

Article

Adaptive Large Neighborhood Search Algorithm for Electric Vehicle Routing Problem with Capacitated Charging Stations and Queueing

Zhuoti Huang, Senlai Zhu * and Yuming Wang

School of Transportation and Civil Engineering, Nantong University, Nantong 226019, China; 2433320050@stmail.ntu.edu.cn (Z.H.); wangyuming@ntu.edu.cn (Y.W.)

* Correspondence: zhusenlai@ntu.edu.cn

Abstract

With growing emphasis on green and low-carbon development and rising urban delivery demand, electric vehicles (EVs) have been increasingly adopted in logistics distribution systems. However, their limited driving range, relatively long charging durations, and the limited capacity of charging stations pose substantial challenges to real-world electric delivery operations. When multiple vehicles arrive at a station with a limited number of chargers, queueing delays may disrupt subsequent customer service and increase total operating costs. To address this issue, this study investigates an electric vehicle routing problem with capacitated charging stations and queueing delays. A mixed-integer linear programming model is formulated, and an enhanced adaptive large neighborhood search (ALNS) algorithm is developed to efficiently solve medium- and large-scale instances. In the proposed model, each vehicle visit to a charging station is represented as a charging event, while finite station capacity is enforced through charging-event assignment and temporal non-overlap constraints. Computational results show that the enhanced ALNS matches the proven optimal solution for the 10-customer instance. For the 15- and 20-customer instances, the best objective values obtained by the enhanced ALNS were 0.39% and 4.84% lower than the corresponding time-limited Gurobi incumbents, respectively. For the 30-, 50-, and 100-customer instances, the enhanced ALNS consistently generates feasible solutions within the prescribed computational budget, whereas Gurobi does not obtain a feasible incumbent within substantially longer time limits. Compared with the baseline ALNS, the enhanced version generally achieves lower mean objective values and more favorable convergence behavior. Sensitivity analyses further show that increasing the number of chargers and improving the charging rate can reduce queueing delays and total charging duration. The proposed approach provides practical decision support for reliable and sustainable urban electric freight operations.

Academic Editor: Giovanni Leonardi

Received: 8 June 2026

Revised: 21 July 2026

Accepted: 23 July 2026

Published: 25 July 2026

Keywords: electric vehicle distribution; capacitated charging stations; queueing delays; mixed-integer linear programming; adaptive large neighborhood search; sustainable logistics

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In recent years, growing emphasis on green and sustainable development has accelerated the adoption of electric vehicles. Owing to their lower local emissions and reduced

noise levels, electric vehicles are increasingly replacing conventional fuel-powered vehicles and have become an important component of sustainable urban freight distribution [1]. Electric delivery vehicles are projected to account for approximately 35% of the urban delivery fleet in Chinese cities by 2035 [2]. Despite these advantages, their large-scale deployment continues to face several practical challenges, including limited driving ranges, inadequate charging infrastructure, and the uneven spatial distribution of charging facilities [3,4]. Continued research on the electric vehicle routing problem (EVRP) is therefore essential for improving both the economic efficiency and environmental sustainability of urban delivery systems [5].

Existing research on the EVRP can be broadly classified into two main streams. The first focuses on the basic problem, extending conventional vehicle routing models by incorporating limited battery capacity and charging-station visit decisions to determine how vehicles replenish energy while serving customers [6,7]. The second introduces more realistic operational features, including customer time windows, nonlinear charging functions, time-dependent travel times, and stochastic waiting times [8–11]. These extensions move the EVRP beyond static and idealized assumptions toward more complex and practically relevant operating environments.

Most existing EVRP studies assume that sufficient charging capacity is always available, allowing vehicles to begin charging immediately upon arrival. This assumption may not hold in practice because the number of available chargers is often limited. When all chargers are occupied, arriving vehicles must wait before charging can begin. A survey of 441 battery electric vehicle users in Japan found that, at publicly accessible charging facilities, approximately 15–30% of respondents experienced average waiting times ranging from 5 to 30 min [12]. In commercial delivery operations, such delays leave both vehicles and drivers idle, increasing operating costs and reducing service reliability [11]. Incorporating charging queues into route planning is therefore necessary to improve both model realism and operational decision-making [8,13]. Moreover, exact optimization methods face considerable computational challenges when solving medium- and large-scale EVRPs [14–17], while directly applying neighborhood search to complete routes that already include charging visits can rapidly expand the search space [18–21].

Motivated by these observations, this study investigates an EVRP with limited charging capacity and queueing delays to support more reliable and sustainable urban logistics operations. Unlike many conventional EVRP formulations, the proposed model explicitly accounts for finite charging-station capacity and represents the assignment, sequencing, and temporal coordination of charging events among multiple vehicles visiting the same station. To address the resulting computational complexity, an enhanced ALNS algorithm is developed. The algorithm adopts a hierarchical solution framework that coordinates customer-route search with charging-insertion and queue-scheduling evaluation.

The main contributions of this study are summarized as follows:

- (a) This study investigates an EVRP with capacitated charging stations and queueing delays for sustainable urban logistics, and represents station capacity limitations through charging-event assignment and temporal non-overlap constraints.
- (b) This study develops a mixed-integer linear programming model that incorporates vehicle-capacity, battery-energy, customer time-window, and charging-station queueing constraints.
- (c) To address the combinatorial complexity of the proposed model, this paper designs an enhanced ALNS algorithm that integrates customer-level route search, charging-station insertion, and queue-scheduling evaluation, thereby improving solution adaptability in complex charging-queue scenarios.

The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 describes the EVRP with capacitated charging stations and queueing

delays and presents the corresponding mixed-integer linear programming model. Section 4 introduces the enhanced ALNS algorithm and its main components, including initial-solution construction, destruction and repair operators, local search, charging-station insertion, queue scheduling, and the acceptance criterion. Section 5 presents the computational experiments and evaluates the proposed model and algorithm. Finally, Section 6 summarizes the main findings, discusses their managerial implications, and outlines directions for future research.

2. Literature Review

2.1. Vehicle Routing Problem

The vehicle routing problem (VRP) is a fundamental problem in distribution and transportation optimization. Its origins can be traced to the truck dispatching problem introduced by Dantzig and Ramser [22]. As real-world distribution systems have become increasingly complex, the classical VRP has evolved into a wide range of variants.

The capacitated vehicle routing problem (CVRP) considers vehicle load-capacity constraints and requires that the total demand assigned to each vehicle does not exceed its carrying capacity [23]. The vehicle routing problem with time windows (VRPTW) further introduces customer service-time requirements, requiring vehicles to serve customers within specified time intervals [24]. The multi-depot vehicle routing problem (MDVRP) considers vehicle scheduling and route coordination across multiple distribution centers [25]. In the open vehicle routing problem (OVRP), vehicles are not required to return to their original depot after completing their delivery routes [26]. The pickup and delivery vehicle routing problem (PDVRP) jointly considers pickup and delivery requests and imposes additional requirements on visit sequences, load evolution, and request matching [27].

The growing uncertainty and complexity of urban distribution systems have also led to the development of stochastic, dynamic, and multi-objective VRPs. The stochastic vehicle routing problem (SVRP) considers uncertain parameters such as customer demand, travel time, and service duration, thereby supporting routing decisions under uncertain operating conditions [28]. The dynamic vehicle routing problem (DVRP) incorporates real-time information, including newly arriving orders and changing traffic conditions, into vehicle-dispatching decisions [29]. The multi-objective vehicle routing problem (MOVRP) simultaneously considers several performance criteria, such as transportation cost, service quality, fleet size, carbon emissions, and workload balance [30].

2.2. Electric Vehicle Routing Problem

Conrad and Figliozzi [6] were among the first to introduce the recharging vehicle routing problem. Erdoğan and Miller-Hooks [7] subsequently formulated the electric vehicle routing problem by representing charging stations as nodes that are distinct from customer locations. Schneider et al. [8] further developed the electric vehicle routing problem with time windows and recharging stations, incorporating battery-capacity constraints and charging-station visits into the routing process.

A substantial body of research has focused on the representation of electric-vehicle charging operations. Montoya et al. [9] introduced nonlinear charging functions and showed that a simple linear relationship may not accurately capture the actual charging process. Wang and Zhao [31] investigated a piecewise-linear partial-recharging strategy and demonstrated that both charging amount and charging duration can influence route selection. Shao et al. [10] considered charging time together with time-varying travel conditions. Kim and Chung [32] modeled energy consumption using state-of-charge-dependent discharge rates, while Xiong et al. [33] developed a more detailed energy-consumption

model under traffic congestion. Beyond short-term energy feasibility, Klein and Schiffer [34] jointly considered battery degradation, nonlinear charging, and time-of-use electricity prices in electric-fleet charging and service scheduling, showing that charging decisions can also affect long-term battery and infrastructure costs.

Other studies have examined charging-infrastructure characteristics and their effects on routing decisions. Fan et al. [35] considered differences in charging performance across stations. Li et al. [36] investigated a sustainable electric vehicle routing problem involving charging-station deployment. Kim [37] incorporated the operating status of charging stations into routing decisions, thereby accounting for dynamic charging availability. More complex settings, including shared charging facilities, mobile charging, and collaborative distribution, have also been studied [38,39]. Huang et al. [40] proposed an en-route navigation and charging-reservation method based on real-time communication between vehicles and charging stations.

At the strategic planning level, Lee and Lee [41] developed a generative-AI framework that integrates geospatial data for charging-station location planning, while Ecer et al. [42] proposed a fuzzy-rough decision-analysis framework incorporating environmental, economic, technological, and social criteria. These studies provide useful support for long-term charging-infrastructure planning. However, they do not jointly optimize vehicle routes, charging-event timing, charger assignment, and queueing interactions at stations with a finite number of chargers.

Despite these advances, many EVRP models still assume that vehicles can begin charging immediately upon arrival at a charging station. Relatively limited attention has been given to queueing delays caused by restricted station capacity. Keskin et al. [11] considered time-dependent and stochastic waiting times at charging stations, but their approach primarily estimated expected waiting times over different periods using a queueing model. Kullman et al. [13] examined charging-station availability and queueing dynamics and showed that waiting conditions can influence charging decisions. Froger et al. [43] studied an electric vehicle routing problem with capacitated charging stations and demonstrated that ignoring charging-station scheduling may result in infeasible routing plans when charger availability is limited. Lam et al. [14] incorporated piecewise-linear charging and capacitated charging stations into an integrated routing and charging-scheduling framework, although the resulting model and solution procedure are computationally demanding.

In contrast to these studies, the present research jointly represents customer routing, charging-station visits, charging-event assignment, charger allocation, and deterministic queue timing. The proposed framework therefore captures the temporal interactions among vehicles competing for a limited number of chargers at the same real station.

2.3. Solution Algorithms for the Electric Vehicle Routing Problem

Existing solution approaches for the EVRP can generally be classified into exact, heuristic, and metaheuristic methods. Exact algorithms are mainly suitable for small-scale instances or relatively simplified problem settings. Lam et al. [14] and Lera-Romero et al. [17] developed branch-and-cut-and-price algorithms for time-dependent electric vehicle routing problems. Desaulniers et al. [15] and Duman et al. [16] applied branch-price-and-cut methods to the electric vehicle routing problem with time windows.

Although exact methods can provide optimality guarantees, their computational requirements increase rapidly with the instance size and the complexity of charging operations. Heuristic and metaheuristic methods are therefore widely used to obtain high-quality solutions within practical computation times. Schneider et al. [8] developed a hybrid variable neighborhood search and tabu search method for the EVRPTW with recharging stations. Rastani et al. [20] proposed a large-neighborhood-search-based heuristic for an

EVRP with load-dependent energy consumption. Souza et al. [44] designed a variable neighborhood search framework that can be adapted to different EVRP variants. Jia et al. [45], Dong et al. [46], and Comert et al. [47] applied ant colony optimization, a memetic algorithm, and a hybrid swarm-intelligence algorithm, respectively.

Adaptive large neighborhood search, originally proposed by Ropke and Pisinger [48], has also been widely applied to electric vehicle routing problems [49]. Keskin and Çatay [18] developed an ALNS algorithm for the EVRPTW with partial-recharging strategies. Keskin et al. [11] considered time-dependent waiting at charging stations and combined ALNS with mathematical programming techniques to improve solution efficiency. They later proposed a two-stage simulation-based heuristic for stochastic charging-station waiting times [19]. Duan et al. [21] developed an ALNS heuristic for the EVRPTW under different charging strategies and charging technologies.

Nevertheless, conventional ALNS frameworks may become less effective when charging-station capacity constraints and inter-vehicle queueing interactions are explicitly considered. Many existing approaches apply destruction, repair, and local-search operators directly to complete routes that contain both customer and charging-station nodes. When charging-station insertion, battery-energy propagation, and customer time windows are jointly considered, even a small route modification may cause cascading changes in downstream battery levels, charging requirements, and arrival times. Moreover, when each station has only a limited number of chargers, charging events belonging to different vehicles become temporally interdependent. This additional coupling further increases the difficulty of solution evaluation and neighborhood exploration.

To address these challenges, the enhanced ALNS proposed in this study adopts a hierarchical solution framework. Customer-route search is performed on the routing structure, while charging-station insertion and queue-scheduling evaluation are handled in subsequent stages. This design reduces the direct coupling between customer-route modifications and charging-event scheduling and facilitates the evaluation of solutions under limited charging capacity.

Table 1 summarizes the main characteristics of the most relevant studies and highlights the differences between the existing literature and the present research.

Table 1. Comparison of representative studies on vehicle routing and charging-station operations.

Reference	Main Factors Considered				Modeling Method	Solution Approach
	E	TW	CS	QW		
Dantzig and Ramser [22]					Classical VRP integer programming model	Exact method
Schneider et al. [8]	✓	✓			MILP	Hybrid VNS/TS
Keskin and Çatay [18]	✓	✓			MILP	ALNS
Montoya et al. [9]	✓				MINLP	Hybrid heuristic
Keskin et al. [11]	✓	✓		✓	MILP	CPLEX + ALNS
Lam et al. [14]	✓	✓	✓		MILP	Branch-and-cut-and-price
Wang et al. [38]	✓	✓			Bi-objective nonlinear programming model	Hybrid GA/TS
Fan et al. [35]	✓				MIP	IACA
Kim [37]	✓				MIP	Exact method
Froger et al. [43]	✓		✓		Continuous-time model	ILS + Branch-and-cut
This study	✓	✓	✓	✓	MILP	Enhanced ALNS

Note: E: battery-energy constraints; TW: customer time windows; CS: explicitly modeled finite charging-station capacity; QW: explicitly represented queue waiting; MIP: mixed-integer programming; MILP: mixed-integer linear programming; MINLP: mixed-integer nonlinear

programming; Hybrid VNS/TS: hybrid variable neighborhood search and tabu search; ALNS: adaptive large neighborhood search; CPLEX: CPLEX optimizer; IACA: improved ant colony algorithm; ILS: iterated local search; Branch-and-cut-and-price: branch-and-cut-and-price algorithm; Branch-price-and-cut: branch-price-and-cut algorithm.

3. Problem Description and Model Formulation

3.1. Problem Description

Figure 1 illustrates the problem investigated in this study. Specifically, the distribution system consists of a depot, a set of customer nodes, and a set of charging stations. A fleet of homogeneous electric vehicles departs from the depot, serves all customers, and then returns to the depot. Owing to the limited driving range of electric vehicles, vehicles need to select appropriate charging stations for energy replenishment according to their battery state during delivery operations. Meanwhile, since each charging station has a limited number of chargers, charging demands from different vehicles at the same station may interact over time, giving rise to queueing delays and subsequently changing the remaining route schedule.

Each customer is associated with a given demand, a service time, and a service time window. Soft time windows are adopted. Specifically, if a vehicle arrives before the earliest allowable service time, it may wait at the customer location until the time window opens; if the service start time exceeds the latest allowable service time, tardiness is incurred. Vehicle travel on the road consumes electricity, and the battery capacity is limited. Moreover, the state of charge must not fall below a minimum safety threshold during operation. Therefore, route decisions must satisfy not only time-window and load-capacity constraints, but also energy reachability and battery-safety requirements along the route.

The objective of this study is to minimize the total system cost while satisfying route-connectivity, customer-service, load-capacity, battery-safety, and time-window constraints, with each customer served exactly once. The total cost consists of travel distance cost, fixed vehicle-use cost, charging time cost, queueing-delay cost at charging stations, and customer tardiness penalty cost.

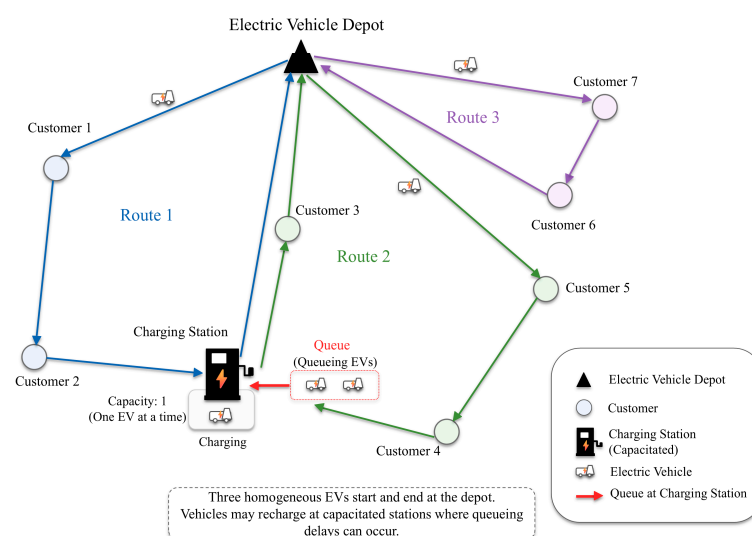


Figure 1. Schematic illustration of the EVRP with capacitated charging stations and queueing.

3.2. Model Formulation

3.2.1. Basic Assumptions and Notation

To establish the mathematical model, the following basic assumptions are adopted:

- (1) Each EV departs from the depot with a fully charged battery and returns to the depot after completing its delivery tasks.
- (2) The energy consumption of an EV is described by arc-specific parameters. The effects of vehicle load, road gradient, speed, congestion, and regenerative braking on battery consumption are not explicitly considered in this study.
- (3) The charging duration at a charging station is linearly proportional to the amount of energy replenished.
- (4) A full-charging policy is adopted at charging stations. After an EV visits a virtual charging-station node, its departure battery level is restored to the battery capacity. Accordingly, the replenished energy amount is determined by the difference between the battery capacity and the battery level upon arrival, rather than being modeled as an independent charging decision.

The problem is defined on a directed network $G = (V, A)$. The node set $V = \{0\} \cup N \cup S$ consists of the depot, customer nodes, and charging nodes. Accordingly, the depot node is denoted by 0, the customer set by N , the set of virtual charging-station nodes by S , the set of real charging stations by P , and the vehicle set by K . The arc set $A \subseteq V \times V$ represents feasible pairs of nodes that vehicles can travel between, where $i, j \in V$ are node indices and $k \in K$ is a vehicle index.

Virtual charging-station nodes are introduced to represent multiple possible charging visits associated with the same real charging station. Each virtual charging-station node corresponds to one possible charging visit. When vehicle k selects virtual charging-station node s , the pair (k, s) forms a candidate charging event. All virtual charging-station nodes mapped to the same real charging station share the same limited charger resources. For any virtual charging-station node $s \in S$, $\phi(s)$ denotes the real charging station to which it belongs, and S_p denotes the set of virtual charging-station nodes associated with real charging station p . The charging service capacity of a station is determined by the number of chargers c_p , where $p \in P$; the charging rate is denoted by ρ_p , which determines the charging duration at the station. Based on this mapping, when a vehicle selects a virtual charging-station node for charging in its route, the corresponding queueing and capacity constraints are imposed at the associated real charging station.

For each customer node $i \in N$, the service time window is denoted by $[a_i, b_i]$. The time window of a customer node constrains the service start time, whereas charging stations are not subject to additional time-window restrictions. For each node $i \in N \cup S$, the service duration and demand are denoted by σ_i and q_i , respectively, where $\sigma_i = q_i = 0$ for charging stations. For any arc $(i, j) \in A$, the associated distance, travel time, and energy consumption are denoted by d_{ij} , τ_{ij} , and e_{ij} , respectively.

A valid choice of the Big-M constants is $M^T = T^{\max} + \max_i \sigma_i + B / \min_{p \in P} \rho_p + \max_{(i,j) \in A} \tau_{ij}$, $M^Q = Q + \max_{i \in N} q_i$, and $M^E = B + \max_{(i,j) \in A} e_{ij}$. These values are sufficiently large to relax the corresponding time, load, and battery propagation constraints when the associated routing arc is not selected.

Vehicle operations are subject to both load and battery constraints. The vehicle load capacity is Q , the battery capacity is B , and the minimum safety battery level is $E^{\min}$. $T^{\max}$ is a pre-specified end of the planning horizon, and a large enough value should be chosen to cover all feasible vehicle operations. M^T , M^Q , and M^E are Big-M constants for time-related constraints, load propagation constraints, and battery propagation constraints, respectively.

In addition, $\lambda_1 \sim \lambda_5$ represent the unit costs associated with travel distance, vehicle use, charging time, queue waiting time, and tardiness penalty, respectively, and are used to construct the integrated objective function.

In the model, binary decision variables are defined as follows. The routing decision is represented by x_{ij}^k ; if vehicle k travels directly from node i to node j , then $x_{ij}^k = 1$, and 0 otherwise. The variable δ_k indicates whether a vehicle is used, where $\delta_k = 1$ means that vehicle k is used. This variable links the fixed vehicle cost with the route structure. The visit indicator F_i^k equals 1 if vehicle k visits node i . For a virtual charging-station node s , z_s^k equals 1 if vehicle k visits s and thus generates a charging event.

To describe arrival, waiting, and service processes simultaneously, A_i^k denotes the arrival time of vehicle k at node i . T_i^k denotes the service start time of vehicle k at node i . When node i is a customer, it represents the time at which customer service starts. Correspondingly, when i is a charging station, it denotes the charging start time. To avoid route-cycle issues in the time recursion at the depot, D_k is separately defined as the return time of vehicle k to the depot. The continuous variable Q_i^k represents the remaining load of vehicle k after leaving node i , and is used to model load-capacity constraints and demand consumption.

The variables $E_i^{a,k}$ and $E_i^{d,k}$ denote the battery levels of vehicle k upon arriving at and departing from node i , respectively. For a virtual charging-station node $s \in S$, the following visit and charging-process variables are introduced: z_s^k indicates whether vehicle k visits s ; H_s^k denotes the charging duration at s ; and W_s^k denotes the queue waiting time caused by charger occupancy. The tardiness variable L_i^k represents the tardiness of vehicle k at customer $i \in N$.

For the h -th charger ($h = 1, \dots, c_p$) at any real charging station $p \in P$, the assignment variable $\mathbb{A}_{p,h,s}^k$ is defined to indicate whether the charging event (k,s) of vehicle k at virtual charging-station node $s \in S$ is assigned to this charger h . To prevent overlap between two charging events on the same charger, the sequencing variable $y_{p,h,s_1,s_2}^{k_1,k_2} \in \{0,1\}$ is introduced. $y_{p,h,s_1,s_2}^{k_1,k_2} = 1$ indicates that event (k_1,s_1) precedes event (k_2,s_2) , whereas $y_{p,h,s_1,s_2}^{k_1,k_2} = 0$ indicates that event (k_2,s_2) precedes event (k_1,s_1) .

Based on the above definitions, the parameters and decision variables are summarized in Table 2.

Table 2. Summary of notation.

Parameters	Description
$V = \{0\} \cup N \cup S$	Set of all nodes
N	Set of customer nodes
$A = \{(i,j) \mid i,j \in V, i \neq j\}$	Set of feasible directed arcs
S	Set of virtual charging-station nodes
P	Set of real charging stations
K	Set of electric vehicles
$\phi(s)$	Real charging station associated with virtual charging-station node s
$S_p = \{s \in S \mid \phi(s) = p\}$	Set of virtual charging-station nodes associated with real charging station
$\mathcal{E}_p = \{(k,s) \mid k \in K, s \in S_p\}$	Set of candidate charging events associated with real charging station p
c_p	Number of parallel chargers at a real charging station p
ρ_p	Charging rate of real charging station p
$[a_i, b_i]$	Service time window of customer node $i \in N$
σ_i	Service duration of node $i \in N \cup S$, $\sigma_i = 0$ for charging stations
q_i	Demand of node $i \in N \cup S$, $q_i = 0$ for charging stations
d_{ij}	Distance of arc (i,j)
τ_{ij}	Travel time of arc (i,j)
e_{ij}	Energy consumption of arc (i,j)
Q	Vehicle load capacity
B	Battery capacity
E^{min}	Minimum battery level

T^{max}	End of the planning horizon, used as an upper bound on time-related variables
M^T	Big-M constant for time-related constraints
M^Q	Big-M constant for load propagation constraints
M^E	Big-M constant for battery propagation constraints
λ_1	Travel-distance cost
λ_2	Vehicle usage cost
λ_3	Charging-time cost
λ_4	Queueing-delay cost
λ_5	Tardiness penalty cost
Decision variables	Description
$x_{ij}^k \in \{0,1\}, \forall (i,j) \in A, k \in K$	Equals 1 if vehicle k travels from node i to node j ; otherwise 0
$\delta_k \in \{0,1\}, \forall k \in K$	Equals 1 if the vehicle k is used; otherwise 0
$F_i^k \in \{0,1\}$	Equals 1 if vehicle k visits node i ; otherwise 0
$z_s^k \in \{0,1\}$	Equals 1 if vehicle k visits virtual charging-station node s ; otherwise 0
$A_i^k \geq 0$	Arrival time of vehicle k at node i
$T_i^k \geq 0$	Service/Charging start time of vehicle k at node i
$D_k \geq 0$	Return time of vehicle k to the depot
$Q_i^k \in [0, Q]$	The remaining load of vehicle k upon departure from node i
$E_i^{a,k} \in [0, B]$	Battery level of vehicle k upon arrival at node i
$E_i^{d,k} \in [0, B]$	Battery level of vehicle k upon departure from node i
$H_s^k \geq 0$	Charging duration of vehicle k at virtual charging-station node s
$W_s^k \geq 0$	Queue waiting time of vehicle k at virtual charging-station node s
$L_i^k \geq 0, \forall i \in N$	Tardiness of vehicle k at customer i
$\mathbb{A}_{p,h,s}^k \in \{0,1\}$	Equals 1 if the charging event of vehicle k at virtual charging-station node s is assigned to charger h of real station p
$y_{p,h,s_1,s_2}^{k_1,k_2} \in \{0,1\}$	Equals 1 if charging event (k_1, s_1) precedes charging event (k_2, s_2) on charger h of real station p ; otherwise 0

3.2.2. Charging Station Capacity Constraints

To model the service capability of a real charging station with a limited number of parallel chargers, each actual charging visit is treated as a charging event. Figure 2 illustrates the assignment of charging events at the same real charging station with two chargers. In the figure, actual charging events (k_1, s_1) and (k_2, s_2) are assigned to the first charger of real charging station p , while actual charging event (k_3, s_3) is assigned to the second charger. Since different chargers can provide service in parallel, event (k_3, s_3) may overlap in time with event (k_1, s_1) or event (k_2, s_2) . However, for event (k_1, s_1) and event (k_2, s_2) , their service intervals cannot overlap because they occupy the same charger.

According to the notation, for a charging event of vehicle k at virtual charging-station node s , T_s^k denotes the charging start time of the event and H_s^k denotes its charging duration. Therefore, the occupied time interval of event (k, s) after being assigned to a charger can be expressed as $[T_s^k, T_s^k + H_s^k]$. Accordingly, for any two charging events (k_1, s_1) and (k_2, s_2) at the same real charging station p , if both are assigned to the same charger h , their occupied intervals must be non-overlapping.

To specify the charger assignment and sequencing domains, we use the candidate charging-event set $\mathcal{E}_p$ defined in Table 2. A candidate charging event $(k, s) \in \mathcal{E}_p$ becomes an actual charging event only when vehicle k visits virtual charging-station node s . The assignment variable is defined as

$$\mathbb{A}_{p,h,s}^k \in \{0,1\}, \forall p \in P, h = 1, \dots, c_p, (k, s) \in \mathcal{E}_p \quad (1)$$

Here, $\mathbb{A}_{p,h,s}^k = 1$ indicates that the candidate charging event (k, s) is assigned to charger h at real charging station p . This domain ensures that a candidate charging event can only be assigned to a charger of its associated real charging station.

Sequencing is considered only between two distinct candidate charging events associated with the same real charging station p and the same charger h . An arbitrary but fixed order “<” is imposed on the elements of $\mathcal{E}_p$, so that the sequencing variable $y_{p,h,s_1,s_2}^{k_1,k_2}$ is defined only for event pairs satisfying $(k_1, s_1) < (k_2, s_2)$:

$$y_{p,h,s_1,s_2}^{k_1,k_2} \in \{0,1\}, \forall p \in P, h = 1, \dots, c_p, (k_1, s_1), (k_2, s_2) \in \mathcal{E}_p, (k_1, s_1) < (k_2, s_2) \quad (2)$$

This definition avoids duplicate sequencing variables for reversed event pairs and excludes self-comparisons of the same candidate charging event.

For two candidate charging events satisfying the above sequencing domain, only one of the following two precedence relations can hold:

$$T_{s_1}^{k_1} + H_{s_1}^{k_1} \leq T_{s_2}^{k_2} \quad (3)$$

$$T_{s_2}^{k_2} + H_{s_2}^{k_2} \leq T_{s_1}^{k_1} \quad (4)$$

Based on the above assignment and sequencing domains, the non-overlap relation can be reformulated in a mixed-integer linear form using the Big-M method as follows:

$$T_{s_1}^{k_1} + H_{s_1}^{k_1} \leq T_{s_2}^{k_2} + M^T (3 - \mathbb{A}_{p,h,s_1}^{k_1} - \mathbb{A}_{p,h,s_2}^{k_2} - y_{p,h,s_1,s_2}^{k_1,k_2}) \quad (5)$$

$$T_{s_2}^{k_2} + H_{s_2}^{k_2} \leq T_{s_1}^{k_1} + M^T (2 - \mathbb{A}_{p,h,s_1}^{k_1} - \mathbb{A}_{p,h,s_2}^{k_2} + y_{p,h,s_1,s_2}^{k_1,k_2}) \quad (6)$$

where M^T is the time-related Big-M constant defined in Table 2. Constraints (5) and (6) are imposed for all $p \in P, h = 1, \dots, c_p, (k_1, s_1), (k_2, s_2) \in \mathcal{E}_p$, and $(k_1, s_1) < (k_2, s_2)$. When $\mathbb{A}_{p,h,s_1}^{k_1} = \mathbb{A}_{p,h,s_2}^{k_2} = 1$, the two events are assigned to the same charger h at the same real charging station p , and the constraints become active. If $y_{p,h,s_1,s_2}^{k_1,k_2} = 1$, the first inequality is tightened to $T_{s_1}^{k_1} + H_{s_1}^{k_1} \leq T_{s_2}^{k_2}$, indicating that event (k_1, s_1) must finish before event (k_2, s_2) starts. If $y_{p,h,s_1,s_2}^{k_1,k_2} = 0$, the second inequality is tightened to $T_{s_2}^{k_2} + H_{s_2}^{k_2} \leq T_{s_1}^{k_1}$, indicating that event (k_2, s_2) must finish before event (k_1, s_1) starts. Conversely, if the two events are not simultaneously assigned to the same charger, the above Big-M terms relax the constraints.

The queueing mechanism in this formulation represents a deterministic schedule-based charging operation. The charging start time is endogenously determined by the route arrival time, charger assignment, and non-overlap constraints. Therefore, the waiting time W_s^k captures the delay caused by charger unavailability at the planned arrival time. This setting is suitable for reservation-based or centrally coordinated logistics operations, where vehicle routes and charging schedules can be planned in advance. More complex operational settings, such as stochastic first-come-first-served queues, charger-type heterogeneity, and uncertain vehicle arrivals, are not explicitly considered in the present formulation and can be incorporated in future extensions.

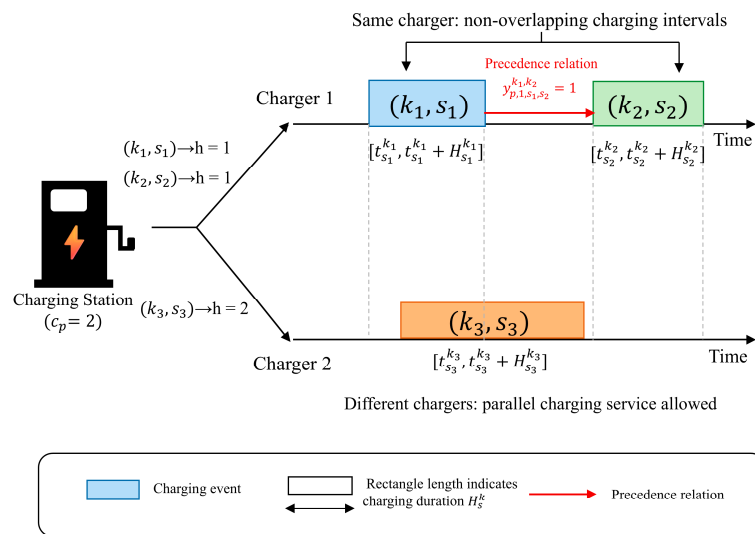


Figure 2. Schematic illustration of the charging station capacity constraints.

3.2.3. Mathematical Model

The objective function (7) minimizes the total cost, which consists of five components. The first term represents the total travel distance cost of vehicles along their routes, the second term represents the total fixed cost of used vehicles, the third term corresponds to the total charging time cost at charging stations, the fourth term is the total queueing-delay time cost at charging stations, and the final term is the total customer tardiness penalty cost.

$$\min \lambda_1 \sum_{k \in K} \sum_{(i,j) \in A} d_{ij} x_{ij}^k + \lambda_2 \sum_{k \in K} \delta_k + \lambda_3 \sum_{k \in K} \sum_{s \in S} H_s^k + \lambda_4 \sum_{k \in K} \sum_{s \in S} W_s^k + \lambda_5 \sum_{k \in K} \sum_{i \in N} L_i^k \quad (7)$$

Subject to

$$\sum_{j:(0,j) \in A} x_{0j}^k = \delta_k, \quad \sum_{i:(i,0) \in A} x_{i0}^k = \delta_k, \quad \forall k \in K \quad (8)$$

$$\sum_{j:(j,i) \in A} x_{ji}^k = \sum_{j:(i,j) \in A} x_{ij}^k, \quad \forall i \in N \cup S, k \in K \quad (9)$$

$$\sum_{k \in K} \sum_{j:(j,i) \in A} x_{ji}^k = 1, \quad \forall i \in N \quad (10)$$

$$F_i^k = \sum_{j:(j,i) \in A} x_{ji}^k, \quad \forall i \in N \cup S, k \in K \quad (11)$$

$$z_s^k = F_s^k, \quad \forall s \in S, k \in K \quad (12)$$

$$\sum_{s \in S_p} z_s^k \leq 1, \quad \forall p \in P, k \in K \quad (13)$$

$$A_i^k \leq T^{\max} F_i^k, \quad T_i^k \leq T^{\max} F_i^k, \quad \forall i \in N \cup S, k \in K \quad (14)$$

$$T_i^k \geq A_i^k, \quad \forall i \in N \cup S, k \in K \quad (15)$$

$$T_i^k \geq a_i - M^T (1 - F_i^k), \quad \forall i \in N, k \in K \quad (16)$$

$$L_i^k \geq T_i^k - b_i - M^T(1 - F_i^k), \forall i \in N, k \in K \quad (17)$$

$$L_i^k \leq M^T F_i^k, \forall i \in N, k \in K \quad (18)$$

$$A_j^k \geq \tau_{0j} - M^T(1 - x_{0j}^k), \forall (0, j) \in A, j \in N \cup S, k \in K \quad (19)$$

$$A_j^k \geq T_i^k + \sigma_i + \tau_{ij} - M^T(1 - x_{ij}^k), \forall (i, j) \in A, i \in N, j \in N \cup S, k \in K \quad (20)$$

$$A_j^k \geq T_s^k + H_s^k + \tau_{sj} - M^T(1 - x_{sj}^k), \forall (s, j) \in A, s \in S, j \in N \cup S, k \in K \quad (21)$$

$$D_k \geq T_i^k + \sigma_i + \tau_{i0} - M^T(1 - x_{i0}^k), \forall (i, 0) \in A, i \in N, k \in K \quad (22)$$

$$D_k \geq T_s^k + H_s^k + \tau_{s0} - M^T(1 - x_{s0}^k), \forall (s, 0) \in A, s \in S, k \in K \quad (23)$$

$$D_k \leq T^{\max} \delta_k, \forall k \in K \quad (24)$$

$$W_s^k = T_s^k - A_s^k, \forall s \in S, k \in K \quad (25)$$

$$Q_0^k = Q \delta_k, \forall k \in K \quad (26)$$

$$Q_i^k \leq Q F_i^k, \forall i \in N \cup S, k \in K \quad (27)$$

$$Q_j^k \geq Q_i^k - q_j - M^Q(1 - x_{ij}^k), \forall (i, j) \in A, j \in N \cup S, k \in K \quad (28)$$

$$Q_j^k \leq Q_i^k - q_j + M^Q(1 - x_{ij}^k), \forall (i, j) \in A, j \in N \cup S, k \in K \quad (29)$$

$$E_0^{a,k} = B \delta_k, E_0^{d,k} = B \delta_k, \forall k \in K \quad (30)$$

$$E_j^{a,k} \geq E_i^{d,k} - e_{ij} - M^E(1 - x_{ij}^k), \forall (i, j) \in A, j \in N \cup S, k \in K \quad (31)$$

$$E_j^{a,k} \leq E_i^{d,k} - e_{ij} + M^E(1 - x_{ij}^k), \forall (i, j) \in A, j \in N \cup S, k \in K \quad (32)$$

$$E^{\min} F_i^k \leq E_i^{a,k} \leq B F_i^k, E^{\min} F_i^k \leq E_i^{d,k} \leq B F_i^k, \forall i \in N \cup S, k \in K \quad (33)$$

$$E_i^{d,k} - e_{i0} \geq E^{\min} - M^E(1 - x_{i0}^k), \forall (i, 0) \in A, i \in N \cup S, k \in K \quad (34)$$

$$E_i^{d,k} = E_i^{a,k}, \forall i \in N, k \in K \quad (35)$$

$$E_s^{d,k} = B z_s^k, \forall s \in S, k \in K \quad (36)$$

$$\rho_p H_s^k = B z_s^k - E_s^{a,k}, \forall p \in P, s \in S_p, k \in K \quad (37)$$

$$\sum_{h=1}^{c_p} \mathbb{A}_{p,h,s}^k = z_s^k, \forall p \in P, s \in S_p, k \in K \quad (38)$$

$$T_{s_1}^{k_1} + H_{s_1}^{k_1} \leq T_{s_2}^{k_2} + M^T(3 - \mathbb{A}_{p,h,s_1}^{k_1} - \mathbb{A}_{p,h,s_2}^{k_2} - y_{p,h,s_1,s_2}^{k_1,k_2}) \quad (39)$$

$$T_{s_2}^{k_2} + H_{s_2}^{k_2} \leq T_{s_1}^{k_1} + M^T(2 - \mathbb{A}_{p,h,s_1}^{k_1} - \mathbb{A}_{p,h,s_2}^{k_2} + y_{p,h,s_1,s_2}^{k_1,k_2}) \quad (40)$$

Constraints (39) and (40) are imposed over the sequencing domain defined in Section 3.2.1, namely $p \in P$, $h = 1, \dots, c_p$, $(k_1, s_1), (k_2, s_2) \in \mathcal{E}_p$, and $(k_1, s_1) < (k_2, s_2)$.

$$x_{ij}^k, \delta_k, F_i^k, z_s^k, A_{p,h,s}^k, y_{p,h,s_1,s_2}^{k_1,k_2} \in \{0,1\} \quad (41)$$

$$A_i^k, T_i^k, D_k, Q_i^k, E_i^{a,k}, E_i^{d,k}, H_s^k, W_s^k, L_i^k \geq 0 \quad (42)$$

Constraint (8) ensures that each used vehicle departs from the depot and eventually returns to the depot. Constraint (9) enforces flow conservation at non-depot nodes. Constraint (10) guarantees that each customer is served exactly once, thereby avoiding duplicate service or omission. Constraint (11) defines the node-visit indicator from the routing variables, while Constraint (12) links the charging-visit variable to the visit indicator of virtual charging-station node s . Together, these constraints ensure that a charging visit is counted only when vehicle k actually visits the corresponding virtual charging-station node in its route. Constraint (13) states that each vehicle is allowed to visit at most one virtual charging-station node associated with the same real charging station during a single route. This restriction is consistent with the full-charging policy adopted in this study. Once a vehicle visits a charging station, its battery is restored to capacity; therefore, repeated visits to the same real charging station within the same route are regarded as redundant. Meanwhile, the virtual-station representation still allows multiple vehicles and multiple candidate charging events to be associated with the same real charging station, and these charging events compete for the limited chargers through the capacity and sequencing constraints. Constraints (14)–(18) define the time-variable bounds, waiting behavior, and soft time-window requirements. Specifically, service or charging cannot start before arrival; early arrivals may wait until the lower bound of the time window, and tardiness is incurred when customer service starts after the upper time-window bound. Constraints (19)–(24) describe time propagation along selected arcs and define the return time to the depot. Together with the depot departure/return constraints and the flow conservation constraints, these time propagation constraints implicitly eliminate disconnected subtours. For any selected cycle, the arrival or service start time would have to increase along the cycle and eventually return to its starting value, which is impossible under positive travel times and non-negative service/charging durations. Constraint (25) calculates the queue waiting time at charging stations. Constraints (26)–(29) formulate the initial load and load propagation using a valid Big-M linearization. Constraints (30)–(34) define the initial battery level, battery propagation, and battery-safety requirements. Constraint (35) ensures that no energy is replenished at customer nodes. Constraints (36) and (37) enforce the full-charging policy at virtual charging-station nodes: if vehicle k visits station s , its departure battery level is restored to B , and the corresponding charging duration is determined by the required energy replenishment. Constraint (38) assigns each actual charging event to exactly one charger at its associated real charging station. Constraints (39) and (40) impose non-overlapping occupied charging intervals for two distinct candidate charging events when they are assigned to the same charger, with the sequencing variable determining their service order. Constraints (41) and (42) define the domains of binary and continuous decision variables.

4. Enhanced Adaptive Large Neighborhood Search Algorithm

The problem studied in this paper simultaneously involves route optimization, time-window constraints, battery-state evolution, charging-station insertion, and queueing delays under a limited number of chargers. If neighborhood search is directly performed on complete routes containing both customer nodes and charging nodes, the search space becomes excessively large. Moreover, each local adjustment has to update charging operations and queueing schedules simultaneously, which leads to high computational

overhead. Therefore, the route representation is decomposed into two layers, and a two-layer solution framework consisting of skeleton-route search and evaluation-stage expansion is adopted. Specifically, vehicle assignment and customer visiting sequences are first searched only at the customer-route level. Then, during the evaluation of a candidate solution, charging-station insertion and queue scheduling are carried out to compute the actual objective value, which is further used to guide solution acceptance and update.

The proposed algorithm follows an adaptive large neighborhood search (ALNS) framework. In each iteration, the destroy operator is selected according to adaptive probabilities derived from the current operator weights. The weights are periodically updated according to the historical contribution of each operator, including whether it generates a new best solution, improves the current solution, produces an accepted solution, or is rejected. In this way, the algorithm can strengthen effective operators while retaining a minimum selection probability for diversification.

The first layer is the skeleton route, which contains only the depot and customer nodes. The skeleton route of vehicle k is denoted as

$$R_k = [0, i_1, i_2, \dots, i_{m_k}, 0], i_\ell \in N, \ell = 1, \dots, m_k \quad (43)$$

where 0 denotes the distribution depot, and $i_1, \dots, i_{m_k}$ denote the sequence of customers served by vehicle k . During the destroy, repair, local search, and cross-route perturbation procedures of ALNS, the direct search object is the skeleton route.

The second layer is the extended route, namely the complete travel route $\tilde{R}_k$ obtained by inserting virtual charging-station nodes into the skeleton route according to battery feasibility. The extended route does not directly participate in destroy and repair operations. Instead, it is dynamically generated by the charging-insertion module during candidate-solution evaluation. Subsequently, all charging events on the extended routes are mapped to their corresponding real charging stations, and a global queue scheduling procedure is conducted under the capacity limitation of each real station. In this way, the actual charging start time, queue waiting time, and the arrival times at downstream nodes can be determined.

$$\tilde{R}_k = [0, i_1, i_2, s_\ell, \dots, i_{m_k}, 0], i_\ell \in N, s_\ell \in S \quad (44)$$

To facilitate the understanding of the overall procedure in Algorithm 1, the main notation used in the algorithm is summarized as follows. Let $\mathcal{B}$ denote the problem-instance data, including the customer set, vehicle parameters, information on virtual and real charging stations, distance and travel-time parameters, and the relevant cost coefficients. Let $w_{\max}$ be the maximum number of iterations. T_0 is the initial temperature of the simulated annealing acceptance criterion, and α is the cooling coefficient. x^0 denotes the initial skeleton solution generated by the initial-solution construction method. G is the stagnation time threshold, g denotes the elapsed time since the last improvement of the historical best solution, and $\mathcal{T}^{imp}$ records the last-improvement time. In the implementation, g is updated according to the difference between the current time and the last-improvement time $\mathcal{T}^{imp}$. When $g \geq G$, the restart mechanism is triggered. $x^{current}$ denotes the current solution, namely the skeleton solution being searched and updated during the iteration. x^{best} denotes the historical best solution, that is, the solution with the minimum comprehensive objective value among all generated solutions since the start of the algorithm. x^{part} denotes the partial solution obtained after applying a destroy operator. x^{new} is the candidate skeleton solution generated from the current solution after destroy, repair, cross-route perturbation, and local search. T denotes the current temperature. T_{end} is the lower bound of the temperature. w is the current iteration counter, and Ω is the destroy-operator set. The function $f(x)$ denotes the

objective value of solution x under the unified evaluation framework, which comprehensively accounts for travel distance, vehicle activation, charging duration, queueing delay, and tardiness penalty costs. $\omega_{\odot}$ is the adaptive weight of operator $\odot$, $\pi_{\odot}$ is the corresponding selection probability, and $\mathbb{p}$ is the reaction factor used in weight updating.

The framework of the enhanced ALNS is shown in Algorithm 1.

Algorithm 1. Framework of the proposed ALNS

Input: Instance $\mathcal{I}$, maximum number of iterations $w_{\max}$, time limit $\mathcal{T}_{\max}$, initial temperature T_0 , terminal temperature T_{end} , cooling factor α , destroy-operator set Ω , weight-update interval $\mathbb{W}$, reaction factor $\mathbb{p}$, stagnation time threshold $\mathbb{G}$, and number of perturbation moves $\mathcal{S}$

Output: Best-so-far solution x^{best}

- 1: Construct the initial skeleton solution x^0 using Algorithm 2
 - 2: Perform an initial 2-opt improvement on x^0
 - 3: Apply Algorithms 3 and 4 to x^0 , and compute $f(x^0)$
 - 4: $x^{\text{current}} \leftarrow x^0$, $x^{\text{best}} \leftarrow x^0$
 - 5: $f(x^{\text{current}}) \leftarrow f(x^0)$, $f(x^{\text{best}}) \leftarrow f(x^0)$
 - 6: Initialize $T \leftarrow T_0$, $w \leftarrow 0$, $g \leftarrow 0$, and record the last-improvement time $\mathcal{T}^{\text{imp}}$
 - 7: Initialize the weight $\omega_{\odot}$, accumulated score $\theta_{\odot}$, and usage count $w_{\odot}$ for each destroy operator $\odot \in \Omega$
 - 8: **while** $w < w_{\max}$ and elapsed time $< \mathcal{T}_{\max}$ **do**
 - 9: Update the elapsed stagnation time g as the elapsed time since $\mathcal{T}^{\text{imp}}$
 - 10: **if** $g \geq \mathbb{G}$ **then**
 - 11: Restart from x^{best} , apply $\mathcal{S}$ cross-route perturbations, and mildly reheat the temperature
 - 12: $x^{\text{current}} \leftarrow x^{\text{best}}$, $g \leftarrow 0$, and update $\mathcal{T}^{\text{imp}}$
 - 13: **end if**
 - 14: Compute adaptive selection probabilities $\pi_{\odot}$ according to the current operator weights
 - 15: Select a destroy operator $\odot \in \Omega$ by roulette-wheel selection based on $\pi_{\odot}$
 - 16: Destroy x^{current} by operator $\odot$ and obtain a partial solution x^{part}
 - 17: Greedy repair x^{part} and obtain a candidate skeleton solution x^{new}
 - 18: Perform cross-route customer exchange with a given probability pr^{cross}
 - 19: Perform 2-opt local search at specified intervals
 - 20: Apply Algorithms 3 and 4 to x^{new} and compute $f(x^{\text{new}})$
 - 21: **if** x^{new} is temporarily irreparable **then**
 - 22: Reject the candidate solution
 - 23: **else if** $f(x^{\text{new}}) \leq f^{\text{current}}$ or the SA acceptance criterion is satisfied **then**
 - 24: $x^{\text{current}} \leftarrow x^{\text{new}}$, $f^{\text{current}} \leftarrow f(x^{\text{new}})$
 - 25: **end if**
 - 26: Assign a score to the selected operator $\odot$ according to the search outcome of x^{new}
 - 27: Update the accumulated score $\theta_{\odot}$ and usage count $w_{\odot}$ of operator $\odot$
 - 28: **if** $f(x^{\text{current}}) < f^{\text{best}}$ **then**
 - 29: $x^{\text{best}} \leftarrow x^{\text{current}}$, $f^{\text{best}} \leftarrow f(x^{\text{current}})$, and update $\mathcal{T}^{\text{imp}}$
 - 30: $g \leftarrow 0$
 - 31: **end if**
 - 32: **if** $w \bmod \mathbb{W} = 0$ **then**
 - 33: Update the destroy-operator weights using the accumulated scores and usage counts
 - 34: Reset $\theta_{\odot} \leftarrow 0$ and $w_{\odot} \leftarrow 0$ for each $\odot \in \Omega$
 - 35: **end if**
-

```

36:   Update the temperature  $T \leftarrow \max\{T_{\text{end}}, \alpha T\}$ 
37:    $\mathbb{W} \leftarrow \mathbb{W} + 1$ 
38: end while
39: return  $x^{\text{best}}$ 

```

4.1. Initial Solution Construction

To construct an evaluable initial solution of reasonable quality within a short computational time, the initialization procedure follows the principle of “skeleton routes first and charging expansion later.” First, Algorithm 2 determines only the vehicle assignment and visiting order of customers. Specifically, under a preset number of activated vehicles, all customers are sorted according to the starting time of their time windows and their distances from the depot. They are then inserted one by one into feasible positions of the current routes according to a greedy insertion rule. The insertion decision is mainly based on the distance increment, and the vehicle-load capacity feasibility of the resulting route is checked after insertion. This process generates a customer visiting sequence without charging nodes. After construction, a lightweight 2-opt improvement can be performed on each route. That is, a small-scale 2-opt local search is applied to each initial customer route. By exchanging two edges and reversing the intermediate segment, obvious crossings and redundant detours can be quickly removed, thereby improving the quality of the initial solution with low computational effort. Unlike the more intensive local search used in subsequent iterations, this step serves only as a fast post-processing operation after initial construction.

Algorithm 2. Construction of Customer-Level Skeleton Routes

Input: Instance $\mathcal{B}$, pre-specified number of activated vehicles m

Output: Customer-level route solution x^{route}

```

1: Initialize  $m$  empty routes  $R_k = [0,0]$ 
2:  $U \leftarrow$  set of unassigned customers
3: while  $U \neq \emptyset$  do
4:    $\zeta^{\text{best}} \leftarrow \zeta^\emptyset$ 
5:    $\varphi^{\text{cost}} \leftarrow +\infty$ 
6:   for each  $i \in U$  do
7:     for each route  $R_k$  do
8:       for each candidate insertion  $p$  do
9:         Compute the distance increment  $\Delta(\mathcal{L})$  caused by inserting  $i$ 
10:        Check load-capacity feasibility
11:        if feasible and the increment is the smallest then
12:          Update  $\zeta^{\text{best}}$  and  $\varphi^{\text{cost}}$ 
13:        end if
14:      end for
15:    end for
16:  end for
17:  if  $\zeta^{\text{best}} \neq \zeta^\emptyset$  then
18:    Execute  $\zeta^{\text{best}}$ 
19:    Remove the corresponding customer from  $U$ 
20:  else
21:    Insert at the feasible position with minimum increment
22:    Remove the corresponding customer from  $U$ 
23:  end if
24: end while
25: for each route  $R_k$  do
26:    $R_k \leftarrow$  light 2-opt improvement ( $R_k$ )
27: end for

```

28: $x^{route} \leftarrow$ customer-level route solution composed of all R_k
 29: **return** x^{route}

It should be noted that the solution generated above is only an initial skeleton solution rather than a final executable route. For this skeleton solution, Algorithm 1 subsequently performs charging-station insertion in the evaluation stage and further constructs the corresponding extended route with queue scheduling. The resulting comprehensive objective value is used to initialize the current solution and the historical best solution. In Algorithm 2, R_k is the k -th vehicle route, U is the set of unassigned customers, i is the customer to be inserted, and p denotes a candidate insertion position. ζ^{best} records the best insertion scheme identified during scanning. It usually contains a set of information, $\zeta^{best} = (i^*, k^*, p^*)$, where i^* is the customer to be inserted, k^* indicates the target route into which the customer should be inserted, and p^* is the insertion position. $\zeta^\emptyset$ indicates that no feasible insertion scheme has been found. The minimum incremental cost corresponding to the scheme is denoted as φ^{cost} . During initialization, $\zeta^{best} \leftarrow \zeta^\emptyset$ indicates that no feasible insertion action has yet been identified, and $\varphi^{cost} \leftarrow +\infty$ assigns a sufficiently large upper bound to the current best cost, so that it can be continuously updated during the subsequent traversal.

4.2. Destroy Operators

In the destroy phase, a subset of customers is removed from the current solution to partially disrupt the solution structure and help the search escape local optima. This paper employs several customer-removal mechanisms, including random removal, related removal, and large-scale destruction. The destroy operator is selected according to adaptive probabilities derived from operator weights. To keep the neighborhood operations focused on the customer-routing structure, the destroy phase removes only customer nodes. Charging nodes are generated and adjusted in a unified manner by the charging-insertion module during the evaluation stage. This design avoids the repeated handling of charging and queueing details during neighborhood operations and thus reduces computational burden. The destroy-operator set is defined as $\Omega = \{\mathbb{O}^{rand}, \mathbb{O}^{rel}, \mathbb{O}^{large}\}$, where $\mathbb{O}^{rand}$, $\mathbb{O}^{rel}$, and $\mathbb{O}^{large}$ denote the random removal operator, Shaw removal operator, and large-scale destruction operator, respectively. The initial weights of these three operators are set to 0.55, 0.30, and 0.15, respectively. These weights are used only to initialize the roulette-wheel selection process and are subsequently updated according to the adaptive weight-update mechanism described in Section 4.6.

Let $\mathcal{C}(x)$ be the customer set in the current skeleton solution. For the normal random removal and related removal operators, the removal size is first randomly determined from a pre-specified integer interval:

$$n_d \sim U_{\mathbb{Z}}(\{n_{min}, n_{min} + 1, \dots, \tilde{n}_{max}\}), \tilde{n}_{max} = \min\{n_{max}, |\mathcal{C}(x)|\} \quad (45)$$

where n_{min} and n_{max} are the lower and upper bounds of the normal removal size, respectively. The notation $U_{\mathbb{Z}}$ denotes a discrete uniform distribution over a finite integer set, and $\tilde{n}_{max}$ ensures that the removal size does not exceed the number of customers in the current skeleton solution. In random removal, n_d customers are uniformly sampled from $\mathcal{C}(x)$ to form the removal set D . In related removal, a seed customer is first randomly selected, and the remaining customers are selected according to their relatedness to the seed customer or to the already removed customers.

For related removal, a seed customer i is first selected at random. Then, the remaining customers are ranked according to their relatedness to the seed customer or to the already removed customers, and customers with stronger relatedness are more likely to be removed together. In this study, the relatedness between customers i and j is defined as

$$Rel(i, j) = d_{ij} + \beta_q |q_i - q_j| + \beta_t |\bar{t}_i - \bar{t}_j| \quad (46)$$

where d_{ij} denotes the travel distance between customers i and j , q_i denotes the demand of customer i , and $\bar{t}_i = (a_i + b_i)/2$ denotes the center of the time window of customer i . The parameters β_q and β_t are used to adjust the influence of demand difference and time-window difference, respectively. In the implementation, they are set to $\beta_q = 0.3$ and $\beta_t = 0.1$. A smaller value of $Rel(i, j)$ indicates that the two customers are closer in spatial location, demand level, or service-time attribute, and are therefore more suitable to be removed simultaneously.

After the relatedness values are calculated, the candidate customers are sorted in ascending order of $Rel(i, j)$. A bias parameter χ is introduced to control the preference for highly related customers during the selection process. In this implementation, $\chi = 6.0$, so that customers with stronger relatedness to the seed customer are more likely to be selected and removed together. This mechanism enables the algorithm to destroy a locally related customer group and thus promotes more targeted structural reorganization.

To enlarge the search radius and enhance global exploration, a large-scale destruction operator is further introduced. Compared with the normal removal operators, this operator removes a larger number of customers, and its removal size is randomly selected from a larger integer interval:

$$n_d^{large} \sim U_z(\{n_{min}^{large}, n_{min}^{large} + 1, \dots, \tilde{n}_{max}^{large}\}), \tilde{n}_{max}^{large} = \min\{n_{max}^{large}, |C(x)|\} \quad (47)$$

where n_{min}^{large} and n_{max}^{large} are the lower and upper bounds of the large-scale removal size, respectively. The condition $n_{min}^{large} \geq n_{max}$ indicates that the large-scale destruction operator uses a larger removal interval than the normal removal operators. Accordingly, a stronger perturbation can be generated to improve the global exploration capability of the algorithm.

4.3. Repair Operators

After customer removal, the removed customers need to be reinserted into the partial skeleton routes to construct a new candidate skeleton solution. Suppose that customer i is to be inserted between adjacent nodes u and v on skeleton route r . The distance increment caused by this insertion is defined as

$$\Delta_i^{ins}(u, v) = d_{ui} + d_{iv} - d_{uv} \quad (48)$$

where d_{ui} , d_{iv} , and d_{uv} denote the corresponding distances between node pairs. If the currently assigned demand of route r is l_r , the capacity constraint after inserting customer i should satisfy

$$l_r + q_i \leq Q \quad (49)$$

where Q is the vehicle-load capacity and q_i is the demand of customer i .

Therefore, for any customer i to be reinserted, its best insertion position can be expressed as

$$(r^*, u^*, v^*) = \arg \min_{r, (u, v)} \{\Delta_i^{ins}(u, v) \mid l_r + q_i \leq Q\} \quad (50)$$

Among all routes and positions satisfying capacity feasibility, the position with the smallest distance increment is selected for insertion. The repair phase restores the removed customers one by one according to the above greedy rule and thus produces a new candidate skeleton solution. Since soft time windows are adopted, time-window violations are not forcibly excluded in the repair phase, but are reflected by tardiness penalties in the subsequent evaluation function. Similarly, battery feasibility and the

impact of charging queues are not directly handled in this phase; instead, they are handled and evaluated by the subsequent charging-insertion and queue-scheduling modules.

4.4. Local Search and Cross-Route Perturbation

After the destroy and repair phases, the obtained candidate skeleton solution has been restored as a complete set of customer routes. However, the intra-route visiting order and the customer allocation among different routes may still be suboptimal. To further improve candidate-solution quality and strengthen the ability of the algorithm to escape local optima, local search and lightweight cross-route perturbation are embedded into the main search framework. Local search is mainly used to optimize the internal structure of a single route, whereas cross-route perturbation enhances the reorganization capability among different routes. The two mechanisms complement each other and jointly improve the search quality of skeleton solutions.

First, a 2-opt local search operator is introduced at the single-route level. If two edges (i_{a-1}, i_a) and (i_b, i_{b+1}) are selected from vehicle skeleton route R_k and then disconnected and reconnected, where $1 \leq a < b \leq m$, the route-distance variation associated with the 2-opt operation can be expressed as

$$\Delta_{ab}^{2\text{-opt}} = (d_{i_{a-1}, i_b} + d_{i_a, i_{b+1}}) - (d_{i_{a-1}, i_a} + d_{i_b, i_{b+1}}) \quad (51)$$

When $\Delta_{ab}^{2\text{-opt}} < 0$, the operation can reduce the route length and is therefore accepted as a local adjustment. The adjusted route can be written as

$$R'_k = (0, i_1, \dots, i_{a-1}, i_b, i_{b-1}, \dots, i_a, i_{b+1}, \dots, i_m, 0) \quad (52)$$

This operation reconnects the segment $(i_a, \dots, i_b)$ in reverse order. It can effectively reduce redundant detours in the route and improve the internal structure of a single route.

In addition to intra-route optimization, the algorithm performs cross-route customer exchanges with a given probability pr^{cross} to enhance structural recombination among different routes. Suppose that customer i is located on the skeleton route R_k of vehicle k , and customer j is located on the skeleton route R_h of vehicle h . After exchanging the two customers, capacity feasibility should first be satisfied, namely

$$l_k - q_i + q_j \leq Q, l_h - q_j + q_i \leq Q \quad (53)$$

where l_k and l_h denote the total customer demands on routes R_k and R_h before the exchange, respectively. Q is the vehicle-load capacity, while q_i and q_j denote the demands of customers i and j , respectively.

Furthermore, let $pre(i)$ and $suc(i)$ denote the predecessor and successor of customer i in its original route R_k , respectively. Similarly, let $pre(j)$ and $suc(j)$ denote the predecessor and successor of customer j in its original route R_h . The skeleton-route distance variation caused by this exchange can then be expressed as

$$\Delta_{ij}^{swap} = (d_{pre(i), j} + d_{j, suc(i)} - d_{pre(i), i} - d_{i, suc(i)}) + (d_{pre(j), i} + d_{i, suc(j)} - d_{pre(j), j} - d_{j, suc(j)}) \quad (54)$$

If the capacity constraints are satisfied, the exchange is retained as a cross-route perturbation move. The resulting candidate solution is then evaluated by the charging-insertion and queue-scheduling modules, and its final acceptance is determined by the acceptance rule in Algorithm 1.

4.5. Charging-Station Insertion and Queue Scheduling

4.5.1. Charging-Station Insertion

The purpose of charging insertion is to ensure that the battery-state evolution of a vehicle satisfies the minimum battery threshold on a given customer route. For any skeleton route, the algorithm starts from the depot with a fully charged battery and calculates the battery consumption segment by segment along the route. If the battery level after traversing a certain arc would fall below the safety threshold, a virtual charging-station node is inserted before that arc.

Virtual charging-station nodes are used to distinguish different charging visits. These visits are ultimately mapped to their corresponding real charging stations and compete for the limited charger resources at each station. In the experimental setting of this paper, each time a vehicle arrives at a virtual charging-station node, it performs one charging operation, and the battery level is restored to the upper battery-capacity limit B after charging is completed.

The notation related to Algorithm 3 is described as follows. x^{route} denotes the input customer-level initial skeleton solution, which consists of several vehicle routes containing only the depot and customer nodes. x^{ext} denotes the extended-route solution obtained after inserting charging nodes into x^{route} . For any route, $\mathcal{R}$ is the vehicle route currently being processed, and E is the remaining battery level of the vehicle at the current node, while B is the battery capacity and the full-charge battery level. ID_x is the index of the current scanning position in the route, and $|\mathcal{R}|$ is the total number of nodes in the route. Let (i, j) be the adjacent arc currently under consideration, where i is the current node and j is its successor. e_{ij} denotes the battery consumption when the vehicle travels along arc (i, j) , and E^{min} denotes the preset minimum safety battery threshold. If the vehicle still satisfies $E - e_{ij} \geq E^{min}$ after traveling along the current arc, then the arc is battery-feasible; otherwise, a charging station needs to be inserted before this arc.

Algorithm 3. Charging-Station Insertion

Input: Customer-level skeleton solution x^{route} , instance $\mathcal{I}$

Output: Expanded-route solution x^{ext} with charging nodes

```

1:  $x^{ext} \leftarrow x^{route}$ 
2: for each route  $\mathcal{R} \in x^{ext}$  do
3:    $E \leftarrow$  Initial battery capacity  $B$ 
4:   Initialize the visited real-station set of the current vehicle
5:    $ID_x \leftarrow 1$ 
6:   while  $ID_x < |\mathcal{R}|$  do
7:     Let the current arc be  $(i, j)$ 
8:     Compute the energy consumption  $e_{ij}$  on this arc  $(i, j)$ 
9:     if  $E - e_{ij} \geq E^{min}$  then
10:       $E \leftarrow E - e_{ij}$ 
11:       $ID_x \leftarrow ID_x + 1$ 
12:     else
13:       $s^* \leftarrow s^\emptyset$ 
14:       $\Delta d_{i,s,j}^{min} \leftarrow +\infty$ 
15:      for each candidate virtual charging-station node  $s$  do
16:        if  $s$  satisfies the dual energy-feasibility condition and its
          associated real station has not been visited by the current vehicle
          then
17:           $\Delta d_{i,s,j} \leftarrow d(i, s) + d(s, j) - d(i, j)$ 
18:          if  $\Delta d_{i,s,j} < \Delta d_{i,s,j}^{min}$  then
19:             $s^* \leftarrow s$ 

```

```

20:            $\Delta d_{i,s,j}^{min} \leftarrow \Delta d_{i,s,j}$ 
21:       end if
22:   end if
23: end for
24: if  $s^* \neq s^\emptyset$  then
25:     Insert  $s^*$  between  $i$  and  $j$ 
26:     Record the associated real station of  $s^*$  as visited by the current
        vehicle
27:     Compute the charging duration according to the charging rate
28:      $E \leftarrow B$ 
29: else
30:     Mark the current route as temporarily irreparable and exit
31: end if
32: end if
33: end while
34: end for
35: return  $\chi^{ext}$ 

```

During charging-station selection, s^* denotes the currently recorded best candidate virtual charging-station node. Its initial value is $s^\emptyset$, indicating that no feasible station has yet been found. $\Delta d_{i,s,j}^{min}$ denotes the currently recorded minimum detour increment, and is initialized as positive infinity so that it can be updated during the traversal. For any candidate virtual charging-station node s , if the vehicle can reach s from the current node i and can further reach the successor node j after being fully recharged at s , then the station is said to satisfy the dual energy-feasibility condition ($E - e_{is} \geq E^{\min}$ and $B - e_{sj} \geq E^{\min}$). If the associated real charging station of s has not been visited by the current vehicle, then station s is regarded as a feasible insertion candidate. In this case, the detour increment associated with station s is given by

$$\Delta d_{i,s,j} = d(i, s) + d(s, j) - d(i, j) \quad (55)$$

where $d(\cdot, \cdot)$ denotes the travel distance between two nodes. Among all candidate stations satisfying the condition, the algorithm selects the one with the minimum detour increment as the insertion station. If $s^* \neq s^\emptyset$, a feasible charging station has been identified; this station can be inserted into arc (i, j) , and the corresponding charging duration is calculated according to the charging rate.

$$H_s^k = \frac{B - E_s^{a,k}}{\rho_p}, s \in S_p \quad (56)$$

Otherwise, the current route cannot be repaired under the available candidate stations, and the algorithm marks it as currently unrepaired.

4.5.2. Queue Scheduling Under Charging-Station Capacity Constraints

In the evaluation stage of the heuristic algorithm, the station-level queue is scheduled according to a deterministic arrival-time-priority rule. Charging events at the same real station are sorted by their updated arrival times, and each event is assigned to the charger that enables the earliest charging start time.

After the extended routes $\{\tilde{R}_k\}$ of all vehicles have been obtained, the queueing process is scheduled for each real charging station. For any real charging station p currently being processed, let $\mathcal{E}_p(x)$ denote the set of selected charging events associated with p in the current solution, where $\mathcal{E}_p(x)$ is a subset of the candidate charging-event set $\mathcal{E}_p$ defined in Section 3. Specifically, if vehicle k visits a virtual charging-station node $s \in S_p$ in the extended route, then the pair (k, s) forms a selected charging event in

$\mathcal{E}_p(x)$. For ease of exposition, a selected charging event $q = (k, s) \in \mathcal{E}_p(x)$ is used in the following queue-scheduling procedure. For each charger within the station, let $t_{p,h}^{\text{free}}$ denote the current available time of the h -th charger at station p . It is initialized as 0, indicating that all chargers are idle at the beginning of scheduling. For any charging event q , A_s^k denotes the time at which the vehicle arrives at the corresponding charging station, H_s^k denotes the required charging duration, T_s^k denotes the charging start time, W_s^k denotes the queue waiting time, and $t_s^{e,k}$ denotes the charging completion time. Among all the chargers in the real charging station p , select the charger that enables the current charging event (k, s) to start charging at the earliest, and denote its index by h^* . The charging start time T_s^k is the maximum of the arrival time A_s^k and the available time t_{p,h^*}^{free} of the assigned charger, namely

$$T_s^k = \max\{A_s^k, t_{p,h^*}^{\text{free}}\} \quad (57)$$

This means that a vehicle can start charging only after it has arrived at the charging station and the assigned charger is available.

$$W_s^k = T_s^k - A_s^k \quad (58)$$

Equation (58) represents the waiting time generated by this charging event at the station.

$$t_s^{e,k} = T_s^k + H_s^k \quad (59)$$

Equation (59) indicates the end time of charging.

Let x^{ext} denote the input extended-route solution, which contains both customer nodes and charging nodes because charging-station insertion has already been completed. x^{sched} denotes the complete time-stamped solution obtained by further considering queueing delays at charging stations on the basis of x^{ext} . $\mathcal{B}$ denotes the problem-instance data, including the set of real charging stations, the number of chargers at each station, vehicle operating parameters, and information related to charging events. ε is the convergence threshold used to determine whether the change between two adjacent schedules is sufficiently small. w_{queue} denotes the current iteration counter. Δt^{max} denotes the maximum change between the current schedule and the previous schedule; it is initialized as positive infinity to ensure that the iteration can start. x^{old} denotes the schedule saved from the previous iteration.

After each round of station-level scheduling, Algorithm 4 writes the charging start time and waiting time of each charging event back to the corresponding vehicle route, and updates the arrival and service times at subsequent nodes accordingly. Finally, Δt^{max} measures the maximum change in the current schedule relative to the previous one. When Δt^{max} is no larger than the convergence threshold ε , or when the number of iterations reaches the upper limit $w_{\text{queue}}^{\text{max}}$, the iteration stops, and the complete time-stamped solution with queueing-delay information, x^{sched} , is output.

Algorithm 4. Queue Scheduling

Input: Expanded-route solution x^{ext} , instance $\mathcal{B}$, maximum number of queue iterations $w_{\text{queue}}^{\text{max}}$, convergence threshold ε

Output: Full time-stamped solution x^{sched} with queueing delays

- 1: $x^{\text{sched}} \leftarrow x^{\text{ext}}$
 - 2: Ignore queueing delays and initialize the arrival times of all charging events
 - 3: $w_{\text{queue}} \leftarrow 0$; $\Delta t^{\text{max}} \leftarrow +\infty$
 - 4: **while** $w_{\text{queue}} < w_{\text{queue}}^{\text{max}}$ and $\Delta t^{\text{max}} > \varepsilon$ **do**
 - 5: Save the previous schedule x^{old}
 - 6: **for** each real charging station p **do**
-

```

7:      Extract the selected charging events associated with  $p$  as  $\mathcal{E}_p(x)$ 
8:      Sort  $\mathcal{E}_p(x)$  in ascending order of updated arrival time
9:      Initialize the availability time of each charger  $t_{p,h}^{\text{free}} \leftarrow 0$ 
10:     for each charging event  $q \in \mathcal{E}_p(x)$  do
11:         Select the currently earliest available charger  $h^*$ 
12:          $T_s^k \leftarrow \max\{A_s^k, t_{p,h^*}^{\text{free}}\}$ 
13:          $W_s^k \leftarrow T_s^k - A_s^k$ 
14:          $t_s^{e,k} \leftarrow T_s^k + H_s^k$ 
15:          $t_{p,h^*}^{\text{free}} \leftarrow t_s^{e,k}$ 
16:         Record the charging start time and waiting duration of  $q$ 
17:     end for
18: end for
19: // Time propagation //
20: Write back the waiting times and charging start times to the vehicle routes
21: Update the arrival times and service times of all subsequent nodes
22:  $\Delta t^{\text{max}} \leftarrow$  Maximum change between the current schedule and  $x^{\text{old}}$ 
23:  $w_{\text{queue}} \leftarrow w_{\text{queue}} + 1$ 
24: end while
25: return  $x^{\text{sched}}$ 

```

In the implementation, the maximum number of queue-scheduling iterations is set to $w_{\text{queue}}^{\text{max}} = 40$, and the convergence threshold is set to $\varepsilon = 10^{-6}$. The threshold is used to judge whether the maximum change in charging start times between two consecutive scheduling iterations is sufficiently small.

The evaluation function calculates the comprehensive objective value under the unified time schedule and uses it as the solution-quality measure in ALNS for candidate-solution comparison, acceptance judgment, and historical-best-solution update.

4.6. Adaptive Operator Weight Update, Acceptance Criterion, Stagnation Restart, and Termination Conditions

In the proposed ALNS, the destroy operator is selected according to adaptive probabilities derived from operator weights. Let Ω denote the set of destroy operators, and let $\omega_{\odot}$ denote the current weight of destroy operator $\odot \in \Omega$. Since a single greedy repair operator is used in this study, the adaptive weight update is applied to the destroy-operator set. To prevent any destroy operator from being completely excluded during the search, a minimum selection probability $\pi_{\min}$ is retained. The selection probability of operator $\odot$, denoted by $\pi_{\odot}$, is defined as

$$\tilde{\omega}_{\odot} = \max\{\omega_{\min}, \omega_{\odot}\}, \pi_{\odot} = \pi_{\min} + (1 - |\Omega| \pi_{\min}) \frac{\tilde{\omega}_{\odot}}{\sum_{\odot' \in \Omega} \tilde{\omega}_{\odot'}} \quad (60)$$

where $\omega_{\min}$ is the minimum operator weight, and $\pi_{\min}$ is the minimum selection probability. During the search, the selected destroy operator receives a score according to the quality of the candidate solution it generates. Specifically, higher scores are assigned when the operator generates a new historical best solution or improves the current solution, while lower scores are assigned when the candidate solution is accepted without improvement or rejected.

The scores and usage counts are accumulated within a fixed update interval. Let $\vartheta_{\odot}$ and $w_{\odot}$ denote the accumulated score and usage count of operator $\odot$ in the current interval, respectively. If $w_{\odot} > 0$, the average score of operator $\odot$ is calculated as

$$\bar{\theta}_{\odot} = \frac{\vartheta_{\odot}}{w_{\odot}} \quad (61)$$

Then, the weight of operator $\mathbb{O}$ is updated by

$$\omega_{\mathbb{O}} \leftarrow \max\{\omega_{\min}, (1 - \mathbb{p})\omega_{\mathbb{O}} + \mathbb{p}\bar{\theta}_{\mathbb{O}}\} \quad (62)$$

where $\mathbb{p}$ is the reaction factor. After each weight update, the accumulated scores and usage counts are reset for the next update interval. This mechanism enables effective destroy operators to be selected more frequently, while maintaining the exploration ability of less frequently selected operators.

To balance global exploration and local convergence, a simulated annealing (SA) acceptance criterion is adopted to determine whether a candidate solution should replace the current solution. Let the objective value of the current solution be $f(x^{\text{current}})$, and let the objective value of the candidate solution be $f(x^{\text{new}})$. If $f(x^{\text{new}}) \leq f(x^{\text{current}})$, the candidate solution is directly accepted. If $f(x^{\text{new}}) > f(x^{\text{current}})$, the candidate solution is accepted with the following probability:

$$Pr_{\text{accept}} = \exp\left(-\frac{f(x^{\text{new}}) - f(x^{\text{current}})}{T}\right) \quad (63)$$

where T is the current temperature parameter. As the iteration proceeds, the temperature gradually decreases according to the preset cooling coefficient:

$$T \leftarrow \max\{T_{\text{end}}, \alpha T\} \quad (64)$$

where T_{end} is the terminal temperature and α is the cooling coefficient. Consequently, the algorithm can accept inferior solutions with a certain probability in the early search stage to escape local optima, while gradually shifting toward stable convergence in the later stage.

When the historical best solution is not improved within the stagnation time threshold $\mathbb{G}$, the stagnation restart mechanism is triggered. Specifically, let g denote the elapsed time since the last improvement of the historical best solution. If $g \geq \mathbb{G}$, the current historical best solution is used as the new starting point of the search, and $\mathcal{S}$ cross-route customer exchanges are applied as perturbations. Meanwhile, the temperature is moderately reheated to restore search diversity:

$$T \leftarrow \max\{T, \eta T_0\} \quad (65)$$

where η is the reheating factor, T_0 is the initial temperature, and $\mathcal{S}$ denotes the number of perturbation moves applied after restart. After the restart is performed, $\mathfrak{T}^{\text{imp}}$ is reset to the current time to avoid repeatedly triggering the restart mechanism. This restart strategy preserves the historical best solution while introducing moderate perturbations to improve search diversification.

When the algorithm reaches the preset maximum running time or the given maximum number of iterations, the main loop terminates and outputs the historical best solution, together with its corresponding extended routes, queue-scheduling results, and objective-decomposition information.

5. Experimental Design and Results Analysis

5.1. Experimental Design

To validate the proposed model and the enhanced adaptive large neighborhood search (ALNS) algorithm, computational experiments are conducted on synthetically generated instances of different customer scales. The experiments examine model validity, the contribution of the enhancement modules, algorithmic convergence and solution quality, and sensitivity to charging-infrastructure settings. The test instances comprise six customer scales: 10, 15, 20, 30, 50, and 100 customers. The customer-related data are generated following the general structure of Solomon-type vehicle-routing benchmark

instances with time windows [24], rather than being directly extracted from the original Solomon instances. The depot is located at (50,50), while customer locations are generated around predefined cluster centers using scale-specific cluster configurations and random seeds. Specifically, the 10- and 15-customer instances adopt a two-cluster structure, the 20-customer instance adopts a three-cluster structure, and the 30-, 50-, and 100-customer instances adopt generalized multi-cluster structures. The corresponding random seeds are 1504, 1008, 3005, 4003, 5007, and 6003, respectively. Customer demands are generated as integers ranging from 5 to 15, the customer service time is fixed at 5, and customer time windows are generated within the depot planning horizon of [0,500].

Each instance contains one depot, a set of customer nodes, several real charging stations, and multiple virtual charging-station nodes associated with each real station. The real charging stations are positioned near the predefined cluster centers. The virtual charging-station nodes represent candidate charging visits, whereas charger capacity and queueing interactions are determined at the corresponding real station. For each real charging station, three associated virtual charging-station nodes are generated. The charging-infrastructure configurations are summarized in Table 3, while the common instance-generation settings are reported in Table 4.

Table 3. Numbers of real charging stations and virtual charging-station nodes in different instances.

Instance	Number of Real Charging Stations	Number of Virtual Charging-Station Nodes
10c	2	6
15c	2	6
20c	3	9
30c	4	12
50c	6	18
100c	10	30

Table 4. Common generation settings for the synthetic test instances.

Parameter	Description	Value
(X_0, Y_0)	Depot coordinates	(50, 50)
T^{max}	Depot planning horizon	500
q_i	Customer demand	5–15
σ_i	Customer service time	5
$[a_i, b_i]$	Customer time window	$[a_i, b_i] \in [0, 500]$

The key algorithmic parameters and objective-function coefficients are summarized in Table 5 and remain unchanged across the tested instances unless otherwise stated. The coefficients λ_1 – λ_5 correspond to travel distance, vehicle use, charging duration, queueing delay, and customer tardiness, and are set to 1.0, 5.0, 0.5, 2.0, and 5.0, respectively. The initial destroy-operator weights only initialize roulette-wheel selection and are subsequently updated using the adaptive mechanism described in Section 4.6. The baseline ALNS uses the same solution representation, charging-insertion procedure, queue-scheduling rule, stopping time, and random seeds as the enhanced ALNS; it differs only in the additional search-enhancement modules considered in the comparison.

Table 5. Key parameter settings of the enhanced ALNS and objective-function cost coefficients.

Parameter	Description	Value
n_{min}, n_{max}	Bounds for normal removal	2, 6
$n_{min}^{large}, n_{max}^{large}$	Bounds for large-scale removal	6, 10
ω_{rand}	Initial weight of random removal operator	0.55

ω_{rel}	Initial weight of related removal operator	0.30
ω_{large}	Initial weight of large-scale destruction operator	0.15
ω_{min}	Minimum operator weight	0.05
pr^{cross}	Probability of performing a cross-route customer exchange	0.55
π_{min}	Minimum selection probability	0.05
$\mathfrak{B}$	Weight-update interval	100
$\mathbb{P}$	Reaction factor	0.20
T_0	Initial temperature	2.0
T_{end}	Terminal temperature	0.05
α	Cooling factor	0.9995
$\mathfrak{T}_{\text{max}}$	Maximum running time	120 s
$\mathbb{G}$	Stagnation threshold	35 s
$\mathcal{S}$	Number of perturbation moves after restart	3
η	Reheating factor	0.50
$w_{\text{queue}}^{\text{max}}$	Maximum number of queue-scheduling iterations	40
ε	Queue-scheduling convergence threshold	10^{-6}
λ_1	Travel-distance cost	1.0
λ_2	Vehicle usage cost	5.0
λ_3	Charging-time cost	0.5
λ_4	Queueing-delay cost	2.0
λ_5	Tardiness penalty cost	5.0

Note: The listed values of n_{min} , n_{max} , $n_{\text{min}}^{\text{large}}$, $n_{\text{max}}^{\text{large}}$ are the default settings; the specific removal-size bounds are adjusted according to the instance scale in the experiments.

5.2. Model Validity Analysis

This section uses a small-scale instance to verify the validity of the proposed model and algorithm. The instance consists of one depot, 15 customer nodes, two real charging stations, and six virtual charging-station nodes. Each real charging station is equipped with one charger, so that the limited charging-resource setting and the resulting queueing phenomenon can be clearly illustrated.

In Figure 3, routes in different colors correspond to different vehicles. Three electric delivery vehicles depart from the depot, visit charging stations according to their battery states while satisfying customer service requirements, and finally return to the depot.

Because Figure 3 cannot show the details of queueing and waiting at the charging station, Figure 4 presents the queueing process at charging station S8. The horizontal axis represents time, while the vertical axis indicates the charging sequence at the same station. Each horizontal bar corresponds to the complete charging process of one vehicle at the station, where the first segment denotes queue waiting and the second segment denotes actual charging. It can be observed that vehicles V1, V2, and V3 arrive at the same charging station. Because the station has only a limited number of chargers, the vehicles must be charged sequentially, resulting in queueing delays.

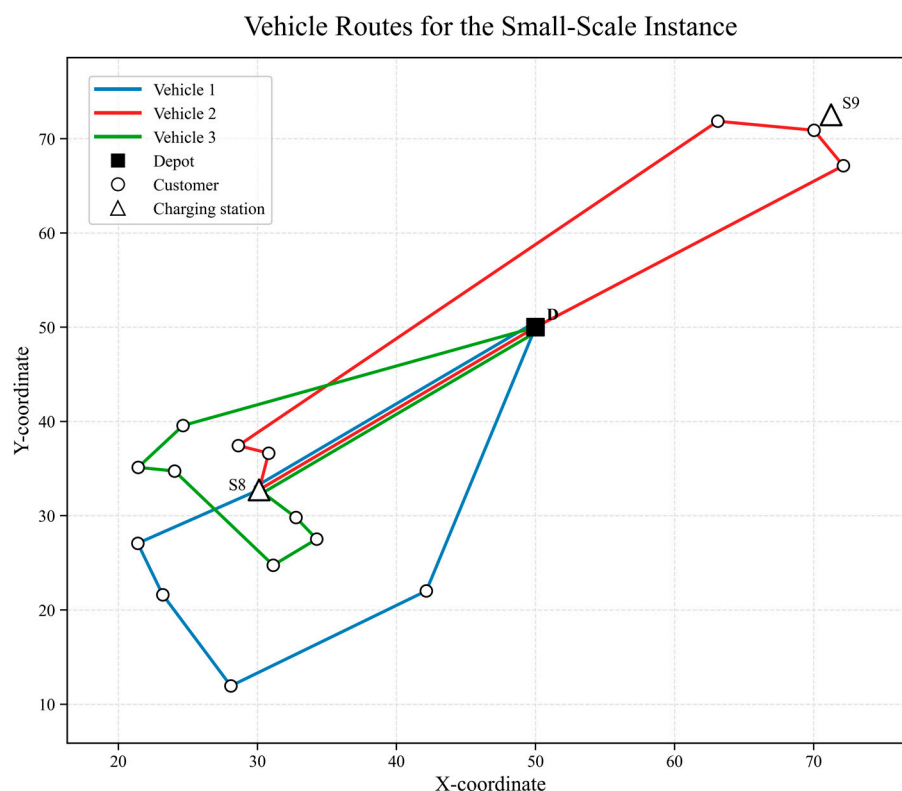


Figure 3. Vehicle routes for the small-scale instance.

Specifically, V1 can start charging immediately after arrival, whereas V2 and V3 must wait in sequence until the preceding vehicle has completed charging before they can begin charging. Vehicles arriving later experience longer waiting times. This result indicates that, when the number of available chargers is limited, charging-station visits affect not only the duration of individual charging operations but also the subsequent timing of vehicle routes through the queueing mechanism. In other words, charging-station capacity constraints create significant temporal coupling among different vehicles. These results verify the necessity of explicitly modeling queueing delays at charging stations and show that approaches ignoring charging-station capacity constraints may fail to capture recharging conflicts in actual operations. From an operational perspective, the proposed formulation prevents overlapping use of the same charger and produces executable route and charging schedules. For centrally coordinated electric delivery fleets, this integrated representation can reduce unplanned vehicle idling and prevent charging delays from propagating to subsequent customer services.

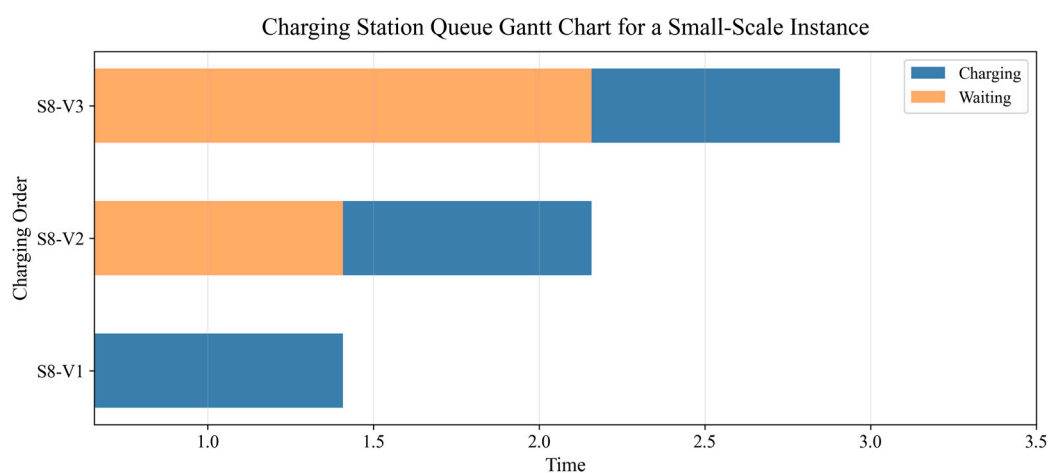


Figure 4. Queueing Gantt chart at the charging station in the small-scale instance.

5.3. Contributions of the Three Enhancement Modules

To evaluate the contribution of each enhancement strategy within the current algorithmic framework, three ablation variants are constructed. For each variant, one enhancement module is removed while the other modules are retained, thereby isolating the marginal contributions of the 2-opt local search, stagnation restart mechanism, and cross-route perturbation strategy. The performance of each ablation variant is then compared with that of the complete enhanced ALNS. In Table 6, f_1 and $\bar{f}_1$, f_2 and $\bar{f}_2$, and f_3 and $\bar{f}_3$ denote the best and mean objective values obtained over 10 independent runs by the variants without 2-opt, stagnation restart, and cross-route perturbation, respectively. Similarly, f and $\bar{f}$ denote the best and mean objective values obtained by the complete enhanced ALNS. The relative contribution of module i is evaluated based on the mean objective values:

$$\Delta_i = \frac{\bar{f}_i - \bar{f}}{\bar{f}} \times 100\% \quad (66)$$

Table 6. Objective values of the complete enhanced ALNS and its ablation variants on instances of different scales.

Instance	Enhanced ALNS								Δ_1 (%)	Δ_2 (%)	Δ_3 (%)
	f_1	$\bar{f}_1$	f_2	$\bar{f}_2$	f_3	$\bar{f}_3$	f	$\bar{f}$			
10c	209.093	209.093	209.093	209.093	209.093	209.093	209.093	209.093	0.00%	0.00%	0.00%
15c	325.914	325.914	325.914	325.914	325.914	325.914	325.914	325.914	0.00%	0.00%	0.00%
20c	517.293	520.342	519.120	520.872	519.120	519.649	519.594	519.660	0.13%	0.23%	−0.002%
30c	742.315	751.538	742.315	757.361	742.315	754.639	742.315	750.790	0.10%	0.87%	0.51%
50c	1181.930	1186.617	1186.002	1194.263	1181.365	1189.182	1180.156	1186.598	0.00%	0.64%	0.22%
100c	2415.457	2480.973	2473.513	2489.135	2467.352	2490.650	2429.723	2480.573	0.02%	0.34%	0.41%

As shown in Figure 5 and Table 6, the differences among the algorithmic variants are negligible for the 10- and 15-customer instances, for which all four versions obtain identical best and mean objective values. For the 20-customer instance, removing the 2-opt local search or the stagnation restart mechanism leads to a slight deterioration in the mean objective value. By contrast, removing the cross-route perturbation produces a relative change of only −0.002%, indicating a negligible effect. As the instance size increases, the differences among the variants become more evident. In particular, the stagnation restart and cross-route perturbation modules generally yield positive improvements for the 30-, 50-, and 100-customer instances, demonstrating that the enhancement modules are more effective for larger instances on average.

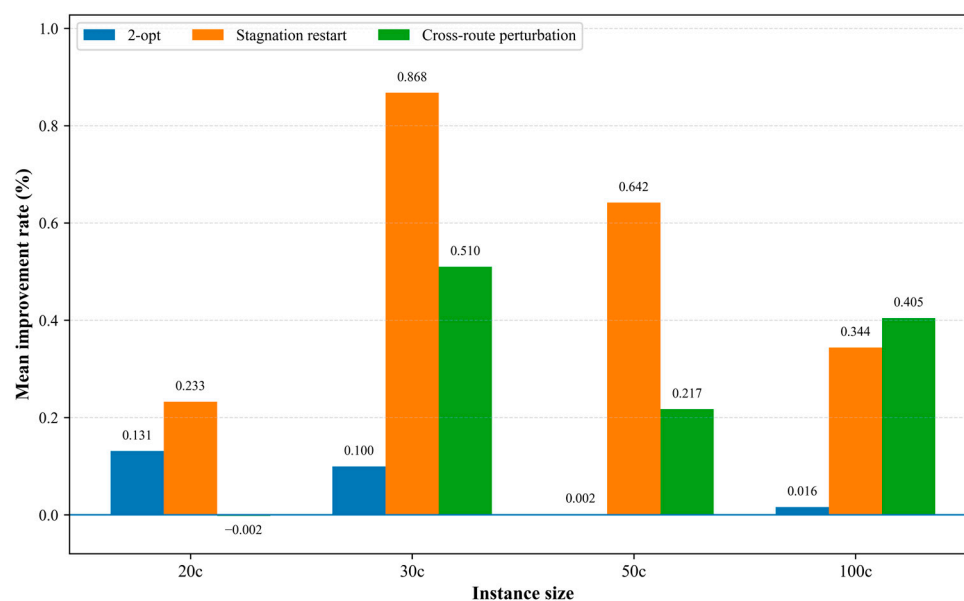


Figure 5. Effects of the three enhancement modules.

The contribution of each module is nevertheless instance-dependent. The 2-opt local search provides only limited improvements in the tested instances, while the stagnation restart mechanism shows the most consistent benefit across medium- and large-scale cases. The cross-route perturbation has little influence on the 20-customer instance but becomes more useful as the problem size grows. Overall, no single module dominates across all instances. The results suggest that the performance of the complete enhanced ALNS mainly arises from the complementary effects of the three enhancement strategies.

To further examine the roles of the hierarchical evaluation components, two diagnostic variants are tested on six representative instances. The charging-insertion and queue-scheduling components are not conventional search operators; rather, they serve as feasibility-recovery and solution-evaluation components within the proposed ALNS framework. Therefore, the variants obtained by removing these components are treated as diagnostic settings rather than as complete algorithms for direct performance comparison.

For the variant without the charging-insertion module, all charging-station visits are removed from the routes produced by the proposed ALNS, and the resulting customer-only routes are then checked for battery feasibility. As shown in Table 7, the routes remain feasible for the 10-customer instance because of the relatively short travel distances. For the larger instances, however, removing the charging visits causes an increasing number of used vehicle routes to become battery-infeasible. Specifically, 2 of 3, 2 of 4, 4 of 5, 6 of 9, and 7 of 18 routes are infeasible for the 15-, 20-, 30-, 50-, and 100-customer instances, respectively. These results confirm that the charging-insertion module is essential for maintaining battery feasibility, particularly for medium- and large-scale instances.

For the variant without the queue-scheduling module, the same complete routes are re-evaluated without considering charger-capacity conflicts or the resulting waiting times. This relaxed evaluation therefore excludes the waiting-related cost caused by limited charging capacity. As reported in Table 7, the omitted queue-waiting penalty increases from 1.487 for the 10-customer instance to 235.381 for the 100-customer instance. This substantial increase shows that the queue-scheduling module is necessary for accurately evaluating solutions when charging-station capacity is limited.

Table 7. Diagnostic verification of the charging-insertion and queue-scheduling components.

Instance	Battery Feasible	Used Vehicle Routes	Battery-Infeasible Routes	Omitted Queue-Waiting Penalty
10c	✓	2	0	1.487
15c	×	3	2	4.499
20c	×	4	2	11.328
30c	×	5	4	20.660
50c	×	9	6	56.275
100c	×	18	7	235.381

Note: Battery feasible indicates whether all used vehicle routes remain battery-feasible after removing charging-station visits. The omitted waiting penalty denotes the waiting-related objective component that would be excluded if the queue-scheduling module were removed.

5.4. Algorithmic Convergence Comparison

After evaluating the contributions of the internal enhancement strategies, Gurobi, the baseline ALNS, and the enhanced ALNS are further compared in terms of solution quality and convergence performance. All MILP experiments are performed sequentially on the same computer using Gurobi Optimizer 13.0.0. The relative MIP gap tolerance is set to 10^{-4} , while the remaining solver parameters are retained at their default values. A Gurobi run terminates when either the prescribed time limit is reached or the relative MIP gap satisfies the specified tolerance. Because the computational difficulty of the MILP model increases rapidly with the instance size, progressively longer time limits are assigned to Gurobi. Specifically, the time limits are set to 300 s for the 10- and 15-customer instances, 600 s for the 20-customer instance, 1800 s for the 30-customer instance, and 3600 s for the 50- and 100-customer instances. The baseline ALNS and the enhanced ALNS are each independently executed 10 times for every instance, with a time limit of 120 s per run and the same set of random seeds. The internal queue-scheduling procedure terminates when the maximum change in charging start times between two consecutive scheduling iterations is no greater than 10^{-6} , or when 40 queue-scheduling iterations have been completed. For the enhanced ALNS, $\text{Time}_{\text{Avg.}}$ denotes the average elapsed time required to first reach the final best solution of each run. All three methods use the same instance data, objective-function coefficients, charging policy, and vehicle-use rules.

As shown in Table 8 and Figure 6, Gurobi proves the optimal solution of 209.093 for the 10-customer instance within 5.51 s, while the enhanced ALNS obtains the same best and mean objective values over 10 independent runs. For the 15- and 20-customer instances, Gurobi reaches the time limit with feasible incumbent solutions and relative MIP gaps of 15.00% and 15.65%, respectively. The best objective values obtained by the enhanced ALNS are 0.39% and 4.84% lower than the corresponding Gurobi incumbents. These results show that the enhanced ALNS identifies better feasible solutions within the tested computational budgets. However, the Gurobi lower bounds remain below the ALNS solutions, and global optimality is not proven for these two instances. Therefore, the results should not be interpreted as the heuristic outperforming proven optimal solutions. For the 30-, 50-, and 100-customer instances, Gurobi does not obtain a feasible incumbent within the specified time limits, whereas the enhanced ALNS consistently generates feasible solutions.

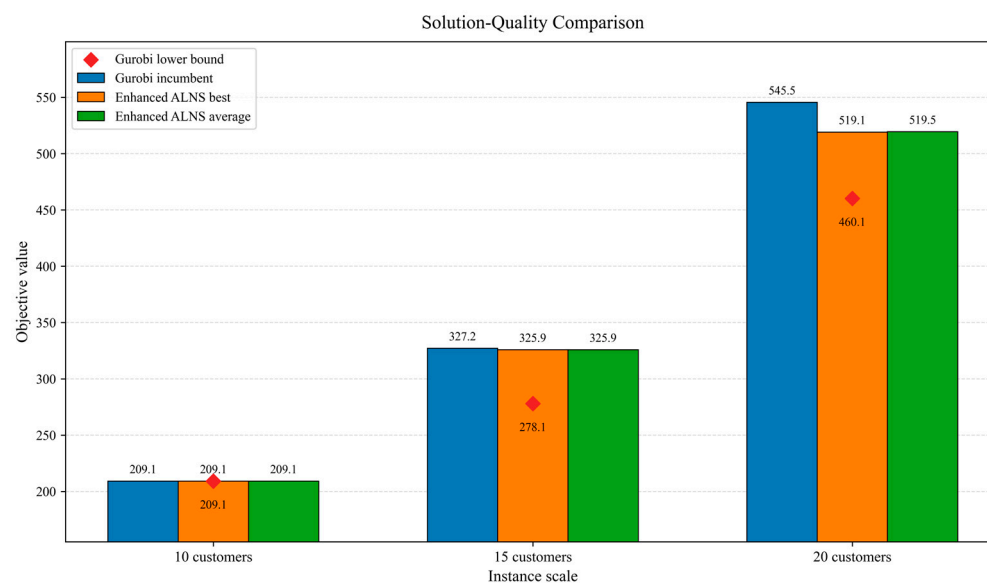


Figure 6. Solution-quality comparison between Gurobi and the enhanced ALNS for the 10-, 15-, and 20-customer instances.

Table 8. Comparison of Gurobi and the enhanced ALNS on instances of different sizes.

Instance	Gurobi					Enhanced ALNS				
	Z_{Gurobi}	LB^G	Time	Gap ^G (%)	Status	Z_{ALNS}^2 (Best)	Time _{Avg.}	Z_{ALNS}^2 (Avg.)	Δ_4 (%)	Δ_5 (%)
10c	209.093	209.093	5.51	0.00%	OPT	209.093	0.71	209.093	0.00%	0.00%
15c	327.201	278.107	300	15.00%	TL-FI	325.914	15.48	325.914	0.39%	0.01%
20c	545.514	460.138	600	15.65%	TL-FI	519.120	43.56	519.467	4.84%	0.07%
30c	N/A	601.232	1800	-	TL-NI	742.315	63.44	748.397	-	0.82%
50c	N/A	N/A	3600	-	TL-NI	1179.937	83.57	1190.420	-	0.89%
100c	N/A	N/A	3600	-	TL-NI	2433.434	95.86	2482.065	-	2.00%

Note: Z_{Gurobi} : the best feasible incumbent objective value obtained by Gurobi at termination. LB^G : the best lower bound established by Gurobi. Gap^G: the relative MIP gap at termination. Status: OPT indicates that global optimality was proven; TL-FI indicates that the time limit was reached with a feasible incumbent; and TL-NI indicates that the time limit was reached without a feasible incumbent. Z_{ALNS}^2 (best) and Z_{ALNS}^2 (Avg.): the best and average objective value obtained by the enhanced ALNS over 10 independent runs. Δ_4 : the relative improvement of the best enhanced ALNS solution over the Gurobi incumbent. Δ_5 : the gap between the Z_{ALNS}^2 (best) and Z_{ALNS}^2 (Avg.).

The computational-time comparison in Figure 7 further shows that the mean time required by the enhanced ALNS to first reach the final best solution of each run increases with the instance size, from 0.71 s for the 10-customer instance to 95.86 s for the 100-customer instance. For the 15- and 20-customer instances, the corresponding mean times are 15.48 and 43.56 s, which are substantially shorter than the Gurobi termination times of 300 and 600 s. Even for the 30-, 50-, and 100-customer instances, the enhanced ALNS reaches its run-specific final best solutions in an average of 63.44, 83.57, and 95.86 s, respectively, while Gurobi terminates without a feasible incumbent after much longer computation. In all tested instances, Time_{Avg.} remains below the 120-s ALNS time limit, indicating that the enhanced ALNS can obtain its final run-specific best solutions within the prescribed computational budget. From a practical planning perspective, these results indicate that the enhanced ALNS can generate feasible route, charging-station visit, and charging-scheduling decisions within a short decision window. This feature is particularly useful for medium- and large-scale urban delivery networks, where exact optimization may not return a feasible incumbent within the available planning time.

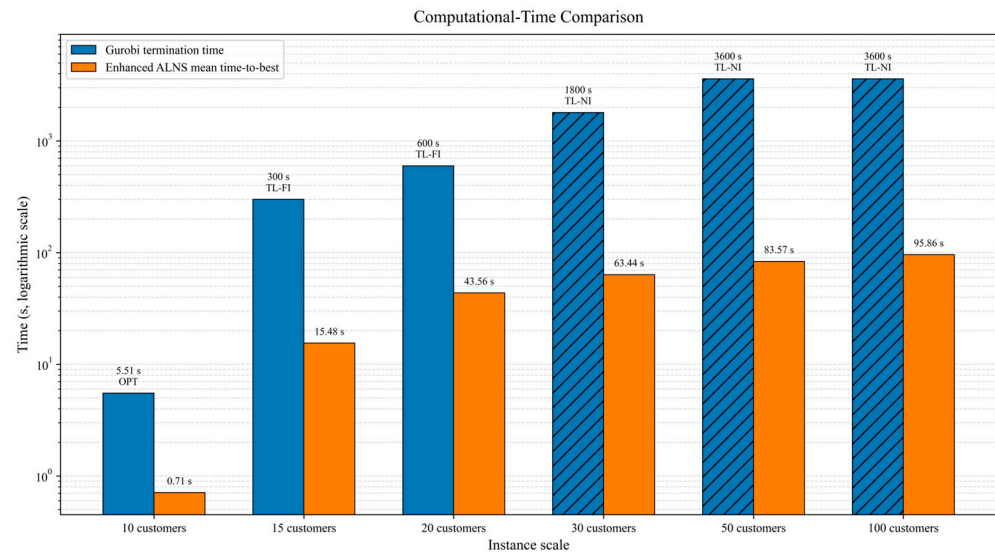


Figure 7. Computational-time comparison between Gurobi and the enhanced ALNS.

It should be noted that the baseline ALNS used in this study is not a direct reproduction of a conventional algorithm from a specific previous study. Instead, it is constructed as a benchmark version under the same solution representation, charging-insertion procedure, queue-scheduling mechanism, and evaluation framework as the enhanced ALNS. As shown in Table 9 and Figure 8, the two ALNS variants obtain the same best objective values for the 10-, 15-, 20-, and 30-customer instances. However, the enhanced ALNS achieves lower mean objective values for five of the six instance sizes, with relative improvements ranging from 0.37% to 2.49%. The largest mean improvement is observed for the 30-customer instance, while no difference is found for the 15-customer instance. For the 50- and 100-customer instances, the enhanced ALNS also obtains lower best and mean objective values. Overall, the enhancement strategies provide moderate but generally positive improvements across the tested instances.

Table 9. Comparison between the baseline ALNS and the enhanced ALNS over 10 independent runs.

Instance	Baseline ALNS			Enhanced ALNS			Δ_6 (%)
	Z_{ALNS}^1 (Best)	Z_{ALNS}^1 (Avg.)	Std^1	Z_{ALNS}^2 (Best)	Z_{ALNS}^2 (Avg.)	Std^2	
10c	209.093	213.543	6.056	209.093	209.093	0.000	2.08%
15c	325.914	325.914	0.000	325.914	325.914	0.000	0.00%
20c	519.120	525.397	8.989	519.120	519.467	1.095	1.13%
30c	742.315	767.505	9.772	742.315	748.397	10.119	2.49%
50c	1187.902	1194.850	10.666	1179.937	1190.420	8.024	0.37%
100c	2442.768	2508.806	30.536	2433.434	2482.065	21.718	1.07%

Note: Z_{ALNS}^1 (best) and Z_{ALNS}^1 (Avg.): the best and average objective value obtained by the baseline ALNS over 10 independent runs. Z_{ALNS}^2 (best) and Z_{ALNS}^2 (Avg.): the best and average objective value obtained by the enhanced ALNS over 10 independent runs. Std^1 and Std^2 : the standard deviation obtained by the baseline ALNS and enhanced ALNS over 10 independent runs, respectively. Δ_6 : the relative improvement of the enhanced ALNS over the baseline ALNS based on the mean objective values.

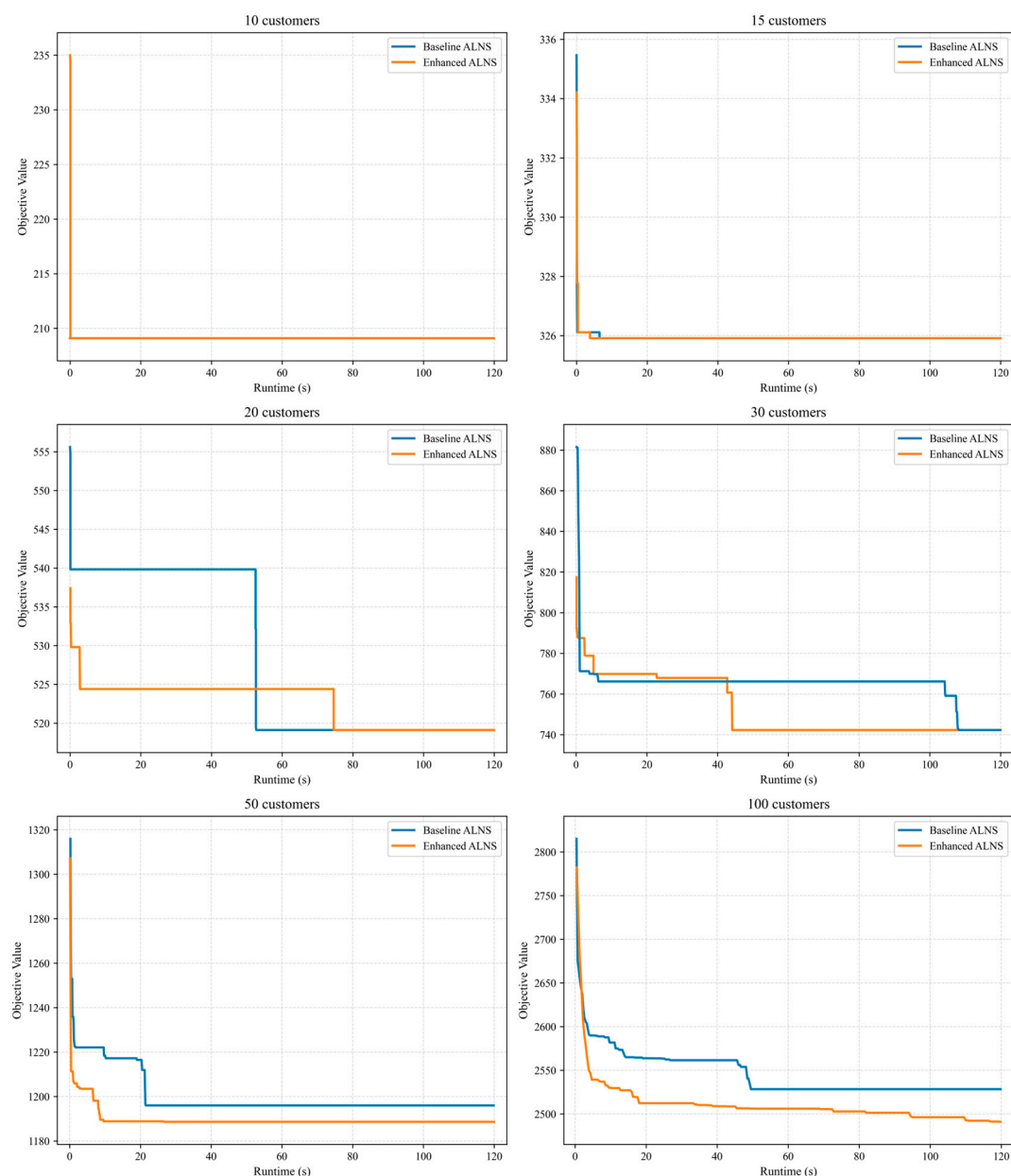


Figure 8. Convergence comparison between the baseline ALNS and the enhanced ALNS on instances of different sizes.

The convergence curves in Figure 8 further show that both algorithms converge rapidly for the 10- and 15-customer instances. For the 20-customer instance, the representative baseline run reaches the common best objective value earlier, although the enhanced ALNS achieves a substantially lower mean objective value and standard deviation over the 10 independent runs. For the 30-customer instance, the enhanced ALNS reaches the common best objective value considerably earlier than the baseline version. For the 50- and 100-customer instances, the convergence curve of the enhanced ALNS remains generally below that of the baseline ALNS and terminates at a lower objective value. The enhanced ALNS also exhibits lower standard deviations for the 10-, 20-, 50-, and 100-customer instances, the same standard deviation for the 15-customer instance, and a slightly higher standard deviation for the 30-customer instance. These results indicate that the enhanced ALNS generally provides better mean performance and a more favorable convergence trend, although the magnitude of improvement remains instance-dependent.

5.5. Sensitivity Analysis

5.5.1. Sensitivity to the Number of Chargers

To examine the effect of charging-station capacity on system performance, the number of chargers at each real charging station is varied from one to three, while all other instance and algorithm settings remain unchanged. As shown in Table 10 and Figure 9, increasing the number of chargers generally reduces the total, average, and maximum queue waiting times, as well as the overall objective value. These results demonstrate that additional charging capacity can effectively alleviate station-level congestion and improve system performance.

Table 10. Performance indicators of the enhanced ALNS under different numbers of chargers.

Instance	c_p	Total Chargers	Z_{ALNS}^2	Total Distance	Used Vehicles	ΣW	$\bar{W}$	$W_{\max}$
10c	1	2	209.093	197.139	2	0.651	0.326	0.651
15c	1	2	325.914	305.291	3	2.249	0.750	1.500
20c	1	3	519.120	483.687	4	5.664	0.944	2.832
30c	1	4	742.315	690.749	5	10.330	1.291	4.132
50c	1	6	1185.563	1076.939	9	28.138	2.345	6.253
100c	1	10	2501.987	2162.178	18	117.690	4.708	13.077
10c	2	4	207.790	197.139	2	0.000	0.000	0.000
15c	2	4	322.915	305.291	3	0.750	0.250	0.750
20c	2	6	516.859	488.996	4	1.888	0.315	0.944
30c	2	8	729.919	690.749	5	4.132	0.516	2.066
50c	2	12	1154.413	1077.039	9	12.506	1.042	3.126
100c	2	20	2368.070	2169.887	18	55.384	2.130	6.154
10c	3	6	207.790	197.139	2	0.000	0.000	0.000
15c	3	6	321.415	305.291	3	0.000	0.000	0.000
20c	3	9	514.971	488.996	4	0.944	0.157	0.944
30c	3	12	725.788	690.749	5	2.006	0.258	1.033
50c	3	18	1143.795	1077.384	9	7.034	0.586	1.563
100c	3	30	2334.286	2160.263	18	34.615	1.385	3.846

Note: c_p : the number of chargers per real charging station. Z_{ALNS}^2 : the objective value obtained by the enhanced ALNS. ΣW : the total queue waiting time at charging stations. $\bar{W}$: the average queue waiting time. $W_{\max}$: the maximum queue waiting time.

A closer examination of Table 10 shows that, for the 10-customer instance, increasing the number of chargers from one to two completely eliminates queue waiting, while adding a third charger provides no further improvement. For the 15-customer instance, queue waiting is eliminated when three chargers are available at each real station. For the 20-customer instance, the total queue waiting time decreases from 5.664 under the one-charger configuration to 0.944 under the three-charger configuration. The effect of capacity expansion becomes more pronounced in larger instances. For the 50- and 100-customer instances, the total queue waiting time decreases from 28.138 and 117.690 to 7.034 and 34.615, respectively, while the corresponding objective values decrease from 1185.563 and 2501.987 to 1143.795 and 2334.286. For all tested instance sizes, the improvement obtained by increasing the number of chargers from one to two is greater than that obtained by increasing it from two to three, indicating diminishing marginal returns from capacity expansion. Therefore, charger deployment should be coordinated with route planning and charging scheduling rather than relying solely on infrastructure expansion.

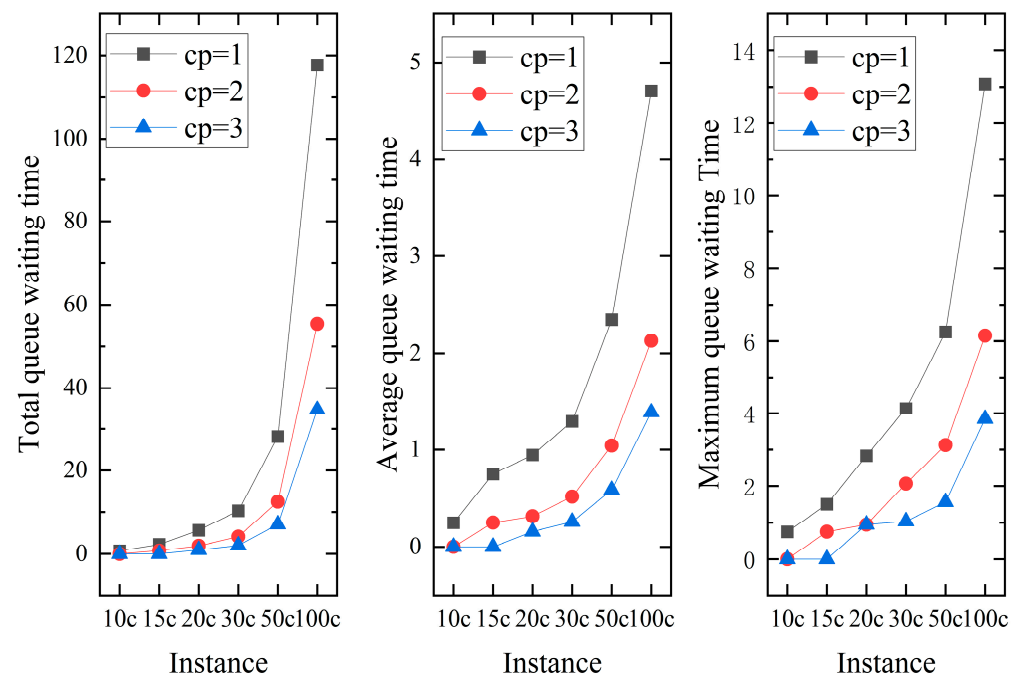


Figure 9. Total, average, and maximum queue waiting times under different numbers of chargers.

5.5.2. Sensitivity to the Charging Rate

After examining the effect of the number of chargers, the charging rate is varied to evaluate its influence on system performance. The customer distribution, charging-station layout, vehicle parameters, cost coefficients, and all other experimental settings remain unchanged. Let ρ_0 denote the baseline charging rate specified in the original instance. Three charging-rate levels are considered: $0.8\rho_0$, $1.0\rho_0$, and $1.2\rho_0$. The resulting objective values, charging durations, and queue waiting times are reported in Table 11 and Figure 10.

As shown in Table 11 and Figure 10, increasing the charging rate consistently reduces the total charging duration and the total, average, and maximum queue waiting times across all tested instance sizes. The objective value also decreases as the charging rate increases. In particular, the charging- and queue-related indicators exhibit a clear downward trend. A higher charging rate reduces charger occupation time, enabling subsequent vehicles to begin charging earlier and thereby decreasing queue waiting.

The absolute effect becomes more pronounced as the instance size increases. Specifically, for the 100-customer instance, increasing the charging rate from $0.8\rho_0$ to $1.2\rho_0$ reduces the total queue waiting time from 130.767 to 98.075 and the total charging duration from 35.177 to 24.237. The corresponding objective value decreases from 2501.822 to 2450.746. These results indicate that improving charging efficiency can enhance system performance, particularly in large-scale delivery networks where charging demand is concentrated and multiple vehicles compete for limited charging resources. Accordingly, fast-charging upgrades should be prioritized at stations with concentrated charging demand and frequent temporal overlap among vehicle arrivals, rather than being uniformly implemented across the entire network.

Table 11. Performance indicators of the enhanced ALNS under different charging rates.

Instance	ρ_p	n	Z_{ALNS}^2	Total Distance	Used Vehicles	$\sum H_s^k$	$\sum W$	$\bar{W}$	$W_{\max}$
10c	$0.8\rho_0$	3	209.582	197.139	2	1.629	0.814	0.407	0.814
15c	$0.8\rho_0$	3	327.320	305.291	3	2.812	2.812	0.937	1.874

20c	$0.8\rho_0$	6	522.978	483.687	4	10.263	7.080	1.180	3.540
30c	$0.8\rho_0$	9	772.500	713.091	5	17.169	12.912	1.435	5.165
50c	$0.8\rho_0$	12	1201.906	1077.384	9	18.357	35.172	2.931	7.816
100c	$0.8\rho_0$	25	2501.822	2160.700	18	35.177	130.767	5.449	15.384
10c	$1.0\rho_0$	3	209.093	197.139	2	1.303	0.651	0.326	0.651
15c	$1.0\rho_0$	3	325.914	305.291	3	2.249	2.249	0.750	1.500
20c	$1.0\rho_0$	6	519.120	488.996	4	8.210	5.664	0.944	2.832
30c	$1.0\rho_0$	9	759.147	706.732	5	13.510	10.330	1.148	4.132
50c	$1.0\rho_0$	12	1186.002	1077.384	9	14.686	28.138	2.345	6.253
100c	$1.0\rho_0$	25	2496.726	2156.946	18	28.799	117.690	4.708	13.077
10c	$1.2\rho_0$	3	208.768	197.139	2	1.086	0.543	0.271	0.543
15c	$1.2\rho_0$	3	324.977	305.291	3	1.874	1.874	0.625	1.250
20c	$1.2\rho_0$	6	516.548	488.996	4	6.842	4.720	0.787	2.360
30c	$1.2\rho_0$	8	737.888	690.749	5	9.844	8.608	1.076	3.443
50c	$1.2\rho_0$	12	1175.399	1077.384	9	12.238	23.448	1.954	5.211
100c	$1.2\rho_0$	25	2450.746	2152.477	18	24.237	98.075	3.923	10.897

Note: ρ_p : the charging rate. n : the number of charging visits. Z_{ALNS}^2 : the objective value obtained by the enhanced ALNS. $\sum H_s^k$: the total charging duration at charging stations. $\sum W$: the total queue waiting time at charging stations. $\bar{W}$: the average queue waiting time. $W_{\max}$: the maximum queue waiting time.

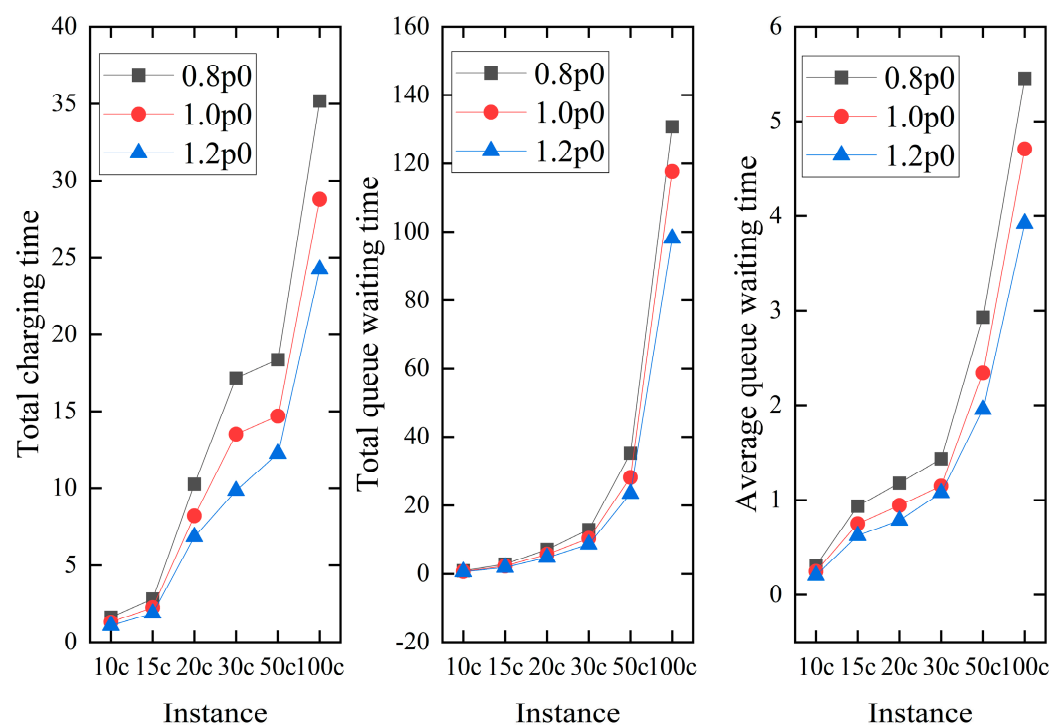


Figure 10. Total charging time, total queue waiting time, and average queue waiting time under different charging rates.

6. Conclusions

This study investigates an electric vehicle routing problem with capacitated charging stations and queueing delays. A mixed-integer linear programming model is developed to jointly represent vehicle routing, battery-energy evolution, charging operations, charger assignment, and temporal conflicts among vehicles visiting the same real charging station. Virtual charging-station nodes are used to represent potential charging visits, while the corresponding real stations determine charger capacity and queue interactions.

This representation enables charging events associated with the same real station to be coordinated through assignment and temporal non-overlap constraints.

An enhanced adaptive large neighborhood search algorithm is developed to solve the resulting problem. The algorithm integrates customer-route search, charging-station insertion, queue-scheduling evaluation, 2-opt local search, stagnation restart, and cross-route perturbation. The computational results show that the enhanced ALNS matches the proven optimal solution for the 10-customer instance. For the 15- and 20-customer instances, it produces better feasible solutions than the time-limited Gurobi incumbents. For the 30-, 50-, and 100-customer instances, the enhanced ALNS consistently generates feasible solutions within the prescribed computational budget, whereas Gurobi does not obtain a feasible incumbent within the specified time limits. Compared with the baseline ALNS, the enhanced version generally achieves lower mean objective values and more favorable convergence behavior, although the magnitude of improvement remains instance-dependent.

The sensitivity analysis provides several practical implications for electric logistics operations. Limited charger availability can generate substantial queue waiting times, particularly in larger delivery networks. Increasing the number of chargers can alleviate congestion, although each additional charger generally produces a smaller reduction in waiting time. Improving the charging rate also shortens charger occupation time and reduces queueing delays. Therefore, charger-capacity expansion, fast-charging investment, and queue-aware route scheduling should be considered jointly. Infrastructure investment should be prioritized at heavily utilized stations rather than applied uniformly across the network. Within limited planning horizons, the enhanced ALNS can provide feasible routing, charging-station visit, and charging-scheduling decisions for medium- and large-scale systems, thereby supporting more reliable and sustainable urban delivery operations.

Several limitations remain. The current model assumes homogeneous vehicles, deterministic travel and charging conditions, linear charging, and full recharging at every charging visit. Although the full-charging policy simplifies charging coordination and may be applicable to fleets that follow standardized charging rules, it does not capture the operational flexibility of partial charging. In particular, restoring the battery to full capacity at every visit may overestimate charging duration and cost and may exclude potentially more efficient routing and charging decisions. Future research should therefore introduce the charging amount as an explicit decision variable and systematically compare full- and partial-charging policies in terms of route feasibility, charging time, queueing delay, and total operating cost. Further extensions may incorporate nonlinear charging functions, heterogeneous vehicle fleets, time-dependent traffic conditions, uncertain vehicle arrivals, charger-type heterogeneity, and reservation or first-come-first-served operating rules. Testing the proposed approach with real operational data would also help evaluate its practical applicability.

Author Contributions: Conceptualization, Z.H., S.Z. and Y.W.; methodology, Z.H.; software, Z.H.; validation, Z.H. and S.Z.; formal analysis, Z.H.; investigation, Z.H.; resources, S.Z.; data curation, Z.H.; writing—original draft preparation, Z.H.; writing—review and editing, Z.H.; visualization, Z.H.; supervision, S.Z.; project administration, S.Z., Y.W.; funding acquisition, Z.H., S.Z. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Humanities and Social Science Foundation of the Ministry of Education of China, grant number 24YJCZH476, and the Natural Science Foundation of Nantong City, grant number JC2024031.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study were generated by the authors for the computational experiments. The processed instances and computational results are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EV	Electric vehicle
EVRP	Electric vehicle routing problem
VRP	Vehicle routing problem
CVRP	Capacitated vehicle routing problem
VRPTW	Vehicle routing problem with time windows
EVRPTW	Electric vehicle routing problem with time windows
MDVRP	Multi-depot vehicle routing problem
OVRP	Open vehicle routing problem
PDVRP	Pickup and delivery vehicle routing problem
SVRP	Stochastic vehicle routing problem
DVRP	Dynamic vehicle routing problem
MOVRP	Multi-objective vehicle routing problem
CS	Charging-station capacity
TW	Time window
MIP	Mixed-integer programming
MILP	Mixed-integer linear programming
MINLP	Mixed-integer nonlinear programming
ALNS	Adaptive large neighborhood search
VNS	Variable neighborhood search
TS	Tabu search
GA	Genetic algorithm
IACA	Improved ant colony algorithm
SA	Simulated annealing
Gurobi	Gurobi optimizer
CPLEX	CPLEX optimizer

References

1. Huang, Y. A review on the electric vehicle routing problem and its variations. *Front. Soc. Sci. Technol.* **2023**, *5*, 85–90. <https://doi.org/10.25236/FSST.2023.050514>.
2. Li, F.; Zhao, Z.; Li, F.; Xie, W. Carbon reduction effects of electric delivery vehicle adoption in Chinese cities: Gradient pattern and scenario analysis. *Energy* **2024**, *309*, 133119. <https://doi.org/10.1016/j.energy.2024.133119>.
3. Dadashzadeh, I.; Garza-Reyes, J.A.; Roy Ghatak, R.; Gonzalez-Aleu, F. An investigation into the major barriers to the adoption of electric vehicles in last-mile deliveries for sustainable transport. *Int. J. Logist. Manag.* **2025**, *36*, 906–938. <https://doi.org/10.1108/IJLM-01-2024-0007>.
4. Lu, Y.; Zhang, R.; Wang, R.; Fu, B.; Lu, K.; Fan, P.; Yu, L.; Song, G. Trajectory data-driven modeling and optimization of urban freight charging networks. *Transp. Res. Part D Transp. Environ.* **2026**, *154*, 105288. <https://doi.org/10.1016/j.trd.2026.105288>.
5. Kucukoglu, I.; Dewil, R.; Cattrysse, D. The electric vehicle routing problem and its variations: A literature review. *Comput. Ind. Eng.* **2021**, *161*, 107650. <https://doi.org/10.1016/j.cie.2021.107650>.
6. Conrad, R.G.; Figliozzi, M.A. The recharging vehicle routing problem. In Proceedings of the 2011 Industrial Engineering Research Conference, Reno, NV, USA, 21–25 May 2011.
7. Erdoğan, S.; Miller-Hooks, E. A green vehicle routing problem. *Transp. Res. Part E Logist. Transp. Rev.* **2012**, *48*, 100–114. <https://doi.org/10.1016/j.tre.2011.08.001>.
8. Schneider, M.; Stenger, A.; Goeke, D. The electric vehicle-routing problem with time windows and recharging stations. *Transp. Sci.* **2014**, *48*, 500–520. <https://doi.org/10.1287/trsc.2013.0490>.

9. Montoya, A.; Guéret, C.; Mendoza, J.E.; Villegas, J.G. The electric vehicle routing problem with nonlinear charging function. *Transp. Res. Part B Methodol.* **2017**, *103*, 87–110. <https://doi.org/10.1016/j.trb.2017.02.004>.
10. Shao, S.; Guan, W.; Ran, B.; He, Z.; Bi, J. Electric vehicle routing problem with charging time and variable travel time. *Math. Probl. Eng.* **2017**, *2017*, 5098183. <https://doi.org/10.1155/2017/5098183>.
11. Keskin, M.; Laporte, G.; Çatay, B. Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Comput. Oper. Res.* **2019**, *107*, 77–94. <https://doi.org/10.1016/j.cor.2019.02.014>.
12. Hanni, U.E.; Yamamoto, T.; Nakamura, T. Modeling of the acceptable waiting time for EV charging in Japan. *Sustainability* **2024**, *16*, 2536. <https://doi.org/10.3390/su16062536>.
13. Kullman, N.D.; Goodson, J.C.; Mendoza, J.E. Electric vehicle routing with public charging stations. *Transp. Sci.* **2021**, *55*, 637–659. <https://doi.org/10.1287/trsc.2020.1018>.
14. Lam, E.; Desaulniers, G.; Stuckey, P.J. Branch-and-cut-and-price for the electric vehicle routing problem with time windows, piecewise-linear recharging and capacitated recharging stations. *Comput. Oper. Res.* **2022**, *145*, 105870. <https://doi.org/10.1016/j.cor.2022.105870>.
15. Desaulniers, G.; Errico, F.; Irnich, S.; Schneider, M. Exact algorithms for electric vehicle-routing problems with time windows. *Oper. Res.* **2016**, *64*, 1388–1405. <https://doi.org/10.1287/opre.2016.1535>.
16. Duman, E.N.; Taş, D.; Çatay, B. Branch-and-price-and-cut methods for the electric vehicle routing problem with time windows. *Int. J. Prod. Res.* **2022**, *60*, 5332–5353. <https://doi.org/10.1080/00207543.2021.1955995>.
17. Lera-Romero, G.; Bront, J.J.M.; Soullignac, F.J. A branch-cut-and-price algorithm for the time-dependent electric vehicle routing problem with time windows. *Eur. J. Oper. Res.* **2024**, *312*, 978–995. <https://doi.org/10.1016/j.ejor.2023.06.037>.
18. Keskin, M.; Çatay, B. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transp. Res. Part C Emerg. Technol.* **2016**, *65*, 111–127. <https://doi.org/10.1016/j.trc.2016.01.013>.
19. Keskin, M.; Çatay, B.; Laporte, G. A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations. *Comput. Oper. Res.* **2021**, *125*, 105060. <https://doi.org/10.1016/j.cor.2020.105060>.
20. Rastani, S.; Çatay, B. A large neighborhood search-based matheuristic for the load-dependent electric vehicle routing problem with time windows. *Ann. Oper. Res.* **2023**, *324*, 761–793. <https://doi.org/10.1007/s10479-021-04320-9>.
21. Duan, Y.; Hu, Y.; Wu, P. An adaptive large neighborhood search heuristic for the electric vehicle routing problems with time windows and recharging strategies. *J. Adv. Transp.* **2023**, *2023*, 1200526. <https://doi.org/10.1155/2023/1200526>.
22. Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. <https://doi.org/10.1287/mnsc.6.1.80>.
23. Toth, P.; Vigo, D. (Eds.) *The Vehicle Routing Problem*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2002. <https://doi.org/10.1137/1.9780898718515>.
24. Solomon, M.M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **1987**, *35*, 254–265. <https://doi.org/10.1287/opre.35.2.254>.
25. Cordeau, J.F.; Gendreau, M.; Laporte, G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* **1997**, *30*, 105–119. [https://doi.org/10.1002/\(SICI\)1097-0037\(199709\)30:2<105::AID-NET5>3.0.CO;2-G](https://doi.org/10.1002/(SICI)1097-0037(199709)30:2<105::AID-NET5>3.0.CO;2-G).
26. Li, F.; Golden, B.; Wasil, E. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Comput. Oper. Res.* **2007**, *34*, 2918–2930. <https://doi.org/10.1016/j.cor.2005.11.018>.
27. Savelsbergh, M.W.P.; Sol, M. The general pickup and delivery problem. *Transp. Sci.* **1995**, *29*, 17–29. <https://doi.org/10.1287/trsc.29.1.17>.
28. Gendreau, M.; Laporte, G.; Séguin, R. Stochastic vehicle routing. *Eur. J. Oper. Res.* **1996**, *88*, 3–12. [https://doi.org/10.1016/0377-2217\(95\)00050-X](https://doi.org/10.1016/0377-2217(95)00050-X).
29. Pillac, V.; Gendreau, M.; Guéret, C.; Medaglia, A.L. A review of dynamic vehicle routing problems. *Eur. J. Oper. Res.* **2013**, *225*, 1–11. <https://doi.org/10.1016/j.ejor.2012.08.015>.
30. Jozefowicz, N.; Semet, F.; Talbi, E.G. Multi-objective vehicle routing problems. *Eur. J. Oper. Res.* **2008**, *189*, 293–309. <https://doi.org/10.1016/j.ejor.2007.05.055>.
31. Wang, W.; Zhao, J. Partial linear recharging strategy for the electric fleet size and mix vehicle routing problem with time windows and recharging stations. *Eur. J. Oper. Res.* **2023**, *308*, 929–948. <https://doi.org/10.1016/j.ejor.2022.12.011>.
32. Kim, Y.J.; Chung, B.D. Energy consumption optimization for the electric vehicle routing problem with state-of-charge-dependent discharging rates. *J. Clean. Prod.* **2023**, *385*, 135703. <https://doi.org/10.1016/j.jclepro.2022.135703>.
33. Xiong, H.; Xu, Y.; Yan, H.; Guo, H.; Zhang, C. Optimizing electric vehicle routing under traffic congestion: A comprehensive energy consumption model considering drivetrain losses. *Comput. Oper. Res.* **2024**, *168*, 106710. <https://doi.org/10.1016/j.cor.2024.106710>.

34. Klein, P.S.; Schiffer, M. Electric vehicle charge scheduling with flexible service operations. *Transp. Sci.* **2023**, *57*, 1605–1626. <https://doi.org/10.1287/trsc.2022.0272>.
35. Fan, L.; Liu, C.; Dai, B.; Li, J.; Wu, Z.; Guo, Y. Electric vehicle routing problem considering energy differences of charging stations. *J. Clean. Prod.* **2023**, *418*, 138184. <https://doi.org/10.1016/j.jclepro.2023.138184>.
36. Li, Y.; Lim, M.K.; Xiong, W.; Huang, X.; Shi, Y.; Wang, S. An electric vehicle routing model with charging stations consideration for sustainable logistics. *Ind. Manag. Data Syst.* **2024**, *124*, 1076–1106. <https://doi.org/10.1108/IMDS-08-2023-0581>.
37. Kim, G. Electric vehicle routing problem with states of charging stations. *Sustainability* **2024**, *16*, 3439. <https://doi.org/10.3390/su16083439>.
38. Wang, Y.; Zhou, J.; Sun, Y.; Fan, J.; Wang, Z.; Wang, H. Collaborative multidepot electric vehicle routing problem with time windows and shared charging stations. *Expert Syst. Appl.* **2023**, *219*, 119654. <https://doi.org/10.1016/j.eswa.2023.119654>.
39. Xiao, J.; Liu, X.; Liu, T.; Li, N.; Martinez-Sykora, A. The electric vehicle routing problem with synchronized mobile partial recharging and non-strict waiting strategy. *Ann. Oper. Res.* **2026**, *359*, 2189–2233. <https://doi.org/10.1007/s10479-024-06069-3>.
40. Huang, Y.; Liu, M.; Zhang, Y. Electric vehicle entire-trip navigation and charging reservation method based on a high-speed communication network. *Int. J. Electr. Power Energy Syst.* **2023**, *150*, 109070. <https://doi.org/10.1016/j.ijepes.2023.109070>.
41. Lee, E.H.; Lee, E. Electric vehicle charging station location selection using generative artificial intelligence. *Transp. Res. Part E Logist. Transp. Rev.* **2026**, *213*, 104930. <https://doi.org/10.1016/j.tre.2026.104930>.
42. Ecer, F.; Pamucar, D.; Demir, G. Decision-analytics-based electric vehicle charging station location selection: A cutting-edge fuzzy rough framework. *Energy Rep.* **2025**, *14*, 711–735. <https://doi.org/10.1016/j.egyr.2025.06.035>.
43. Froger, A.; Jabali, O.; Mendoza, J.E.; Laporte, G. The electric vehicle routing problem with capacitated charging stations. *Transp. Sci.* **2022**, *56*, 460–482. <https://doi.org/10.1287/trsc.2021.1111>.
44. Souza, A.L.S.; Papini, M.; Penna, P.H.V.; Souza, M.J. A flexible variable neighbourhood search algorithm for different variants of the Electric Vehicle Routing Problem. *Comput. Oper. Res.* **2024**, *168*, 106713. <https://doi.org/10.1016/j.cor.2024.106713>.
45. Jia, Y.H.; Mei, Y.; Zhang, M. Confidence-based ant colony optimization for capacitated electric vehicle routing problem with comparison of different encoding schemes. *IEEE Trans. Evol. Comput.* **2022**, *26*, 1394–1408. <https://doi.org/10.1109/TEVC.2022.3144142>.
46. Dong, J.; Wang, H.; Zhang, S. Dynamic electric vehicle routing problem considering mid-route recharging and new demand arrival using an enhanced memetic algorithm. *Sustain. Energy Technol. Assess.* **2023**, *58*, 103366. <https://doi.org/10.1016/j.seta.2023.103366>.
47. Comert, S.E.; Yazgan, H.R. A new approach based on hybrid ant colony optimization-artificial bee colony algorithm for multi-objective electric vehicle routing problems. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106375. <https://doi.org/10.1016/j.engappai.2023.106375>.
48. Ropke, S.; Pisinger, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **2006**, *40*, 455–472. <https://doi.org/10.1287/trsc.1050.0135>.
49. Ye, C.; He, W.; Chen, H. Electric vehicle routing models and solution algorithms in logistics distribution: A systematic review. *Environ. Sci. Pollut. Res.* **2022**, *29*, 57067–57090. <https://doi.org/10.1007/s11356-022-21559-2>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.