

Article

Multi-Depot Electric Vehicle–Drone Collaborative-Delivery Routing Optimization with Time-Varying Vehicle Travel Time

Yong Peng ¹, Wenjing Zhu ¹, Dennis Z. Yu ^{2,*}, Song Liu ¹ and Yali Zhang ¹

¹ School of Traffic and Transportation, Chongqing Jiaotong University, Chongqing 400074, China; pengyong@cqjtu.edu.cn (Y.P.); 622230950041@mails.cqjtu.edu.cn (W.Z.); song@cqjtu.edu.cn (S.L.); 622220110010@mails.cqjtu.edu.cn (Y.Z.)

² The David D. Reh School of Business, Clarkson University, Potsdam, NY 13699, USA

* Correspondence: dyu@clarkson.edu

Abstract: This paper presents an electric vehicle-drone (EV–drone) collaborative-delivery routing optimization model that leverages the time-varying characteristics of electric vehicles and drones across multiple distribution centers (i.e., central depots) to address the logistics industry’s low-carbon transformation in the last-mile delivery. The model aims to minimize total delivery costs by formulating a mixed-integer programming (MIP) model that accounts for essential constraints such as nonlinear charging time, time-varying EV travel time, delivery time window, payload capacity, and maximum range. An improved adaptive large-neighborhood search (ALNS) algorithm is developed to solve the model. Experimental results validate the effectiveness of the proposed algorithm and highlight the impact of EV and drone technology parameters, along with the time-varying EV travel times, on the economic efficiency of delivery distribution and route planning.

Keywords: low-carbon logistics; last-mile delivery; time-varying travel time; electric vehicle; drone; adaptive large-neighborhood search algorithm



Citation: Peng, Y.; Zhu, W.; Yu, D.Z.; Liu, S.; Zhang, Y. Multi-Depot Electric Vehicle–Drone Collaborative-Delivery Routing Optimization with Time-Varying Vehicle Travel Time. *Vehicles* **2024**, *6*, 1812–1842. <https://doi.org/10.3390/vehicles6040088>

Academic Editor: Patrick Gruber

Received: 25 September 2024

Revised: 24 October 2024

Accepted: 25 October 2024

Published: 27 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As the global logistics industry advances, last-mile delivery has become a central focus in optimizing logistics networks. Traditionally, this segment has relied heavily on manual labor, but with the rapid growth of e-commerce and express delivery services, such methods are increasingly unable to meet the rising demand. Companies like JD.com, SF Express, and Meituan are actively exploring drone technology to overcome these limitations. While drones hold promise, their current payload capacity and range limitations prevent them from independently supporting urban delivery networks. A more practical approach is the integration of vehicles and drones in collaborative-distribution strategies. Historically, research has focused on optimizing the traditional truck and drone model for such systems. However, as environmental concerns intensify and efforts to conserve energy and reduce emissions gain momentum, electric vehicles (EVs) emerge as a viable alternative to conventional delivery trucks for urban logistics. Although research on the “EV–drone” model is beginning to emerge, it remains limited, due to the various commercial challenges that EVs continue to face [1].

The objective of this study is to develop a sustainable, low-carbon urban delivery scheme that leverages electric vehicles (EVs) and drones to optimize delivery routes, reduce energy consumption, and minimize transportation costs. To achieve this, we propose a multi-depot EV–drone collaborative-delivery routing optimization model that incorporates practical constraints such as EVs’ nonlinear charging times, time-varying travel speeds, and drones’ payload and range limitations. The model is solved using an improved adaptive large-neighborhood search (ALNS) algorithm. This research aims to provide an efficient, cost-effective solution for EV–drone collaborative delivery, advancing the practical

application of green logistics in urban distribution, enhancing overall delivery efficiency, and supporting the low-carbon development goals of the modern logistics industry.

The remainder of this paper is structured as follows: Section 2 reviews the relevant literature on EV–drone collaborative-delivery route optimization. Section 3 outlines the proposed model, including the problem definition, mathematical formulation, and key assumptions. Section 4 describes the design and implementation of the improved adaptive large-neighborhood search (ALNS) algorithm. In Section 5, we present the experimental results and analysis, demonstrating the effectiveness and efficiency of the proposed algorithm and exploring the impact of time-varying factors on route optimization. Finally, Section 6 concludes the paper with key insights and suggestions for future research directions.

2. Literature Review

In recent years, electric vehicle (EV) distribution has gained significant attention in the fields of operations research and computer science, due to its environmental advantages. The electric vehicle routing problem (EVRP), a critical variant of the traditional vehicle routing problem (VRP), introduces essential considerations such as battery capacity constraints and charging operations [2]. In two foundational studies of the electric vehicle routing problem (EVRP), Conrad et al. [3] introduced the concept of incorporating charging stations into vehicle routes, requiring vehicles to recharge at customer nodes. Meanwhile, Erdogan et al. [4] proposed a model that replaced traditional fuel vehicles with alternative fuel vehicles. Since these pioneering works, many researchers have sought to deepen the study of the EVRP by continuously expanding the model to account for the specific operational characteristics of EVs within distribution routes. Schneider et al. [5] examined the EV routing problem with time-window constraints, focusing on minimizing the number of charging station visits. They proposed a hybrid variable neighborhood tabu-search heuristic and demonstrated its effectiveness in solving large-scale instances. Yang et al. [6] investigated the integration of battery swap stations into EV routes, highlighting the efficiency gains from rapid battery exchange despite the high initial costs of battery replacement infrastructure. Keskin et al. [7] further considered incomplete charging strategies during EV charging, relaxing the assumption of complete charging strategies in the study by Schneider et al. [5]. Desaulniers et al. [8] explored various EV distribution scenarios, comparing the effects of different charging strategies on route optimization. Paz et al. [9] addressed the multi-depot EV routing problem with time windows, incorporating both conventional charging and battery-swapping technologies into their decision-making model. Froger et al. [10] extended previous research by constructing a more realistic EVRP model, using a concave linear function to capture the nonlinear nature of charging time. Zhang et al. [11] developed a fuzzy optimization model based on plausibility theory, using fuzzy numbers to represent uncertainties in service time, energy consumption, and travel time across different regions and customer types. Lee [12] focused on optimizing both EV route mileage and charging time by considering a nonlinear charging-time function. Wang et al. [13] proposed a multi-depot EV routing model with time windows and shared charging stations and designed an improved multi-objective genetic algorithm with nested tabu search to optimize EV routes and charging station placement.

Murray and Chu [14] introduced the flying sidekick traveling salesman problem (FSTSP) to minimize completion time in a traditional traveling salesman problem (TSP) with the assistance of drones. Building on this work, Agatz et al. [15] proposed the drone traveling salesman problem (TSPD) as an extension of the FSTSP. Since then, numerous studies have aimed to expand upon the TSPD, with one of the most prominent variants being the vehicle routing problem with drones (VRPDs). The VRPDs integrates drone-assisted delivery into the classic vehicle routing problem (VRP). Wang et al. [16] were the first to conceptualize a model where a fleet of trucks carries one or more drones for deliveries. In this model, drones can be launched and retrieved from either a distribution center or customer nodes, with customers served by either trucks or drones exclusively. Sacramento et al. [17] developed a mixed-integer optimization model for the VRPDs, incorporating

vehicle load capacity and drone-endurance constraints. They designed an adaptive large neighborhood search (ALNS) algorithm to solve large-scale instances, demonstrating its effectiveness for the VRPDs. Pugliese et al. [18] further investigated the VRPDs considering the constraints of customers' time windows, and experimentally concluded that the introduction of drones would improve the delivery efficiency. Tamke et al. [19] developed an optimization model targeting different time orientations for drone flight mileage and speed constraints. Kuo et al. [20] proposed a mixed-integer programming model considering the customer time windows and designed an adapted variable neighborhood search algorithm, which proved to be a superior VRPD solution method in addition to the ALNS algorithm proposed by Sacramento et al. [17].

Although there has been progress in research on optimizing EV routing and drone-assisted vehicle delivery routing, the exploration of the synergy between EVs and drones remains in its early stages. Baek et al. [21] were the first to propose a collaborative-delivery system involving a single EV and a drone to reduce energy consumption and emissions. In their model, the drone is limited to serving one customer at a time, and its battery is non-replaceable, restricting its effectiveness to its range. Zhu et al. [22] extended this by incorporating scenarios where the EV must visit charging stations during delivery while allowing the drone to share energy with the EV. In their model, the EV's remaining power decreases each time the drone completes a delivery. They demonstrated that an adaptive large-neighborhood search algorithm outperformed a variable neighborhood search for solving the EV–drone collaborative-delivery problem. Kyriakakis et al. [23] introduced an MIP model that optimizes energy consumption for EV–drone collaborative delivery, solving it with a hybrid ant-colony optimization algorithm. However, in their approach, the EVs primarily served as platforms for launching and recovering drones, with all customers serviced by the drones. Building on this, Mara et al. [24] proposed a more comprehensive cooperative delivery model where EVs and drones deliver to customers simultaneously. In this model, the EVs must visit charging stations to ensure the feasibility of the delivery route.

Although significant progress has been made in studying the electric vehicle routing problem (EVRP) and the vehicle routing problem with drones (VRPDs), several research gaps remain in the exploration of EV–drone collaborative-delivery routing. First, many existing studies adopt a linear charging model for EVs, overlooking the fact that charging rates are influenced by external factors such as temperature, which limits the practical application of these models. Second, most research focuses on single distribution-center scenarios, with limited attention to resource sharing and collaborative optimization in multi-depot environments. Furthermore, studies on EV–drone collaboration often assume that EVs serve merely as landing platforms for drones, failing to fully leverage the delivery potential of EVs and overlooking the integration of charging operations with time window constraints.

In this study, we propose an innovative EV–drone collaborative-delivery routing optimization model that incorporates nonlinear EV charging times alongside several realistic constraints, such as time-varying vehicle travel times, customer time windows, and the payload and range limitations of drones. By solving this model, we offer practical solutions for EV–drone collaborative distribution, contributing to more efficient and sustainable logistics operations in urban environments.

3. Problem Description and Modeling Framework

3.1. Problem Description and Notation

Our proposed multi-depot EV–drone collaborative-delivery routing problem builds on the foundations of the electric vehicle routing problem with time windows (EVRPTWs), as introduced by Schneider et al. [5], and the electric vehicle–drone routing problem (EVDRP), as developed by Mara et al. [24]. In this study, we refer to our model as MD-EVDRPTW, as it incorporates both multiple distribution centers and customers' time windows. Additionally, our model accounts for time-varying EV travel times due to the fluctuating speeds of EVs

in a time-dependent road network. These travel times are modeled based on the work of Ichoua et al. [25]. Our modeling environment involves multiple distribution centers, customer nodes, and electric vehicles (EVs) equipped with drones of identical specifications. In particular, customer nodes are classified into two categories: those that can only be served by drones and those that can be served by either drones or electric vehicles. The EVs transport drones from the distribution centers to deliver customer orders within specific time windows. While the EV performs its delivery at one customer node, the onboard drone can serve another customer, simultaneously. However, each customer is served by either the EV or the drone, but not both. After the drone is deployed at a customer node, it completes its delivery independently and later reunites with the EV at a subsequent node. Once all deliveries are completed, the EV and the drone may return to the original or an alternative distribution center. The overall delivery process is illustrated in Figure 1.

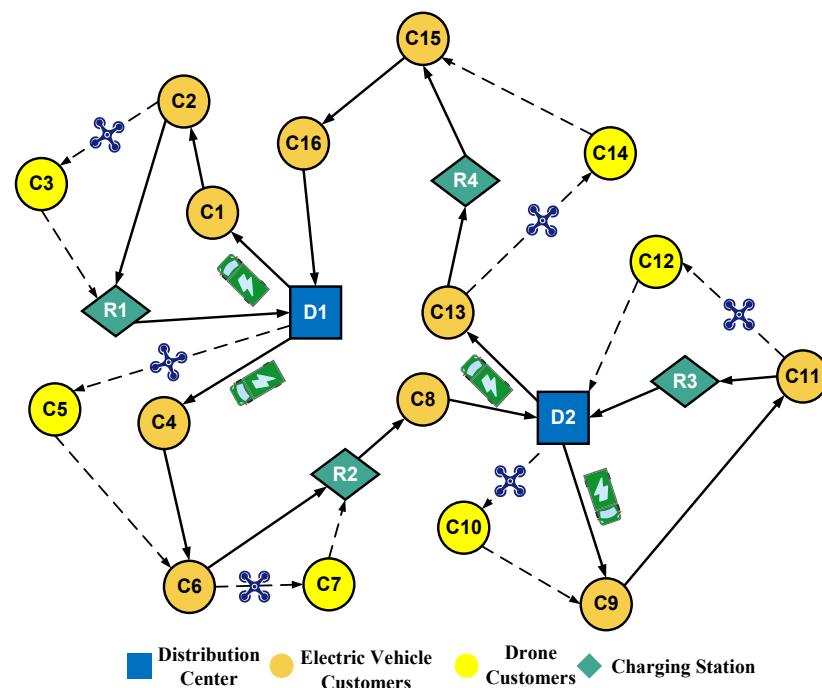


Figure 1. Diagram of multi-depot EV–drone collaborative delivery.

During the distribution process, EVs must visit charging stations to ensure they have sufficient power for continued operation. With the inclusion of charging stations, two additional factors must be considered: (i) the EV charging process is nonlinear, due to variables such as battery performance and heating effects (as described by Montoya et al. [26]); and (ii), while the EV is charging, the station can be treated as a special customer node, without delivery demand or time window constraints, and can be used as a launch and recovery point for drones as part of the drone deployment and retrieval strategy.

The following assumptions are made for the proposed MD-EVDRPTW model:

- Each EV is equipped with a drone, and while the delivery process begins and ends at a distribution center, the EV and the drone are allowed to return to different distribution centers.
- The EV departs fully charged, and the energy consumption while servicing customer nodes is ignored.
- The drone always takes off fully charged and operates in an instant power-swap mode.
- The drone must be launched and recovered from the same EV, and these operations can be performed at distribution centers, customer nodes, or charging stations.
- All charging stations have standardized equipment and provide full charges, but can only serve a limited number of EVs at any time.

- Each distribution center has sufficient capacity to meet all customer demands, and the centers are homogeneous.
- A tuple $\langle i, j, k \rangle$ is used to represent a drone's delivery route, in which the first node represents the drone's launch node, the second node j represents the customer node served by the drone, and the third node k represents the drone's recovery node.

The definitions of symbols and variables are provided in Table 1.

Table 1. Definitions of symbols and variables.

	Symbol	Definition
Sets	m, n, r	Number of distribution centers, number of customer nodes, number of charging stations, respectively
	V	Set of all nodes, $V = V_0 \cup V_S \cup V_C \cup V_0^+$
	V_0	Set of departure distribution centers, $V_0 = \{0, 1, \dots, m\}$
	V_C	Set of customer nodes, $V_C = \{m+1, \dots, m+n\}$
	V_S	Set of charging stations, $V_S = \{m+n+1, \dots, m+n+r\}$
	V_0^+	Set of arrival distribution centers, $V_0^+ = \{m+n+r+1, \dots, 2m+n+r+1\}$
	V_D	Set of drone customers, $V_D \subset V_C$
	V_L	Set of drone launch nodes, $V_L = V_0 \cup V_S \cup V_C$
	V_R	Set of drone recovery nodes, $V_R = V_S \cup V_C \cup V_0^+$
	V_N	Set of all nodes except for distribution centers, $V_N = V_S \cup V_C$
Parameters	A	Set of arcs that EVs travel, $A = \{(i, j) i, j \in V\}$
	F	Set of EVs that are equipped with designated drones, $F = \{1, \dots, f, \dots, G\}$
	d_{ij}^t, d_{ij}^d	Distance traveled by the EV from node i to node j , and distance flown by the drone from node i to node j , respectively
	t_{ij}^t, t_{ij}^d	EV's travel time from node i to node j and the drone's flying time from node i to node j , respectively
	s_i^t, s_i^d	EV's service time at customer node i , and drone's service time at node i , respectively
	$[ET_i, LT_i]$	Service time window for customer node i
	q_i, Q_t	Customer node i 's demand, and EV's payload, respectively
	Q_d, B_t	Drone's payload, and EV's battery capacity, respectively
	B_d, v_d	Drone's maximum endurance, and drone's flight speed, respectively
	h, lt	EV's energy consumption coefficient per kilometer, and drone's single-launch operation time, respectively
	rt, c_1	Drone's single-recovery operation time, and cost-per-unit distance traveled by EV
	c_2, c_3	Charging station's unit charging cost, and drone's transportation cost-per-unit distance, respectively
	c_t, U_{if}	EV's fixed cost, and EV f 's charging time at the charging station i , respectively
	M	A large positive number
Decision variables	x_{ijf}	Binary variable indicating if EV f travels from node i to node j
	x_{ijkf}	Binary variable indicating if drone f takes the route $\langle i, j, k \rangle$
	y_{if}	Binary variable indicating if EV f is charged at charging station i
	p_{ijf}	Binary variable indicating if node i is visited before node j
Non-decision variables	a_{if}^t, a_{if}^d	Time when EV f , and drone f arrive at node i , respectively.
	$e_{if}^{(1)}, e_{if}^{(2)}$	EV f 's remaining amount of electricity when it arrives at node i , and EV f 's remaining amount of electricity when it leaves node i , respectively
	ec_{if}, tc_{if}	EV f 's charge amount at node i , and EV f 's charging time at node i , respectively
	u_{if}	Node i 's order in EV f 's route coding

3.2. Optimization Model

The objective function of the proposed multi-depot EV–drone collaborative-delivery routing problem is to minimize the total delivery cost Z , including EV's transportation cost $Z_{transport}^t$, EV's charging cost $Z_{charging}^t$, and drone's transportation cost $Z_{transport}^d$, which are provided in the following.

$$Z_{transport}^t = c_1 \sum_{f \in F} \sum_{i \in V, i \neq j} \sum_{j \in V} d_{ij}^t x_{ijf} + c_t \sum_{f \in F} \sum_{i \in V_0} \sum_{j \in V_N \cup V_0^+} x_{ijf} \quad (1)$$

$$Z_{charging}^t = c_2 \sum_{f \in F} \sum_{i \in V_S} e c_{if} y_{if} \quad (2)$$

$$Z_{transport}^d = c_3 \sum_{f \in F} \sum_{i \in V_L} \sum_{j \in V_R} d_{ij}^d x_{ijkf} \quad (3)$$

Therefore, our MD-EVDRPTW model can be formulated as follows.

Objective function:

$$\min Z = Z_{transport}^t + Z_{charging}^t + Z_{transport}^d \quad (4)$$

Constraints:

$$\sum_{f \in F} \sum_{i \in V_L \setminus \{j\}} x_{ijf} + \sum_{f \in F} \sum_{i \in V_L \setminus \{j,k\}} \sum_{k \in V_R \setminus \{i,j\}} y_{ijkf} = 1, \forall j \in V_C \quad (5)$$

$$\sum_{j \in V_N \cup V_0^+} x_{ijf} \leq 1, \forall f \in F, i \in V_0 \quad (6)$$

$$\sum_{i \in V_0 \cup V_N} x_{ijf} \leq 1, \forall f \in F, j \in V_0^+ \quad (7)$$

$$\sum_{i \in V_L \setminus \{j\}} x_{ijf} - \sum_{k \in V_R \setminus \{j\}} x_{jkf} = 0, \forall f \in F, j \in V_N \quad (8)$$

$$u_{jf} \leq M \sum_{i \in V_L \setminus \{j\}} x_{ijf}, \forall f \in F, j \in V_R \quad (9)$$

$$u_{if} - u_{jf} \leq M(1 - x_{ijf}) - 1, \forall f \in F, i \in V_L \setminus \{j\}, j \in V_R \setminus \{i\} \quad (10)$$

$$u_{jf} - u_{if} \leq M p_{ijf}, \forall f \in F, i \in V_L \setminus \{j\}, j \in V_N \setminus \{i\} \quad (11)$$

$$u_{jf} - u_{if} \leq M(p_{ijf} - 1) + 1, \forall f \in F, i \in V_L \setminus \{j\}, j \in V_N \setminus \{i\} \quad (12)$$

$$\sum_{j \in V_C} \sum_{k \in V_R \setminus \{j\}} q_j x_{jkf} + \sum_{j \in V_C} \sum_{i \in V_L \setminus \{j,k\}} \sum_{k \in V_R \setminus \{i,j\}} q_j y_{ijkf} \leq Q_t, \forall f \in F \quad (13)$$

$$y_{ijkf} = 0, \forall f \in F, i \in V_L \setminus \{j,k\}, j \in V_C \setminus \{V_D, i, k\}, k \in V_R \setminus \{i,j\} \quad (14)$$

$$\sum_{j \in V_D \setminus \{i,k\}} \sum_{k \in V_R \setminus \{i,j\}} y_{ijkf} \leq 1, \forall f \in F, i \in V_L \quad (15)$$

$$\sum_{i \in V_L \setminus \{j,k\}} \sum_{j \in V_D \setminus \{i,k\}} y_{ijkf} \leq 1, \forall f \in F, k \in V_R \quad (16)$$

$$\sum_{n \in V_R \setminus \{i\}} x_{inf} + \sum_{n \in V_L \setminus \{k\}} x_{nkf} \geq 2y_{ijkf}, \forall f \in F, i \in V_L \setminus \{j,k\}, j \in V_C \setminus \{i,k\}, k \in V_R \setminus \{i,j\} \quad (17)$$

$$a_{if}^t + t_{ik}^t + s_k^t + lt \sum_{j \in V_D \setminus \{i,n\}} \sum_{n \in V_R \setminus \{i,j\}} y_{ijnf} + rt \sum_{n \in V_L \setminus \{j,k\}} \sum_{j \in V_D \setminus \{n,k\}} y_{njkf} - M * (1 - x_{ikf}) \leq a_{kf}^t \quad (18)$$

$$\forall f \in F, i \in V_L \setminus \{k\}, k \in V_R \setminus \{i, V_S\}$$

$$a_{if}^t + t_{ik}^t + lt \sum_{j \in V_D \setminus \{i,n\}} \sum_{n \in V_R \setminus \{i,j\}} y_{ijnf} + rt \sum_{n \in V_L \setminus \{j,k\}} \sum_{j \in V_D \setminus \{n,k\}} y_{njkf} + tc_{kf} - M * (1 - x_{ikf}) \leq a_{kf}^t \quad (19)$$

$$\forall f \in F, i \in V_L \setminus \{k\}, k \in (V_R \cap V_S) \setminus \{i\}$$

$$a_{if}^t + lt + t_{ij}^d + s_i^d - M(1 - \sum_{k \in V_R \setminus \{i,j\}} y_{ijkf}) \leq a_{if}^d, \forall f \in F, i \in V_L \setminus \{j\}, j \in V_D \setminus \{i\} \quad (20)$$

$$a_{jf}^d + t_{jk}^d + rt + s_k^t - M(1 - \sum_{i \in V_L \setminus \{j,k\}} y_{ijkf}) \leq a_{kf}^d, \forall f \in F, j \in V_D \setminus \{k\}, k \in V_R \setminus \{j, V_S\} \quad (21)$$

$$a_{jf}^d + t_{jk}^d + rt + tc_{kf} - M(1 - \sum_{i \in V_L \setminus \{j,k\}} y_{ijkf}) \leq a_{kf}^d, \forall f \in F, j \in V_D \setminus \{k\}, k \in (V_R \cap V_S) \setminus \{j\} \quad (22)$$

$$a_{if}^t - M(1 - \sum_{j \in V_D \setminus \{i,k\}} \sum_{k \in V_R \setminus \{i,j\}} y_{ijkf}) \leq a_{if}^d, \forall f \in F, i \in V_L \quad (23)$$

$$a_{if}^t + M(1 - \sum_{j \in V_D \setminus \{i,k\}} \sum_{k \in V_R \setminus \{i,j\}} y_{ijkf}) \leq a_{if}^d, \forall f \in F, i \in V_L \quad (24)$$

$$a_{kf}^t - M(1 - \sum_{i \in V_L \setminus \{j,k\}} \sum_{j \in V_D \setminus \{i,k\}} y_{ijkf}) \leq a_{kf}^d, \forall f \in F, k \in V_R \quad (25)$$

$$a_{kf}^t + M(1 - \sum_{i \in V_L \setminus \{j,k\}} \sum_{j \in V_D \setminus \{i,k\}} y_{ijkf}) \leq a_{kf}^d, \forall f \in F, k \in V_R \quad (26)$$

$$a_{kf}^d + M(3 - p_{imf} - \sum_{i \in V_C \setminus \{i,k\}} y_{ijkf} - \sum_{p \in V_C \setminus \{m,q\}} \sum_{q \in V_R \setminus \{m,p\}} y_{mpqf}) \leq a_{mf}^d, \quad \forall f \in F, i \in V_L \setminus \{m,k\}, m \in V_N \setminus \{i,k\}, k \in V_R \setminus \{i,m\} \quad (27)$$

$$ET_i \leq a_{if}^t \leq LT_i, \forall f \in F, i \in V_C - V_D \quad (28)$$

$$ET_i \leq a_{if}^d \leq LT_i, \forall f \in F, i \in V_D \quad (29)$$

$$tc_{if} = U_{if}(e_{if}^{(1)}), \forall f \in F, i \in V_S \quad (30)$$

$$ec_{if} = (B_t - e_{if}^{(1)}) * x_{if}, \forall f \in F, i \in V_S \quad (31)$$

$$e_{if}^{(2)} - h * d_{ij} - M * (1 - x_{ijf}) \leq e_{if}^{(1)}, \forall f \in F, i \in V_L \setminus \{j\}, j \in V_R \setminus \{i\} \quad (32)$$

$$e_{if}^{(1)} \geq 0, \forall f \in F, i \in V \quad (33)$$

$$e_{if}^{(2)} = B_t, \forall f \in F, i \in V_R \quad (34)$$

$$e_{if}^{(1)} = e_{if}^{(2)}, \forall f \in F, i \in V_C \quad (35)$$

$$e_{if}^{(2)} = B_t, \forall f \in F, i \in V_0 \quad (36)$$

$$a_{if}^t = 0, \forall f \in F, i \in V_0 \quad (37)$$

$$a_{if}^d = 0, \forall f \in F, i \in V_0 \quad (38)$$

$$t_{ij}^d + s_i^d + t_{jk}^d - M * (1 - y_{ijkf}) \leq B_d, \forall f \in F, i \in V_L \setminus \{j,k\}, j \in V_D \setminus \{i,k\}, k \in V_R \setminus \{i,j\} \quad (39)$$

$$x_{ijf} \in \{0,1\}, \forall f \in F, i, j \in A \quad (40)$$

$$y_{ijkf} \in \{0,1\}, \forall f \in F, i \in V_L \setminus \{j,k\}, j \in V_D \setminus \{i,k\}, k \in V_R \setminus \{i,j\} \quad (41)$$

$$p_{ijf} \in \{0,1\}, \forall f \in F, i \in V_L \setminus \{j\}, j \in V_N \setminus \{i\} \quad (42)$$

$$u_{if} \in \mathbb{N}, \forall f \in F, i \in V \quad (43)$$

Equation (4) defines the objective function as minimizing the total delivery costs. Equation (5) ensures that each customer is visited by an EV or a drone exactly once. Constraints (6) and (7) specify that every route begins and ends at one of the distribution centers. Equation (8) represents the node flow-conservation constraint, ensuring the continuity of the delivery route. Constraints (9) and (10) are sub-loop elimination constraints to prevent unnecessary loops in the routing.

Constraints (11) and (12) restrict the order of the EV routes, ensuring no turnback occurs. Constraint (13) sets the payload capacity constraint for the EV. Equation (14) specifies that drones only visit customer nodes that meet the criteria for drone service. Constraints (15) and (16) ensure that each drone launch and recovery node is used only once. Constraint (17) stipulates that drones can only be launched and retrieved at two distinctive nodes that the designated EV has visited.

Constraints (18) and (19) ensure that when the EV visits both charging and non-charging station nodes, the arrival time must be greater than the previous node's visit time. Constraint (20) ensures that the drone arrives at a customer node after the EV reaches the launch node. Constraints (21) and (22) apply the same timing constraints for drone visits to charging and non-charging station nodes. Constraints (23) and (24) ensure synchronization of the arrival times of both the EV and drone at the launch node, while Constraints (25) and (26) ensure the synchronization of their arrival times at the recovery node.

Constraint (27) restricts each EV to carrying only one drone. Constraints (28) and (29) enforce customer time window constraints for EVs and drones. Equation (30) calculates the charging time for the EV, while Equation (31) determines the amount of charge needed at the charging station. Constraint (32) calculates the EV's energy consumption between any two points, and Constraint (33) ensures the EV's charge level remains non-negative.

Equation (34) defines a full-charge strategy for EV charging, and Equation (35) specifies that there is no energy consumption while the EV is at a customer node. Equation (36) ensures that the EV departs from the distribution center fully charged. Equations (37) and (38) set the starting time for the EV and drone at the distribution center to zero. Constraint (39) imposes the range constraint for the drone. Finally, Constraints (40) through (43) specify the decision variable attributes.

4. Algorithm Design

Our proposed multi-depot EV-drone collaborative-delivery routing optimization problem is apparently classified as NP-hard. We have developed an improved adaptive large-neighborhood search (ALNS) algorithm to solve the MD-EVDRPTW model. The framework of the proposed algorithm is presented in Figure 2.

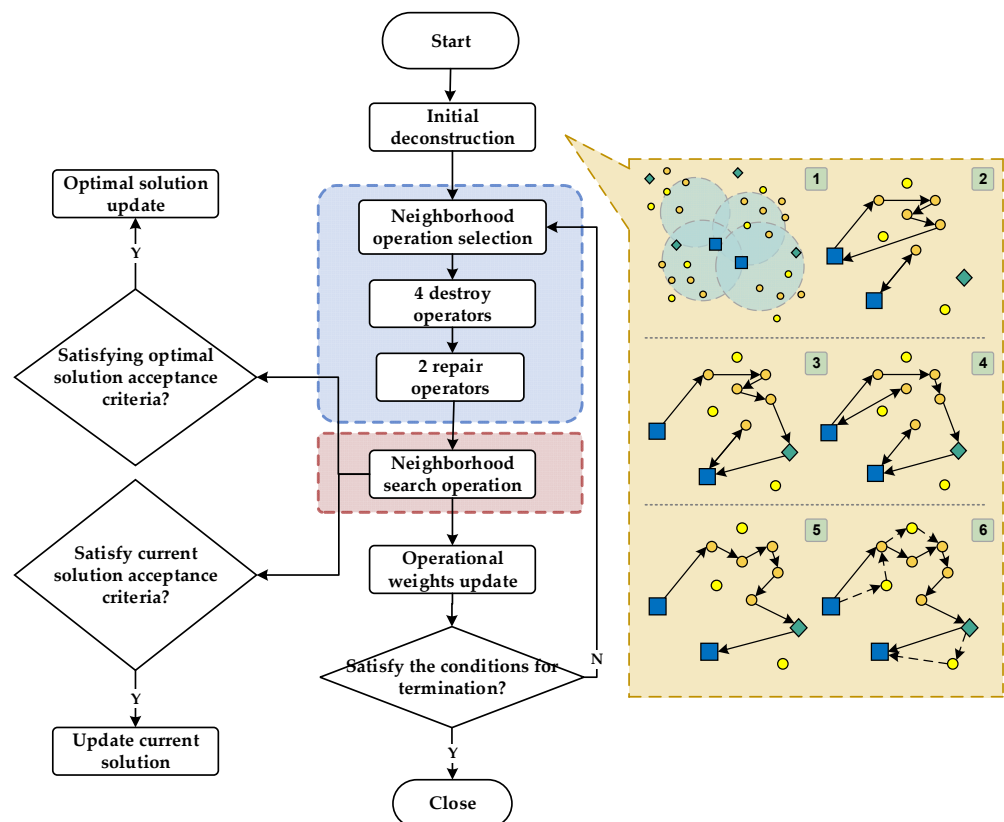


Figure 2. An improved ALNS algorithm framework.

4.1. Solution Description and Initial Solution Construction

4.1.1. Solution Description

Using integer arrangement coding, we assume that the road network contains n customer and charging station nodes, as well as m distribution centers, represented by the natural numbers $0, 1, \dots, m$. An initial solution consists of two codes: an upper code, denoted by $\mathbb{U}$, representing the vehicle path, and a lower binary code of the same length, denoted by $\mathbb{L}$, which is used to determine the service type for each customer node. A non-negative integer value $i \in \{2, 3, \dots, n + 1\}$ is assigned to distinguish customer nodes based on the following rules:

- If the i -th node of $\mathbb{L}$ is coded as 0, the corresponding i -th customer node in $\mathbb{U}$ will be served by an EV;
- If the i -th node of $\mathbb{L}$ is coded as 1 and the $(i - 1)$ -th node is coded as 0, the i -th customer node in $\mathbb{U}$ will be served by a drone;
- If the i -th node of $\mathbb{L}$ is coded as 1 and the $(i - 1)$ -th node is also coded as 1, then the i -th customer node in $\mathbb{U}$ will be served by an EV.

From the three rules described above, the following can be inferred: when a sequence of $\mathbb{L}$ contains a coded “1”, the first “1” represent a customer node served by a drone. In this case, the drone’s launch and recovery nodes correspond to the nodes of $\mathbb{U}$ that are immediately preceded and followed by “0”s in the sequence. Figure 3 illustrates an example of a delivery route solution with 2 distribution centers, 16 customer nodes, and 4 charging stations, along with the corresponding route encoding method.

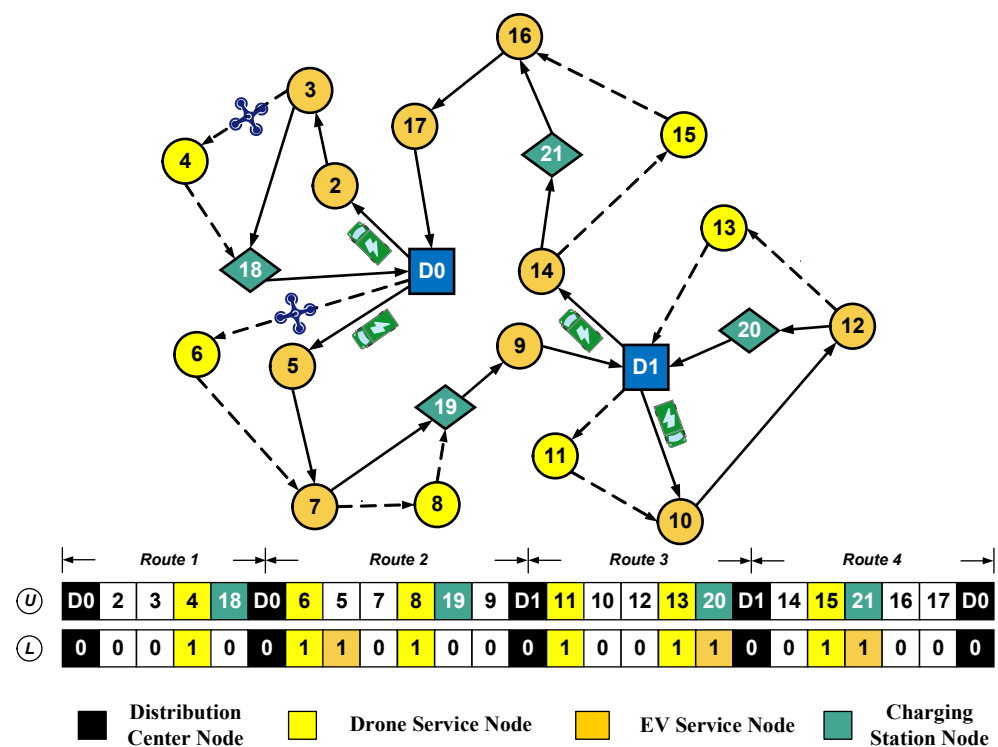


Figure 3. Illustration of a multi-depot EV-drone cooperative-delivery solution coding.

It is important to note that in a multi-depot, time-dependent network, the insertion of drones and charging stations into the EV route expands the solution space in two key ways: (1) charging stations can simultaneously accommodate EV charging, drone launch, and synchronization between the EV and drone; and (2) drones cannot independently visit charging stations to perform launching and recovering functions. Figure 4 illustrates the expansion of the solution space following the insertion of a charging station.

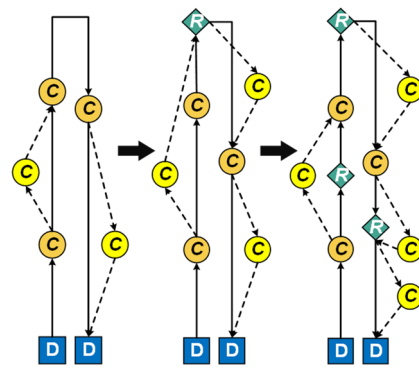


Figure 4. Solution space expansion due to charging station insertion.

4.1.2. Initial Solution Construction

Constructing a high-quality initial solution is critical for the efficiency and effectiveness of the improved ALNS algorithm, as it directly impacts both the search process and the final solution. In this paper, we generate a high-quality initial solution through six key steps: customer node grouping, EV customer node insertion, charging station insertion, EV route decomposition, route integration optimization, and drone route insertion. The overall construction process is illustrated in Figure 5, with the detailed steps outlined in the following.

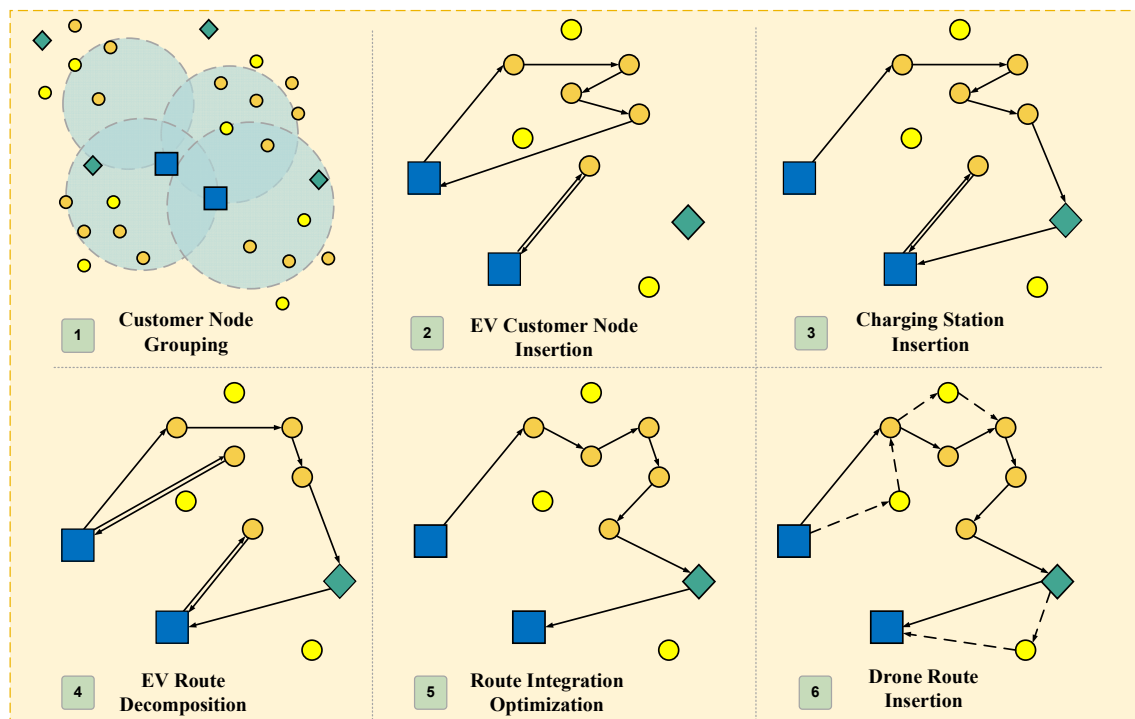


Figure 5. The initial solution construction process.

Step 1. Customer node grouping

First, the customer nodes that meet the technical specifications of the drone are separated from the set of customer nodes V_C and classified as the alternative set of drone-servicable customers V_{D0} . Next, the minimum round-trip flight distance between each node in this alternative set and other customer nodes is calculated. If a node falls within the drone's mileage range, it is included in the set of drone-servicable customers V_{D1} . The nodes in V_{D1} are then randomly shuffled and divided into two subsets: $V_{D1}^{(1)}$, which represents the nodes to be directly served by drones, and $V_{D1}^{(2)}$, which includes nodes

not served by drones. The final set of drone-served customers is $V_D = V_{D1}^{(1)}$, while the remaining customer nodes are placed into V_T to be served by EVs. This method allows the greedy insertion operator in the subsequent ALNS neighborhood operation to reintroduce drone nodes into the EV route, establishing the “trinity” relationship illustrated in Figure 6.

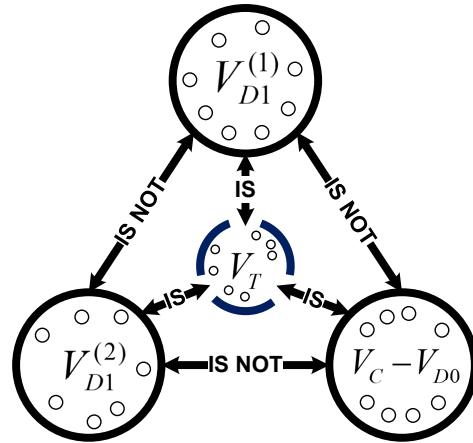


Figure 6. Trinity of customer node subsets.

Step 2. EV customer node insertion

A distribution center is randomly selected, and an EV is dispatched to begin delivery by finding the nearest customer node to the current position. Starting from the first customer node, the EV’s capacity and the customer’s time window requirements are checked for feasibility. If both conditions are satisfied, the EV proceeds to that customer; otherwise, the EV returns to the nearest distribution center from the last served customer node. A new EV is then dispatched from the closest distribution center to the next customer node, to continue the service. This process is repeated until all EV-served customer nodes are inserted into the service. The process of EV customer node insertion is illustrated in Figure 7.

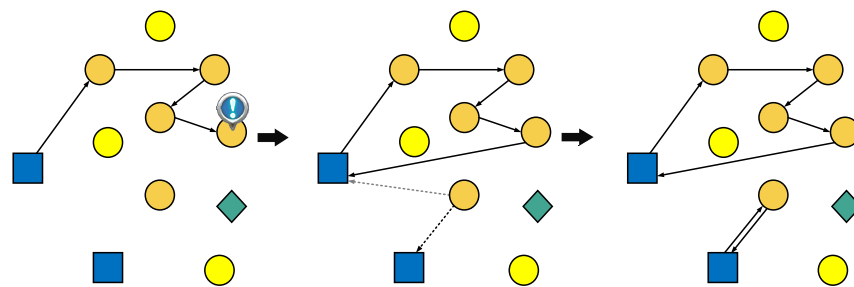


Figure 7. Illustration of EV customer node insertion.

Step 3. Charging station insertion

The previously mentioned EV customer node insertion process does not account for the vehicle’s energy feasibility along the route. To maintain sufficient power throughout the journey, the greedy charging-station insertion strategy, as proposed by Keskin et al. [7], is applied.

Step 4. EV route decomposition

During the charging station insertion process, infeasible routes need to be decomposed to handle scenarios where the EV’s remaining charge is insufficient to reach a charging station, or the time window constraints of subsequent customer nodes cannot be met. For the set of infeasible-route customer nodes, the process begins by inserting each node into a new route, starting from the distribution center closest to the first customer node. A charging station is added if the route remains infeasible after inserting the customer nodes. If the insertion of a charging station still fails to create a feasible route, the longest currently

feasible portion of the route is retained. The customer node that could not be inserted becomes the starting point for a new route, beginning at the nearest distribution center. This process is repeated until all infeasible customer nodes are incorporated into a valid route. The EV route decomposition process is illustrated in Figure 8.

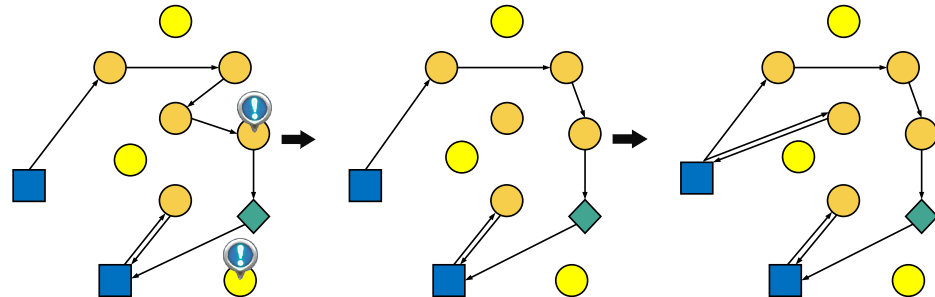


Figure 8. Illustration of EV route decomposition.

Step 5. Route integration optimization

First, a threshold value is established for the number of customer nodes to be merged. Customer nodes that do not meet this threshold are extracted from the routes, leading to the deletion of those routes. The removed customer nodes are then reinserted into the existing routes using the customer node insertion method described in Step 2. New routes are created using the same method if this insertion fails. Once the integration and optimization process is complete, the objective function values of the old- and new-route sets are compared, and the more efficient set of routes is retained. The route integration optimization process is illustrated in Figure 9.

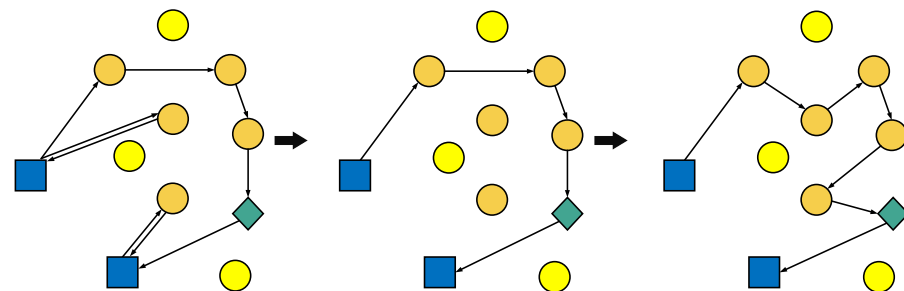


Figure 9. Illustration of route integration optimization.

Step 6. Drone route insertion

This step primarily involves the strategies for launching and recovering drones within the complete EV route.

- **Drone launch strategy:** For each drone customer node in the complete EV route, identify the nearest node as the launch point for the drone. Each EV customer node must be selected only once to prevent drones carried by different EVs from sharing the same launch node.
- **Drone recovery strategy:** After assigning a launch node to each drone, search for a suitable recovery node based on proximity. Traverse each EV route to locate the drone's launch node, then examine all potential recovery nodes—including charging stations and distribution centers—moving toward the route's end node. Suppose the flight distance from the launch node to the customer node, followed by the distance to the potential recovery node, does not exceed the drone's maximum flight distance: in that case, the node is designated as the recovery point.

The EV route insertion process is illustrated in Figure 10.

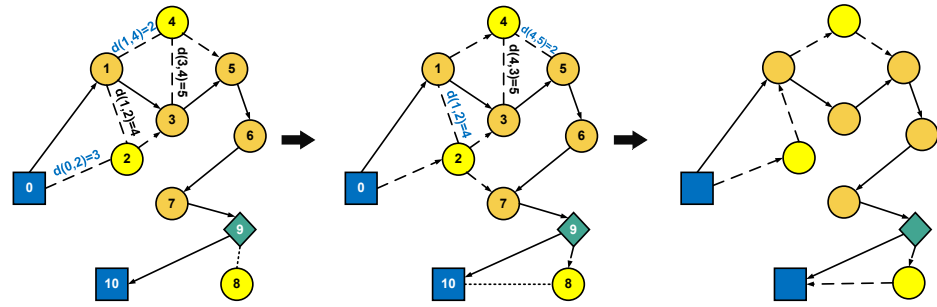


Figure 10. Illustration of drone route insertion process.

4.2. Destruction and Reconstruction Operations

After the improved ALNS algorithm generates the initial solution, it proceeds to create a new solution through a process of destruction (“destroy”) and reconstruction (“repair”) [17]. This paper aims to minimize the delivery costs of the EV and drone routes. To facilitate this, we have designed four removal operators (i.e., destroy operators) and two insertion operators (i.e., repair operators), which address delivery costs and randomness during the destruction and reconstruction processes. Additionally, these operators incorporate perturbation mechanisms to enhance solution diversity in the neighborhood search process.

4.2.1. Removal Operators

(1) Random Removal Operator

Randomly select a certain number of customers from each currently solved EV route for removal, ensuring that all drone customers are excluded. If a route ends up with zero customer nodes, it will be deleted as invalid. For example, as illustrated in Figure 11, EV customer nodes 5, 10, 12, and 16 are randomly removed from the route. Consequently, the invalid route $1 \rightarrow 20 \rightarrow 1$, which consists solely of travel to the charging station, is eliminated.

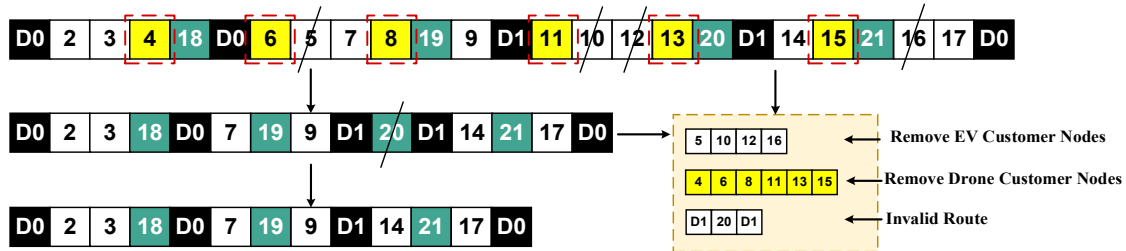


Figure 11. Illustration of random removal operator.

(2) Saving-Degree Removal Operator

Calculate the saving degree cost by comparing the original objective function value $f(s)$ of the current solution s with the objective function value $f(s)_{-i}$ after removing the EV customer node i . Remove a specified number of customer nodes, ranked from largest to smallest based on their savings degree, while excluding all drone customers and eliminating invalid routes. The formula for calculating the saving degree of customer nodes is presented in Equation (44).

$$\text{cost}(i, s) = f(s) - f(s)_{-i} \quad (44)$$

As illustrated in Figure 12, the cost value $\text{cost}(i, s)$ for each EV customer node is calculated and sorted after multiplying each value by a random number between 0 and 1. Nodes 16, 9, 2, and 3 are then selected for removal in order from the highest to the

lowest cost. After removing the drone customer node, a new invalid route $0 \rightarrow 18 \rightarrow 0$ is generated, which is subsequently eliminated.

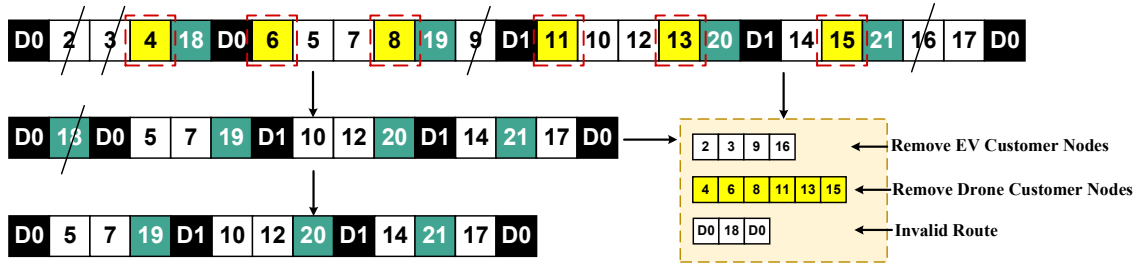


Figure 12. Illustration of the saving-degree removal operator.

(3) Correlation Removal Operator

When the correlation $r(i, j)$ between two customer nodes in the current solution s exhibits a negative relationship with the distance d_{ij} , customer nodes with higher correlation within the route are removed, along with all drone customers. The method for calculating the correlation is outlined in Equation (45), where $D(s)$ represents the total mileage of route s .

$$r(i, j) = 1 - \frac{d_{ij}}{D(s)} \quad (45)$$

As illustrated in Figure 13, we randomly select customer nodes 5 and 16 from the current solutions of Routes 2 and 4. Taking Route 2 as an example, the correlation between neighboring customer nodes 5 and 7 is denoted as $r(5, 7)$. For non-neighboring nodes, the correlation is calculated by summing the computed values across multiple segments; for instance, the correlation between nodes 5 and 9 is given by $r(5, 9) = r(5, 7) + r(7, 19) + r(19, 9)$, which includes the distances to and from any charging stations, if applicable. After calculating the correlation for the current node against all subsequent nodes, each correlation value is multiplied by a random number in the range of 0 to 1. Subsequently, customer nodes 9 and 17 are removed in descending order of their calculated values, followed by the removal of the drone customer node.

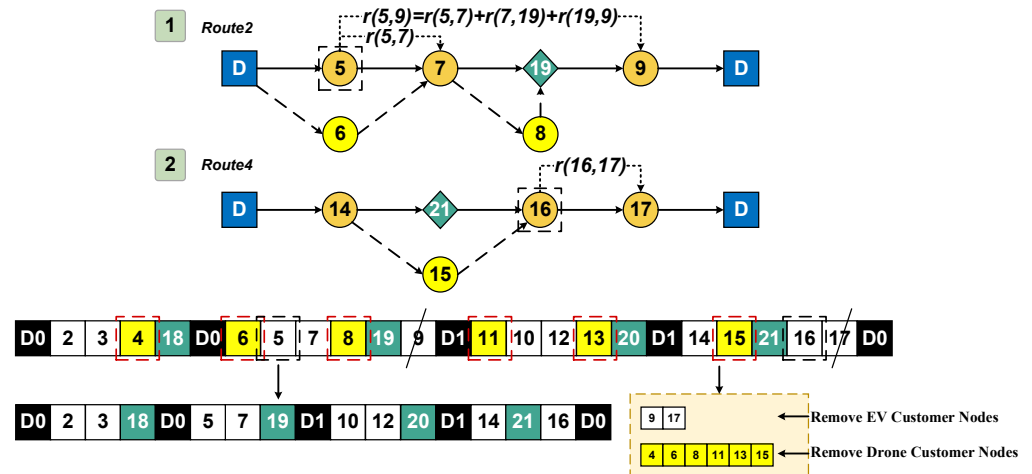


Figure 13. Illustration of the correlation removal operator.

(4) Extreme-Distance Removal Operator

Filter out customers with the largest total distance between each EV route in the current solution and their immediate preceding, and following nodes for removal. This includes all drone customers and invalid routes. As illustrated in Figure 14, we take Routes 2 and 3 as examples. We calculate the sum of distances between each customer node and its

neighboring nodes (which may include distribution centers and charging stations). In this process, we identify and remove the extreme-distances of customer nodes 7 and 10 from the routes, extract the drone customer nodes, and delete the invalid route: $1 \rightarrow 20 \rightarrow 1$.

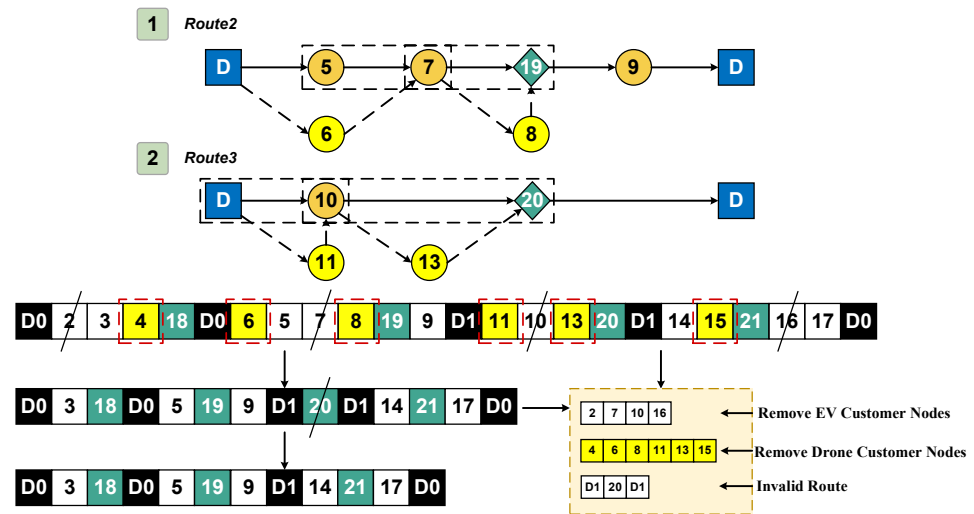


Figure 14. Illustration of the extreme-distance removal operator.

4.2.2. Insertion Operator

(1) Random Insertion Operator

For the customer nodes removed by the operators mentioned above, the insertion points for EV customer nodes are randomly selected and inserted back into the EV route. Drone customer nodes are reinserted into the EV route using the same method described for drone route insertion in Section 4.1.2. This process is repeated until all customer nodes are added back to the route. Using the corrupted code in Figure 15 as an example, the removed EV customer nodes (5, 10, 12, and 16) are reinserted after randomly selecting their insertion points. Additionally, the drone launch and recovery nodes are adjusted as the drone customer nodes are reinserted, resulting in a newly optimized solution.

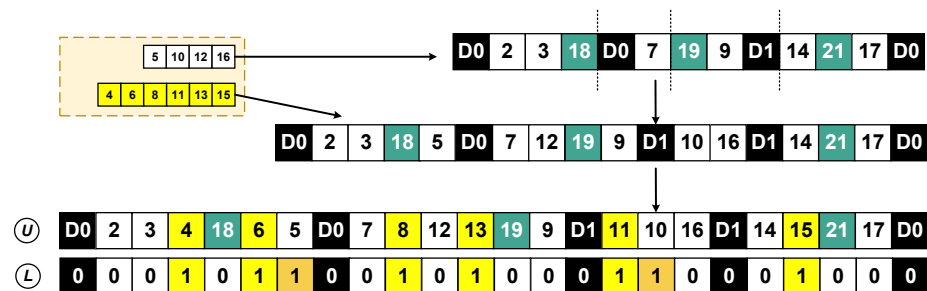


Figure 15. Illustration of the random insertion operator.

(2) Greedy Insertion Operator

In our study, some customers suitable for drone delivery are deliberately kept as EV customers during the initial solution construction phase of the ALNS algorithm. After merging the removed EV and original drone customers, the algorithm filters them and inserts those that exceed the drone's maximum carrying capacity or flight distance back into the EV route, aiming to minimize the cost increment while satisfying all constraints. Using the solution coding in Figure 16 as an example, the algorithm checks whether the removed customer nodes {5, 10, 12, 16, 4, 6, 8, 11, 13, 15} can be reinserted into the current EV route under drone delivery conditions. In this case, the original drone customer nodes {8, 13} no longer meet the drone delivery criteria, while the original EV customer nodes {10, 12} do. The remaining nodes {5, 8, 13, 16} are reinserted into the EV route by selecting the insertion

points with the smallest difference in route cost before and after insertion, both multiplied by random numbers between 0 and 1, as outlined in the EV route construction method in Section 4.1.2. Finally, the drone customers are inserted to restore a feasible solution while ensuring the EV's payload capacity and the customer's time window constraints are met.

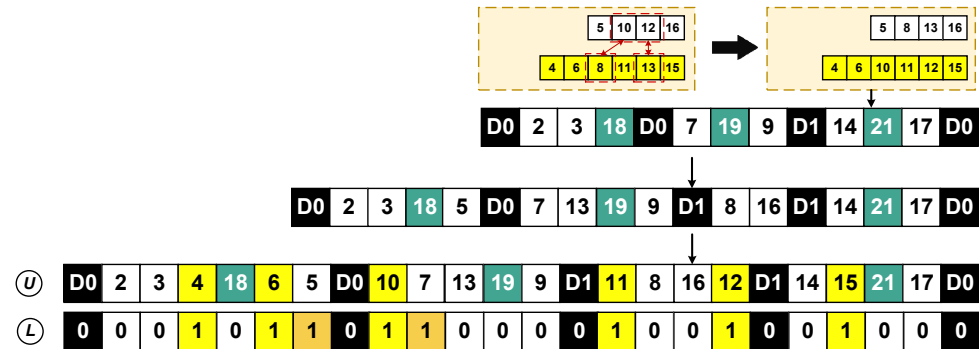


Figure 16. Illustration of the greedy insertion operator.

To better illustrate the destruction and reconstruction operations process using removal and insertion operators, Figure 17 demonstrates how a new route is generated from the original route after applying a random removal operator followed by a random insertion operator.

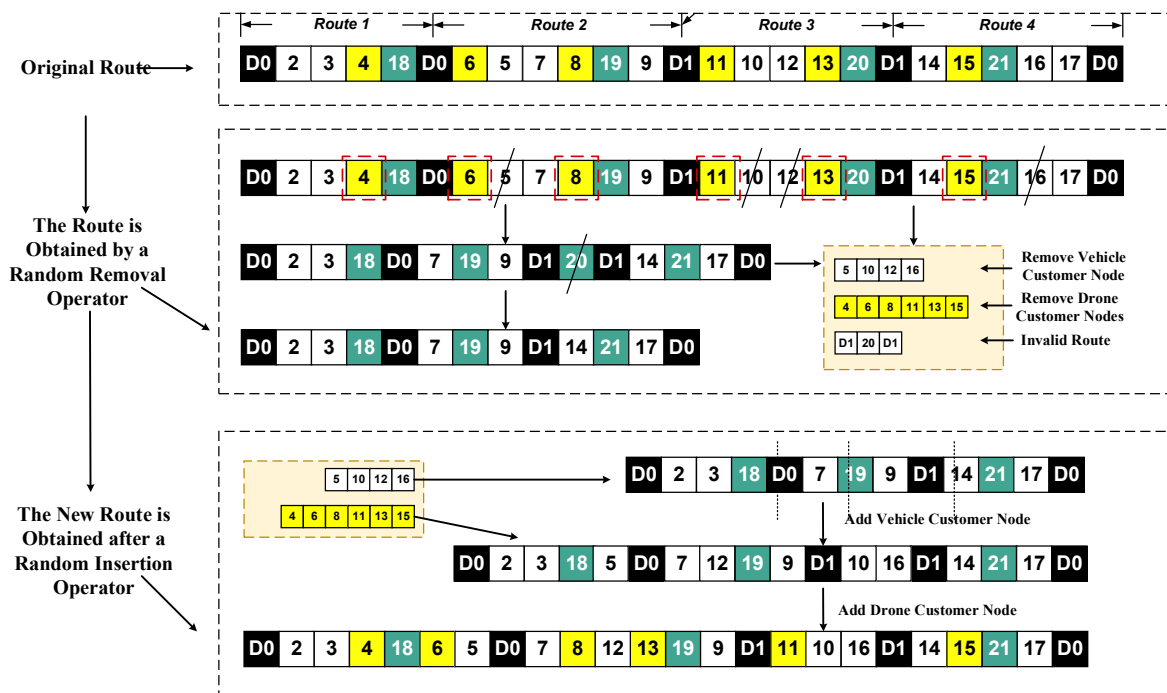


Figure 17. Illustration of applying random removal and insertion operators.

4.3. Choice of Operator

During each algorithm iteration, the destruction and repair operators are selected using a roulette wheel method. Let Op represent the set of all operators, $\omega_i (i \in Op)$, represent the weights of the operators, and $\delta_i (i \in Op)$ represent their scores. The number of times each operator is used is recorded as v_i . Operators are scored based on the quality of the new solutions they generate, following three criteria:

- Global optimal solution: +1.5 points
- Suboptimal solution: +1.2 points
- Feasible solution: +0.8 points

Let the update coefficient, ξ , for the operator weights range between 0 and 1. Based on the operator scores, the weights are updated according to Equation (46).

$$\omega_i = \begin{cases} \omega_i, & \text{if } v_i = 0 \\ (1 - \xi) * \omega_i + \xi * \frac{\delta_i}{v_i}, & \text{if } v_i > 0 \end{cases}, \forall i \in Op \quad (46)$$

4.4. Stopping and Acceptance Criteria

(1) The Metropolis criterion in the simulated annealing algorithm is employed to accept inferior solutions. To effectively prevent the algorithm from prematurely converging to a local optimum, a neighborhood solution $f(x)_{new}$ which is worse than the current solution $f(x)_{old}$, is accepted with a certain probability p . At the start of the simulated annealing algorithm, an initial temperature T_0 is set, with a decay rate within the range of $[0, 1]$. The temperature T at each iteration is the product of the previous iteration's temperature and the decay rate. The probability p is calculated using Equation (47).

$$p = \begin{cases} 1, & f(x)_{new} < f(x)_{old} \\ \exp\left(\frac{f(x)_{old} - f(x)_{new}}{T}\right), & f(x)_{new} \geq f(x)_{old} \end{cases} \quad (47)$$

(2) The algorithm's iteration is terminated using an unimproved-threshold policy. When the number of consecutive iterations without improvement, denoted as num_con , reaches the preset threshold num_end , the iteration stops, and the satisfactory solution is returned. Both the preset threshold num_end and the maximum number of iterations $num_iterations$ are determined based on the population size, ensuring that the number of adaptive neighborhood searches is suitable for the problem being solved.

5. Experimental Analysis

All experiments in this study were conducted on a computer equipped with an Intel Core i7-10875H 2.30GHz CPU, 16GB of DDR4 RAM, running the Windows 10 operating system, and using Python 3.7 for programming.

5.1. Experimental Design

Due to the complexity of the MD-EVDRPTW model, no existing dataset contains instances that directly meet its requirements. To address this, we utilize the EVRPTW dataset C101_21, developed by Schneider et al. [5] and made available through a link provided by Goeke [27]. We apply the K-means clustering algorithm, using profile coefficients, to group all nodes and identify the distribution centers. This process transforms the EVRPTW dataset into one suitable for the MD-EVDRPTW model, which we label as C101_21M. The new dataset consists of 4 distribution centers, and 21 charging stations.

(1) Parameter settings related to the nonlinear charging function for EVs:

The EV nonlinear charging function is represented as shown in Figure 18. The charging time, t_k , and the state of charge, α_k , corresponding to the turning point k in the function are set as follows: $t_1, \alpha_1 = [226.7814, 65.3047]$, $t_2, \alpha_2 = [276.6302, 73.4023]$, and $t_3, \alpha_3 = [378.6145, 77.75]$.

(2) Setting of parameters related to EV's time-varying travel speed:

To analyze the impact of speed variation on delivery route planning under time-varying conditions, three different EV travel speed scenarios were set up for comparison. In the experiment conducted by Sacramento et al. [17], the EV travels at a fixed speed of 56.3 km/h (Case 1). To incorporate time-varying characteristics, this section uses a computer-generated set of seven random speed values with an average of 56.3 km/h, representing the EV's speed across different time intervals (Case 2). A holiday scenario

featuring widespread congestion is also introduced (Case 3). The specific time periods and corresponding speeds for each scenario are provided in Table 2.

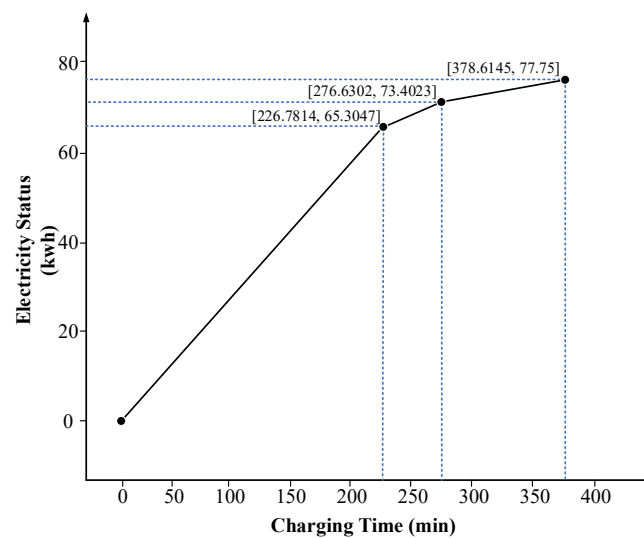


Figure 18. EV's nonlinear charging function.

Table 2. Time-varying EV travel speed in different scenarios.

Time-Varying Scenario	Travel Speed (km/h)						
	7:00–9:30	9:30–13:30	13:30–15:30	15:30–17:30	17:30–20:30	20:30–24:00	24:00–31:00
Case 1	56.3	56.3	56.3	56.3	56.3	56.3	56.3
Case 2	43.4	64.7	68.1	52.4	33.8	56.3	75.4
Case 3	21.7	32.25	34.05	26.2	16.9	28.15	37.7

(3) Initial parameter settings for the remaining experiment variables:

The initial values for the remaining experiment-related parameters are outlined in Table 3 below.

Table 3. Initial values of experiment-related parameters.

Parameters	Values
EV payload (kg)	200
EV battery capacity (kWh)	79.69
EV power-consumption coefficient per kilometer	1
drone payload (kg)	10
drone endurance time (min)	55
drone flight speed (km/h)	80
drone launch and recovery operation time (min)	1
EV unit fixed cost (CNY)	100
EV cost per kilometer of electricity consumption (CNY)	0.6
EV unit charging cost (CNY)	0.4
EV transportation cost-per-unit distance (CNY)	0.3
Initial temperature of the simulated-annealing acceptance criterion	100
Decay rate of the simulated-annealing acceptance criterion	0.94
Initial weight of ALNS operator	0
Initial score of ALNS operator	1
ALNS weight-update factor	0.3
ALNS unimproved threshold	200
ALNS maximum number of iterations	10,000

5.2. Analysis of Experimental Results

Ablation experiments will be conducted to evaluate the performance of the proposed algorithm and assess the impact of the four removal operators and two insertion operators. These experiments will involve masking the perturbation strategies within the extremum-distance removal operator module and the greedy insertion operator. The results of these masked experiments will be analyzed by comparing them with those of the original algorithm. Each of the latter three experiments will be executed 10 times using the C101_21M dataset, which includes 100 customers. Here, G represents the number of dispatched EVs, rs denotes the number of charging stations utilized, Z_{ALNS} indicates the satisfactory solution generated by the algorithm, while T_{ALNS} and I_{ALNS} represent the algorithm's solution time and the optimal number of iterations, respectively.

(1) The experimental results of using our proposed algorithm to solve the instances in the dataset are listed in Table 4.

Table 4. Results of using the proposed algorithm.

Series Number	G	rs	Z_{ALNS}	T_{ALNS} (s)	I_{ALNS}
1	13	3	1895.08	25.41	212
2	13	5	1890.57	35.65	244
3	13	5	1916.16	37.79	238
4	13	4	1886.38	33.61	231
5	13	4	1884.13	30.50	234
6	13	5	1876.80	33.69	207
7	13	4	1881.54	31.20	217
8	13	2	1911.81	33.07	221
9	13	3	1897.89	36.08	239
10	13	4	1900.49	29.73	224

Table 4 shows that in the 10 operations of the ALNS algorithm applied to the C101_21M example, the maximum and minimum costs are 1916.16 and 1876.80, respectively. The deviation of these values from the median is 1.23% and -0.85% , while their deviation from the mean is 1.17% and -0.91% , all within acceptable ranges. Additionally, the number of dispatched EVs stabilized at 13, with the utilization rates of charging stations remaining low. The algorithm consistently produces satisfactory solutions within 250 iterations. Figure 19 illustrates the convergence curve when the known optimal solution of 1876.80 is reached, and Figure 20 presents the route maps derived from two of the optimal solutions. Overall, the ALNS algorithm developed in this study effectively solves the multi-depot EV-drone collaborative-delivery routing problem under time-varying EV travel times.

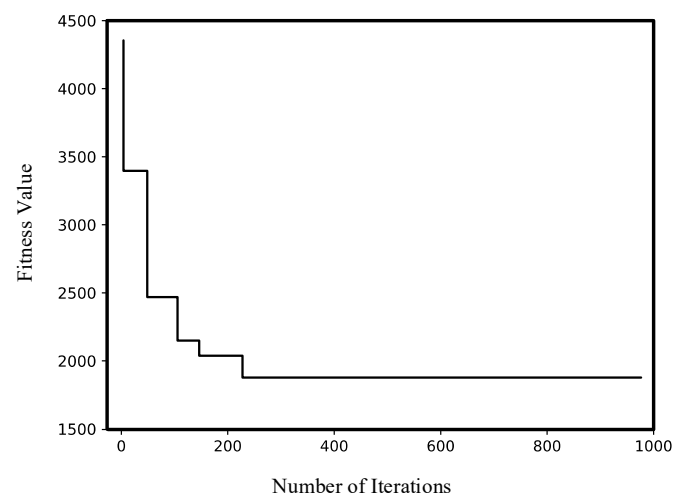


Figure 19. Iterative convergence curve of the ALNS algorithm.

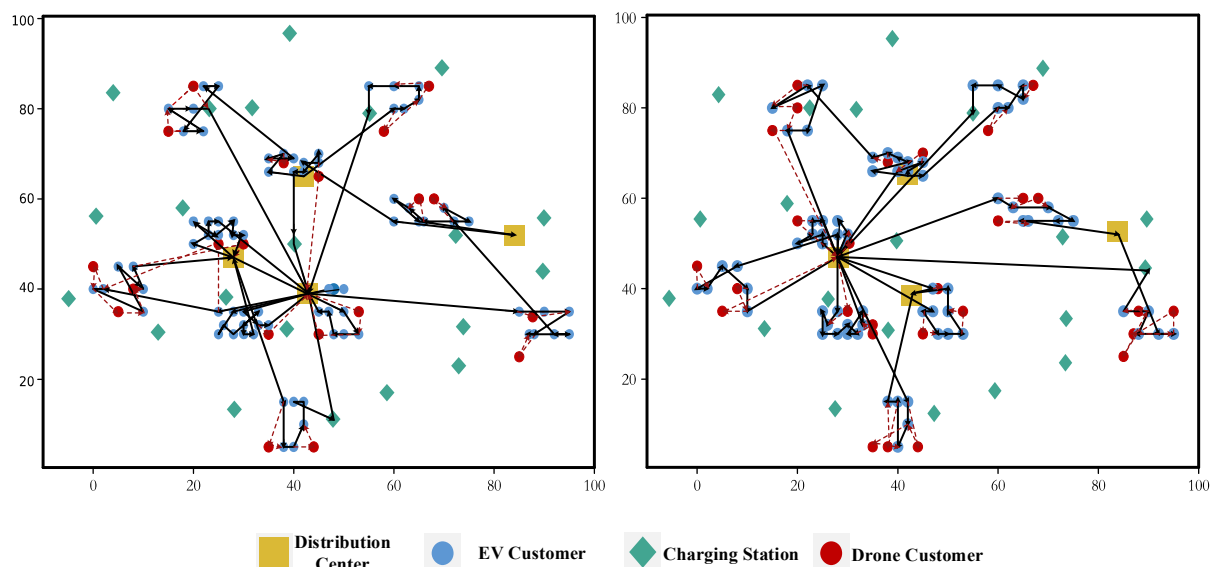


Figure 20. Route maps of two optimal solutions for C101_21M.

(2) The results of masking extreme-distance removal operator are shown in Table 5.

Table 5. Solution results of masking extreme-distance removal operator.

Series Number	G	rs	Z_{ALNS}	T_{ALNS} (s)	I_{ALNS}
1	13	3	1955.06	64.91	368
2	14	4	1950.32	59.52	304
3	13	2	1987.12	52.09	300
4	14	2	1971.62	70.07	317
5	14	2	1966.40	72.99	323
6	14	3	1972.13	80.76	410
7	14	4	1958.25	69.94	320
8	14	3	1959.25	53.68	448
9	13	3	1990.56	74.26	352
10	14	2	1987.27	50.85	333

Table 5 shows that in the results of solving the C101_21M algorithm 10 times after masking the extreme-distance removal operator, the maximum and minimum costs are 1990.56 and 1950.32, respectively. The deviations from the median are 1.09% and -0.95% , while the deviations from the mean are 1.05% and -0.99% , all falling within acceptable ranges. Additionally, the number of dispatched EVs remains stable between 13 and 14, and the utilization rates of charging stations stay low. The algorithm consistently produces satisfactory solutions within 450 iterations. This indicates that masking the extreme-distance removal operator effectively solves the multi-depot EV–drone collaborative-delivery routing problem. Figure 21 compares the results obtained after masking the operator with those of the original algorithm presented in this study.

Figure 20 illustrates the impact of the extreme-distance removal operator on the algorithm's performance, specifically regarding satisfactory solutions, algorithm solution time, and optimal number of iterations. The results are as follows:

(i) Satisfactory solution: the original algorithm achieves a minimum cost of 1876.80 and a maximum cost of 1916.16. However, after masking the extreme-distance removal operator, the minimum cost rises to 1950.32, and the maximum cost increases to 1990.56. This indicates that the cost of obtaining a solution has increased following the removal of this operator.

(ii) Algorithm solution time: the average solution time of the original algorithm is shorter than that of the algorithm after masking the extreme-distance removal operator, suggesting that this operator contributes to reducing the overall solution time.

(iii) Optimal number of iterations: the original algorithm obtains a satisfactory solution within 250 generations, while the modified algorithm requires up to 450 generations to achieve an acceptable solution. Additionally, the original algorithm reaches a stable state after 250 iterations, whereas the convergence speed of the modified algorithm slows down, necessitating more iterations to attain stability.

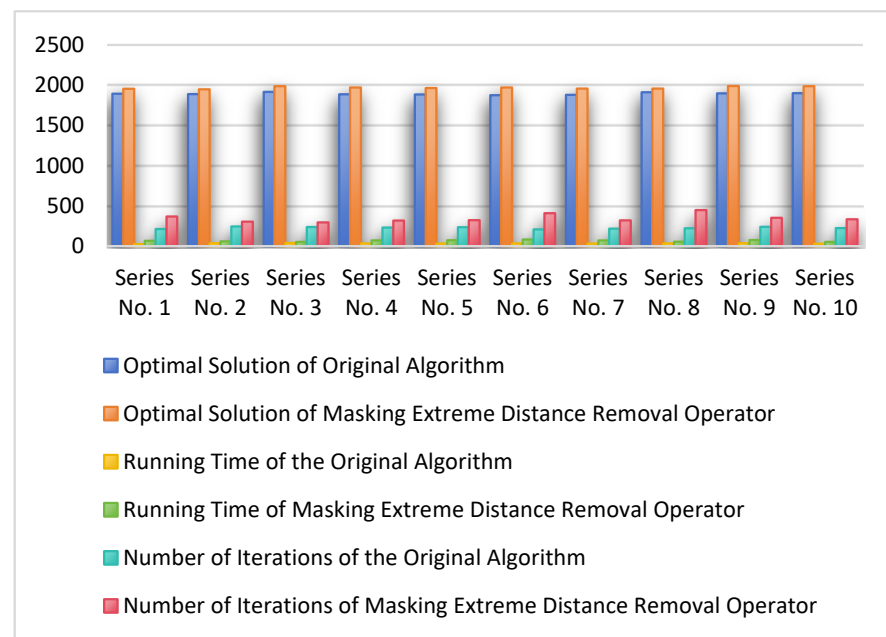


Figure 21. Comparison of the algorithm with extreme-distance removal operator masked vs. the original algorithm.

It can be concluded that the performance of the ALNS algorithm declines after masking the extreme-distance removal operator, yet it remains within an acceptable range. This underscores the operator's critical role in enhancing both the efficiency of the algorithm and the quality of the solutions obtained. The extreme-distance removal operator contributes to the algorithm's global search capability by removing specific elements from the current solution during iterations, helping to escape local optima. By increasing the exploitability of the solution space, this operator enables the algorithm to transition from a local optimum to new regions, potentially uncovering superior solutions.

In the proposed ALNS algorithm, the extreme-distance removal operator works in conjunction with other operators, forming a comprehensive set of tools. The algorithm dynamically adjusts the score of each operator based on its performance during the iteration process, prioritizing those with higher scores. This adaptive selection mechanism allows the algorithm to choose the most suitable operator for the current state at various stages, thereby enhancing both search efficiency and solution quality.

The adaptive selection of operators in the ALNS algorithm is achieved through a dynamic scoring system. The extreme-distance removal operator's score fluctuates according to its effectiveness within the algorithm, influencing its probability of being selected. This mechanism enables the algorithm to adjust its search strategy adaptively, aligning with the different stages and characteristics of the problem.

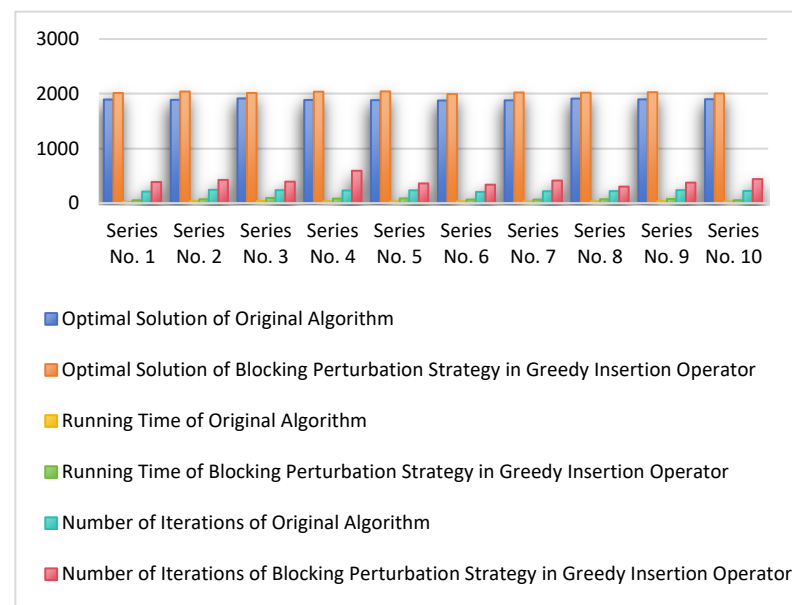
(3) The results of blocking the perturbation strategy in the greedy insertion operator are shown in Table 6.

Table 6. Results of blocking the perturbation strategy in the greedy insertion operator.

Series Number	G	rs	Z_{ALNS}	T_{ALNS} (s)	I_{ALNS}
1	14	3	2014.69	52.30	388
2	15	3	2041.81	70.85	425
3	14	2	2014.56	94.45	393
4	15	3	2039.72	82.79	592
5	15	2	2044.19	85.86	360
6	15	2	1993.70	66.67	337
7	15	3	2022.50	65.72	414
8	15	3	2021.34	72.44	302
9	15	3	2028.72	76.92	375
10	15	2	2007.44	52.72	441

Table 6 shows that after blocking the perturbation strategy in the greedy insertion operator for solving the C101_21M example, the maximum and minimum costs across 10 runs are 2044.19 and 1993.70, respectively. The deviations from the median are 1.10% and -1.40% , while the deviations from the mean are 1.05% and -1.44% , all within the acceptable range. The number of EV dispatches remains stable, between 14 and 15, with the utilization rates of charging stations consistently low.

Regarding the algorithm's iterations, a satisfactory solution is reached within 600 generations, demonstrating that the algorithm remains effective in solving the multi-depot EV-drone collaborative-delivery routing problem with time-varying characteristics, even with the perturbation strategy in the greedy insertion operator blocked. A comparison of the results between blocking the perturbation strategy and the original algorithm is shown in Figure 22.

**Figure 22.** Comparison of the algorithm with perturbation strategy blocked in the greedy insertion operator vs. the original algorithm.

From Figure 21, it is evident that the perturbation strategy affects the algorithm's performance in several key areas:

(i) Satisfactory solution: after blocking the perturbation strategy, both the maximum and minimum costs increase—the maximum rising from 1916.16 to 2044.19, and the minimum from 1876.80 to 1993.70. Despite this increase, the percentage deviation remains within an acceptable range. This suggests that the algorithm struggles to escape local optima without the perturbation strategy, impacting the solution quality. In contrast, the

perturbation strategy introduces randomness, helping the algorithm explore new regions of the solution space, thereby increasing the likelihood of finding a globally optimal solution.

(ii) Algorithm solution time: blocking the perturbation strategy significantly increases the solution time. Without the strategy, the algorithm focuses too much on the neighborhood of the current solution, slowing the convergence and increasing runtime. By enhancing exploration, the perturbation strategy allows the algorithm to escape local optima, ultimately reducing the solution time.

(iii) Optimal number of iterations: the original algorithm achieves a satisfactory solution in 250 generations, but after blocking the perturbation strategy, it requires 600 iterations to reach the same outcome. This highlights the fact that the lack of a perturbation strategy limits the algorithm's ability to search efficiently, requiring more iterations to achieve comparable results. This effect is particularly pronounced in larger solution spaces or more complex problems, where the perturbation strategy helps reduce the iterations needed to find a satisfactory solution.

To sum up, the perturbation strategy is crucial in reducing costs, accelerating convergence, and minimizing the number of iterations in the ALNS algorithm. The improved ALNS algorithm designed in this paper integrates the perturbation strategy with an adaptive selection mechanism, which dynamically adjusts the operators' usage frequency and weights, based on their historical performance. This mechanism ensures that effective perturbation strategies are applied more frequently, further optimizing the algorithm performance. It also tailors the intensity and frequency of perturbations based on the current state and historical performance, achieving a more efficient search process.

5.3. Sensitivity Analysis

5.3.1. Sensitivity Analysis of Drone Parameters

The experiment examines the impact of drone parameters on delivery costs by analyzing the results of the MD-EVDRPTW. The number of customers that meet the drone payload requirement is set at $V_D = 25$, $V_D = 50$, and $V_D = 75$. The drone's flight speed, v_d , varies from 50 km/h to 110 km/h in increments of 30 km/h, while its endurance, E_d , ranges from 30 min to 80 min, increasing by 25 min intervals. All other parameters are kept at their default values. In this context, V_D' represents the actual number of customers served by the drone, D_d and D_t denote the drone's flight mileage and the EV's driving mileage, respectively, and ERC refers to the EV's power consumption and charging cost. The results of the algorithm are presented in Table 7.

Table 7. Solution results with different drone parameters.

Series Number	V_D	V_D'	v_d	E_d	D_d	D_t	ERC	Z_{ALNS}
1	25	22	80	30	240.11	1071.20	668.96	1940.99
2	50	24	80	30	291.51	975.09	594.06	1881.51
3	75	27	80	30	353.96	785.89	478.77	1784.96
4	25	22	80	55	288.52	981.51	617.90	1904.46
5	50	23	80	55	357.55	879.40	543.82	1851.09
6	75	27	80	55	402.82	665.66	419.40	1740.25
7	25	22	80	80	274.04	978.81	613.69	1895.90
8	50	25	80	80	363.53	817.28	507.88	1816.94
9	75	28	80	80	457.43	657.46	411.48	1748.71
10	25	23	50	55	267.24	1043.73	638.27	1918.44
11	50	25	50	55	291.39	977.28	596.37	1883.79
12	75	27	50	55	370.11	702.80	427.62	1738.65
13	25	22	110	55	316.78	1028.55	635.14	1930.17
14	50	26	110	55	397.40	804.73	492.88	1812.10
15	75	29	110	55	480.52	612.97	374.86	1719.02

Based on the experiments, the sensitivity analysis of drone parameters is presented in Figure 22. By comparing the data from Table 7 and Figure 23, the following can be

observed: as the drone's flight speed increases from 50 km/h to 110 km/h, its flight mileage grows by an average of 28.64%, the EV's driving mileage decreases by 10.19% on average, electricity consumption and charging costs improve by 9.59%, and the total delivery cost is reduced by 1.44%, on average. The algorithm produces better solutions when the drone's speed surpasses the upper limit of the EV's traveling speed and the proportion of customers served by the drones increases. This occurs because the optimal solution to the MD-EVDRTW problem favors maximizing drone dispatch to improve delivery efficiency, which also helps minimize EV mileage. However, as the drone's mileage increases, the associated transportation costs rise, leading to less noticeable optimization in the overall delivery cost. Similarly, when the drone's battery life extends from 30 to 80 min, the drone's flight mileage increases by an average of 23.65%, EV driving mileage decreases by 13.37%, electricity consumption and charging costs are optimized by 11.98%, and the total delivery cost is reduced by 2.60%. These results align with the previous conclusions.

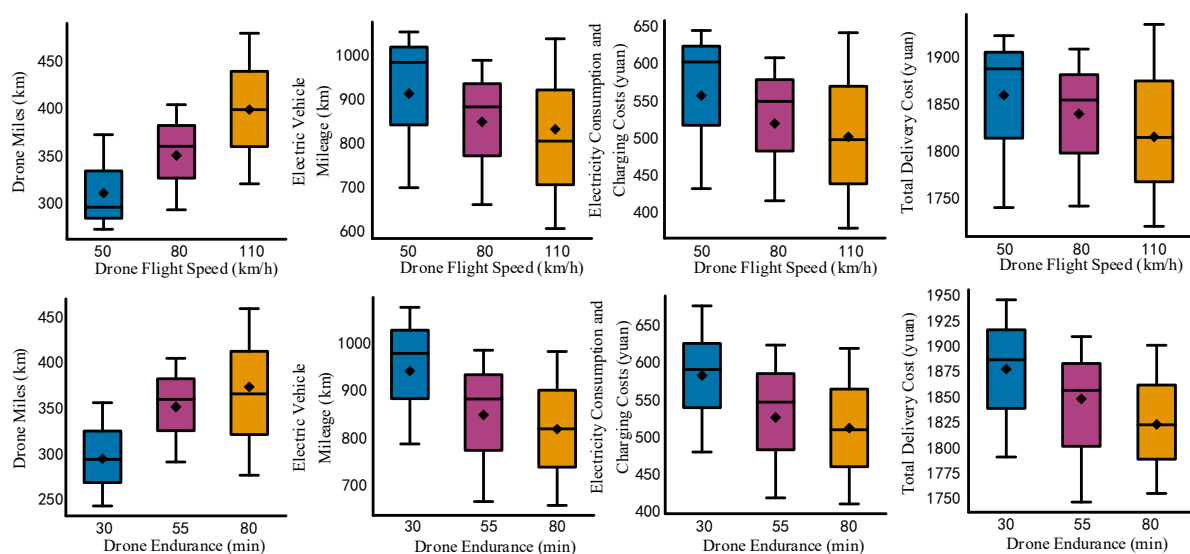


Figure 23. Sensitivity analysis of drone parameters.

As the number of customers meeting the drone payload requirements increases, the algorithm will more likely assign EV target customers to the drone service during the customer-type screening process. From the analysis results in Figure 24, it is evident that when the proportion of customers meeting the drone payload rises from 25% to 75%, the drone's flight mileage increases by an average of 48.90%. In comparison, EV mileage decreases by 32.90%, on average. Additionally, electricity consumption and charging costs improve by 33.45%, and the total delivery cost is reduced by 8.95%, demonstrating a notable optimization effect.

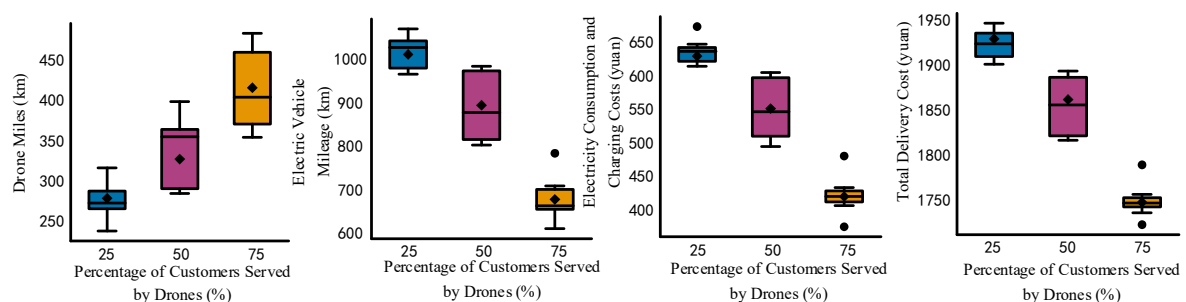


Figure 24. Sensitivity analysis of drone customers' percentage.

5.3.2. Sensitivity Analysis of EV Parameters

The experiment focuses on EV battery capacity and the charging efficiency of stations, analyzing the impact of EV parameters on delivery costs by examining the solution results of the MD-EVDRPTW. Since EV charging efficiency varies with charging time, the experiment sets the charging efficiency as a fixed value to observe the impact of different charging times on delivery costs. The EV battery capacity, E_t , is increased by 20 kWh increments, ranging from 60 kWh to 100 kWh. The charging efficiency, R_t , is represented by the slopes of nonlinear charging functions: 3.47 kWh/min, 3.77 kWh/min, and 4.87 kWh/min. All other parameters are set to default values. The results of the algorithm are presented in Table 8.

Table 8. Solution results with different EV parameters.

Series Number	V_D	V_D'	v_d	E_d	D_d	D_t	ERC	Z_{ALNS}
1	25	23	60	4.87	245.87	1022.08	664.25	1938.01
2	50	24	60	4.87	291.39	952.83	611.70	1899.12
3	75	25	60	4.87	338.50	880.55	555.33	1856.88
4	25	21	80	4.87	281.24	956.36	609.82	1894.19
5	50	23	80	4.87	364.75	862.90	546.74	1856.16
6	75	26	80	4.87	517.89	750.68	468.41	1823.77
7	25	22	100	4.87	300.82	898.79	570.27	1860.52
8	50	24	100	4.87	395.51	815.60	514.36	1833.01
9	75	25	100	4.87	482.38	716.07	445.64	1790.36
10	25	23	80	3.77	279.75	942.88	596.73	1880.65
11	50	24	80	3.77	377.61	870.77	553.46	1866.74
12	75	25	80	3.77	509.08	746.30	477.78	1830.50
13	25	22	80	3.47	277.82	954.07	604.44	1887.79
14	50	24	80	3.47	370.91	871.13	553.68	1864.95
15	75	26	80	3.47	515.58	748.86	481.32	1835.99

Based on the experiments, Figure 25 presents the sensitivity analysis results of EV parameters. The following conclusions can be drawn by combining Table 8 with Figure 25. When EV battery capacity increases from 60 kWh to 100 kWh, the drone flight range improves by an average of 34.59%, while EV driving range decreases by an average of 14.88%. Electricity consumption and charging costs are optimized by an average of 16.44%, and the total delivery cost is reduced by an average of 3.69%. When the charging efficiency increases from 3.47 kWh/min to 4.87 kWh/min, drone flight mileage shows a minimal increase of 0.04%, EV driving mileage decreases by an average of 0.16%, electricity consumption and charging costs improve by 0.88%, on average, and the total delivery cost is optimized by only 0.26%.

It can be observed that increasing EV battery capacity significantly reduces driving mileage and optimizes electricity consumption and charging costs. This is because a higher battery range reduces the reliance on charging stations during trips, minimizing detours for recharging. However, improving battery capacity and charging efficiency does not significantly reduce total delivery costs. This is due to the nature of the MD-EVDRPTW problem, which treats drones as a key decision variable, resulting in lower EV dispatch rates compared to EVDRP-type problems [17]. Consequently, the impact of battery capacity and charging technology is diminished. Additionally, the experimental settings assume an average EV charging rate of 200 kWh/h, classified as fast-charging technology, which may be cost-prohibitive for general commercial use [28]. These findings suggest that focusing on drone energy-consumption strategies is a more practical approach for enhancing EV logistics and distribution system performance, rather than relying heavily on advancements in battery and charging technologies.

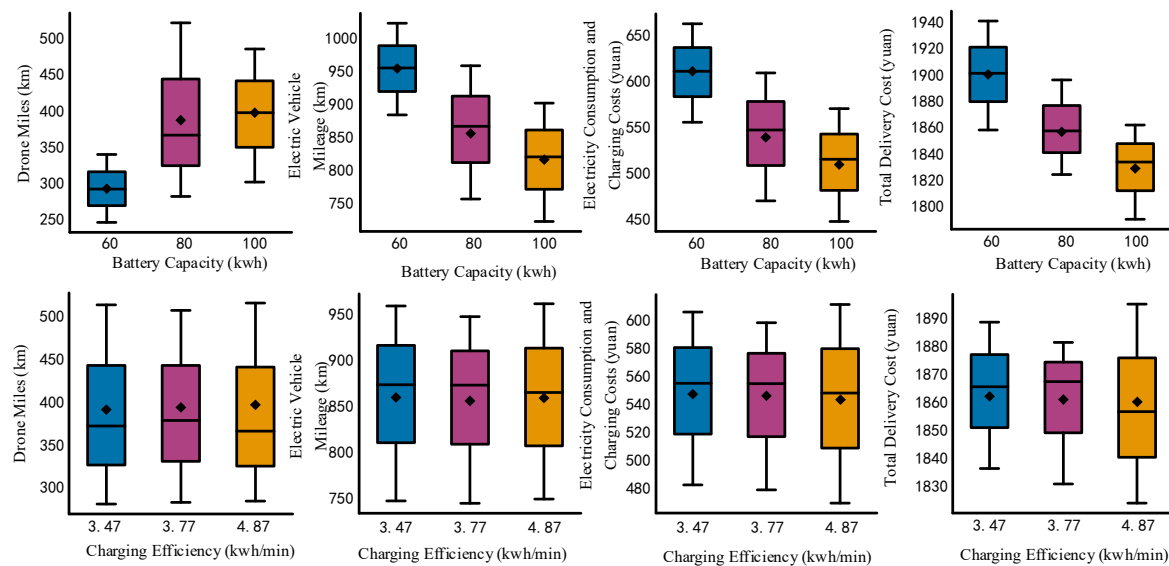


Figure 25. Sensitivity analysis of EV parameters.

5.3.3. Time-Varying Characteristics

The experiments evaluate the impact of time-varying characteristics on delivery route planning by analyzing the solution results across three EV driving scenarios: Cases 1, 2, and 3, as described in Section 3.1. All other experimental parameters are set to their default values, and the algorithm's results are presented in Table 9.

Table 9. Solution results for different time-varying scenarios.

	V_D	V_D'	G	D_d	D_t	ERC	Z_{ALNS}
Case 1	25	22	12	280.64	961.88	599.13	1883.32
	50	24	12	370.59	842.01	525.21	1836.39
	75	27	12	410.97	728.30	451.98	1775.27
Case 2	25	22	12	288.52	981.51	617.90	1904.46
	50	24	12	357.55	879.40	543.82	1851.09
	75	27	12	402.82	765.66	489.40	1810.24
Case 3	25	25	14	384.77	706.15	440.69	1956.12
	50	32	14	486.05	601.60	369.96	1915.78
	75	35	14	522.19	518.38	320.03	1876.68

Based on the experiments conducted, Figure 26 presents the sensitivity analysis results regarding the time-varying characteristics of EV driving speed. Examining Table 9 alongside Figure 26 shows that the results vary across the three time-varying scenarios. When transitioning from Case 1 to Case 2, the number of drone customers remains constant, and there is no significant change in drone flight mileage. However, the EV driving mileage, electricity consumption, and charging costs experience modest increases, averaging 3.73% and 4.75%, respectively, while the total delivery cost rises by an average of 1.28%. In contrast, when moving from Case 1 to Case 3, the number of drone customers increases, resulting in an average flight mileage increase of 31.14% for drones. Conversely, the driving mileage of EVs and the costs associated with electricity consumption and charging show significant decreases, averaging 27.88% and 28.27%, respectively, while the total delivery cost increases by an average of 4.62%.

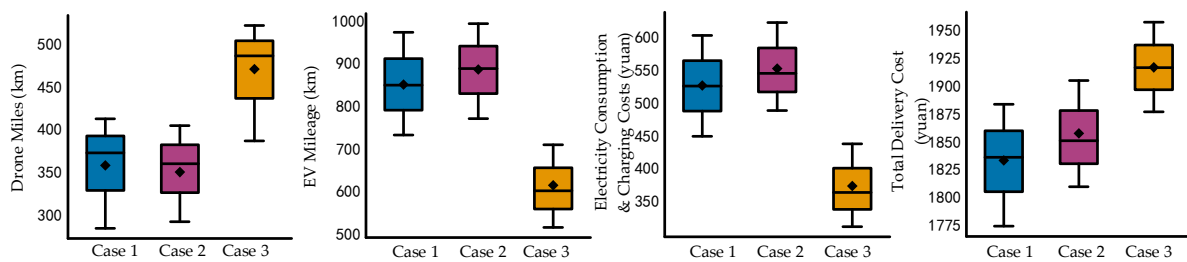


Figure 26. Sensitivity analysis results of time-varying EV travel speeds.

Overall, incorporating time-varying characteristics significantly impacts the delivery route planning, ultimately influencing delivery costs. In the shift from Case 1 to Case 2, certain customer nodes that could be served by EVs at normal speeds become invalidated, causing EVs to take detours, which leads to increased mileage and higher electricity consumption and charging costs. Despite these changes, the number of EV dispatches remains stable, due to minimal speed fluctuations between the two scenarios.

When transitioning from Case 1 to Case 3, substantial speed fluctuations result in the failure of some original routes constrained by time windows for demanding customer nodes, necessitating the re-dispatch of EVs for deliveries and increasing fixed costs. However, since the drone is a crucial decision variable, the algorithm optimizes EV delivery costs by increasing the number of drone customers. It is important to note that dispatching an additional EV also enhances drone flight mileage, which helps to mitigate the costs incurred by the EV's driving process and high fixed costs, ultimately reducing the total delivery cost despite significant fluctuations in EV travel speed. In short, the time-varying characteristics of the road network contribute to an increase in total delivery costs. Decision-makers should proactively plan routes based on actual road congestion to optimize distribution resources and avoid unnecessary expenses.

5.4. Comparative Analysis of Various Models

5.4.1. Comparison of Time-Varying Speed Model vs. Fixed-Speed Model

To demonstrate the advantages of the MD-EVDRPTW model with time-varying vehicle travel speeds, we conducted ten experiments using the computational dataset C101_21. These experiments compare the performance of the fixed-speed model and the time-varying speed model across different scenarios. In the fixed-speed model, EVs are assumed to travel at a constant speed of 60 km/h, while the time-varying speed model follows the conditions outlined in Case 2 of the time-varying scenarios. The objective of these experiments is to assess the difference in total delivery costs between the two models. This includes evaluating the costs associated with using the same route plan generated by the optimal solution of the fixed-speed model, as well as comparing it to the optimal solution of the time-varying speed model. The results, shown in Table 10, include three key cost measures: the delivery cost of the fixed-speed model's optimal solution, the delivery cost of applying the time-varying speeds to the fixed-speed optimal routes, and the delivery cost of the optimal solution from the time-varying speed model.

As shown in Table 10, there are significant differences in delivery costs between the fixed-speed model and the time-varying speed model, with the latter demonstrating clear advantages in optimizing delivery costs by accounting for traffic flow variations. The average delivery cost of the fixed-speed model's optimal solution is 1851.86. However, when these optimal routes are applied to the time-varying scenario, the cost rises substantially to an average of 1982.71, reflecting an increase of approximately 7.07%. This highlights the fixed-speed model's limitations in addressing complex traffic conditions. To further validate the time-varying speed model's effectiveness, we optimized the routes specifically for the time-varying scenario, considering traffic fluctuations and peak hours. The result was an average delivery cost of 1895.72, a reduction of 4.59% compared to using the fixed-speed model's routes in the same time-varying scenario. This demonstrates that incorporating

time-varying characteristics into routing optimization not only allows for better adaptation to complex traffic environments, but also significantly reduces total delivery costs.

Table 10. Comparative results of time-varying speed model vs. fixed-speed model.

Series Number	Delivery Costs of the Fixed-Speed Model's Optimal Solution	Delivery Costs of the Fixed-Speed Model's Optimal Routes with Time-Varying Speeds	Delivery Costs of the Time-Varying Speed Model's Optimal Solution
1	1806.41	1914.42	1885.13
2	1865.21	1999.00	1896.14
3	1870.45	2010.71	1898.44
4	1863.63	1989.32	1907.78
5	1871.51	1999.05	1911.04
6	1831.26	1998.84	1890.98
7	1856.77	1989.73	1894.31
8	1845.74	1963.47	1889.08
9	1869.33	2005.13	1896.89
10	1838.24	1957.44	1887.45

In summary, while the fixed-speed model performs better under ideal, stable traffic conditions, the time-varying speed model proves more effective in real-world scenarios where traffic dynamics are more pronounced. These findings suggest that fully considering dynamic traffic-flow changes in logistics and distribution route planning enhances both the rationality and cost-efficiency of route optimization.

5.4.2. Comparison of Single-Depot Model vs. Multi-Depot Model

To investigate the differences in delivery costs between the single-distribution center and multiple-distribution center models, we conducted ten comparative experiments to calculate the total delivery costs for each model. In the single-depot model, all deliveries are dispatched from one central distribution center, and both EVs and drones must return to that same center. In contrast, the multi-depot model allows EVs and drones to depart from multiple distribution centers, with the flexibility to return to the nearest one, providing more options for optimizing delivery routes. The experimental results, presented in Table 11, illustrate the performance of both models across ten computational experiments.

Table 11. Comparative results of single-depot model vs. multi-depot model.

Series Number	Delivery Costs of Multi-Depot Model	Delivery Costs of Single-Depot Model
1	1895.08	2320.73
2	1890.57	2337.74
3	1916.16	2279.52
4	1886.38	2306.35
5	1884.13	2200.86
6	1876.80	2343.34
7	1811.54	2323.19
8	1911.81	2287.56
9	1879.89	2334.17
10	1900.49	2345.23

The experimental results show that the average delivery cost in the multi-depot model is 1894.09, significantly lower than the 2307.87 average cost in the single-depot model. In every test case, the multi-depot model outperforms the single-depot model, demonstrating clear advantages in delivery-route optimization and cost reduction. In the single-depot model, EVs must travel longer distances for deliveries and return trips. Conversely, in the multi-depot model, EVs and drones can depart from the distribution center closest to

the customers and return to the most convenient one, allowing for greater flexibility and shorter routes, which helps minimize unnecessary return distances and total delivery costs.

Furthermore, spreading delivery tasks across multiple distribution centers reduces the load on any single center, improving the overall efficiency of EVs. This not only reduces the number of dispatches, but also lowers operational costs. Shorter routes also lead to decreased energy consumption and reduced dependency on charging stations, further cutting down on energy and charging expenses. In contrast, the centralized nature of the single-depot model results in longer routes, lower EV utilization, and higher total delivery costs.

6. Conclusions

This paper proposes a multi-depot EV–drone collaborative-delivery routing problem that incorporates time-varying EV travel times. We establish a mixed-integer programming (MIP) model to minimize the transportation costs associated with electric vehicles (EVs), charging, and drone operations. The model considers several complexities, including nonlinear charging times, time-varying travel speeds, customer time windows, carrying capacities, and operational ranges. This approach enhances the distribution process's realism and extends existing research on drone-assisted vehicle routing problems.

We develop an improved adaptive large-neighborhood search (ALNS) algorithm to solve the proposed model. This algorithm enhances global search capability and convergence speed by integrating an extremum-distance removal operator, a perturbation strategy, and an adaptive selection mechanism. The result significantly reduces computational costs and processing time, while preserving solution quality, particularly for complex optimization scenarios. The algorithm's adaptive nature allows it to dynamically adjust its search strategy based on changing problem conditions, making it particularly effective for addressing uncertainties and complexities inherent in real-world applications.

Through sensitivity analyses of drone parameters and the examination of time-varying characteristics of EVs, we demonstrate that EV–drone collaboration can lead to substantial reductions in delivery costs. However, we also observe that time-varying characteristics of the road network can increase overall delivery costs. Therefore, decision-makers must engage in proactive route planning that considers actual road congestion, optimizes distribution-resource utilization, and minimizes unnecessary expenses.

In summary, this research contributes valuable insights into integrating EVs and drones for collaborative delivery, offering practical solutions for enhancing efficiency in logistics and distribution systems. Future work could explore additional time-dependent variables or constraints to refine the model further and improve the algorithm's performance in diverse operational contexts.

While our proposed model effectively addresses the EV–drone collaborative-delivery routing problem under various constraints, several limitations remain in the current framework. First, we assume a constant unloading time across all customer nodes, which overlooks the complexity of varying unloading requirements. Future research could enhance route planning by incorporating minimum and maximum unloading times based on customer-specific factors. Second, the study is built on fixed customer demand, without considering dynamic scheduling for drones. Future studies should explore more sophisticated drone-scheduling strategies that account for real-time changes in delivery conditions. Additionally, our model does not account for drone flight-restriction areas, such as no-fly zones, which should be incorporated to better align with real-world operational constraints. Finally, due to the unavailability of real-world datasets, this study relies on simulated data. Future research should aim to validate the model using real-world case studies to assess its practical applicability.

Author Contributions: Conceptualization, Y.P.; methodology, Y.P., W.Z., D.Z.Y., S.L. and Y.Z.; formal analysis, Y.P., W.Z. and D.Z.Y.; data curation, Y.P., W.Z., S.L. and Y.Z.; writing—original draft, W.Z. and S.L.; writing—review and editing, D.Z.Y.; visualization, W.Z. and D.Z.Y.; supervision, D.Z.Y.; funding acquisition, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the Graduate Student Tutor Team Building Program in Chongqing, No. JDDSTD2022004; Graduate Student Research Innovation Program of Chongqing Jiaotong University, No. 2023S0060; Transport Science and Technology Project in Chongqing, No. CQJT-2024CZ28-3 and No. CQJT-2024CZ31-1.

Data Availability Statement: The data are available upon request.

Conflicts of Interest: All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Cantillo, V.; Amaya, J.; Serrano, I. Influencing factors of trucking companies willingness to shift to alternative fuel vehicles. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *163*, 102753. [\[CrossRef\]](#)
2. Kucukoglu, I.; Dewil, R.; Cattrysse, D. The electric vehicle routing problem and its variations: A literature review. *Comput. Ind. Eng.* **2021**, *161*, 107650. [\[CrossRef\]](#)
3. Conrad, R.G.; Figliozzi, M.A. The Recharging Vehicle Routing Problem. In Proceedings of the 2011 Industrial Engineering Research Conference, IISE Norcross, GA, USA, 21 May 2011.
4. Erdoğan, S.; Miller-Hooks, E. A green vehicle routing problem. *Transp. Res. Part E Logist. Transp. Rev.* **2012**, *48*, 100–114. [\[CrossRef\]](#)
5. Schneider, M.; Stenger, A.; Goeke, D. The electric vehicle-routing problem with time windows and recharging stations. *Transp. Sci.* **2014**, *48*, 500–520. [\[CrossRef\]](#)
6. Yang, J.; Sun, H. Battery swap station location-routing problem with capacitated electric vehicles. *Comput. Oper. Res.* **2015**, *55*, 217–232. [\[CrossRef\]](#)
7. 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. [\[CrossRef\]](#)
8. Desaulniers, G.; Errico, F.; Irnich, S. Exact algorithms for electric vehicle-routing problems with time windows. *Oper. Res.* **2016**, *64*, 1388–1405. [\[CrossRef\]](#)
9. Paz, J.C.; Granada-Echeverri, M.; Escobar, J. The multi-depot electric vehicle location routing problem with time windows. *Int. J. Ind. Eng. Comput.* **2018**, *9*, 123–136. [\[CrossRef\]](#)
10. Froger, A.; Mendoza, J.E.; Jabali, O. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. *Comput. Oper. Res.* **2019**, *104*, 256–294. [\[CrossRef\]](#)
11. Zhang, J.; Yan, J.; Liu, Y. Daily electric vehicle charging load profiles considering demographics of vehicle users. *Appl. Energy* **2020**, *274*, 115063. [\[CrossRef\]](#)
12. Lee, C. An exact algorithm for the electric-vehicle routing problem with nonlinear charging time. *J. Oper. Res. Soc.* **2021**, *72*, 1461–1485. [\[CrossRef\]](#)
13. Wang, Y.; Zhou, J.; Sun, Y. Collaborative multi-depot electric vehicle routing problem with time windows and shared charging stations. *Expert Syst. Appl. Int. J.* **2023**, *219*, 119654. [\[CrossRef\]](#)
14. Murray, C.C.; Chu, A.G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 86–109. [\[CrossRef\]](#)
15. Agatz, N.; Bouman, P.; Schmidt, M. Optimization approaches for the traveling salesman problem with drone. *Transp. Sci.* **2018**, *52*, 965–981. [\[CrossRef\]](#)
16. Wang, X.; Poikonen, S.; Golden, B. The vehicle routing problem with drones: Several worst-case results. *Optim. Lett.* **2017**, *11*, 679–697. [\[CrossRef\]](#)
17. Sacramento, D.; Pisinger, D.; Ropke, S. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transp. Res. Part C Emerg. Technol.* **2019**, *102*, 289–315. [\[CrossRef\]](#)
18. Pugliese, L.D.P.; Guerriero, F.; Macrina, G. Using drones for parcels delivery process. *Procedia Manuf.* **2020**, *42*, 488–497. [\[CrossRef\]](#)
19. Tamke, F.; Buscher, U. A branch-and-cut algorithm for the vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **2021**, *144*, 174–203. [\[CrossRef\]](#)
20. Kuo, R.J.; Lu, S.H.; Lai, P.Y. Vehicle routing problem with drones considering time windows. *Expert Syst. Appl. Int. J.* **2022**, *191*, 116264. [\[CrossRef\]](#)
21. Baek, D.; Chen, Y.; Chang, N. Energy-efficient coordinated electric truck-drone hybrid delivery service planning. In Proceedings of the International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), Turin, Italy, 18–20 November 2020; pp. 1–6.
22. Zhu, T.; Boyles, S.D.; Unnikrishnan, A. Electric vehicle traveling salesman problem with drone with partial recharge policy. *arXiv* **2022**, arXiv:2205.13735.
23. Kyriakakis, N.A.; Stamadianos, T.; Marinaki, M. The electric vehicle routing problem with drones: An energy minimization approach for aerial deliveries. *Clean. Logist. Supply Chain* **2022**, *4*, 100041. [\[CrossRef\]](#)
24. Mara, S.T.W.; Sarker, R.; Essam, D. Solving electric vehicle–drone routing problem using memetic algorithm. *Swarm Evol. Comput.* **2023**, *79*, 101295. [\[CrossRef\]](#)
25. Ichoua, S.; Gendreau, M.; Potvin, J.Y. Vehicle dispatching with time-dependent travel times. *Eur. J. Oper. Res.* **2003**, *2*, 379–396. [\[CrossRef\]](#)

26. Montoya, A.; Guéret, C.; Mendoza, J.E. The electric vehicle routing problem with nonlinear charging function. *Transp. Res. Part B Methodol.* **2017**, *103*, 87–110. [[CrossRef](#)]
27. Goeke, D. E-VRPTW Instances. Mendeley Data, V1, Published 15 April 2019. Available online: <https://data.mendeley.com/datasets/h3mrm5dhxw/1> (accessed on 12 June 2024). [[CrossRef](#)]
28. LaMonaca, S.; Ryan, L. The state of play in electric vehicle charging services—A review of infrastructure provision, players, and policies. *Renew. Sustain. Energy Rev.* **2022**, *154*, 111733. [[CrossRef](#)]

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.