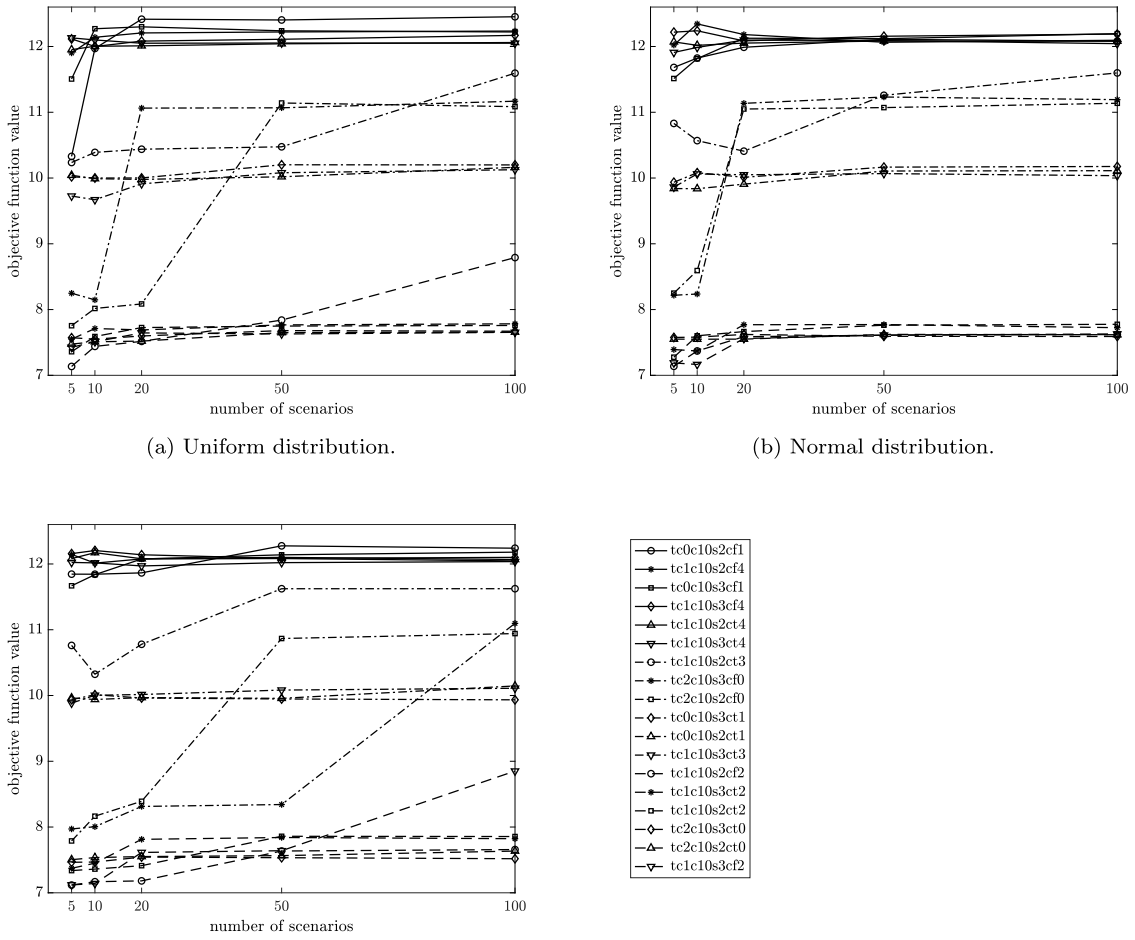
When we compare the outcomes of the three probability distributions, we note that all the indicators are broadly similar. We only observe that, when the size of the scenario tree is small (i.e., $|S| = \{5, 10\}$), the number of routes for the exponential case is the lowest one, followed by the normal and the uniform distributions. For medium scenario trees (i.e., $|S| = \{20, 50\}$), the results are consistent across the three distributions. Overall, these findings indicate that while the choice of distribution has some influence when only a limited number of scenarios is considered, the performance of the ILS-SP heuristic and the model, with respect to both solution quality and computational time, is largely in line across different probabilistic settings.

6.3.1. *In-sample stability*

In this section, we discuss the minimum number of scenarios guaranteeing stability and the impact of adopting the scenario reduction technique, presented in Section 5, when dealing with the SEVRP-T. The results of an in-sample stability analysis (see Kaut and Wallace, 2007) over the 10-customer instances for the three considered distributions are displayed in Fig. 6. The detailed results are reported in Table D.26 in Appendix D. We consider the following sizes of the scenario tree: $|S| = \{5, 10, 20, 50, 100\}$. As the objective function value, we consider the one provided by the solver whenever Gurobi finds an optimal solution within the runtime limit. Otherwise, we use the value obtained through the ILS-SP heuristic. As empirically shown in the previous sections, these two values coincide if the model is solved to optimality by the solver.

Regardless of the probability distributions, three different sets of six instances each can be identified, represented in Fig. 6 with solid, dash-dotted and dotted line, respectively. The first set (solid line) converges to an average objective function value of 12.14. The second set (dash-dotted line) exhibits a higher variability, with three different subsets. Indeed, three instances (tc0c10s3ct1, tc0c10s2ct1, tc1c10s3ct3) converge to 10.11, two instances (tc2c10s3cf0, tc2c10s2cf0) converge to 11.10, while the remaining one (tc1c10s2ct3) reaches an average value of 11.61. Finally, the third set (dotted line) shows a similar behavior to the first one, with an average value of 7.81 with 100 scenarios. We identify two outliers, tc1c10s2ct3 and tc1c10s3ct3, in the case of uniform and exponential distribution, respectively. We notice that, in the 83 % of the instances (fifteen out of eighteen), after 20 scenarios the objective function value becomes fairly stable on average. Therefore, we decided to consider this number as benchmark for the scenario tree size in the following discussion.

In Table 6 we report the results on medium- and large-sized instances of 40 and 80 customers obtained through the ILS-SP heuristic and with a uniformly distributed energy consumption (see Tables E.28–E.32 in Appendix E for details). We observe that increasing the size of the scenario tree generally leads to a higher average objective function value. However, for larger cases, the runtime limit is often reached, resulting in a limited number of iterations of the ILS-SP heuristic. This is particularly pronounced when the scenario tree size is $|S| = 50$. To address this issue, we apply the *forward selection algorithm* as a scenario reduction technique described in Section 5 and implemented in Python via the *ScenarioReducer* library (see Gioia, 2023). Specifically, we reduce a scenario tree with 50 scenarios to a smaller one with 20 scenarios. These results are shown in Table 6 and denoted with the symbol ‡. From a computational perspective, the CPU time, the number of iterations of the ILS-SP heuristic, and the number of routes are comparable with the base case of 20 scenarios on average for the cases with 40 and 80 customers. We note that the average objective function value obtained with 20 reduced scenarios is slightly higher than that of the base case of 20 scenarios and more inline with the results obtained using the 50 scenarios. However, in the latter case, the CPU time is often close or even equal to the runtime limit of 10800s on average.

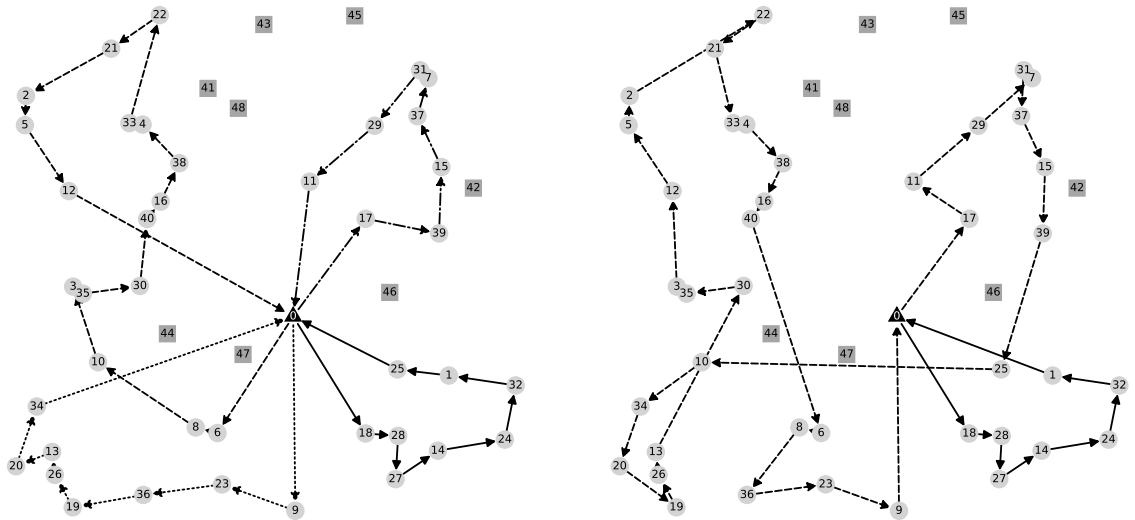


(a) Uniform distribution.

(b) Normal distribution.

(c) Exponential distribution.

Fig. 6. Approximate in-sample stability analysis for different energy consumption probability distributions.



(a) $Q^T = 20\%Q^{max}$.

(b) $Q^T = 40\%Q^{max}$.

Fig. 7. First-stage routes for instance tc0c40s8cf0 with $Q^T = 20\,\%Q^{max}$ (left panel) and $Q^T = 40\,\%Q^{max}$ (right panel). Depot, customers and charging stations are depicted as a triangle, circles and squares, respectively.

Table 6
Solutions obtained through the ILS-SP heuristic (Algorithm 1) for 40- and 80-customer instances, with a uniformly distributed energy consumption. Average results are reported. ‡ denote results for the scenario reduction case from 50 to 20.

Number of customers	$\|S\|$	Time (s)	Number of iterations	Number of routes	Objective
40	1	731.68	2000.00	1.90	17.89
	10	2593.75	2000.00	2.10	18.29
	20	3972.28	1969.00	2.30	18.57
	20‡	4029.39	1969.00	2.30	18.87
	50	10613.88	1252.10	2.40	19.36
80	20	10800.00	343.00	2.75	26.85
	20‡	10800.00	360.85	2.75	27.16
	50	10800.00	174.35	2.65	28.19

Table 7
Stochastic measures with 10-customer instances and $|S| = 20$. Average results are reported.

Distribution	RP	WS	%EVPI	EVP	EEV	%VSS
Uniform	9.90	9.18	6.91 %	9.24	inf	inf
Normal	10.05	9.22	7.83 %	9.25	inf	inf
Exponential	9.71	9.09	6.17 %	9.25	inf	inf

For this reason, we conclude that the application of scenario reduction technique is effective in this context since it provides good solutions by enabling the heuristic to execute a larger number of iterations.

6.3.2. Impact of uncertainty

To quantify the quality of the stochastic solution, we conducted an analysis with classical stochastic measures (see Maggioni and Wallace, 2012; Crainic et al., 2018; Maggioni and Pflug, 2019). Based on the results from Section 6.3.1, we perform the current analysis with $|S| = 20$. The average results are reported in Table 7, while details are in Table D.27 in Appendix D. The results on the so-called Recourse Problem (RP, see Eqs. (1)–(51)) are obtained through the ILS-SP heuristic, while all other results were obtained by the solver, since it converged to optimality in these cases.

Considering uniformly distributed energy consumption, on average the %Expected Value of Perfect Information (%EVPI) is 6.91 % of the optimal objective function value of the RP (see Eqs. (1)–(51)). This measure quantifies the potential benefit of having perfect information on energy consumption at the beginning of the planning. Similar conclusions can be drawn for the other two probability distributions.

As a simpler approach, the decision maker may replace the energy consumption rate with its expected value, computed as the average of the sampled values, and solve the Expected Value Problem (EVP). The %Value of Stochastic Solution (%VSS) measures the expected benefit of solving the stochastic model RP (see Eqs. (1)–(51)) rather than the EVP. We compute it as the difference between the EEV and the RP, divided by the RP, where the EEV is the Expected result of using the EVP solution, i.e., the objective function value of the RP model with the first-stage decision variables fixed at the optimal values obtained from the EVP (see Maggioni et al., 2014). In twelve out of eighteen instances the EVP solution leads to infeasibility under the stochastic setting. These cases are reported as having an infinite EEV and, consequently, an infinite %VSS. This highlights the fact that ignoring energy consumption uncertainty during route planning may result in solutions that cannot be executed in the stochastic setting, causing infeasibility of the model.

6.4. Managerial insights

We conclude our analysis by providing some managerial insights. Specifically, we evaluate the model's behaviour under varying conditions. First, we relax the hypothesis on arc-level detours by allowing only node-level detours (see Section 6.4.1). Second, we analyze different spatial distributions of customers and charging stations (see Section 6.4.2). Third, we investigate the impact of different energy threshold parameters (see Section 6.4.3).

6.4.1. Impact of arc-level detours

In this subsection, we quantify the added value of allowing arc-level detours compared to node-level detours. To this end, we perform numerical experiments by imposing that variable z_{ijs}, which represents the percentage of arc (i, j) covered before performing a detour under scenario s (see Table 2), is not continuous in $[0, 1]$ but binary. Therefore, if $z_{ijs} = 0$, the detour point is node i; otherwise, it is node j (see Constraints (39)–(40)). Since this modification is made at the model level, we restrict our attention to the 10-customer instances that can be solved to optimality by Gurobi, i.e., the 18, 16 and 8 instances reported in Tables 3 and 5 for the uniform distribution.

The results are summarized in Table 8, while extensive results are in Tables F.33, F.34, and F.35 in Appendix F.1. First, we notice that only a small subset of instances yields an optimal solution, while all the remaining ones are infeasible. This outcome is expected,

Table 8
Average results obtained by Gurobi under the node-level detour assumption for 10-customer instances and uniform energy consumption distribution. Relative changes are computed with respect to the arc-level policy and refer only to instances solved to optimality by both the node-level and arc-level recourse policies.

$\|S\|$	# Opt / # Inst	Relative changes	
		Number of routes	Objective
1	8 / 18	75.00 %	24.20 %
5	4 / 16	75.00 %	24.63 %
10	3 / 8	116.67 %	23.83 %

Table 9
Average results, considering uniform energy consumption distribution, obtained over different customers and charging stations distributions through the ILS-SP heuristic (Algorithm 1) for instances with 40 customers and $|S| = 20$, reduced from an initial set of 50 scenarios through the FFS procedure (Algorithm 6).

	Customers distribution			CSs distribution	
	Random uniform	Clustered random	Mixture of both	Random	p-median heuristic
Number of routes	2.38	2.75	2.00	2.70	1.90
Objective	21.18	22.57	14.70	19.45	18.28

Table 10
Average results, considering uniform energy consumption distributions, obtained by varying the values of Q^T and Q^G through the ILS-SP heuristic (Algorithm 1) for instances with 40 customers and $|S| = 20$, reduced from an initial set of 50 scenarios through the FFS procedure (Algorithm 6).

	Q^T			Q^G		
	$20\,\%Q^{max}$	$30\,\%Q^{max}$	$40\,\%Q^{max}$	$70\,\%Q^{max}$	$80\,\%Q^{max}$	$90\,\%Q^{max}$
Number of routes	2.45	2.30	2.10	2.05	2.30	3.16
Objective	18.83	18.87	20.27	18.66	18.87	20.40

since the EV may run out of battery before reaching a CS from a customer node. Second, for the instances that are solved to optimality under the node-based recourse policy, the average number of routes increases by 75 % or more compared to the arc-based policy. Finally, the objective function value also increases substantially, by more than 24 % on average. These findings show that introducing arc-level detours not only allows the model to obtain optimal solution for a larger set of instances but also yields more effective solutions compared to the node-level alternative.

6.4.2. Impact of customer and charging station spatial distributions

In this subsection, we evaluate the impact of the spatial distribution of customers and charging stations on both the number of routes and the expected objective value. We consider the 40-customer instances with $|S| = 20$, which are reduced from an initial set of 50 scenarios following a uniform energy consumption distribution. The results are reported in Table 9. We notice that the clustered random distribution produces the highest values, with on average 2.75 routes and an objective of 22.57, indicating less efficient operations. The random uniform case yields intermediate results, with 2.38 routes and an objective of 21.18. By contrast, the mixed distribution performs best, requiring only 2 routes on average and achieving the lowest objective value (14.70).

Considering the distribution of charging stations, the results in the fifth and sixth columns of Table 9 show that the random allocation leads to higher objective function values, with 2.70 routes and an objective of 19.45. In contrast, the p-median heuristic (see Montoya et al., 2017 for further details) produces the most efficient configurations, requiring on average only 1.90 routes and yielding a lower objective of 18.28. Overall, these findings confirm that the results strongly depend on the spatial distribution of both customers and charging stations, underlining their decisive role in determining least-cost routes.

6.4.3. Impact of energy threshold parameters

Finally, we conclude the managerial insights by discussing the results of a sensitivity analysis on Q^T and Q^G, which are two key SEVRP-T parameters. We adopt the same setting as in the previous analysis, i.e., 40-customer instances and 20 scenarios reduced from an initial set of 50. Average results are reported in Table 10, while extensive results are in Tables F.36 and F.37 in Appendix F.2.

First, we account for three different possibilities of Q^T: $20\,\%Q^{max}$, $30\,\%Q^{max}$ and $40\,\%Q^{max}$. Increasing the value of Q^T causes more frequent stops of the EVs, imposed by a stricter threshold recourse policy. This implies higher expected durations due to the charging operations, as confirmed by the results in Table 10 (see the second, third and fourth columns).

We also consider Q^G values of $70\,\%, 80\,\%, 90\,\%$ of Q^{max}. We recall that Q^G is the SoC at the subsequent node after each detour. Therefore, when Q^G is close to Q^{max}, the EVs require more time to be recharged, given that the slope of the nonlinear charging function

is less steep in this final phase. Indeed, in our instances, the average percentage increase of the total time when Q^G goes from 70 % to 90 % of Q^{max} is 9.32 %. We note that an infeasibility issue occurs in one instance (tc1c40s5cf1, see Table F.37 in Appendix F) when $Q^G = 90\,\% Q^{max}$. This arises because it is not possible to reach nodes with a high level of Q^G, unless the EVs recharge to a SoC exceeding Q^{max}.

Finally, we observe that increasing the value of Q^T results in a decrease in the average number of routes. Conversely, increasing Q^G leads to an increase in the average number of routes. This observation is not directly related to the specific application under investigation but rather mainly depends on the combinatorial nature of the problem. For the sake of illustration, we depict in Fig. 7 the first-stage routes obtained for the instance tc0c40s8cf0 when $Q^T = 20\,\% Q^{max}$ (left panel, four routes) and $Q^T = 40\,\% Q^{max}$ (right panel, two routes).

7. Conclusions

In this paper, we considered the stochastic electric vehicle routing problem with a threshold recourse policy under uncertain energy consumption. We modeled the problem as a two-stage stochastic mixed-integer second-order cone model. In the first stage the sequence of customers visits is determined, while in the second stage a threshold recourse charging policy is implemented. According to this policy, if the state of charge of the battery reaches a predefined threshold level, the EV detours towards an appropriate CS. After a partial recharging operation, the EV completes its first-stage route. As such, the threshold recourse policy allows detours towards CSs along arcs, rather than being confined to nodes.

Given the computational complexity of the model, we proposed a heuristic algorithm based on an Iterated Local Search procedure combined with a Set Partitioning phase. Specifically, we built a pool of high-quality routes through the ILS. To include charging operations in a given first-stage route, we proposed an exact algorithm to solve the stochastic fixed route vehicle charging problem with a threshold policy. Furthermore, to filter unpromising moves during the search phase, we derived two lower bounds on the duration of fixed routes. The most promising combination of routes is then selected by the Set Partitioning formulation. To handle cases involving large-sized instances and several scenarios, we applied a scenario reduction technique to the problem.

We conducted extensive computational experiments on instances adapted from the EVRP literature. We assessed the effectiveness of the heuristic in the deterministic and stochastic settings, showing good performances both in terms of solution quality and computational time. We analyzed the impact of different probability distributions. Moreover, we conducted an in-sample stability analysis to determine the number of scenarios yielding convergence. In addition, we evaluated the impact of uncertainty through the computation of stochastic measures, showing the importance of considering energy consumption uncertainty in this setting. Finally, we discussed managerial insights by examining the model's behaviour under varying conditions.

Regarding future developments, different streams of research can originate from this work. First of all, the generation of the scenario trees could be performed on the basis of real data, taking into account spatial and temporal correlations among travel speeds and energy consumptions. To this end, the increasingly available spatio-temporal data coming from different sources could be used to identify a suitable scenarios representation by applying Deep Learning techniques. Secondly, alternative recourse policies could be examined in order to relax some of the SEVRP-T assumptions, e.g., uniform energy consumption per unit of distance. Thirdly, tighter lower bounds could be derived to further strengthen the filtering phase of the heuristic. Furthermore, exact decomposition strategies, such as L-shaped methods, could be employed. Finally, a dynamic version of the stochastic problem monitoring energy consumption at an instantaneous level merits future investigation.

CRediT authorship contribution statement

Andrea Spinelli: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis; **Dario Bezzi:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Ola Jabali:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization; **Francesca Maggioni:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The author Ola Jabali is an Associate Editor for Transportation Research Part C: Emerging Technologies and was not involved in the editorial review or the decision to publish this article.

Acknowledgements

The authors would like to thank the review team whose comments contributed to enhancing the quality of the manuscript. This work has been supported by "ULTRA OPTYMAL - Urban Logistics and sustainable TRAnsportation: OPtimization under uncertainTY

and MAchine Learning", a PRIN2020 project funded by the Italian University and Research Ministry (grant number 20207C8T9M). This study was also carried out within the MOST - Sustainable Mobility National Research Center and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) - MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 - D.D. 1033 17/06/2022, CN00000023), Spoke 5 "Light Vehicle and Active Mobility" and Spoke 10 "Logistics and Freight". This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. Andrea Spinelli acknowledges the organizers and the experts of the EURO Summer Institute ESI2024 "Decision-making under uncertainty for commodities and financial markets", held in Ischia (Italy) from September 15, to 25, 2024. Finally, the authors acknowledge the support received from Gruppo Nazionale per il Calcolo Scientifico (GNCS-INdAM).

Appendix A. Algorithm for the selection of non-dominated charging stations

See Algorithm 5.

Algorithm 5 NDCS(i, j).

Input: Nodes $i, j \in \mathcal{I}^+$
Output: A set C_{ij} of non-dominated CSs

function NDCS
  $C_{ij} \leftarrow \mathcal{K}$ ▷ All CSs are inserted
  $\widetilde{\mathcal{K}} \leftarrow \mathcal{K}$ ▷ Auxiliary set to span over all CSs
  for every CS $k_1 \in \mathcal{K}$ **do**
    $\widetilde{\mathcal{K}} \leftarrow \widetilde{\mathcal{K}} \setminus \{k_1\}$
    $d_{ij}^{(k_1)} \leftarrow$ minimum distance between CS k_1 and arc (i, j)
    if $j = 0$ **then**
      $\overline{q}_{k_1} \leftarrow e_{k_1 j}$ ▷ SoC at CS k_1 after charging, before traversing the last arc
    else
      $\overline{q}_{k_1} \leftarrow Q^G + e_{k_1 j}$
    if $Q^T < \frac{e_{ij}}{d_{ij}} d_{ij}^{(k_1)}$ or $\overline{q}_{k_1} > Q^{max}$ **then**
      $C_{ij} \leftarrow C_{ij} \setminus \{k_1\}$ ▷ Remove CS k_1 from the set C_{ij}
    else
      for every CS $k_2 \in \widetilde{\mathcal{K}}$ **do**
        $d_{ij}^{(k_2)} \leftarrow$ minimum distance between CS k_2 and arc (i, j)
        if k_1 and k_2 have the same technology **then**
          if $d_{k_1 j} > d_{k_2 j}$ **then**
            if $d_{ij}^{(k_1)} > \max\{d_{ik_2}, d_{k_2 j}\}$ **then**
              $C_{ij} \leftarrow C_{ij} \setminus \{k_1\}$
          else
            if $d_{ij}^{(k_2)} > \max\{d_{ik_1}, d_{k_1 j}\}$ **then**
              $C_{ij} \leftarrow C_{ij} \setminus \{k_2\}$
        else if k_1 has a slower technology than k_2 **then**
          if $d_{ij}^{(k_1)} + d_{k_1 j} > \max\{d_{ik_2}, d_{k_2 j}\} + d_{k_2 j}$ **then**
            $C_{ij} \leftarrow C_{ij} \setminus \{k_1\}$
        else if k_1 has a faster technology than k_2 **then**
          if $d_{ij}^{(k_2)} + d_{k_2 j} > \max\{d_{ik_1}, d_{k_1 j}\} + d_{k_1 j}$ **then**
            $C_{ij} \leftarrow C_{ij} \setminus \{k_2\}$
  return C_{ij}

Appendix B. Fast forward selection algorithm for scenario reduction

See Algorithm 6.

In the initial step of the procedure (lines 2-9), problem Eq. (64) is solved, identifying scenario $\overline{s}_1$ and set $\mathcal{S}_{[1]}^{del}$. Next, the recursive phase (lines 10-19) selects the remaining scenarios $\overline{s}_2, \ldots, \overline{s}_{|\mathcal{S}^{red}|}$ to retain according to Eq. (65). Upon completion, sets $\mathcal{S}^{del}$ and $\mathcal{S}^{red}$ are determined (line 20). Finally, the optimal redistribution rule Eq. (63) is applied to compute the probabilities of the reduced scenarios (lines 21-29).

Appendix C. Detailed results for instances with 10, 15, 20 customers - deterministic setting

In Tables C.11–C.13, we report the results of the numerical experiments for the instances with 10, 15 and 20 customers, in the deterministic setting ($|S| = 1$). Specifically, we provide a comparison between ILS-SP heuristic (Algorithm 1) and Gurobi solver, in terms of CPU time, number of routes and optimality of the solution. The acronym "LB" stands for Lower Bound provided by Gurobi and the asterisk indicates that the time limit of 10800 seconds has been reached.

Algorithm 6 FFS (see Heitsch and Römisch, 2003 and Gioia, 2023).

Input: A set S of scenarios e_s with probabilities p_s, the cardinality $|S^{red}|$ of the set of reduced scenarios
Output: The set S^{red} of reduced scenarios with probabilities p^{red}

function FFS
  $z^{[1]} \leftarrow \infty, \overline{s}_1 \leftarrow 0$
  for every couple of scenarios $(o, \tilde{s}) \in S$ **do**
    $c^{[1]}_{o\tilde{s}} \leftarrow \left\| e_o - e_{\tilde{s}} \right\|_2$ ▷ Distance between scenarios
  for every scenario $\tilde{s} \in S$ **do** ▷ See problem (64)
    $z_{\tilde{s}} \leftarrow \sum_{o \in S \setminus \{\tilde{s}\}} p_o c^{[1]}_{o\tilde{s}}$
    if $z_{\tilde{s}} < z^{[1]}$ **then**
      $z^{[1]} \leftarrow z_{\tilde{s}}, \overline{s}_1 \leftarrow \tilde{s}$
  $S^{del}_{[1]} \leftarrow S \setminus \{\overline{s}_1\}$
  $m \leftarrow 2$
  while $m \leq |S^{red}|$ **do**
    $z^{[m]} \leftarrow \infty, \overline{s}_m \leftarrow 0$
    for every couple of scenarios $(o, \tilde{s}) \in S^{del}_{[m-1]}$ **do**
      $c^{[m]}_{o,\tilde{s}} \leftarrow \min\{c^{[m-1]}_{o\tilde{s}}, c^{[m-1]}_{o\overline{s}_{m-1}}\}$ ▷ Update distance between scenarios
    for every scenario $\tilde{s} \in S^{del}_{[m-1]}$ **do** ▷ See (65)
      $z_{\tilde{s}} \leftarrow \sum_{o \in S^{del}_{[m-1]} \setminus \{\tilde{s}\}} p_o c^{[m]}_{o\tilde{s}}$
      if $z_{\tilde{s}} < z^{[m]}$ **then**
        $z^{[m]} \leftarrow z_{\tilde{s}}, \overline{s}_m \leftarrow \tilde{s}$
    $S^{del}_{[m]} \leftarrow S^{del}_{[m-1]} \setminus \{\overline{s}_m\}$
  $S^{del} \leftarrow S^{del}_{[|S^{red}|]}, S^{red} \leftarrow S \setminus S^{del}$ ▷ Sets of deleted and reduced scenarios
  for every scenario $s \in S^{red}$ **do** ▷ See (63)
    $c^*_s \leftarrow \infty, S^{del}_s \leftarrow \emptyset$
    for every scenario $\tilde{s} \in S^{del}$ **do**
      $c_{s\tilde{s}} \leftarrow \left\| e_s - e_{\tilde{s}} \right\|_2$
      if $c_{s\tilde{s}} < c^*_s$ **then**
        $c^*_s \leftarrow c_{s\tilde{s}}, S^{del}_s \leftarrow \tilde{s}$
      else if $c_{s\tilde{s}} = c^*_s$ **then**
        $S^{del}_s \leftarrow S^{del}_s \cup \{\tilde{s}\}$
    $p^{red}_s \leftarrow p_s + \sum_{\tilde{s} \in S^{del}_s} p_{\tilde{s}}$ ▷ Optimal redistribution rule
  return S^{red}, p^{red}

Table C.11
10 customers - deterministic setting.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	11.05	0.76	2	2	9.97	9.97	9.97	0 %	0 %
tc0c10s2ct1	10.78	9.51	2	2	9.84	9.84	9.84	0 %	0 %
tc0c10s3cf1	11.71	2.54	2	2	9.97	9.97	9.97	0 %	0 %
tc0c10s3ct1	11.99	16.75	2	2	9.84	9.84	9.84	0 %	0 %
tc1c10s2cf2	11.04	10.76	1	1	7.08	7.08	7.08	0 %	0 %
tc1c10s2cf4	12.36	6.70	2	2	11.85	11.85	11.85	0 %	0 %
tc1c10s2ct2	10.18	12.72	1	1	7.25	7.25	7.25	0 %	0 %
tc1c10s2ct3	11.66	354.42	1	1	11.04	11.04	11.04	0 %	0 %
tc1c10s2ct4	11.80	7.30	1	1	11.36	11.36	11.36	0 %	0 %
tc1c10s3cf2	11.27	4.02	1	1	7.08	7.08	7.08	0 %	0 %
tc1c10s3cf4	13.26	11.96	2	2	11.85	11.85	11.85	0 %	0 %
tc1c10s3ct2	11.01	19.04	1	1	7.25	7.25	7.25	0 %	0 %
tc1c10s3ct3	13.68	634.33	1	1	9.83	9.83	9.83	0 %	0 %
tc1c10s3ct4	12.47	15.51	1	1	11.36	11.36	11.36	0 %	0 %
tc2c10s2cf0	8.67	2.41	2	2	8.21	8.21	8.21	0 %	0 %
tc2c10s2ct0	11.74	41.03	2	2	7.45	7.45	7.45	0 %	0 %
tc2c10s3cf0	8.85	2.71	2	2	8.21	8.21	8.21	0 %	0 %
tc2c10s3ct0	11.71	136.18	2	2	7.53	7.53	7.53	0 %	0 %

Table C.12
15 customers - deterministic setting.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c15s3cf2	45.18	268.10	2	2	12.32	12.32	12.32	0 %	0 %
tc0c15s3ct2	35.53	142.13	2	2	11.21	11.21	11.21	0 %	0 %
tc0c15s4cf2	48.01	2817.04	2	2	12.32	12.32	12.32	0 %	0 %
tc0c15s4ct2	40.79	387.32	2	2	11.21	11.21	11.21	0 %	0 %
tc1c15s3cf1	25.88	4064.50	1	1	7.45	7.45	7.45	0 %	0 %
tc1c15s3cf3	42.24	3938.97	2	2	10.71	10.71	10.71	0 %	0 %
tc1c15s3cf4	60.42	268.63	2	2	11.80	11.80	11.80	0 %	0 %
tc1c15s3ct1	24.30	4404.25	1	1	7.75	7.75	7.75	0 %	0 %
tc1c15s3ct3	33.11	883.37	1	1	9.62	9.62	9.62	0 %	0 %
tc1c15s3ct4	50.50	1680.79	1	1	11.50	11.50	11.50	0 %	0 %
tc1c15s4cf1	24.69	6332.66	1	1	7.45	7.45	7.45	0 %	0 %
tc1c15s4cf3	40.10	4477.57	2	2	10.71	10.71	10.71	0 %	0 %
tc1c15s4cf4	74.77	431.53	2	2	11.80	11.80	11.80	0 %	0 %
tc1c15s4ct1	23.92	6833.78	1	1	7.75	7.75	7.75	0 %	0 %
tc1c15s4ct3	38.16	10800*	1	1	10.60	10.60	8.92	15.80 %	0 %
tc1c15s4ct4	34.06	128.83	1	1	11.34	11.34	11.34	0 %	0 %
tc2c15s3cf0	37.00	10800*	1	1	9.30	9.30	6.55	29.52 %	0 %
tc2c15s3ct0	32.37	10800*	1	1	9.60	9.60	6.42	33.13 %	0 %
tc2c15s4cf0	37.57	10800*	1	1	9.30	9.30	6.55	29.57 %	0 %
tc2c15s4ct0	32.65	10800*	1	1	9.26	9.26	6.39	30.96 %	0 %

Table C.13
20 customers - deterministic setting.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c20s3cf2	98.32	10800*	2	2	13.24	13.24	12.41	6.28 %	0 %
tc0c20s3ct2	71.52	8994.82	2	2	12.27	12.27	12.27	0 %	0 %
tc0c20s4cf2	104.51	10800*	2	2	13.10	13.10	11.76	10.19 %	0 %
tc0c20s4ct2	75.69	10790.63	2	2	12.20	12.20	12.20	0 %	0 %
tc1c20s3cf1	117.35	10800*	2	2	12.99	12.99	11.82	9.03 %	0 %
tc1c20s3cf3	68.04	10800*	1	1	11.28	11.28	9.55	15.27 %	0 %
tc1c20s3cf4	133.06	10800*	3	3	14.22	14.22	13.65	4.02 %	0 %
tc1c20s3ct1	120.25	10800*	2	2	13.93	14.19	11.48	19.13 %	−1.86 %
tc1c20s3ct3	73.23	10800*	1	1	10.11	10.11	9.21	8.92 %	0 %
tc1c20s3ct4	94.37	10800*	2	2	14.12	14.12	13.63	3.44 %	0 %
tc1c20s4cf1	81.71	10800*	1	1	12.15	12.29	10.18	17.12 %	−1.14 %
tc1c20s4cf3	69.05	10800*	1	1	11.28	11.28	9.62	14.73 %	0 %
tc1c20s4cf4	151.06	10800*	3	3	14.22	14.22	13.59	4.41 %	0 %
tc1c20s4ct1	86.74	10800*	2	2	12.56	12.91	10.25	20.59 %	−2.70 %
tc1c20s4ct3	63.89	10800*	1	1	10.71	10.71	9.53	11.07 %	0 %
tc1c20s4ct4	101.93	10800*	2	2	14.05	14.05	13.67	2.66 %	0 %
tc2c20s3cf0	82.92	10800*	1	1	10.68	10.68	8.56	19.86 %	0 %
tc2c20s3ct0	71.89	10800*	1	1	10.86	10.86	7.86	27.58 %	0 %
tc2c20s4cf0	88.40	10800*	1	1	10.68	10.68	8.54	20.02 %	0 %
tc2c20s4ct0	75.89	10800*	1	1	10.86	10.86	7.76	28.55 %	0 %

Appendix D. Detailed results for instances with 10 customers - stochastic setting

In Tables D.14–D.27, we report the results of the numerical experiments for the instances with 10 customers in the stochastic setting. Specifically, we start by providing a comparison between ILS-SP heuristic (Algorithm 1) and Gurobi solver, in terms of CPU time, number of routes and optimality of the solution. We consider an increasing number of scenarios ($|S| = \{5, 10, 20, 50\}$). The acronym "LB" stands for Lower Bound provided by Gurobi and the asterisk indicates that the time limit of 10800s has been reached.

Different probability distributions on the energy consumption are explored: uniform (see Appendix D.1), normal (see Appendix D.2), exponential (see Appendix D.3). Then, we report the detailed results of an in-sample stability analysis with respect to the size of the scenario tree (see Appendix D.4). Finally, we show the values of classic stochastic measures obtained in the case $|S| = 20$ (see Appendix D.5).

D.1. Uniform distribution

See Tables D.14–D.17.

Table D.14
10 customers - $|S| = 5$ - uniform distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	19.34	16.46	2	2	10.33	10.33	10.33	0 %	0 %
tc0c10s2ct1	20.00	86.67	2	2	10.04	10.04	10.04	0 %	0 %
tc0c10s3cf1	20.37	41.37	2	2	11.51	11.51	11.51	0 %	0 %
tc0c10s3ct1	28.78	263.43	2	2	10.02	10.02	10.02	0 %	0 %
tc1c10s2cf2	28.80	115.84	1	1	7.13	7.13	7.13	0 %	0 %
tc1c10s2cf4	24.65	281.74	2	2	11.90	11.90	11.90	0 %	0 %
tc1c10s2ct2	34.30	626.87	1	1	7.36	7.36	7.36	0 %	0 %
tc1c10s2ct3	24.49	3913.02	1	1	10.24	10.24	10.24	0 %	0 %
tc1c10s2ct4	29.63	383.82	2	2	11.95	11.95	11.95	0 %	0 %
tc1c10s3cf2	37.12	283.98	2	2	7.43	7.43	7.43	0 %	0 %
tc1c10s3cf4	30.08	1621.69	2	2	12.12	12.12	12.12	0 %	0 %
tc1c10s3ct2	45.34	2103.30	2	2	7.55	7.55	7.55	0 %	0 %
tc1c10s3ct3	33.99	10800*	1	1	9.73	9.73	8.66	10.95 %	0 %
tc1c10s3ct4	37.95	10800*	2	2	12.13	12.13	11.97	1.34 %	0 %
tc2c10s2cf0	19.73	9.31	2	2	7.75	7.75	7.75	0 %	0 %
tc2c10s2ct0	37.98	1062.10	2	2	7.48	7.48	7.48	0 %	0 %
tc2c10s3cf0	18.99	14.66	2	2	8.25	8.25	8.25	0 %	0 %
tc2c10s3ct0	39.34	2436.22	2	2	7.57	7.57	7.57	0 %	0 %

Table D.15
10 customers - $|S| = 10$ - uniform distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	24.01	170.48	2	2	11.97	11.97	11.97	0 %	0 %
tc0c10s2ct1	30.30	5747.90	2	2	9.99	9.99	9.99	0 %	0 %
tc0c10s3cf1	27.15	309.56	2	2	12.27	12.27	12.27	0 %	0 %
tc0c10s3ct1	48.58	10800*	2	2	10.00	10.00	9.99	0.09 %	0 %
tc1c10s2cf2	62.05	1603.15	2	2	7.44	7.44	7.44	0 %	0 %
tc1c10s2cf4	42.33	10800*	2	2	12.14	12.14	11.88	2.19 %	0 %
tc1c10s2ct2	71.35	10560.38	2	2	7.59	7.59	7.59	0 %	0 %
tc1c10s2ct3	44.90	10800*	1	1	10.39	10.39	9.89	4.83 %	0 %
tc1c10s2ct4	50.76	10800*	2	2	12.00	12.00	11.84	1.34 %	0 %
tc1c10s3cf2	70.74	1600.98	2	2	7.51	7.51	7.51	0 %	0 %
tc1c10s3cf4	50.97	10800*	2	2	12.00	12.00	11.41	4.91 %	0 %
tc1c10s3ct2	84.34	10800*	2	2	7.71	7.73	7.13	7.74 %	−0.19 %
tc1c10s3ct3	75.80	10800*	1	1	9.67	9.84	6.88	30.16 %	−1.82 %
tc1c10s3ct4	64.80	10800*	2	2	12.10	12.10	11.10	8.29 %	0 %
tc2c10s2cf0	29.15	658.23	2	2	8.02	8.02	8.02	0 %	0 %
tc2c10s2ct0	73.36	10800*	2	2	7.51	7.51	7.34	2.27 %	0 %
tc2c10s3cf0	28.25	384.49	2	2	8.15	8.15	8.15	0 %	0 %
tc2c10s3ct0	65.38	10800*	2	2	7.55	7.63	6.69	12.34 %	−1.04 %

Table D.16
10 customers - $|S| = 20$ - uniform distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	39.84	10800*	2	2	12.42	12.42	12.18	1.90 %	0 %
tc0c10s2ct1	50.44	10800*	2	2	9.98	9.98	9.94	0.41 %	0 %
tc0c10s3cf1	38.86	10800*	2	2	12.30	12.30	11.77	4.31 %	0 %
tc0c10s3ct1	77.55	10800*	2	2	10.00	10.00	9.95	0.46 %	0 %
tc1c10s2cf2	108.03	10800*	2	2	7.52	7.52	7.33	2.49 %	0 %
tc1c10s2cf4	74.11	10800*	2	2	12.21	12.21	11.56	5.32 %	0 %
tc1c10s2ct2	135.72	10800*	2	2	7.73	7.73	6.89	10.84 %	0 %
tc1c10s2ct3	68.96	10800*	1	1	10.44	10.44	7.35	29.63 %	0 %
tc1c10s2ct4	81.00	10800*	2	2	12.01	12.01	11.80	1.73 %	0 %
tc1c10s3cf2	123.01	10800*	2	2	7.64	7.64	7.49	2.03 %	0 %
tc1c10s3cf4	89.40	10800*	2	2	12.08	12.08	10.48	13.29 %	0 %
tc1c10s3ct2	159.62	10800*	2	2	7.69	7.74	6.68	13.68 %	−0.65 %
tc1c10s3ct3	115.46	10800*	1	1	9.91	10.67	6.18	42.05 %	−7.09 %
tc1c10s3ct4	103.33	10800*	2	2	12.05	12.05	10.21	15.29 %	0 %
tc2c10s2cf0	47.07	10800*	2	2	8.09	8.09	7.41	8.34 %	0 %
tc2c10s2ct0	132.98	10800*	2	2	7.52	7.52	5.52	26.55 %	0 %
tc2c10s3cf0	47.50	10800*	3	3	11.06	11.06	10.44	5.59 %	0 %
tc2c10s3ct0	134.02	10800*	2	2	7.60	7.82	5.06	35.29 %	−2.89 %

Table D.17
10 customers - $|S| = 50$ - uniform distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	84.67	10800*	2	2	12.40	12.40	11.66	5.97 %	0 %
tc0c10s2ct1	115.79	10800*	2	2	10.02	10.02	9.86	1.56 %	0 %
tc0c10s3cf1	79.99	10800*	2	2	12.24	12.24	11.79	3.65 %	0 %
tc0c10s3ct1	162.68	10800*	2	2	10.20	12.39	8.60	30.56 %	−17.63 %
tc1c10s2cf2	274.73	10800*	2	2	7.84	7.60	7.00	7.87 %	3.16 %
tc1c10s2cf4	164.11	10800*	2	2	12.22	12.22	10.94	10.51 %	0 %
tc1c10s2ct2	350.54	10800*	2	2	7.75	9.56	6.04	36.82 %	−18.83 %
tc1c10s2ct3	159.61	10800*	1	2	10.47	12.95	6.13	52.68 %	−19.13 %
tc1c10s2ct4	183.76	10800*	2	2	12.04	12.04	10.79	10.42 %	0 %
tc1c10s3cf2	267.07	10800*	2	2	7.63	7.63	6.99	8.34 %	−0.02 %
tc1c10s3cf4	183.45	10800*	2	2	12.11	12.79	9.92	22.39 %	−5.30 %
tc1c10s3ct2	389.91	10800*	2	2	7.77	9.01	5.45	39.58 %	−12.53 %
tc1c10s3ct3	275.79	10800*	1	1	10.08	11.85	5.23	55.88 %	−14.95 %
tc1c10s3ct4	215.45	10800*	1	2	12.05	12.86	9.26	28.01 %	−5.90 %
tc2c10s2cf0	104.34	10800*	3	3	11.14	11.14	10.07	9.59 %	0 %
tc2c10s2ct0	310.73	10800*	2	2	7.65	8.17	4.33	47.00 %	−6.09 %
tc2c10s3cf0	111.19	10800*	3	3	11.07	11.07	10.02	9.45 %	0 %
tc2c10s3ct0	317.84	10800*	2	2	7.68	7.78	3.51	54.82 %	−1.28 %

D.2. Normal distribution

See Tables D.18–D.21.

Table D.18
10 customers - $|S| = 5$ - normal distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	16.46	19.54	2	2	11.68	11.68	11.68	0 %	0 %
tc0c10s2ct1	21.11	72.47	2	2	9.84	9.84	9.84	0 %	0 %
tc0c10s3cf1	18.28	36.73	2	2	11.51	11.51	11.51	0 %	0 %
tc0c10s3ct1	33.38	344.20	2	2	9.94	9.94	9.94	0 %	0 %
tc1c10s2cf2	35.56	243.62	1	1	7.14	7.14	7.14	0 %	0 %
tc1c10s2cf4	25.24	377.06	2	2	12.02	12.02	12.02	0 %	0 %
tc1c10s2ct2	38.41	861.75	1	1	7.28	7.28	7.28	0 %	0 %
tc1c10s2ct3	25.91	3789.47	1	1	10.83	10.83	10.83	0 %	0 %
tc1c10s2ct4	29.44	302.51	2	2	12.07	12.07	12.07	0 %	0 %
tc1c10s3cf2	36.73	250.19	1	1	7.19	7.19	7.19	0 %	0 %
tc1c10s3cf4	31.10	2289.60	3	3	12.22	12.22	12.22	0 %	0 %
tc1c10s3ct2	35.18	1233.05	1	1	7.39	7.39	7.39	0 %	0 %
tc1c10s3ct3	37.96	10800*	1	1	9.86	9.86	9.05	8.19 %	0 %
tc1c10s3ct4	34.60	1946.13	2	2	11.91	11.91	11.91	0 %	0 %
tc2c10s2cf0	19.05	11.82	2	2	8.25	8.25	8.25	0 %	0 %
tc2c10s2ct0	35.28	916.66	2	2	7.55	7.55	7.55	0 %	0 %
tc2c10s3cf0	19.19	10.06	2	2	8.22	8.22	8.22	0 %	0 %
tc2c10s3ct0	40.45	2473.44	2	2	7.58	7.58	7.58	0 %	0 %

Table D.19
10 customers - $|S| = 10$ - normal distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	25.90	84.80	2	2	11.82	11.82	11.82	0 %	0 %
tc0c10s2ct1	31.95	573.66	2	2	9.84	9.84	9.84	0 %	0 %
tc0c10s3cf1	27.95	2472.73	2	2	11.82	11.82	11.82	0 %	0 %
tc0c10s3ct1	57.71	10800*	2	2	10.08	10.08	10.07	0.11 %	0 %
tc1c10s2cf2	60.75	708.13	1	1	7.37	7.37	7.37	0 %	0 %
tc1c10s2cf4	43.56	10800*	2	2	12.34	12.34	12.15	1.60 %	0 %
tc1c10s2ct2	80.66	10800*	2	2	7.60	7.60	7.43	2.28 %	0 %
tc1c10s2ct3	44.34	10800*	1	1	10.57	10.57	9.89	6.43 %	0 %
tc1c10s2ct4	49.14	10800*	2	2	12.02	12.02	11.93	0.76 %	0 %
tc1c10s3cf2	58.36	1153.97	1	1	7.17	7.17	7.17	0 %	0 %
tc1c10s3cf4	50.25	10800*	2	2	12.24	12.24	11.11	9.21 %	0 %
tc1c10s3ct2	62.68	10800*	1	1	7.37	7.37	7.33	0.56 %	0 %
tc1c10s3ct3	76.17	10800*	1	1	10.06	10.33	6.91	33.13 %	−2.57 %
tc1c10s3ct4	59.70	10800*	2	2	11.99	11.99	10.88	9.22 %	0 %
tc2c10s2cf0	26.79	622.84	2	2	8.59	8.59	8.59	0 %	0 %
tc2c10s2ct0	65.50	10800*	2	2	7.55	7.55	7.41	1.80 %	0 %
tc2c10s3cf0	28.20	398.56	2	2	8.24	8.24	8.24	0 %	0 %
tc2c10s3ct0	72.29	10800*	2	2	7.58	7.71	6.44	16.45 %	−1.68 %

Table D.20
10 customers - $|S| = 20$ - normal distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	38.78	10800*	2	2	11.99	11.99	11.97	0.13 %	0 %
tc0c10s2ct1	53.79	10800*	2	2	9.91	9.91	9.89	0.18 %	0 %
tc0c10s3cf1	36.44	10800*	2	2	12.13	11.96	11.88	0.73 %	1.35 %
tc0c10s3ct1	99.69	10800*	2	2	10.01	10.01	9.97	0.40 %	0 %
tc1c10s2cf2	122.20	10800*	2	2	7.59	7.59	7.28	4.01 %	0 %
tc1c10s2cf4	74.79	10800*	2	2	12.18	12.18	11.45	6.03 %	0 %
tc1c10s2ct2	170.25	10800*	2	2	7.67	7.71	6.80	11.81 %	−0.54 %
tc1c10s2ct3	78.43	10800*	1	1	10.41	10.44	7.34	29.66 %	−0.28 %
tc1c10s2ct4	86.48	10800*	2	2	12.05	12.05	11.73	2.67 %	0 %
tc1c10s3cf2	122.50	10800*	2	2	7.56	7.56	7.39	2.27 %	0 %
tc1c10s3cf4	92.91	10800*	2	2	12.09	12.09	11.20	7.33 %	0 %
tc1c10s3ct2	174.34	10800*	2	2	7.77	7.86	6.47	17.67 %	−1.10 %
tc1c10s3ct3	125.17	10800*	1	1	10.05	10.64	6.36	40.23 %	−5.54 %
tc1c10s3ct4	98.57	10800*	2	2	12.10	12.10	10.38	14.21 %	0 %
tc2c10s2cf0	45.46	10800*	3	3	11.05	11.05	10.57	4.32 %	0 %
tc2c10s2ct0	138.97	10800*	2	2	7.55	7.61	5.47	28.10 %	−0.78 %
tc2c10s3cf0	47.94	10800*	3	3	11.14	11.14	10.41	6.52 %	0 %
tc2c10s3ct0	130.82	10800*	2	2	7.62	8.03	5.29	34.16 %	−5.15 %

Table D.21
10 customers - $|S| = 50$ - normal distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	83.88	10800*	2	2	12.11	12.11	11.40	5.89 %	0 %
tc0c10s2ct1	109.10	10800*	2	2	10.11	10.11	9.92	1.89 %	0 %
tc0c10s3cf1	79.29	10800*	2	2	12.12	12.12	11.87	2.03 %	0 %
tc0c10s3ct1	201.46	10800*	2	1	10.17	11.88	8.54	28.09 %	−14.44 %
tc1c10s2cf2	290.62	10800*	2	2	7.61	7.62	7.04	7.63 %	−0.09 %
tc1c10s2cf4	169.22	10800*	2	2	12.06	12.06	10.96	9.19 %	0 %
tc1c10s2ct2	365.59	10800*	2	2	7.76	8.50	5.94	30.18 %	−8.34 %
tc1c10s2ct3	153.56	10800*	1	1	11.26	12.81	6.09	52.48 %	−12.08 %
tc1c10s2ct4	176.25	10800*	2	2	12.08	12.08	11.14	7.82 %	0 %
tc1c10s3cf2	277.53	10800*	2	2	7.61	7.61	7.14	6.26 %	0 %
tc1c10s3cf4	212.29	10800*	2	3	12.16	12.50	10.19	18.46 %	−2.73 %
tc1c10s3ct2	347.34	10800*	2	2	7.77	8.63	5.52	36.09 %	−9.03 %
tc1c10s3ct3	272.49	10800*	1	2	10.07	15.95	5.24	67.16 %	−34.81 %
tc1c10s3ct4	232.08	10800*	2	1	12.09	12.81	9.26	27.67 %	−5.60 %
tc2c10s2cf0	100.81	10800*	3	3	11.07	11.07	9.99	9.76 %	0 %
tc2c10s2ct0	313.85	10800*	2	2	7.62	7.80	4.69	39.86 %	−2.29 %
tc2c10s3cf0	112.86	10800*	3	3	11.23	11.23	9.99	11.05 %	0 %
tc2c10s3ct0	315.33	10800*	2	1	7.59	8.03	3.73	53.52 %	−5.38 %

D.3. Exponential distribution

See Tables D.22–D.25.

Table D.22
10 customers - $|S| = 5$ - exponential distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	16.93	18.53	2	2	11.85	11.85	11.85	0 %	0 %
tc0c10s2ct1	20.68	87.77	2	2	9.96	9.96	9.96	0 %	0 %
tc0c10s3cf1	18.00	22.86	2	2	11.67	11.67	11.67	0 %	0 %
tc0c10s3ct1	39.64	326.96	2	2	9.94	9.94	9.94	0 %	0 %
tc1c10s2cf2	29.53	144.33	1	1	7.12	7.12	7.12	0 %	0 %
tc1c10s2cf4	26.84	436.43	2	2	12.15	12.15	12.15	0 %	0 %
tc1c10s2ct2	37.81	816.91	1	1	7.34	7.34	7.34	0 %	0 %
tc1c10s2ct3	28.53	5602.28	1	1	10.76	10.76	10.76	0 %	0 %
tc1c10s2ct4	26.13	133.95	2	2	12.08	12.08	12.08	0 %	0 %
tc1c10s3cf2	31.46	140.90	1	1	7.12	7.12	7.12	0 %	0 %
tc1c10s3cf4	33.29	1277.58	2	2	12.16	12.16	12.16	0 %	0 %
tc1c10s3ct2	32.40	1739.31	1	1	7.37	7.37	7.37	0 %	0 %
tc1c10s3ct3	40.71	10800*	1	1	9.88	9.88	7.78	21.25 %	0 %
tc1c10s3ct4	34.68	3114.16	1	1	12.02	12.02	12.02	0 %	0 %
tc2c10s2cf0	18.52	9.84	2	2	7.79	7.79	7.79	0 %	0 %
tc2c10s2ct0	34.08	1228.23	2	2	7.50	7.50	7.50	0 %	0 %
tc2c10s3cf0	19.12	14.33	2	2	7.97	7.97	7.97	0 %	0 %
tc2c10s3ct0	35.27	2420.12	2	2	7.46	7.46	7.46	0 %	0 %

Table D.23
10 customers - $|S| = 10$ - exponential distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	24.20	182.87	2	2	11.84	11.84	11.84	0 %	0 %
tc0c10s2ct1	34.89	9119.73	2	2	9.94	9.94	9.94	0 %	0 %
tc0c10s3cf1	26.88	239.33	2	2	11.84	11.84	11.84	0 %	0 %
tc0c10s3ct1	50.49	10800*	2	2	10.01	10.01	10.01	0.04 %	0 %
tc1c10s2cf2	57.31	1572.35	1	1	7.17	7.17	7.17	0 %	0 %
tc1c10s2cf4	41.89	10800*	2	2	12.01	12.01	11.75	2.21 %	0 %
tc1c10s2ct2	74.09	4604.89	1	1	7.36	7.36	7.36	0 %	0 %
tc1c10s2ct3	41.61	10800*	1	1	10.32	10.32	9.29	9.99 %	0 %
tc1c10s2ct4	43.42	10800*	2	2	12.17	12.17	11.92	2.06 %	0 %
tc1c10s3cf2	56.56	507.94	1	1	7.14	7.14	7.14	0 %	0 %
tc1c10s3cf4	50.81	10800*	2	2	12.20	12.20	11.32	7.24 %	0 %
tc1c10s3ct2	64.47	10800*	1	1	7.45	7.45	7.12	4.38 %	0 %
tc1c10s3ct3	69.16	10800*	1	1	10.00	10.47	6.73	35.70 %	−4.45 %
tc1c10s3ct4	66.45	10800*	2	2	12.02	12.02	11.23	6.57 %	0 %
tc2c10s2cf0	28.11	139.37	2	2	8.16	8.16	8.16	0 %	0 %
tc2c10s2ct0	62.22	10800*	2	2	7.54	7.54	7.21	4.32 %	0 %
tc2c10s3cf0	29.53	575.31	2	2	8.01	8.01	8.01	0 %	0 %
tc2c10s3ct0	69.14	10800*	2	2	7.47	7.47	6.90	7.64 %	0 %

Table D.24
10 customers - $|S| = 20$ - exponential distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	38.54	10800*	2	2	11.86	11.86	11.85	0.15 %	0 %
tc0c10s2ct1	51.73	10800*	2	2	9.97	9.97	9.93	0.40 %	0 %
tc0c10s3cf1	40.92	10800*	2	2	12.08	12.08	11.70	3.10 %	0 %
tc0c10s3ct1	83.79	10800*	2	2	9.96	9.96	9.89	0.69 %	0 %
tc1c10s2cf2	107.00	10800*	1	1	7.18	7.18	6.93	3.50 %	0 %
tc1c10s2cf4	72.84	10800*	2	2	12.08	12.08	11.32	6.25 %	0 %
tc1c10s2ct2	137.79	10800*	1	1	7.41	7.41	6.91	6.82 %	0 %
tc1c10s2ct3	69.36	10800*	1	1	10.78	10.78	6.92	35.76 %	0 %
tc1c10s2ct4	81.84	10800*	2	2	12.08	12.08	11.38	5.82 %	0 %
tc1c10s3cf2	113.55	10800*	2	2	7.61	7.61	7.47	1.92 %	0 %
tc1c10s3cf4	84.10	10800*	2	2	12.14	12.14	11.06	8.91 %	0 %
tc1c10s3ct2	158.51	10800*	2	2	7.81	8.30	6.22	25.10 %	−5.91 %
tc1c10s3ct3	132.73	10800*	1	1	10.02	10.74	6.29	41.46 %	−6.73 %
tc1c10s3ct4	101.84	10800*	2	2	11.97	12.16	10.46	13.99 %	−1.56 %
tc2c10s2cf0	44.64	10800*	2	2	8.39	8.39	8.14	2.95 %	0 %
tc2c10s2ct0	130.50	10800*	2	2	7.55	7.55	6.03	20.18 %	0 %
tc2c10s3cf0	42.86	10800*	2	2	8.31	8.31	7.82	5.93 %	0 %
tc2c10s3ct0	130.64	10800*	2	2	7.54	7.54	5.21	30.90 %	0 %

Table D.25
10 customers - $|S| = 50$ - exponential distribution.

Instance	Time (s)		Number of routes		Objective			Gap	
	Algorithm	Model	Algorithm	Model	Algorithm	Model		Gurobi	Algorithm-Model
						Incumbent	LB		
tc0c10s2cf1	85.77	10800*	2	2	12.28	12.28	11.95	2.66 %	0 %
tc0c10s2ct1	114.01	10800*	2	2	9.96	9.96	9.80	1.55 %	0 %
tc0c10s3cf1	88.55	10800*	2	2	12.14	12.14	12.04	0.86 %	0 %
tc0c10s3ct1	184.09	10800*	2	1	9.95	11.93	8.61	27.85 %	−16.61 %
tc1c10s2cf2	272.67	10800*	2	2	7.64	7.64	6.98	8.59 %	−0.05 %
tc1c10s2cf4	155.74	10800*	2	2	12.08	12.08	11.08	8.31 %	0 %
tc1c10s2ct2	277.32	10800*	2	2	7.86	8.67	6.10	29.69 %	−9.09 %
tc1c10s2ct3	155.11	10800*	2	2	11.62	12.65	6.26	50.51 %	−8.13 %
tc1c10s2ct4	173.35	10800*	2	2	12.10	12.10	10.57	12.62 %	0 %
tc1c10s3cf2	253.92	10800*	2	2	7.64	8.02	6.84	14.67 %	−4.74 %
tc1c10s3cf4	183.11	10800*	2	2	12.08	13.14	9.95	24.25 %	−5.54 %
tc1c10s3ct2	351.08	10800*	2	2	7.84	9.07	5.56	38.71 %	−13.16 %
tc1c10s3ct3	245.48	10800*	1	1	10.08	13.77	5.33	61.31 %	−26.77 %
tc1c10s3ct4	229.08	10800*	2	2	12.02	13.75	9.44	31.32 %	−12.56 %
tc2c10s2cf0	92.67	10800*	3	3	10.87	10.87	10.43	4.04 %	0 %
tc2c10s2ct0	292.45	10800*	2	2	7.57	8.01	4.52	43.62 %	−5.60 %
tc2c10s3cf0	114.21	10800*	2	2	8.34	8.34	7.18	13.89 %	0 %
tc2c10s3ct0	318.53	10800*	2	2	7.53	8.11	3.42	57.84 %	−7.01 %

D.4. In-sample stability analysis

See Table D.26.

D.5. Stochastic measures

See Table D.27.

Appendix E. Detailed results for instances with 40, 80 customers

In Tables E.28–E.32, we report the results of the numerical experiments for the instances with 40 and 80 customers. Specifically, for the 40-customer instances, we consider an increasing number of scenarios ($|S| = \{1, 10, 20, 50\}$) and exploring the case where an initial set of 50 scenarios is reduced to 20 through the FFS algorithm (Algorithm 6). For the instances with 80 customers, we set $|S| = 20$. All the results are obtained through ILS-SP heuristic (Algorithm 1) with a maximum number of iterations equal to 2000 and a runtime limit of 10800 seconds. The asterisk indicates that the time limit has been reached. Uniform distribution on the energy consumption is considered.

Appendix F. Detailed results - managerial insights

In Tables F.33–F.37, we report the results of the numerical experiments aimed at deriving managerial insights for the problem under investigation. Specifically, we start by providing a comparison between the arc-level detour perspective proposed in this work and a more conservative node-level alternative (see Appendix F.1). This analysis is carried out on 10-customer instances with uniform energy consumption distribution. The results are obtained using the Gurobi solver. The asterisk indicates that the runtime limit of 10800 seconds has been reached.

Next, we focus on the instances with 40 customers and different combinations of energy threshold parameters (see Appendix F.2). Here, we consider $Q^T = \alpha^T Q^{max}$, with $\alpha^T = \{20\,\%, 30\,\%, 40\,\%\}$, and $Q^G = \alpha^G Q^{max}$, with $\alpha^G = \{70\,\%, 80\,\%, 90\,\%\}$. The number of scenarios is set to 20, reduced from an initial set of 50 through the FFS algorithm (Algorithm 6). All the results are obtained through ILS-SP heuristic (Algorithm 1) with a maximum number of iterations equal to 2000 and a runtime limit of 10800 seconds. Uniform distribution on the energy consumption is considered.

F.1. Comparison of arc-level and node-level detours

See Tables F.33–F.35.

F.2. Sensitivity analysis on the energy threshold parameters

See Tables F.36–F.37.

Table D.26
Approximate in-sample stability analysis based on the objective function value obtained from ILS-SP heuristic (Algorithm 1) for the 10-customer instances and $|S| = \{5, 10, 20, 50, 100\}$. Three probability distributions for the energy consumption are considered.

Instance	Uniform					Normal					Exponential				
	5	10	20	50	100	5	10	20	50	100	5	10	20	50	100
tc0c10s2cf1	10.33	11.97	12.42	12.40	12.45	11.68	11.82	11.99	12.11	12.08	11.85	11.84	11.86	12.28	12.24
tc0c10s2ct1	10.04	9.99	9.98	10.02	10.16	9.84	9.84	9.91	10.11	10.11	9.96	9.94	9.97	9.96	10.15
tc0c10s3cf1	11.51	12.27	12.30	12.24	12.22	11.51	11.82	12.13	12.12	12.19	11.67	11.84	12.08	12.14	12.18
tc0c10s3ct1	10.02	10.00	10.00	10.20	10.20	9.94	10.08	10.01	10.17	10.18	9.94	10.01	9.96	9.95	9.93
tc1c10s2cf2	7.13	7.44	7.52	7.84	8.79	7.14	7.37	7.59	7.61	7.62	7.12	7.17	7.18	7.64	7.66
tc1c10s2cf4	11.90	12.14	12.21	12.22	12.23	12.02	12.34	12.18	12.06	12.09	12.15	12.01	12.08	12.08	12.05
tc1c10s2ct2	7.36	7.59	7.73	7.75	7.76	7.28	7.60	7.67	7.76	7.78	7.34	7.36	7.41	7.86	7.86
tc1c10s2ct3	10.24	10.39	10.44	10.47	11.59	10.83	10.57	10.41	11.26	11.60	10.76	10.32	10.78	11.62	11.62
tc1c10s2ct4	11.95	12.00	12.01	12.04	12.06	12.07	12.02	12.05	12.08	12.08	12.08	12.17	12.08	12.10	12.07
tc1c10s3cf2	7.43	7.51	7.65	7.63	7.65	7.19	7.17	7.56	7.61	7.63	7.12	7.14	7.61	7.64	8.85
tc1c10s3cf4	12.12	12.00	12.08	12.11	12.17	12.22	12.24	12.09	12.16	12.19	12.16	12.20	12.14	12.08	12.10
tc1c10s3ct2	7.55	7.71	7.69	7.77	7.78	7.39	7.37	7.77	7.77	7.72	7.37	7.45	7.81	7.84	7.82
tc1c10s3ct3	9.73	9.67	9.91	10.08	10.13	9.86	10.06	10.05	10.07	10.04	9.88	10.00	10.02	10.08	10.11
tc1c10s3ct4	12.13	12.10	12.05	12.05	12.05	11.91	11.99	12.10	12.09	12.04	12.02	12.02	11.97	12.02	12.04
tc2c10s2cf0	7.75	8.02	8.09	11.14	11.09	8.25	8.59	11.05	11.07	11.14	7.79	8.16	8.39	10.87	10.94
tc2c10s2ct0	7.48	7.51	7.52	7.65	7.66	7.55	7.55	7.55	7.62	7.62	7.50	7.54	7.55	7.57	7.63
tc2c10s3cf0	8.25	8.15	11.06	11.07	11.17	8.22	8.24	11.14	11.23	11.19	7.97	8.01	8.31	8.34	11.10
tc2c10s3ct0	7.57	7.55	7.60	7.68	7.67	7.58	7.58	7.62	7.59	7.59	7.46	7.47	7.54	7.53	7.52

Table D.27
Stochastic measures for the 10-customer instances and $|S| = 20$. Three probability distributions for the energy consumption are considered.

Instance	Uniform						Normal						Exponential					
	RP	WS	%EVPI	EVP	EEV	%VSS	RP	WS	%EVPI	EVP	EEV	%VSS	RP	WS	%EVPI	EVP	EEV	%VSS
tc0c10s2cf1	12.42	10.48	15.58 %	9.99	inf	inf	11.99	10.60	11.61 %	10.01	inf	inf	11.86	10.32	13.03 %	10.16	inf	inf
tc0c10s2ct1	9.98	9.91	0.66 %	9.83	9.98	0 %	9.91	9.87	0.32 %	9.83	9.91	0 %	9.97	9.92	0.51 %	9.83	9.97	0 %
tc0c10s3cf1	12.30	10.51	14.55 %	10.03	inf	inf	12.13	10.55	13.02 %	9.95	inf	inf	12.08	10.29	14.80 %	10.00	inf	inf
tc0c10s3ct1	10.00	9.91	0.88 %	9.83	10.00	0 %	10.01	9.95	0.59 %	9.84	10.01	0 %	9.96	9.86	1.07 %	9.84	9.96	0 %
tc1c10s2cf2	7.52	7.12	5.31 %	7.09	inf	inf	7.59	7.10	6.38 %	7.08	inf	inf	7.18	7.08	1.40 %	7.07	inf	inf
tc1c10s2cf4	12.21	11.79	3.37 %	11.85	inf	inf	12.18	11.89	2.40 %	11.84	inf	inf	12.08	11.77	2.58 %	11.79	inf	inf
tc1c10s2ct2	7.73	7.27	5.91 %	7.23	inf	inf	7.67	7.27	5.14 %	7.24	inf	inf	7.41	7.28	1.81 %	7.29	inf	inf
tc1c10s2ct3	10.44	9.83	5.79 %	10.45	inf	inf	10.41	9.77	6.12 %	10.41	inf	inf	10.78	9.33	13.42 %	10.43	inf	inf
tc1c10s2ct4	12.01	11.45	4.65 %	11.36	inf	inf	12.05	11.52	4.38 %	11.30	inf	inf	12.08	11.48	4.92 %	11.26	inf	inf
tc1c10s3cf2	7.64	7.11	6.94 %	7.08	inf	inf	7.56	7.10	6.13 %	7.08	inf	inf	7.61	7.09	6.83 %	7.07	inf	inf
tc1c10s3cf4	12.08	11.86	1.87 %	11.83	inf	inf	12.09	11.88	1.68 %	11.87	inf	inf	12.14	11.85	2.35 %	11.82	inf	inf
tc1c10s3ct2	7.69	7.30	5.07 %	7.26	inf	inf	7.77	7.26	6.52 %	7.24	inf	inf	7.81	7.27	6.98 %	7.25	inf	inf
tc1c10s3ct3	9.91	9.27	6.47 %	9.66	10.42	5.10 %	10.05	9.47	5.79 %	9.91	10.06	0.09 %	10.02	9.30	7.17 %	9.88	10.23	2.18 %
tc1c10s3ct4	12.05	11.43	5.14 %	11.36	inf	inf	12.10	11.41	5.67 %	11.37	inf	inf	11.97	11.35	5.22 %	11.33	inf	inf
tc2c10s2cf0	8.09	7.51	7.12 %	8.24	8.09	0.03 %	11.05	7.68	30.47 %	8.20	inf	inf	8.39	7.33	12.60 %	8.22	inf	inf
tc2c10s2ct0	7.52	7.35	2.21 %	7.49	7.52	0 %	7.55	7.41	1.87 %	7.44	7.74	2.53 %	7.55	7.33	2.90 %	7.47	7.61	0.81 %
tc2c10s3cf0	11.06	7.67	30.63 %	8.22	inf	inf	11.14	7.72	30.66 %	8.22	inf	inf	8.31	7.41	10.82 %	8.20	inf	inf
tc2c10s3ct0	7.60	7.43	2.21 %	7.52	7.69	1.16 %	7.62	7.46	2.14 %	7.58	7.62	0 %	7.54	7.34	2.63 %	7.51	7.54	0 %

Table E.28
40 customers - $|S| = \{1, 10\}$.

Instance	$\|S\| = 1$				$\|S\| = 10$			
	Time (s)	Number of iterations	Number of routes	Objective	Time (s)	Number of iterations	Number of routes	Objective
tc0c40s5cf0	661.19	2000	3	22.11	1417.58	2000	3	22.69
tc0c40s5cf4	686.12	2000	2	20.01	1550.06	2000	2	20.03
tc0c40s5ct0	669.10	2000	2	20.35	1484.50	2000	2	21.49
tc0c40s5ct4	692.58	2000	1	19.07	1510.30	2000	2	20.06
tc0c40s8cf0	593.09	2000	3	21.85	1959.07	2000	3	22.30
tc0c40s8cf4	1164.79	2000	3	17.96	8020.82	2000	4	18.56
tc0c40s8ct0	521.03	2000	1	19.51	2858.97	2000	2	19.84
tc0c40s8ct4	567.34	2000	1	18.64	2638.43	2000	1	18.90
tc1c40s5cf1	1261.19	2000	2	21.72	2165.60	2000	3	23.24
tc1c40s5ct1	1230.07	2000	2	22.53	2342.72	2000	2	22.49
tc1c40s8cf1	804.06	2000	1	20.52	3922.73	2000	2	20.94
tc1c40s8ct1	691.05	2000	1	19.41	2796.19	2000	2	19.92
tc2c40s5cf2	853.37	2000	1	15.95	2352.17	2000	1	16.28
tc2c40s5cf3	703.74	2000	2	13.40	2086.98	2000	2	13.56
tc2c40s5ct2	530.02	2000	2	14.52	2413.35	2000	1	15.21
tc2c40s5ct3	497.27	2000	2	13.91	1073.97	2000	2	14.14
tc2c40s8cf2	812.41	2000	1	15.64	3816.96	2000	1	15.53
tc2c40s8cf3	697.58	2000	3	13.32	3043.14	2000	3	13.41
tc2c40s8ct2	514.17	2000	2	14.50	2695.07	2000	1	14.25
tc2c40s8ct3	483.33	2000	3	12.98	1726.36	2000	3	13.04

Table E.29
40 customers - $|S| = \{20, 50\}$.

Instance	$\|S\| = 20$				$\|S\| = 50$			
	Time (s)	Number of iterations	Number of routes	Objective	Time (s)	Number of iterations	Number of routes	Objective
tc0c40s5cf0	2174.05	2000	4	22.68	10800*	1392	4	22.79
tc0c40s5cf4	2216.35	2000	2	20.35	10800*	1761	3	22.61
tc0c40s5ct0	2201.56	2000	3	21.44	10800*	1875	3	21.86
tc0c40s5ct4	2336.83	2000	2	21.00	10800*	1694	2	20.98
tc0c40s8cf0	3048.14	2000	3	22.17	10800*	1355	4	22.61
tc0c40s8cf4	10800*	1380	3	19.03	10800*	199	3	19.06
tc0c40s8ct0	4429.88	2000	2	19.88	10800*	1095	2	20.25
tc0c40s8ct4	3862.97	2000	1	20.06	10800*	1057	2	21.42
tc1c40s5cf1	3340.73	2000	3	23.16	10800*	1291	4	28.70
tc1c40s5ct1	3680.40	2000	3	23.74	10800*	1171	2	24.77
tc1c40s8cf1	6042.46	2000	2	20.94	10800*	771	2	21.04
tc1c40s8ct1	4351.24	2000	2	20.23	10800*	989	1	21.39
tc2c40s5cf2	3022.32	2000	3	17.00	10800*	1521	2	17.25
tc2c40s5cf3	2810.18	2000	2	13.60	10800*	1674	3	14.52
tc2c40s5ct2	4049.67	2000	1	15.62	10800*	1184	1	16.95
tc2c40s5ct3	1569.56	2000	2	14.10	7077.51	2000	2	14.26
tc2c40s8cf2	7030.63	2000	1	15.62	10800*	580	1	15.75
tc2c40s8cf3	4888.71	2000	3	13.43	10800*	962	3	13.41
tc2c40s8ct2	5059.92	2000	1	14.34	10800*	713	1	14.42
tc2c40s8ct3	2529.91	2000	3	13.09	10800*	1758	3	13.12

Table E.30
40 customers - $|S| = 20$, reduced from an initial set of 50 scenarios.

Instance	Time (s)	Number of iterations	Number of routes	Objective
tc0c40s5cf0	2193.93	2000	4	22.38
tc0c40s5cf4	2440.75	2000	2	23.84
tc0c40s5ct0	2321.57	2000	2	21.15
tc0c40s5ct4	2303.74	2000	2	20.96
tc0c40s8cf0	3272.41	2000	3	22.41
tc0c40s8cf4	10800*	1380	3	18.78
tc0c40s8ct0	4656.01	2000	2	20.26
tc0c40s8ct4	3796.65	2000	1	19.63
tc1c40s5cf1	3401.69	2000	4	25.78
tc1c40s5ct1	3532.22	2000	3	23.61
tc1c40s8cf1	6888.19	2000	2	20.66
tc1c40s8ct1	4069.98	2000	2	20.24
tc2c40s5cf2	3206.99	2000	3	16.99
tc2c40s5cf3	2963.66	2000	2	14.49
tc2c40s5ct2	3785.36	2000	1	15.24
tc2c40s5ct3	1459.03	2000	2	14.24
tc2c40s8cf2	7063.48	2000	1	15.71
tc2c40s8cf3	4916.70	2000	3	13.43
tc2c40s8ct2	4899.15	2000	1	14.42
tc2c40s8ct3	2616.29	2000	3	13.11

Table E.31
80 customers - $|S| = \{20, 50\}$.

Instance	$\|S\| = 20$				$\|S\| = 50$			
	Time (s)	Number of iterations	Number of routes	Objective	Time (s)	Number of iterations	Number of routes	Objective
tc0c80s8cf0	10800*	736	5	34.60	10800*	404	4	33.32
tc0c80s8cf1	10800*	749	3	39.05	10800*	294	2	36.29
tc0c80s8ct0	10800*	744	5	34.02	10800*	417	4	38.17
tc0c80s8ct1	10800*	646	4	40.71	10800*	325	4	41.95
tc0c80s12cf0	10800*	159	3	29.37	10800*	90	2	29.50
tc0c80s12cf1	10800*	254	3	32.16	10800*	130	2	32.52
tc0c80s12ct0	10800*	402	3	28.20	10800*	53	2	32.65
tc0c80s12ct1	10800*	233	2	30.73	10800*	110	2	32.33
tc1c80s8cf2	10800*	122	1	22.13	10800*	48	2	26.52
tc1c80s8ct2	10800*	203	2	22.84	10800*	253	1	24.67
tc1c80s12cf2	10800*	182	1	22.29	10800*	69	3	23.16
tc1c80s12ct2	10800*	193	1	20.88	10800*	92	2	23.14
tc2c80s8cf3	10800*	559	3	23.55	10800*	302	4	25.66
tc2c80s8cf4	10800*	172	2	24.59	10800*	57	4	26.07
tc2c80s8ct3	10800*	524	3	23.32	10800*	400	3	25.83
tc2c80s8ct4	10800*	168	4	25.05	10800*	71	3	25.76
tc2c80s12cf3	10800*	145	3	21.22	10800*	110	4	22.83
tc2c80s12cf4	10800*	250	2	21.69	10800*	87	2	21.34
tc2c80s12ct3	10800*	168	3	20.50	10800*	88	2	21.93
tc2c80s12ct4	10800*	251	2	20.17	10800*	87	1	20.12

Table E.32
80 customers - $|S| = 20$, reduced from an initial set of 50 scenarios.

Instance	Time (s)	Number of iterations	Number of routes	Objective
tc0c80s8cf0	10800*	842	3	34.58
tc0c80s8cf1	10800*	637	3	33.01
tc0c80s8ct0	10800*	657	6	36.35
tc0c80s8ct1	10800*	638	2	39.62
tc0c80s12cf0	10800*	178	3	31.00
tc0c80s12cf1	10800*	265	3	31.58
tc0c80s12ct0	10800*	141	3	30.37
tc0c80s12ct1	10800*	281	3	31.39
tc1c80s8cf2	10800*	144	3	25.08
tc1c80s8ct2	10800*	356	3	25.44
tc1c80s12cf2	10800*	159	1	21.67
tc1c80s12ct2	10800*	254	1	20.98
tc2c80s8cf3	10800*	554	4	24.32
tc2c80s8cf4	10800*	158	2	23.72
tc2c80s8ct3	10800*	711	3	25.56
tc2c80s8ct4	10800*	159	2	23.79
tc2c80s12cf3	10800*	396	4	22.01
tc2c80s12cf4	10800*	239	2	21.69
tc2c80s12ct3	10800*	246	2	20.46
tc2c80s12ct4	10800*	202	2	20.60

Table F.33
10 customers - $|S| = 1$.

Instance	Number of routes		Objective		Relative change	
	Arc-level detours	Node-level detours	Arc-level detours	Node-level detours	Number of routes	Objective
tc0c10s2cf1	2	3	9.97	12.22	50.00 %	22.57 %
tc0c10s2ct1	2	3	9.84	12.22	50.00 %	24.26 %
tc0c10s3cf1	2	3	9.97	12.22	50.00 %	22.57 %
tc0c10s3ct1	2	3	9.84	12.22	50.00 %	24.26 %
tc1c10s2cf2	1	2	7.08	8.82	100.00 %	24.62 %
tc1c10s2cf4	2	-	11.85	inf	-	-
tc1c10s2ct2	1	2	7.25	9.09	100.00 %	25.36 %
tc1c10s2ct3	1	-	11.04	inf	-	-
tc1c10s2ct4	1	-	11.36	inf	-	-
tc1c10s3cf2	1	2	7.08	8.82	100.00 %	24.62 %
tc1c10s3cf4	2	-	11.85	inf	-	-
tc1c10s3ct2	1	2	7.25	9.09	100.00 %	25.36 %
tc1c10s3ct3	1	-	9.83	inf	-	-
tc1c10s3ct4	1	-	11.36	inf	-	-
tc2c10s2cf0	2	-	8.21	inf	-	-
tc2c10s2ct0	2	-	7.45	inf	-	-
tc2c10s3cf0	2	-	8.21	inf	-	-
tc2c10s3ct0	2	-	7.53	inf	-	-

Table F.34
10 customers - $|S| = 5$.

Instance	Number of routes		Objective		Relative change	
	Arc-level detours	Node-level detours	Arc-level detours	Node-level detours	Number of routes	Objective
tc0c10s2cf1	2	4	10.33	13.07	100.00 %	26.58 %
tc0c10s2ct1	2	4	10.04	13.07	100.00 %	30.20 %
tc0c10s3cf1	2	3	11.51	12.95	50.00 %	12.53 %
tc0c10s3ct1	2	3	10.02	12.95	50.00 %	29.23 %
tc1c10s2cf2	1	-	7.13	inf	-	-
tc1c10s2cf4	2	-	11.90	inf	-	-
tc1c10s2ct2	1	*	7.36	*	*	*
tc1c10s2ct3	1	-	10.24	inf	-	-
tc1c10s2ct4	2	-	11.95	inf	-	-
tc1c10s3cf2	2	-	7.43	inf	-	-
tc1c10s3cf4	2	-	12.12	inf	-	-
tc1c10s3ct2	2	-	7.55	inf	-	-
tc2c10s2cf0	2	-	7.75	inf	-	-
tc2c10s2ct0	2	-	7.48	inf	-	-
tc2c10s3cf0	2	-	8.25	inf	-	-
tc2c10s3ct0	2	-	7.57	inf	-	-

Table F.35
10 customers - $|S| = 10$.

Instance	Number of routes		Objective		Relative change	
	Arc-level detours	Node-level detours	Arc-level detours	Node-level detours	Number of routes	Objective
tc0c10s2cf1	2	4	11.97	13.07	100.00 %	9.24 %
tc0c10s2ct1	2	4	9.99	13.07	100.00 %	30.92 %
tc0c10s3cf1	2	5	12.27	16.11	150.00 %	31.33 %
tc1c10s2cf2	2	-	7.44	inf	-	-
tc1c10s2ct2	2	*	7.59	*	*	*
tc1c10s3cf2	2	-	7.51	inf	-	-
tc2c10s2cf0	2	-	8.02	inf	-	-
tc2c10s3cf0	2	-	8.15	inf	-	-

Table F.36
40 customers - analysis on $Q^T = \alpha^T Q^{max}$.

Instance	α^T 20 %		30 %		40 %	
	Number of routes	Objective	Number of routes	Objective	Number of routes	Objective
tc0c40s5cf0	4	22.59	4	22.38	4	25.12
tc0c40s5cf4	2	20.11	2	23.84	2	23.27
tc0c40s5ct0	3	20.41	2	21.15	3	23.30
tc0c40s5ct4	3	21.33	2	20.96	2	22.65
tc0c40s8cf0	4	21.94	3	22.41	2	25.14
tc0c40s8cf4	2	17.99	3	18.78	3	19.80
tc0c40s8ct0	2	18.92	2	20.26	2	21.40
tc0c40s8ct4	2	18.91	1	19.63	2	21.48
tc1c40s5cf1	3	27.20	4	25.78	2	26.43
tc1c40s5ct1	3	25.10	3	23.61	3	29.87
tc1c40s8cf1	1	20.88	2	20.66	2	22.43
tc1c40s8ct1	2	22.06	2	20.24	1	23.00
tc2c40s5cf2	3	17.54	3	16.99	1	17.77
tc2c40s5cf3	3	14.64	2	14.49	2	14.81
tc2c40s5ct2	1	15.19	1	15.24	1	16.22
tc2c40s5ct3	2	14.35	2	14.24	3	14.71
tc2c40s8cf2	2	15.94	1	15.71	2	16.46
tc2c40s8cf3	3	13.79	3	13.43	2	13.26
tc2c40s8ct2	1	14.52	1	14.42	1	15.07
tc2c40s8ct3	3	13.30	3	13.11	2	13.13

Table F.37
40 customers - analysis on $Q^G = \alpha^G Q^{max}$.

Instance	α^G 70 %		80 %		90 %	
	Number of routes	Objective	Number of routes	Objective	Number of routes	Objective
tc0c40s5cf0	4	22.55	4	22.38	6	24.76
tc0c40s5cf4	2	21.20	2	23.84	4	23.67
tc0c40s5ct0	2	21.21	2	21.15	4	22.87
tc0c40s5ct4	2	21.02	2	20.96	3	22.42
tc0c40s8cf0	2	22.21	3	22.41	5	24.99
tc0c40s8cf4	2	18.46	3	18.78	3	19.03
tc0c40s8ct0	2	19.64	2	20.26	3	21.46
tc0c40s8ct4	2	20.21	1	19.63	2	22.68
tc1c40s5cf1	3	23.36	4	25.78	-	inf
tc1c40s5ct1	3	23.11	3	23.61	4	27.56
tc1c40s8cf1	2	21.49	2	20.66	2	23.61
tc1c40s8ct1	1	21.01	2	20.24	2	26.08
tc2c40s5cf2	3	16.83	3	16.99	3	19.43
tc2c40s5cf3	2	13.57	2	14.49	3	14.69
tc2c40s5ct2	1	15.64	1	15.24	3	17.83
tc2c40s5ct3	2	14.35	2	14.24	2	14.39
tc2c40s8cf2	2	16.18	1	15.71	4	18.39
tc2c40s8cf3	2	13.28	3	13.43	3	13.53
tc2c40s8ct2	1	14.73	1	14.42	1	16.99
tc2c40s8ct3	1	13.07	3	13.11	3	13.18

References

Afroditi, A., Boile, M., Theofanis, S., Sdoukopoulos, E., Margaritis, D., 2014. Electric vehicle routing problem with industry constraints: trends and insights for future research. Transp. Res. Procedia 3, 452–459.

Almouhanna, A., Quintero-Araujo, C.L., Panadero, J., Juan, A.A., Khosravi, B., Ouelhadj, D., 2020. The location routing problem using electric vehicles with constrained distance. Comput. Oper. Res. 115, 104864.

Andelmin, J., Bartolini, E., 2017. An exact algorithm for the green vehicle routing problem. Transp. Sci. 51, 1288–1303.

Andelmin, J., Bartolini, E., 2019. A multi-start local search heuristic for the green vehicle routing problem based on a multigraph reformulation. Comput. Oper. Res. 109, 43–63.

Asamer, J., Graser, A., Heilmann, B., Ruthmair, M., 2016. Sensitivity analysis for energy demand estimation of electric vehicles. Transp. Res. Part D Transp. Environ. 46, 182–199.

Basso, R., Kulcsár, B., Sanchez-Diaz, I., 2021. Electric vehicle routing problem with machine learning for energy prediction. Transp. Res. Part B Methodol. 145, 24–55.

Basso, R., Kulcsár, B., Sanchez-Diaz, I., Qu, X., 2022. Dynamic stochastic electric vehicle routing with safe reinforcement learning. Transp. Res. Part E Logist. Transp. Rev. 157, 102496.

Bezzi, D., Ceselli, A., Righini, G., 2023. A route-based algorithm for the electric vehicle routing problem with multiple technologies. Transp. Res. Part C Emerg. Technol. 157, 104374.

Bezzi, D., Jabali, O., Maggioni, F., 2024. A threshold recourse policy for the electric vehicle routing problem with stochastic energy consumption. In: Bruglieri, M., Festa, P., Macrina, G., Pisacane, O. (Eds.), Optimization in Green Sustainability and Ecological Transition. Springer Nature, Cham, Switzerland, pp. 219–229. https://doi.org/10.1007/978-3-031-47686-0_20

Birge, J.R., Louveaux, F., 2011. Introduction to Stochastic Programming. Springer Science & Business Media.

Breunig, U., Baldacci, R., Hartl, R.F., Vidal, T., 2019. The electric two-echelon vehicle routing problem. Comput. Oper. Res. 103, 198–210.

Bruglieri, M., Paolucci, M., Pisacane, O., 2023. A matheuristic for the electric vehicle routing problem with time windows and a realistic energy consumption model. Comput. Oper. Res. 157, 106261.

Bruni, M.E., Jabali, O., Khodaparasti, S., 2020. The electric vehicle route planning problem with energy consumption uncertainty. In: 2020 Forum on Integrated and Sustainable Transportation Systems (FISTS). IEEE, pp. 224–229.

Crainic, T.G., Maggioni, F., Perboli, G., Rei, W., 2018. Reduced cost-based variable fixing in two-stage stochastic programming. Ann. Oper. Res. 1–37.

Dastpak, M., Errico, F., Jabali, O., Malucelli, F., 2024. Dynamic routing for the electric vehicle shortest path problem with charging station occupancy information. Transp. Res. Part C Emerg. Technol. 158, 104411.

Desaulniers, G., Errico, F., Irnich, S., Schneider, M., 2016. Exact algorithms for electric vehicle-routing problems with time windows. Oper. Res. 64, 1388–1405.

Dönmez, S., Koç, c., Altıparmak, F., 2022. The mixed fleet vehicle routing problem with partial recharging by multiple chargers: mathematical model and adaptive large neighborhood search. Transp. Res. Part E Logist. Transp. Rev. 167, 102917.

Dupačová, J., Gröwe-Kuska, N., Römisch, W., 2003. Scenario reduction in stochastic programming. Math. Program. 95 (3), 493–511.

Erdoğan, S., Miller-Hooks, E., 2012. A green vehicle routing problem. Transp. Res. Part E Logist. Transp. Rev. 48, 100–114.

Felipe, Á., Ortuño, M.T., Righini, G., Tirado, G., 2014. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transp. Res. Part E Logist. Transp. Rev. 71, 111–128.

Filippi, C., Maggioni, F., Speranza, M.G., 2025. Robust and distributionally robust shortest path problems: a survey. Comput. Oper. Res. 182, 107096. https://doi.org/10.1016/j.cor.2025.107096

Froger, A., Jabali, O., Mendoza, J., Laporte, G., 2022. The electric vehicle routing problem with capacitated charging stations. Transp. Sci. 56, 460–482.

Froger, A., Mendoza, J., Jabali, O., Laporte, G., 2019. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. Comput. Oper. Res. 104, 256–294.

Gioia, D., 2023. ScenarioReducer. https://github.com/DanieleGioia/ScenarioReducer.

Goeke, D., Schneider, M., 2015. Routing a mixed fleet of electric and conventional vehicles. Eur. J. Oper. Res. 245 (1), 81–99.

Golden, B., Raghavan, S., Wasil, E., 2008. The Vehicle Routing Problem: Latest Advances and New Challenges. Vol. 43. Springer.

Han, B., Yan, Y., Chi, T., Park, Y., 2025. The electric vehicle routing problem with travel time and energy consumption uncertainty. Transp. Res. Part E Logist. Transp. Rev. 202, 104211. https://doi.org/10.1016/j.tre.2025.104211

Heitsch, H., Römisch, W., 2003. Scenario reduction algorithms in stochastic programming. Comput. Optim. Appl. 24, 187–206.

IEA, 2023. Global EV Outlook 2023. https://www.iea.org/reports/global-ev-outlook-2023.

Jeong, J., Ghaddar, B., Zufferey, N., Nathwani, J., 2024. Adaptive robust electric vehicle routing under energy consumption uncertainty. Transp. Res. Part C Emerg. Technol. 160, 104529.

Jie, W., Yang, J., Zhang, M., Huang, Y., 2019. The two-echelon capacitated electric vehicle routing problem with battery swapping stations: formulation and efficient methodology. Eur. J. Oper. Res. 272 (3), 879–904. https://doi.org/10.1016/j.ejor.2018.07.002

Juan, A.A., Mendez, C.A., Faulin, J., De Armas, J., Grasman, S.E., 2016. Electric vehicles in logistics and transportation: a survey on emerging environmental, strategic, and operational challenges. Energies 9, 86–106.

Kaut, M., Wallace, S.W., 2007. Evaluation of scenario-generation methods for stochastic programming. Pacific J. Optim. 3 (2), 257–271.

Keskin, M., Çatay, B., Laporte, G., 2021. A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations. Comput. Oper. Res. 125, 105060. https://doi.org/10.1016/j.cor.2020.105060

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

Klein, P.S., Schiffer, M., 2023. Electric vehicle charge scheduling with flexible service operations. Transp. Sci. 57 (6), 1605–1626.

Kucukoglu, I., Dewil, R., Cattrysse, D., 2021. The electric vehicle routing problem and its variations: a literature review. Comput. Indus. Eng. 161, 107650.

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. Comput. Oper. Res. 145, 105870.

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

Liu, X., Wang, D., Yin, Y., Cheng, T.C.E., 2023. Robust optimization for the electric vehicle pickup and delivery problem with time windows and uncertain demands. Comput. Oper. Res. 151, 106119.

Lourenço, H.R., Martin, O.C., Stützle, T., 2003. Iterated Local Search. Springer US, Boston, MA. pp. 320–353.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., 2020. The green-vehicle routing problem: a survey. Model. Optim. Green Logist., 1–26.

Maggioni, F., Allevi, E., Bertocchi, M., 2014. Bounds in multistage linear stochastic programming. J. Optim. Theory Appl. 163, 200–229.

Maggioni, F., Cagnolari, M., Bertazzi, L., 2019. The value of the right distribution in stochastic programming with application to a newsvendor problem. Comput. Manag. Sci. 16 (4), 739–758. https://doi.org/10.1007/s10287-019-00356-2

Maggioni, F., Pflug, G.C., 2019. Guaranteed bounds for general nondiscrete multistage risk-averse stochastic optimization programs. SIAM J. Optim. 29 (1), 454–483. https://doi.org/10.1137/17M1140601

Maggioni, F., Potra, F.A., Bertocchi, M., Allevi, E., 2009. Stochastic second-order cone programming in mobile ad hoc networks. J. Optim. Theory Appl. 143, 309–328.

Maggioni, F., Wallace, S.W., 2012. Analyzing the quality of the expected value solution in stochastic programming. Ann. Oper. Res. 200, 37–54.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Comput. Oper. Res. 24 (11), 1097–1100.

Montanino, M., Natale, I., Fiori, C., Punzo, V., 2025. A traffic-flow-dependent macroscopic model of electric vehicle energy consumption. Transp. Res. Part C Emerg. Technol. 178, 105220. https://doi.org/10.1016/j.trc.2025.105220

Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017. The electric vehicle routing problem with nonlinear charging function. Transp. Res. Part B Methodol. 103, 87–110.

Oyola, J., Arntzen, H., Woodruff, D.L., 2018. The stochastic vehicle routing problem, a literature review, part i: models. EURO J. Transp. Logist. 7 (3), 193–221.

Pelletier, S., Jabali, O., Laporte, G., 2019. The electric vehicle routing problem with energy consumption uncertainty. Transp. Res. Part B Methodol. 126, 225–255.
Pelletier, S., Jabali, O., Laporte, G., Veneroni, M., 2017. Battery degradation and behaviour for electric vehicles: review and numerical analyses of several models. Transp. Res. Part B Methodol. 103, 158–187.
Schiffer, M., Klein, P.S., Laporte, G., Walther, G., 2021. Integrated planning for electric commercial vehicle fleets: a case study for retail mid-haul logistics networks. Eur. J. Oper. Res. 291, 944–960.
Schiffer, M., Stütz, S., Walther, G., 2018. Electric commercial vehicles in mid-haul logistics networks. Behaviour of Lithium-Ion Batteries in Electric Vehicles: Battery Health, Performance, Safety, and Cost, Springer Nature, 153–173.
Schiffer, M., Walther, G., 2018. Strategic planning of electric logistics fleet networks: a robust location-routing approach. Omega 80, 31–42. https://doi.org/10.1016/j.omega.2017.09.003
Schneider, M., Stenger, A., Goeke, D., 2014. The electric vehicle-routing problem with time windows and recharging stations. Transp. Sci. 48, 500–520.
Schulz, A., 2024. Using infeasible path cuts to solve electric vehicle routing problems with realistic charging functions exactly within a branch-and-cut framework. EURO J. Transp. Logist. 13, 100131.
Shapiro, A., 2003. Monte carlo sampling approach to stochastic programming. Esaim Proc. 13, 65–73.
Spinelli, A., Maggioni, F., Ramos, T. R.P., Barbosa-Póvoa, A.P., Vigo, D., 2025. A rolling horizon heuristic approach for a multi-stage stochastic waste collection problem. Eur. J. Oper. Res. 323 (1), 276–296. https://doi.org/10.1016/j.ejor.2024.11.041
Toth, P., Vigo, D., 2014. Vehicle Routing: Problems, Methods, and Applications, Second Edition. Society for Industrial and Applied Mathematics, Philadelphia, PA.
Wang, D., Zhou, H., 2021. The electric vehicle routing problem with time windows under travel time uncertainty. J. Phys. Conf. Ser. 2095 (1), 012032. https://doi.org/10.1088/1742-6596/2095/1/012032
Wang, W., Adulyasak, Y., Cordeau, J.-F., He, G., 2025. The heterogeneous-fleet electric vehicle routing problem with nonlinear charging functions. Transp. Res. Part C Emerg. Technol. 170, 104932. https://doi.org/10.1016/j.trc.2024.104932
Wang, W., Zhao, J., 2023. Partial linear recharging strategy for the electric fleet size and mix vehicle routing problem with time windows and recharging stations. Eur. J. Oper. Res. 308 (2), 929–948.
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. Renew. Sustain. Energy Rev. 151, 111567.
Xiong, H., Xu, Y., Yan, H., Guo, H., Zhang, C., 2024. Optimizing electric vehicle routing under traffic congestion: a comprehensive energy consumption model considering drivetrain losses. Comput. Oper. Res. 168, 106710.
Zhao, J., Zeng, Z., Liu, Y., 2025. Electric vehicle routing problem considering traffic conditions and real-time loads. Transp. Res. Part C Emerg. Technol. 176, 105150. https://doi.org/10.1016/j.trc.2025.105150