applied sciences

MDPI

Article

Vehicle Routing Problem with Drones Considering Time Windows and Dynamic Demand

Jing Han [1,*], Yanqiu Liu [1] and Yan Li [2]

[1] School of Management, Shenyang University of Technology, Shenyang 110870, China; liuyanqiu@sut.edu.cn
[2] School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China; li.yan@whut.edu.cn
* Correspondence: hanjing@smail.sut.edu.cn; Tel.: +86-155-4216-0968

Abstract: As a new delivery mode, the collaborative delivery of packages using trucks and drones has been proven to reduce delivery costs and delivery time. To cope with the huge cost challenges brought by strict time constraints and ever-changing customer orders in the actual delivery process, we established a two-stage optimization model based on different demand response strategies with the goal of minimizing delivery costs. To solve this problem, we designed a simulated annealing chimp optimization algorithm with a sine–cosine operator. The performance of this algorithm is improved by designing a variable-dimensional matrix encode to generate an initial solution, incorporating a sine–cosine operator and a simulated annealing mechanism to avoid falling into a local optimum. Numerical experiments verify the effectiveness of the proposed algorithm and strategy. Finally, we analyze the impact of dynamic degree on delivery cost. The proposed model and algorithm extend the theory of the vehicle routing problem with drones and also provide a feasible solution for route planning, taking into account dynamic demands and time windows.

Keywords: truck–drone; vehicle routing problem with drones; package delivery; time windows; dynamic demand; optimization algorithm

check for updates

Citation: Han, J.; Liu, Y.; Li, Y. Vehicle Routing Problem with Drones Considering Time Windows and Dynamic Demand. *Appl. Sci.* **2023**, *13*, 13086. https://doi.org/10.3390/app132413086

Academic Editor: Alexandre Carvalho

Received: 8 November 2023
Revised: 4 December 2023
Accepted: 4 December 2023
Published: 7 December 2023

Copyright: © 2023 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

With the rapid development of drone technology, logistics companies have become increasingly interested in the field of drone delivery [1]. Drones have numerous advantages, such as not being restricted by ground obstacles, fast delivery, low cost, etc., ensuring that they can avoid traffic congestion and deliver goods to their destinations more quickly. However, the drone's load capacity is small and the battery technology cannot support extended deliveries. These limitations render it impossible for the drone to complete the delivery service independently. Therefore, most logistics and delivery companies are more inclined to adopt a collaborative delivery model of trucks and drones [2]. Under this cooperative model, trucks are not only used for large-scale delivery tasks but also serve as a "mobile platform" for drones, which undertake small-scale delivery tasks. This model can fully exploit the advantages of trucks and drones, effectively avoiding traffic congestion and road restrictions, significantly reducing labor and time costs, and comprehensively improving service efficiency. Different from traditional truck delivery, in this model, trucks and drones are synchronously combined for delivery. Trucks and drones travel at different speeds, hence impacting the delivery time window. Additionally, in the actual delivery process, the distribution center continues to receive new demands after the truck leaves, resulting in constant changes in the status of the delivery system, which requires readjustment of the existing delivery route. If the delivery route cannot be planned and adjusted reasonably and quickly, it will not only greatly increase the operating costs of the enterprise, but also lower the customer service level. In summary, it is of great practical significance to study the vehicle routing problem with dynamic demand and time window under a collaborative delivery model of trucks and drones.

The vehicle routing problem (VRP) is a classic combinatorial optimization problem that has received extensive attention in the research community [3]. With the complexity of the real situation, many varieties of VRPs have been derived, including VRP with capacity constraints (CVRP) [4], VRP with time window constraints (VRPTW) [5], heterogeneous VRP (HVRP) [6], VRP with dynamic demand (DDVRP) [7], and VRP with dynamic demand and time window (DDVRPTW) [8]. Among these, DDVRPTW was introduced relatively late, with Larsen first combining time windows with dynamic vehicle routing problems in 2000 [9]. Since then, scholars have proposed different models and solutions [10,11]. For example, Yao et al. [12] proposed a method based on bender decomposition to apply electric vehicle delivery to the VRP problem with dynamic demand. Based on the intelligent network connection and queuing theory, Wang et al. [13] proposed a predictive charging queuing probability and a path optimization model to reduce the waiting time of electric logistics vehicles, select appropriate charging locations and reduce delivery costs. However, electric vehicles (road-based ground vehicles) often lead to delivery inefficiencies due to factors such as traffic congestion and road network constraints, making it difficult to reduce delivery costs. In addition, dynamic customer orders and strict time constraints make it impossible for dispatchers to efficiently consolidate multiple orders of different sizes, while traditional vehicles can only deliver a small number of packages owing to their poor ability to quickly respond to the ever-changing dynamic demands, resulting in low flexibility and customer coverage in the delivery model. To solve this problem, Dayarian et al. [14] proposed a novel delivery service in which the truck executes the demand of the customer who has placed an order in advance while emerging dynamic demand is replenished by the drone. However, this delivery mode limits the usage scenarios of drones, seriously affects the service scope of drones, and is not conducive to the reduction of total delivery costs. Ulmer and Thomas [15] studied the problem of dynamic vehicle routing with drones under same-day delivery and proposed a geographic partition-based approximation of the policy function to decide whether an order is delivered by drone or vehicle. However, they ignored vehicle capacity constraints, which may lead to unfeasible scheduling situations such as overloaded trucks during the actual delivery.

Although there has been some research on DDVRPTW, most of these studies only considered one vehicle type. However, in DDVRP, which considers the collaborative delivery model of trucks and drones, dynamic customers are often served independently by trucks or drones, in other words, only one kind of vehicle is used to serve dynamic customers. Additionally, time window constraints are not taken into account. With the development in drone technology and the increase in delivery demand, we can more effectively deal with the changing order quantity, capacity, and location and ensure that the delivery time is within the specified time window by comprehensively considering the two factors of dynamic demand and time window. For this reason, we chose to study the Vehicle Routing Problem with Drones considering Time Windows and Dynamic Demand (VRPDTWDD). The purpose was to solve the practical challenges of delivery logistics, improve delivery efficiency, and promote the application of drone delivery. Our research question includes some technically feasible manipulations to increase drone utilization, for example, allowing drones to carry multiple packages per trip, synchronizing with trucks to meet dynamic customer demands, and allowing direct launch and retrieve drones at distribution centers. Some important and practical factors should also be considered, especially the impact of dynamic degrees on vehicle scheduling and routing. The higher the dynamic degree of customers, the greater the challenge of optimizing delivery routes.

The above factors make the VRPDTWDD problem more complicated; therefore, we established a mixed integer programming model for VRPDTWDD and proposed a Simulated Annealing Chimp Optimization Algorithm (SACHOA) with a sine–cosine operator to solve it. The algorithm uses the sine–cosine operator and the simulated annealing mechanism to avoid local optimal and improve the convergence accuracy of the algorithm. To adapt to the Hamiltonian loop routing with a branching structure under the two vehicle types, a variable-dimensional matrix encoding method, which makes the algorithm have stronger

adaptability when dealing with diverse scenarios, is introduced. This study verified the effectiveness and stability of the proposed strategy and algorithm through examples and took a self-operated brand store in Shenyang as a case for verification and analysis. Finally, we conducted a sensitivity analysis on the degree of dynamics to ensure the practical feasibility and applicability of the research results. The results of this study have significant practical implications for enhancing logistics delivery efficiency and reducing logistics costs, particularly in meeting personalized customer demands and enhancing customer experience.

The remainder of this paper is organized as follows: Section 2 presents a review of the relevant literature, followed by a mathematical formulation of the problem in Section 3. Sections 4 and 5 describe the proposed solution approach and computational results, respectively. Section 6 concludes this study and suggests potential extensions for future studies.

2. Related Literature

In this section, we briefly outline the existing literature that is most relevant to our work. Readers who require a more comprehensive summary and investigation can refer to the studies conducted by Otto et al. [16] and Macrina et al. [17].

Murray and Chu [18] were the first scholars to propose the Flying Sidekick Traveling Salesman Problem (FSTSP) and establish a Mixed Integer Linear Programming (MILP) model, which laid the foundation for the drone–vehicle routing problem that has garnered considerable attention from both academia and industry. In this problem, a truck carries a drone to serve various customers. Each drone flight can only serve one customer. Ha et al. [19] solved the FSTSP problem by modifying the objective function from minimizing delivery time to minimizing delivery costs. They proposed two heuristic methods—TSP-Local Search (TSP-LS) and Greedy Rand Adaptive Search Procedure (GRASP)—to address this issue. Agatz et al. [20] presented the traveling salesman problem with drone (TSPD) that confined the drone retrieval node to the same location as the drone launch node. They designed a hybrid algorithm combining local search and dynamic programming to solve the problem. Meanwhile, Marinelli et al. [21] considered conducting along-route operations in TSPD, implying that the implementation of launch and retrieve can be performed along the route segment to enable the drone to serve a wider area. Chang and Lee [22] focused on discovering a better route based on TSPD to achieve a broader drone delivery area. They developed a nonlinear programming model and demonstrated its effectiveness. Yurek and Ozmutlu [23] devised a decomposition-based iterative algorithm for moderately scaled TSPD problems. Kim and Moon [24] added drone stations to TSPD and proposed the traveling salesman problem with drone stations. Wang et al. [25] investigated the bi-objective TSPD problem and proposed an improved non-dominated sorting genetic algorithm to tackle this problem. Additionally, certain studies focused on exploring exact methods [26,27] and heuristic algorithms [28,29] for FSTSP/TSPD.

Kitjacharoenchai et al. [30] extended the FSTSP problem, proposed a novel variant of a truck carrying multiple drones—the multiple traveling salesman problem with drones—and established a mathematical model and heuristic algorithm for it. Murray and Raj [31] conducted a thorough study of the Multiple Flying Sidekicks Traveling Salesman Problem (MFSTSP), incorporating the energy consumption model of drones as a function of package weight, speed, and operation time. They proposed a three-stage iterative heuristic algorithm that can handle instances with up to 100 customers. Moreover, Peng et al. [32] proposed truck-assisted multi-drone package delivery and provided a hybrid genetic algorithm to solve the problem. Penna et al. [33] proposed a novel MFSTSP problem of using a parking lot to assist with truck–drone package delivery services, thereby developing an alternative solution for FSTSP. Dell'amico et al. [34], Cavani et al. [35], Tinic et al. [36], and others investigated exact and heuristic algorithms to solve the MFSTSP problem.

The study by Gonzalez et al. [37] extended the FSTSP problem by adding the factor that the drone can visit multiple customers in a single trip, that is, the truck–drone team

logistics problem, and designed an iterative greedy heuristic to solve it. Luo et al. [38] studied the multi-drone multi-visit traveling salesman problem and assumed that the energy consumption of drones is associated with the flight time, self-weight, and payload; they also proposed a multi-start tabu search to answer up to 100 customers' questions. Mara et al. [39] proposed a novel mathematical formula and a heuristic algorithm based on an Adaptive Large-scale Neighborhood Search algorithm (ALNS) to meet the challenge of path optimization for combined systems. Mahmoudi and Eshghi [40] considered the Energy-constrained Multiple-visit Traveling Salesman Problem (EM-TSPD) of multiple drones at non-customer rendezvous locations, provided a mathematical model for EM-TSPD, and designed a heuristic algorithm to handle medium and large-scale instances.

Wang et al. [41] introduced vehicle routing problem with drone (VRPD), taking into account multiple trucks and drones with the aim of minimizing the operation time, and studied the worst-case VRPD. Masmoudi et al. [42] analyzed the operating costs of VRPD and proposed an ALNS for up to 250 customers' questions. Wang et al. [43] and Tamke et al. [44] adopted exact algorithms to solve the VRPD model. Euchi and Sadok [45] provided a hybrid genetic-sweep algorithm to solve the VRPD problem with up to 200 customers' questions. Lei et al. [46] adopted a novel dynamic artificial bee colony algorithm to minimize the overall operating cost in VRPD. Schermer et al. [47] extended VRPD to each truck carrying multiple drones to provide service to all customers. Kuo et al. [48] established a bi-objective mathematical model for VRPD to minimize delivery completion time and carbon emissions and proposed a non-dominated sorting-based genetic algorithm to solve this problem.

Regarding VRPD combined with practical factors, Meng et al. [49] expanded the scope of VRPD research to incorporate a logistics system with two delivery demands: pick-up and delivery. The author introduced an energy consumption model for the drone which overall considers the flying distance of the drone and the weight of the package. They established a mixed integer programming (MIP) model with a problem customization inequality and provided a novel two-stage heuristic algorithm to solve it. Dayarian et al. [14] considered introducing dynamic demand considerations in VRPD. They suggested that trucks should handle the delivery service for customers who have already placed orders while drones should be employed to replenish any emerging dynamic demands. Ulmer and Thomas [15] explored the issue of dynamic vehicle routing with drones under the same-day delivery scheme and proposed a geographic partition-based approximation of the policy function to determine whether orders should be delivered by drones or vehicles. Kuo et al. [50] investigated the extended VRPD problem with customer time windows (VRPDTW) aimed at minimizing travel costs. They also presented a variable neighborhood search (VNS) algorithm that can tackle an instance comprising 50 customers.

As mentioned above, the current VRPD research is still in the exploratory stage. A novel vehicle routing problem with drone considering dynamic demands and time windows has yet to be examined. Additionally, the proposed VRPDTWDD proves to be highly practical as it enables each drone to serve multiple customers in a single flight. Lastly, VRPDTWDD considers the impact of the dynamic degree on vehicle scheduling and routing. Recent research findings in this field are summarized in Table 1. Among them, T represents the number of trucks, D represents the number of drones carried on each truck or the number of drones when serving customers independently with the truck, and m is the abbreviation for "multiple". SV/MV represents the number of customers per trip (i.e., single visit or multiple visits). Column TW defines whether the problem considers time windows. Column DD defines whether the problem considers dynamic demands.

Table 1. Summary of related studies.

Reference	T	D	SV/MV	TW	DD	Objective	Method
Murray and Chu [18]	1	1	SV	no	no	Min time	MILP, Heuristic
Ha et al. [19]	1	1	SV	no	no	Min cost	MILP, GRASP
Agatz et al. [20]	1	1	SV	no	no	Min time	MILP, Approximation algorithm
Yurek and Ozmutlu [23]	1	1	SV	no	no	min time	MILP, Heuristic
Kitjacharoenchai et al. [30]	m	m	SV	no	no	Min time	MIP, Heuristic
Murray and Raj [31]	1	m	SV	no	no	Min time	MILP, Heuristic
Gonzalez et al. [37]	1	1	MV	no	no	Min time	MIP, Heuristic
Luo et al. [38]	1	m	MV	no	no	Min time	MILP, Heuristic
Wang et al. [41]	m	m	SV	no	no	Min time	Worst-case analysis
Masmoudi et al. [42]	m	1	MV	no	no	Min cost	MILP, ALNS
Schermer et al. [47]	m	m	SV	no	no	Min time	MILP, Metaheuristic
Kuo et al. [48]	m	m	SV	yes	no	Min cost	MIP, VNS
Ulmer and Thomas [15]	m	m	SV	no	yes	Min time	Function approximation
Dayarian et al. [14]	1	1	SV	no	yes	maximize the number of orders delivered	Heuristic
This paper	m	m	MV	yes	yes	Min cost	Two-stage optimization model, SACHOA

3. Model

3.1. Problem Description

VRPDTWDD can be described as follows: there is a distribution center in a city's logistics network. The distribution center has an unlimited number of homogeneous trucks and drones. These trucks and drones provide package delivery services for customers with time window constraints. First, the distribution center formulates a delivery route and delivers the goods to the static customers who booked the previous day. In the process of delivery, it needs to re-plan the route of the vehicles being distributed or dispatch new vehicles to provide services if there is new customer demand. Static customers refer to customers whose demand remains unchanged in the delivery cycle; dynamic customers include new customers added during the delivery process and new customers derived from the new demands of the original customers. The ratio of dynamic customers to all customers is called the dynamic degree. The dynamic degree will increase with time, and the truck must decide on whether to respond to dynamic customers. The goal of this problem is to find the optimal route to minimize the delivery costs.

Before establishing the VRPDTWDD model, certain operational configurations need to be given:

- The logistic network including static and dynamic customers.
- Each customer (node) needs to be served once by a truck or drone.
- Vehicle access customer needs to satisfy time window constraints.
- Each truck and its drone are deployed in pairs and must leave and return to a distribution center once. For each pair, the drone is equipped atop a truck and can only be launched and retrieved from the same truck.
- Drones can be launched and retrieved at the customer node or the distribution center.
- Each drone has multiple non-overlapping flights in a collaborative route, with a maximum of one launch and retrieve for each customer.
- During the delivery process, the drone can pick up one or more packages from the truck (multiple packages need to meet load and battery capacity constraints), take off

to deliver the packages to the customer when the truck stops, and return to the truck at a subsequent customer node (including the distribution center).
- When the truck and its drone regroup at the retrieve node, whoever arrives first will wait for the other.
- The truck and drone both maintain a constant speed during the delivery process, and the drone needs to replace the battery immediately after merging to ensure the next flight.
- To simplify the study, the impact of traffic congestion is not considered.
- In this paper, VRPDTWDD is assumed to be explored in two-dimensional (2D) space.

Figure 1 is a simple schematic diagram of VRPDTWDD. The entire delivery process can be divided into two time periods: $[T_a, T_b]$, $[T_b, T_c]$. First, perform path planning for some static customers in time period $[T_a, T_b]$. That is, one or more trucks carrying configured drones and packages depart from the distribution center, deliver the packages according to the planned route, and return to the distribution center. Assume that dynamic customers appear at time T_b, then perform path planning for the remaining customers in time period $[T_b, T_c]$. The mathematical symbols and meanings used in this paper are given in Table 2.

Table 2. Model symbols and explanations.

Notation	Description
	Sets
V	Collection of all nodes, where 0 and $n+1$ are distribution centers $V = \{0, 1, 2, \cdots, n, n+1\}$
C	Collection of all customers $C = V \backslash \{0, n+1\}$
A	Set of all arcs $A = \{(i,j) \mid i \in V, j \in V, i \neq j\}$
K	Collection of all trucks $K = \{1, 2, \cdots, \mathrm{k}\}$
P	Collection of all drone trips $P = \{1, 2, \cdots, \mathrm{p}\}$
	Parameters
C_k / C_p	The unit delivery cost for the truck/drone
d'_{ij} / d_{ij}	The distance between the truck/drone and customer i and customer j. Due to the limitation of obstacles on the ground, the distance covered by the drone is calculated using the European formula and that covered by the truck is calculated as $d'_{ij} = \eta \cdot d_{ij}$
Q_k / Q_p	Rated load capacity of truck/drone
v_k / v_p	Traveling/flying speed of truck/drone
a_i / b_i	The earliest/latest service start time for customer i
q_i	Demand of customer i
s_i	Service time for customer i
$D_{\max}$	The maximum flight distance of the drone
	Variables
$x_{ij}^k (= 1)$	Truck k drives from customer i to customer j
$y_{ij}^{kp} (= 1)$	The p-th trip of the drone carried by truck k runs along the arc (i, j)
$h_{ip}^{kS} / h_{ip}^{kE} (= 1)$	Customer i is the launch/retrieve node for the drone trip p carried by truck k, where S and E are used to distinguish launch and retrieve nodes
$Z_i^{kp} (= 1)$	The p-th trip service of the drone carried to customer i by truck k
$Z_i^k (= 1)$	Customer i is served by truck k
t_i	Time when the vehicle (truck–drone) arrives at customer i

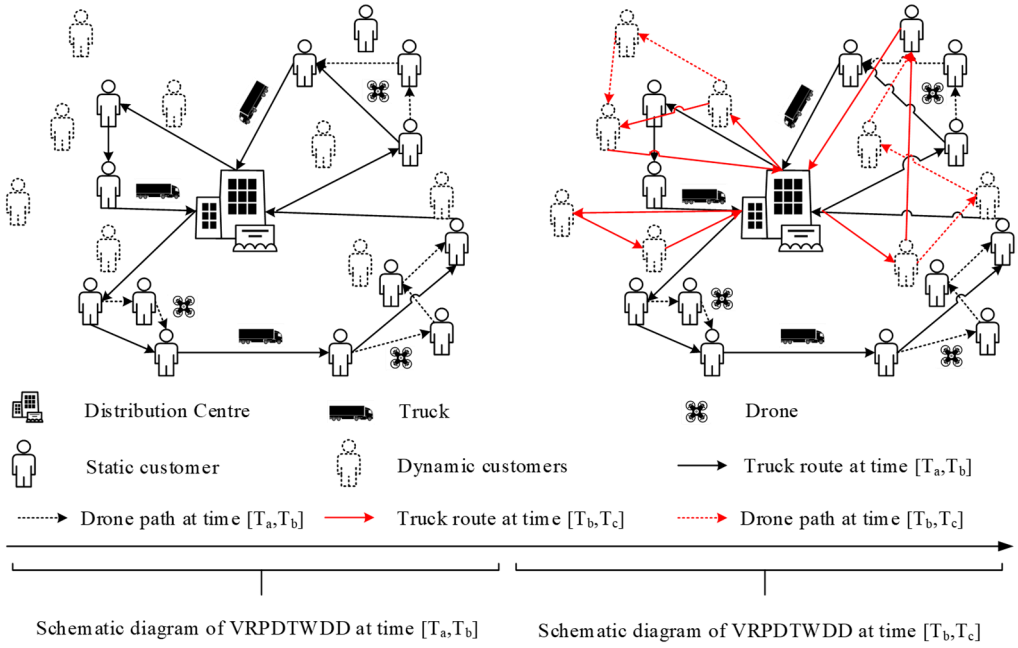


Figure 1. A simple diagram of VRPDTWDD.

3.2. Mathematical Model

3.2.1. Initial Planning Stage

A two-stage vehicle route planning strategy is adopted for VRPDTWDD. The first stage is the initial planning stage. The initial planning stage is mainly for route planning for static customers who have made reservations, which can be regarded as a static VRPDTW problem. The optimization model in the initial stage of construction is as follows:

$$\min f = C_k \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} d'_{ij}x^k_{ij} + C_p \sum_{k\in K}\sum_{p\in P}\sum_{i\in V}\sum_{j\in V} d_{ij}y^{kp}_{ij} \tag{1}$$

s.t.

$$\sum_{k\in K}\sum_{p\in P} Z^{kp}_i + \sum_{k\in K} Z^k_i = 1,\ \forall i \in C, \tag{2}$$

$$\sum_{j\in V} y^{kp}_{ij} \geq Z^{kp}_i,\ \forall i \in C, p \in P, k \in K, \tag{3}$$

$$Z^k_i = \sum_{j\in V} x^k_{ji},\ \forall i \in C, k \in K, \tag{4}$$

$$h^{kE}_{ip} + \sum_{j\in V} y^{kp}_{ij} = h^{kS}_{ip} + \sum_{j\in V} y^{kp}_{ji},\ \forall i \in C, p \in P, k \in K, \tag{5}$$

$$\sum_{j\in V} y^{kp}_{ij} = \sum_{j\in V} y^{kp}_{ji} \leq 1,\ \forall i \in V, p \in P, k \in K, \tag{6}$$

$$\sum_{j\in V} x^k_{ij} = \sum_{j\in V} x^k_{ji} \leq 1,\ \forall i \in V, k \in K, \tag{7}$$

$$Z^{kp}_i + Z^{kp}_j \geq 1 - M(1 - y^{kp}_{ij}),\ \forall (i,j) \in A, p \in P, k \in K, \tag{8}$$

$$h^{kS}_{ip} \geq 1 - M(2 - y^{kp}_{ij} - Z^{kp}_j + Z^{kp}_i), \forall i \in V, j \in C, i \neq j, p \in P, k \in K, \tag{9}$$

$$h^{kE}_{jp} \geq 1 - M(2 - y^{kp}_{ij} - Z^{kp}_i + Z^{kp}_j), \forall j \in V, i \in C, i \neq j, p \in P, k \in K, \tag{10}$$

$$\sum_{p\in P} h_{ip}^{kS} \le M \sum_{j\in V} x_{ji}^{k},\ \forall i \in V, k \in K, \tag{11}$$

$$\sum_{p\in P} h_{ip}^{kE} \le M \sum_{j\in V} x_{ij}^{k},\ \forall i \in V, k \in K, \tag{12}$$

$$h_{ip}^{kS} + h_{ip}^{kE} \le 1,\ \forall i \in V, p \in P, k \in K, \tag{13}$$

$$\sum_{i\in V} h_{ip}^{kS} = \sum_{i\in V} h_{ip}^{kE} \le 1,\ \forall p \in P, k \in K, \tag{14}$$

$$\sum_{i\in C} q_i Z_i^k + \sum_{i\in C} q_i Z_i^{kp} \le Q_k,\ \forall p \in P, k \in K, \tag{15}$$

$$\sum_{i\in C} q_i Z_i^{kp} \le Q_p,\ \forall p \in P, k \in K, \tag{16}$$

$$\sum_{l\in V}\sum_{i\in C} d_{li} h_{lp}^{kS} + \sum_{i\in C}\sum_{j\in C} d_{ij} Z_i^{kp} + \sum_{j\in C}\sum_{m\in V} d_{jm} h_{mp}^{kE} \le D_{\max}, \forall p \in P, k \in K, \tag{17}$$

$$a_i \le t_i \le b_i,\ \forall i \in C, \tag{18}$$

$$t_i = \max\left\{t_l + s_l + d'_{li}/v_k, t_m + s_m + d_{mi}/v_p, a_i\right\}, \forall i \in V, l \in V, m \in V, \tag{19}$$

$$x_{ij}^k \in \{0,1\}, y_{ij}^{kp} \in \{0,1\}, Z_i^k \in \{0,1\}, Z_i^{kp} \in \{0,1\}, h_{ip}^{kS} \in \{0,1\}, h_{ip}^{kE} \in \{0,1\}, \quad \forall i \in V, j \in V, p \in P, k \in K, \tag{20}$$

$$t_i \ge 0,\ \forall i \in V. \tag{21}$$

Objective function 1 indicates that the total distribution cost is minimized. The first part is the delivery cost of the truck, and the second part is the delivery cost of the drone. Constraint 2 means that each customer must be served by a truck service or a drone, and is not allowed to receive both services at the same time. Constraints 3 and 4 give the relationship between the decision variables of trucks and drones. Constraint 5 represents the entry and exit balance constraints of the drone trip, including launch and retrieve nodes. Constraint 6 requires that each edge be visited by the drone at most once. Constraint 7 not only maintains the entry and exit balance constraints of the truck route but also guarantees that the truck can visit a customer point at most once. Constraint 8 requires that at least one customer is served in the drone trip to avoid the drone flying along the truck's route unnecessarily. Constraints 9 and 10 determine the launch and retrieve nodes of the drone trip, respectively. Constraints 11 and 12 ensure that the truck must visit the launch and retrieve nodes of the drone. Constraint 13 ensures that the launch and retrieve nodes in the same drone trip cannot be the same node. Constraint 14 ensures that the number of launch and retrieve nodes in the same drone trip is equal and there is at most one. Constraints 15 and 16 represent the loading constraints of the truck and drone, respectively. Constraint 17 is the flight capability constraint of a drone. Constraint 18 ensures that the time when the vehicle provides service to the customer must be within the time window required by the customer. Constraint 19 gives the calculation formula for the time to arrive at a certain customer node. Constraints 20 and 21 represent the value range for variables.

3.2.2. Dynamic Planning Stage

The second stage is the dynamic planning stage. After the dynamic time window is opened, the distribution center will continue to receive dynamic customer information. The dynamic degree increases as time passes. Therefore, delivery personnel need to decide

whether to respond to dynamic customers. When the dynamic degree is small, choosing to respond to dynamic customers can quickly satisfy their needs, but the delivery cost will also increase. When the dynamic degree is large, responding to dynamic customers can reduce delivery costs, but the demand cannot be met quickly. To better balance cost and service speed, we adopt a regular update strategy that divides time domains. We divide the dynamic time window of the distribution center into time domains with the same duration. At the end of each time domain, the received dynamic demands are processed according to the response strategies of different dynamic demands. Subsequently, based on the location of the current vehicle service customer as the starting point, the route is re-planned for the customers whose deliveries have not been met thus far as well as the dynamic customers in the previous period. In this way, we can transform the dynamic VRPDTWDD problem into multiple static VRPDTW problems.

Dynamic demand includes various situations, including adding new customer demands, canceling customer demands, adding new demands from existing customers, adjusting the time windows, and changing the delivery address. For different types of dynamic demands, the specific response strategy is as follows: we add new customer demands to the delivery network. If the dynamic level is low at this time, we insert the new demand customer node into the original delivery scheme for repair; if the level of dynamic degree is high, then the dynamic problem is converted into a multi-stage static sub-problem based on time domain division for processing. When the new demand has no impact on the vehicle capacity and time windows, the dynamic degree is defined as a lower level. On the contrary, when the new demand violates the vehicle capacity or time windows, the dynamic degree is defined as a higher level. We delete customers with canceled demand directly from the delivery network. New demands from existing customers and changes in delivery addresses are treated as customer nodes that are canceled first and subsequently added. For customers who adjust the time windows, we make the corresponding adjustments in the delivery network.

4. Solution Approach

Compared with other dynamic event update strategies, the two-stage vehicle route planning strategy (as shown in Figure 2) can dynamically adjust the delivery route while reducing the amount of calculation with higher flexibility. The problem in the initial planning stage is an NP-Hard problem. As customer size increases, the number of feasible solutions grows exponentially, making it challenging for exact algorithms to obtain optimal solutions efficiently. To solve this problem, this study chose the chimp optimization algorithm, which is simple in principle and has few parameters, to solve it. However, the traditional chimp optimization algorithm has problems with easily falling into local optimum and low convergence accuracy, and it is not directly applicable to optimizing both trucks and drones. Therefore, this study improves the chimp optimization algorithm by combining the characteristics of the collaborative truck and drone delivery problem to obtain the delivery scheme in the initial planning stage. In the dynamic planning stage, the K-means clustering method is used to cluster all customer nodes, and subsequently, the corresponding strategy is used according to the regional dynamic degree level. For specific strategies, see Section 3.2.2.

4.1. Chimp Optimization Algorithm

The chimp optimization algorithm (CHOA) is a swarm intelligence optimization algorithm based on chimpanzee hunting behavior that is used to solve optimization problems [51]. During hunting, chimpanzees find prey by driving, blocking, chasing, and surrounding them to meet the individual's needs. This process involves four types of chimpanzees, namely attackers, barriers, chasers, and drivers, representing the optimal, suboptimal, third, and fourth optimal solutions of the population, respectively. In other words, the process of using CHOA to search for a solution can be regarded as the four types of chimpanzees, constantly updating their individual positions until a satisfactory

solution is found. Before using the algorithm, the above process needs to be modeled first. The mathematical model is as follows:

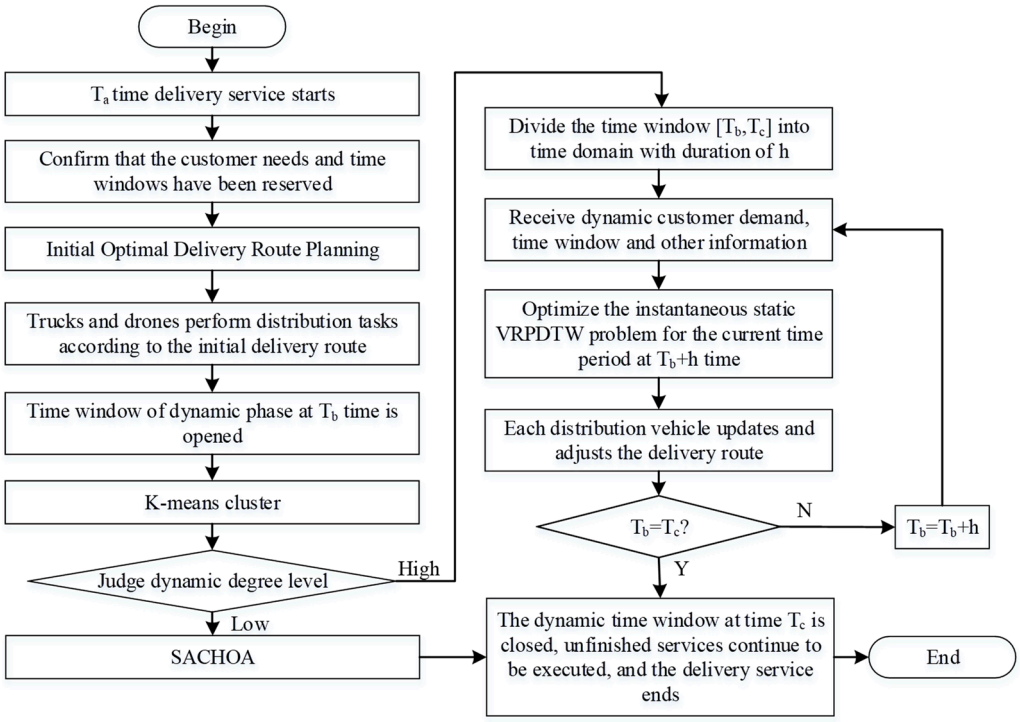


Figure 2. Schematic diagram of the two-stage vehicle route planning strategy.

In the D-dimensional search space, assuming that there is a population composed of NIND chimpanzees $X = \{X_1, X_2, \cdots, X_{NIND}\}$, the position of the *i*-th chimpanzee is defined as $X_i = (X_i^1, X_i^2, \cdots, X_i^D)^T$, and $X_i^j (j = 1, 2, \cdots, D)$ is the component of the *j*-th dimension of the *i*-th chimpanzee. In the process of driving, blocking, and chasing prey, the behavior of chimpanzee *i* approaching the prey can be described by Formulas (22) and (23):

$$d_i^j = \left| c_i^j X_{prey}^j(gen) - m_i^j X_i^j(gen) \right|, \tag{22}$$

$$X_i^j(gen+1) = X_{prey}^j(gen) - a_i^j d_i^j. \tag{23}$$

where d_i^j represents the distance between chimpanzee *i* and prey in dimension *j*; X_{prey}^j is the position of prey; *gen* is the number of iterations; *a*, *m* and *c* are coefficient vectors determined by Formulas (24)–(27).

$$a_i^j = 2fr_1, \tag{24}$$

$$c_i^j = 2r_2, \tag{25}$$

$$f = 2 - \frac{2gen}{MAXGEN}, \tag{26}$$

$$m_i^j = Chaotic_value. \tag{27}$$

where f is a nonlinear disturbance factor; $MAXGEN$ is the maximum number of iterations; r_1, r_2 is the random number within the range [0, 1]; a_i^j is a random variable reflecting the distance between chimpanzees and prey; c_i^j is a random variable reflecting the impact of

prey position on chimpanzees' position; and m_i^j is a chaotic vector representing the impact of chimpanzees' neutral movement during hunting.

Finally, the population satisfies the demand by attacking the prey to obtain the optimal solution. A mathematical model of a chimpanzee attacking prey looks like this:

$$X_{i,1}^j = X_{Attacker}^j - a_{i,1}^j \left| c_{i,1}^j X_{Attacker}^j - m_{i,1}^j X_i^j \right|, \tag{28}$$

$$X_{i,2}^j = X_{Barrier}^j - a_{i,2}^j \left| c_{i,2}^j X_{Barrier}^j - m_{i,2}^j X_i^j \right|, \tag{29}$$

$$X_{i,3}^j = X_{Chaser}^j - a_{i,3}^j \left| c_{i,3}^j X_{Chaser}^j - m_{i,3}^j X_i^j \right|, \tag{30}$$

$$X_{i,4}^j = X_{Driver}^j - a_{i,4}^j \left| c_{i,4}^j X_{Driver}^j - m_{i,4}^j X_i^j \right|, \tag{31}$$

$$X_i^j(gen+1) = \frac{X_{i,1}^j + X_{i,2}^j + X_{i,3}^j + X_{i,4}^j}{4}. \tag{32}$$

where $X_{Attacker}^j, X_{Barrier}^j, X_{Chaser}^j, X_{Driver}^j$ represent the positions of the attacker, barrier, driver, and chaser on the j-th dimension, respectively, and $X_i^j(gen+1)$ represents the updated position of the (gen + 1)-th generation chimpanzee on the j-th dimension.

4.2. Simulated Annealing Chimp Optimization Algorithm with Sine–Cosine Operator

The traditional CHOA is mainly suitable for solving continuous optimization problems. The VRPDTWDD problem is a combinatorial optimization problem, thus CHOA cannot be directly applied to VRPDTWDD. To solve this problem, we need to design an appropriate discrete strategy. Furthermore, there exists a Hamiltonian loop with a branching structure in the solution of the VRPDTWDD problem, i.e., the same nodes appear in the truck route and the drone route (such as customer 5 in Figure 3). However, this branch route structure cannot be directly compiled and expressed. Therefore, we need to design a new initial solution deconstruction method and encoding method. Based on the principle of the algorithm, other chimpanzees estimate the location of the prey based on the location of the attacker and then update their own location to hunt. To further improve the convergence accuracy of the algorithm, this study introduces a sine–cosine operator to update the attacker's position. Meanwhile, to avoid local optimum, this study also introduces a simulated annealing mechanism. To sum up, we propose an improved chimp optimization algorithm named Simulated Annealing Chimp Optimization Algorithm with Sine–Cosine Operator (SACHOA). It includes the following main key steps:

4.2.1. Initial Solution Construction and Variable-Dimension Matrix Encode

The process of generating the initial solution is divided into two stages: the generation of a truck route and the generation of a drone route.

Truck route generation method: Use the "sorting first, then grouping" approach to construct truck routes. That is, first randomly sort all customer nodes, and then divide and determine the specific customer nodes to be served based on truck capacity.

Drone route generation method: First, all customer nodes are divided into two categories based on the maximum flight distance and the maximum load of the drone: One is served by a drone, $N^u = \{i | q_i \le Q_u \& d_{ij} \le D_{\max}/2, \forall i, j \in V\}$; and the other is served by trucks, $N^k = V - N^u$. Then, the algorithm needs to remove the drone service node and its front and rear nodes from the truck route and add them to the drone route. Next, duplicate nodes are removed from the drone route, and the remaining nodes are served by the truck in the original order. Finally, the algorithm judges the loading capacity and the maximum flight distance of the drone route and only keeps the legal drone route. This forms a complete truck–drone solution.

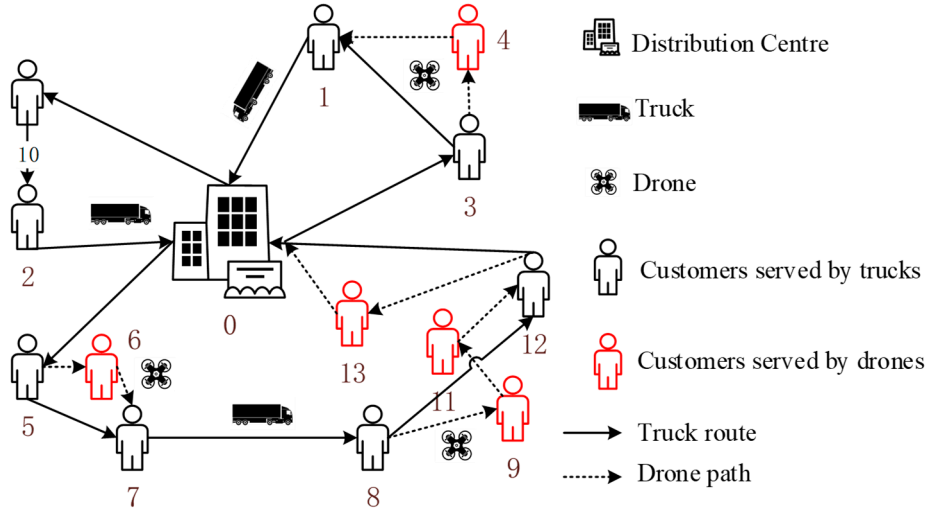


Figure 3. A diagram of possible truck and drone collaborative delivery routes.

The variable-dimension matrix encode is a $p * q$ dimension matrix structure where $p = l + k + 1$ represents the total number of truck customer nodes, and l represents the number of trucks used. $q = \max r(i) + 1$, where $r(i)$ represents the number of drone customers corresponding to the i-th retrieving node. To display the variable-dimensional matrix encode more intuitively, this paper provides the route structure in Figure 3; the corresponding coding is shown in Figure 4.

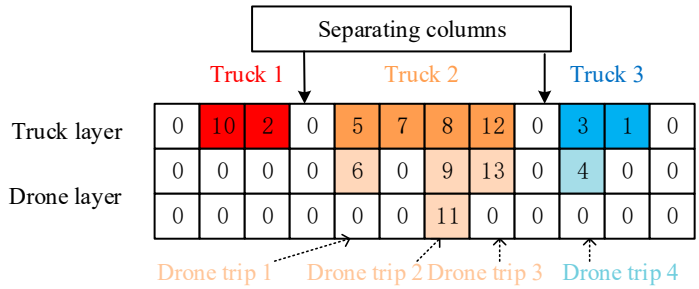


Figure 4. Route encoding diagram.

The encode in Figure 4 is a multi-layer structure. The first layer is the truck layer, which represents the access node sequences for all trucks. The other layers, except the first layer, are the drone layers that represent docking and access node sequences selected by all drones. The first row of the drone column is the launch node and the first row of the subsequent drone column is the retrieval node. The decoded routes in Figure 4 are truck route 1 (0→10→2→10), truck route 2 (0→5→7→8→12→0), drone trip 1 carried by truck 2 (5→6→7), drone trip 2 (8→9→11→12), drone trip 3 (12→13→0), truck route 3 (0→3→1→0), and drone trip 4 carried by truck 3 (3→4→1). The variable-dimension matrix encode can clearly show the collaborative delivery route of multiple trucks and multiple drones. This encoding method can also be extended to other combinatorial optimization problems with branch structures.

4.2.2. Group Initialization

The population is randomly initialized and divided into four independent groups based on the sequence of individual numbers. The number of individuals in the first three subgroups is $\lfloor NIND/4 \rfloor$, and the remaining individuals are put in the last subgroup. Such division criteria help to determine the specific areas where the four types of individuals are attackers, blockers, chasers, and drivers, and allow other chimpanzees to self-renew according to these four types of leaders.

4.2.3. Chimpanzee Location Updates

For continuous optimization problems, individual chimpanzees use Formulas (23) and (32) to update their positions; however, for combinatorial optimization problems, it is obvious that the above formulas cannot be used directly. Therefore, it is necessary to combine the characteristics of VRPDTWDD and chimpanzee positions to update positions. At the same time, the selection, crossover, mutation, and local search operations of the genetic algorithm are used to update the chimpanzee position.

In this stage, we simulated the learned behavior of chimpanzees following the leader's obedience during a hunt and the chaotic behavior that emerges after being socially motivated. First, we select some individuals in each population as leaders based on their degree of fitness and use Order Crossover (OX) to realize that other individuals follow the leader to learn. In this way, we show the guiding role of outstanding individuals over other individuals. In addition, during group hunting, some individuals forget their hunting responsibility due to social incentives when they catch food, for which we propose mutation and local search operations. These operations are conducive to maintaining the diversity of the population and preventing the algorithm from falling into a local optimal solution. It is worth noting that after crossover, mutation, and local search, it is necessary to construct the drone route for the individual based on the method in Section 4.2.1.

OX operation: Randomly select the start and end positions in the two parent chimpanzees, copy the gene in this region of parent chimpanzee 1 to the same position in offspring 1, and then copy the gene in the region of offspring 1 to parent chimpanzee 2. Missing genes are filled in in order. Another offspring is obtained in a similar manner.

Mutation operation: randomly select two gene positions to swap in chimpanzees.

Local search operation: First remove several nodes from the current solution using the correlation removal operator and then reinsert them into the corrupted solution using the repair operator. See (33) for the correlation calculation formula. The idea of the repair operator is to insert the removed nodes back to the insertion position that increases the objective function the least.

$$R(i,j) = 1/(\overline{d_{ij}} + V_{ij}), \tag{33}$$

$$\overline{d_{ij}} = \frac{d_{ij}}{\max d_{ij}}. \tag{34}$$

Among them, $\overline{d_{ij}}$ is the value after normalization; d_{ij} is the Euclidean distance between i and j; V_{ij} is whether i and j are on the same route, and if i and j are on the same route, then $V_{ij} = 1$.

4.2.4. Adding a Sine–Cosine Operator When Updating the Attacker's Position

In traditional CHOA, the position of the attacker is very critical to finding the optimal solution. Other chimpanzees in the population change based on the location of the attackers. When the attackers fall into the local optimum, the population will also fall into the local optimum. In this study, the sine–cosine operator is added to the attacker's position change, and a nonlinear learning factor (see Figure 5 for the value of 200 iterations) is introduced. The formula for the nonlinear learning factor and the sine–cosine operator is as follows:

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \sin(gen \times \pi / MAXGEN), \tag{35}$$

$$p_m = \omega \sin(2\pi r1), \tag{36}$$

$$p_c = \omega \cos(2\pi r2). \tag{37}$$

where p_c is the probability of crossing and p_m is the probability of mutating.

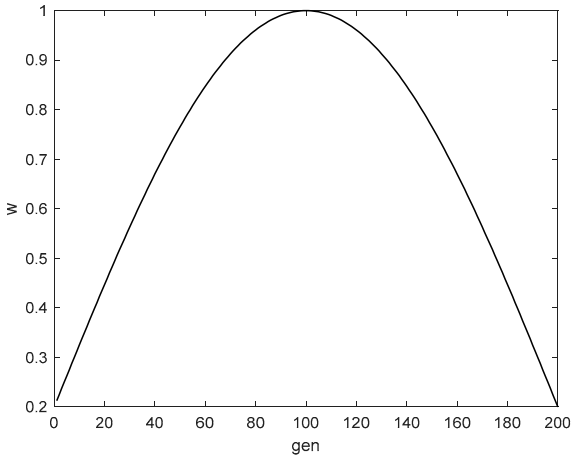


Figure 5. Distribution of nonlinear learning factor values for 200 iterations. During the early stage of the search (0–100 generations), a larger value is helpful for global exploration; during the later stage of the search (100–200 generations), a smaller value is helpful for improving local development capabilities.

The pseudocode of the attacker's position update is shown in Figure 6. First, the crossover probability p_c and mutation probability p_m are obtained according to Formulas (35)–(37). If $p_c > p_m$, then use the OX operation to update the attacker's position. Otherwise, use the mutation operation to update the attacker's location.

Input	$gen, \omega_{\max}, \omega_{\min}, MAXGEN$, and Population size $NIND$
Output	$Chrom_new$
1)	for $j = 1 : NIND$
2)	Update p_m and p_c according to Formula (35–37)
3)	if $p_c > p_m$ then
4)	$Chrom_new(j,:) = \text{cross}(X_{Attacker}(j,:), \text{objv})$;
5)	else
6)	$Chrom_new(j,:) = \text{mutate}(X_{Attacker}(j,:), \text{objv})$;
7)	end if
8)	end for

Figure 6. Pseudocode of the algorithm for updating the attacker's position.

4.2.5. Simulated Annealing Mechanism

Although the traditional CHOA uses randomness, its degree of randomization is not sufficient, and it is prone to getting trapped in a locally optimal solution during the search process and cannot jump out. To address this issue, this study proposes integrating the simulated annealing acceptance criterion into the CHOA. This approach allows the algorithm to accept a novel solution that is worse than the current solution with a certain probability as the current solution and subsequently continue searching for a better solution. If the algorithm discovers a superior solution to the local optima, it indicates that the algorithm has escaped from the local optimum condition. This method not only ensures the power of global exploration but also enhances convergence accuracy. The probability formula for the simulated annealing mechanism to accept the new solution is as follows:

$$P = \begin{cases} 1 & f(S_{new}) < f(S_{curr}) \\ e^{-[f(S_{new}) - f(S_{curr})]/T} & f(S_{new}) \geq f(S_{curr}) \end{cases}. \quad (38)$$

where $f(S_{curr})$ represents the objective function value of the current solution and $f(S_{new})$ represents the objective function value of the new solution.

As the number of iterations increases, the acceptance probability should decrease due to the need to reduce the search time. In other words, temperature T decreases with the number of iterations. See (39) for the formula, where β is the cooling factor $0 < \beta < 1$.

$$T_{gen+1} = \beta T_{gen}. \tag{39}$$

4.2.6. SACHOA Algorithm Steps

To sum up, the calculation steps for SACHOA are as follows:

Step 1: Initialize parameters and population. a is obtained according to Formula (24), as the sequential crossover probability, and c is obtained according to Formula (25), as the mutation probability.
Step 2: Initialize the group and divide it into four populations.
Step 3: Calculate the fitness of all individuals in each population and determine the leader in the population.
Step 4: Update the position of the leader in each population and determine the attacker based on the fitness.
Step 5: Optimize the position of the attacker by incorporating the sine–cosine operator.
Step 6: Use a simulated annealing mechanism to further search the solution space.
Step 7: If the algorithm reaches the termination condition (the maximum number of iterations), output the global optimal solution; otherwise, repeat Steps 2–6.

5. Computational Experiments

Since there is no standard calculation example for the VRPDTWDD problem studied in this paper, the R, C, and RC series test data for the VRPTW problem given by Solomon are selected as the benchmark data. The data are downloaded from the VRP Web link (dorronsoro.es, https://www.bernabe.dorronsoro.es/vrp/ 7 November 2023). The customers and distribution center in the R series test sets are randomly distributed. The C series test sets present aggregated characteristics. The RC series test sets show mixed delivery characteristics of randomness and aggregation, and the three types of test sets are distributed in the 100 by 100 coordinate system. To meet the characteristics of the problem in this study, the unit delivery distance cost ratio of the truck and drone is set to 5:1. The speed ratio of the truck and drone is 1:1.1. The loading capacity of the truck and drone is 200 kg and 12 kg, respectively. The maximum flying distance of the drone is 20 km. The test environment is Matlab 2018b, Intel (R) Core (TM) i5−10200H CPU @ 2.40 GHz processor, Windows 10 operating system. The parameters for the algorithm are set as follows: population size, 100; maximum iterations, 200; probability of crossing, 0.9; probability of mutating, 0.5; generation gap, 0.9; upper and lower bounds of the nonlinear learning factor, [1, 0.2]; chaotic vector, $chaos(3;1;1)$; annealing coefficient, 0.99; and initial annealing temperature, 100.

5.1. Algorithm Performance Test

5.1.1. Comparison of the Results of Different Algorithms

The enumeration method (EM) is an algorithm for finding the optimal solution to a problem by enumerating all possible solutions. It has the characteristics of a simple principle and easy proof of correctness. Genetic algorithm (GA), simulated annealing (SA), and traditional CHOA are widely used to solve various optimization problems and are also suitable for solving drone–vehicle routing problems with time windows. To evaluate the effectiveness of the algorithm, this paper compares the results of SACHOA with those of EM, GA, SA, and CHOA. The results are presented in Table 3, including the name, scale, number of truck service customers (T_C) in the test data set, the optimal solution value of each algorithm, and the difference, GAP^Y, between them. $GAP^Y = (SACHOA - Y)/Y \times 100\%$, where Y represents EM/GA/SA/CHOA.

Table 3. Results of different algorithms.

Test Name	Number _Node	T_C	EM	GA	SA	CHOA	SACHOA	GAP^{EM}	GAP^{GA}	GAP^{SA}	GAP^{CHOA}
R101_10	10	7	250.2162	250.2162	250.2162	250.2162	250.2162	0.00	0.00	0.00	0.00
R102_10	10	7	244.9569	244.9569	244.9569	244.9569	244.9569	0.00	0.00	0.00	0.00
R201_10	10	7	232.2306	232.2306	232.2306	232.2306	232.2306	0.00	0.00	0.00	0.00
C101_10	10	7	97.6206	97.6206	97.6206	97.6206	97.6206	0.00	0.00	0.00	0.00
C102_10	10	7	61.7037	61.7037	61.7037	61.7037	61.7037	0.00	0.00	0.00	0.00
C201_10	10	7	163.0913	163.0913	163.0913	163.0913	163.0913	0.00	0.00	0.00	0.00
RC101_10	10	7	180.8669	180.8669	180.8669	180.8669	180.8669	0.00	0.00	0.00	0.00
RC102_10	10	7	151.4601	151.4601	151.4601	151.4601	151.4601	0.00	0.00	0.00	0.00
RC103_10	10	7	149.4657	149.4657	149.4657	149.4657	149.4657	0.00	0.00	0.00	0.00
RC104_10	10	7	152.6316	152.6316	152.6316	152.6316	152.6316	0.00	0.00	0.00	0.00
RC105_10	10	7	153.3902	153.3902	153.3902	153.3902	153.3902	0.00	0.00	0.00	0.00
RC201_10	10	7	152.7721	154.2408	152.7721	152.7721	152.7721	0.00	−0.95	0.00	0.00
R101_25	25	19	-	584.857	575.5174	575.7988	527.6039	-	−9.79	−8.33	−8.37
R102_25	25	22	-	514.6691	513.6467	489.5837	487.1129	-	−5.35	−5.17	−0.50
R201_25	25	13	-	478.3015	476.4763	486.1999	476.4763	-	−0.38	0.00	−2.00
C101_25	25	19	-	372.1768	423.8668	390.68	337.75	-	−9.25	−20.32	−13.55
C102_25	25	22	-	287.8577	279.9872	280.0399	278.1451	-	−3.37	−0.66	−0.68
C201_25	25	13	-	374.1189	353.567	345.433	335.7829	-	−10.25	−5.03	−2.79
RC101_25	25	19	-	360.2397	361.6697	375.8553	349.6057	-	−2.95	−3.34	−6.98
RC102_25	25	22	-	344.6897	344.6897	340.3574	340.3574	-	−1.26	−1.26	0.00
RC103_25	25	19	-	346.0971	343.5942	343.5942	341.0745	-	−1.45	−0.73	−0.73
RC104_25	25	19	-	350.5197	347.1345	342.585	341.2114	-	−2.66	−1.71	−0.40
RC105_25	25	19	-	352.4976	355.2043	348.3525	348.8076	-	−1.05	−1.80	0.13
RC201_25	25	13	-	348.3884	372.0786	348.3884	345.0395	-	−0.96	−7.27	−0.96
R101_100	100	82	-	1967.4459	1605.4746	1633.5445	1598.6787	-	−18.74	−0.42	−2.13
C101_100	100	82	-	1387.1206	2224.5011	2571.4379	1382.78	-	−0.31	−37.84	−46.23
RC101_100	100	82	-	1682.9865	1883.9339	1823.1095	1619.0319	-	−3.80	−14.06	−11.19

Note: "-" indicates that the optimal solution cannot be obtained within an acceptable time.

Table 3 demonstrates that while EM can produce optimal solutions for VRPDTW problems on smaller scales, it falls short in solving examples with 20 customers within a reasonable timeframe. This is due to the significant increase in solution time as the number of customers increases, resulting from an expanded solution space. Alternatively, GA, SA, CHOA, and SACHOA are better suited for such problems, with SACHOA achieving superior results. In the case of small-scale test datasets, such as R101_10, each algorithm performs consistently and produces optimal solutions. This is attributed to the smaller search space and lower difficulty in these datasets, resulting in relatively similar algorithm performances. Nevertheless, for medium-scale datasets such as C102_25, there are slight differences in the results of the algorithms, although overall performance remains quite similar, demonstrating that the performance of the algorithms starts to show differences for moderate-scale datasets. In contrast, for large-scale datasets like RC101_100, the algorithms vary significantly in performance, with SACHOA outperforming the others. These findings indicate that SACHOA has greater adaptability and robustness when tackling larger-scale problems. In addition, the number and location of customer nodes served by trucks have a direct impact on the delivery costs. When the number of customers served by trucks is low, the delivery cost is small; when the number of customers served by trucks is high, the delivery cost is high, which is the result of the fact that the unit delivery cost of drones is lower than that of trucks. Consequently, the more customer nodes drones serve, the greater the reduction in delivery costs. From the perspective of the delivery location of the customer nodes from the truck service, when delivery to the customer nodes from the truck service is relatively centralized, the delivery cost is smaller. The reason for this is that location dispersion leads to a small change in the length of the truck route, which is not conducive to a reduction in the delivery cost; this is the actual delivery situation.

Figure 7 depicts the convergence diagrams for SACHOA, CHOA, SA, and GA algorithms when solving example RC101_25. The results show that the SACHOA algorithm has the following advantages over other algorithms when dealing with the VRPDTW problem: first, the SACHOA algorithm is outstanding in the speed of convergence to the relatively

optimal solution; second, the quality of the optimal solution it finds is relatively higher, which means that the SACHOA algorithm has higher applicability and accuracy for solving the VRPDTW problem.

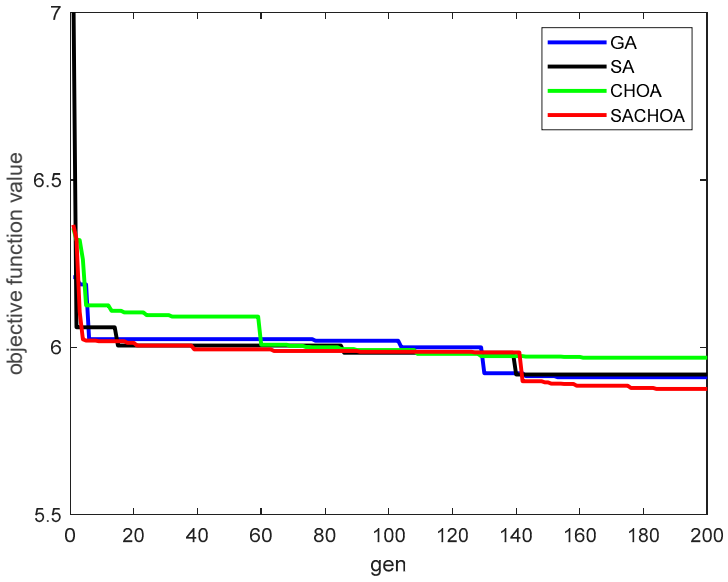


Figure 7. Comparative iteration of different algorithms for solving RC101_25.

5.1.2. Comparison of the Results of Different Strategies

To analyze the impact of fusing the sine–cosine operator and the simulated annealing mechanism on algorithm improvement, the datasets were tested under different customer scales and algorithm combinations. In each example, each algorithm combination is run 20 times; the obtained objective function values are given in Table 4. CHOA denotes the objective function value obtained from the traditional chimp optimization algorithm, while CHOA+CS represents the solution obtained after integrating the sine–cosine operator into CHOA's attacker position change. Similarly, CHOA+SA denotes the solution obtained after incorporating the simulated annealing mechanism into the CHOA objective function value. Finally, GAP^Y indicates the difference between SACHOA and the optimal solution obtained following the introduction of each strategy, $GAP^Y = (SACHOA - Y)/Y \times 100\%$, where Y represents CHOA/(CHOA+CS)/(CHOA+SA).

Table 4. Results of different algorithm strategies.

Name	CHOA	Single Strategy		Multiple Strategies	GAP^{CHOA}	$GAP^{CHOA+CS}$	$GAP^{CHOA+SA}$
		CHOA+CS	CHOA+SA	SACHOA			
R101_25	575.7988	560.5092	543.353	527.6039	−8.37	−5.87	−2.9
R101_100	1633.5445	1607.8378	1713.8326	1598.6787	−2.13	−0.57	−6.72
C101_25	390.68	390.7074	359.4839	337.75	−13.54	−13.55	−6.05
C101_100	2571.4379	2353.2128	2195.8693	1382.78	−46.23	−41.24	−37.03
RC101_25	375.8553	367.7014	350.1739	349.6057	−6.98	−4.92	−0.16
RC101_100	1823.1095	1794.3128	1658.763	1619.0319	−11.2	−9.77	−2.4

The results show that adding a sine–cosine operator and simulated annealing mechanism can effectively improve the performance of the algorithm. The main reason is that the sine–cosine operator strategy can continue to search for better solutions, thereby expanding the search range of feasible solutions, which is beneficial for obtaining relatively better solutions. The simulated annealing mechanism can accept the inferior solution to avoid falling into the local optimum. In addition, when a single strategy is adopted, causing the

algorithm to fall into the local optimum is easy, and the optimization effect is worse than that of the SACHOA algorithm.

Figure 8 displays the convergence diagrams for the calculation of example RC101_25 using SACHOA, CHOA, CHOA+SA, and CHOA+CS algorithms. The results show that the SACHOA algorithm can converge to the relatively optimal solution faster than other algorithm combinations, which further proves the effectiveness of the algorithm.

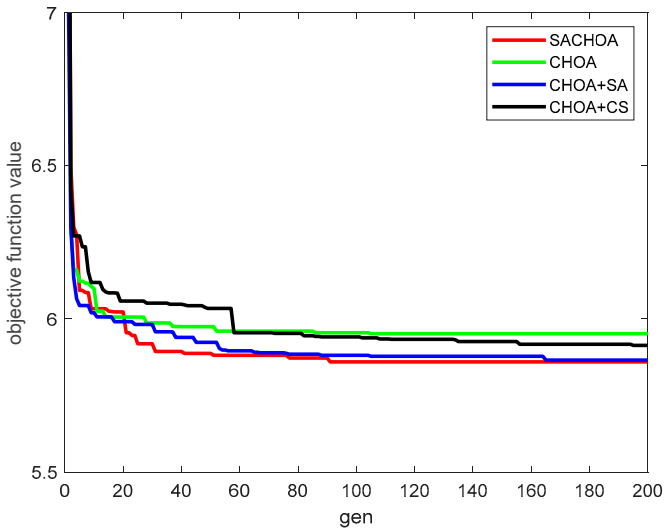


Figure 8. Comparative iteration for solving RC101_25 using different strategies.

5.1.3. Stability Analysis

To further test the stability of the algorithm under the same parameter configurations, examples R101, R105, C101, C105, RC101, and RC105 are tested 10 times each; the results are shown in Figure 9.

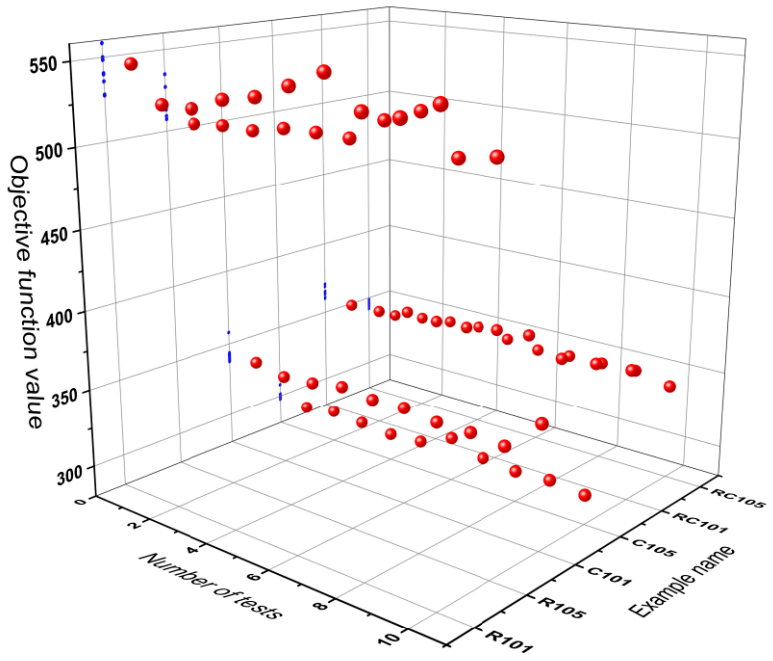


Figure 9. Algorithm stability test results. The red dots represent the numerical results of ten tests, while the blue dots indicate the mapping positions of these red dots on the Z-axis.

Figure 9 shows that during the 10 tests, the maximum value of the objective function of R101 is 558.2347, the minimum value is 527.6039, the average value is 541.8093, and

the average error rate relative to the minimum value is 2.69%. The maximum value of the objective function of R105 is 535.0197, the minimum value is 507.3721, the average value is 516.1252, and the average error rate relative to the minimum value is 1.73%. The maximum value of C101 is 343.7847, the minimum value is 337.7465, the average value is 342.3882, and the average error rate relative to the minimum value is 1.37%. The maximum value of C105 is 301.5938, the minimum value is 297.6879, the average value is 300.4842, and the average error rate relative to the minimum value is 0.94%. The maximum value of RC101 is 370.7237, the minimum value is 349.6057, the average value is 363.3692, and the average error rate relative to the minimum value is 3.94%. The maximum value of RC105 is 348.8725, the minimum value is 341.6005, the average value is 344.94, and the average error rate relative to the minimum value is 0.98%. Of the six tests, R101 had the highest average error rate while C105 had the smallest. Overall, due to the limitations of the heuristic algorithm, the results of each solution fluctuate to a certain extent and the relative average error rate of the 10 tests is controlled within an acceptable range of 5%, indicating the stability of the designed algorithm and showing that it can be used to solve VRPDTW problems.

5.2. Actual Case Study

This study takes the customers of a self-operated store of a certain brand in Shenyang from 8:00 a.m. to 11:00 a.m. as the data for verifying the feasibility of the proposed method. During this period, the company provided services to 30 customers in total. The spatial distribution of customers is shown in Figure 10 and detailed information for the customers is listed in Table 5. The identifiers in the "Customer Type" column are explained as follows: a for canceled customers and b for newly added customers. The maximum loading capacity of the truck is 100 kg, the average speed is 30 km/h, and the unit delivery cost is about \$0.78/km [52]; the maximum loading capacity of the drone is 12 kg, the maximum flight distance is 20 km, the average speed is 45 km/h, and the unit delivery cost is about \$0.078/km [52].

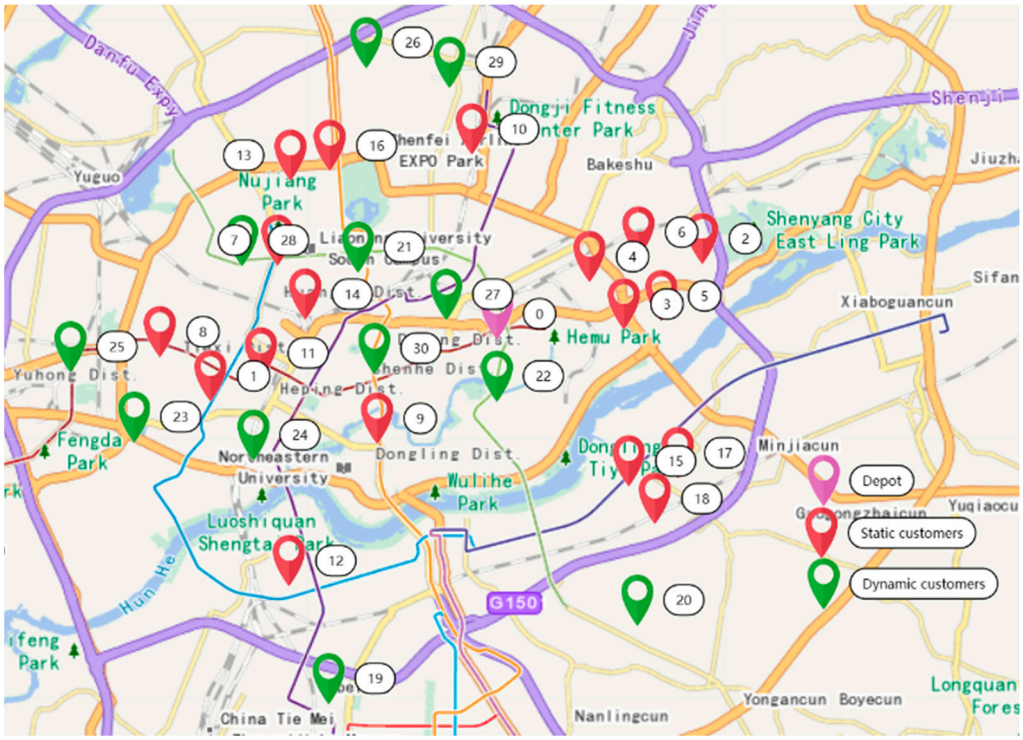


Figure 10. Spatial distribution of customer locations. 0 represents the depot, and 1–30 represents the customer.

Table 5. Information for static customers.

Number	Latitude	Longitude	Demand	Time Window	Number	Latitude	Longitude	Demand	Time Window
0	41.81139	123.4902	0	8:00–11:00	11	41.79927	123.3922	13.9	9:00–10:00
1	41.79223	123.3694	28.4	8:00–9:00	12	41.73469	123.4037	18.5	9:00–10:00
2	41.83274	123.5763	0.8	10:00–11:00	13	41.86065	123.4044	1.3	10:00–11:00
3	41.81372	123.5455	16.4	8:00–9:00	14	41.81685	123.4101	16.8	9:00–10:00
4	41.8289	123.5302	23.5	8:00–9:00	15	41.76482	123.5461	17.2	10:00–11:00
5	41.81375	123.5581	1.5	9:00–10:00	16	41.86355	123.4206	21.1	9:00–10:00
6	41.8369	123.5497	1.5	9:00–10:00	17	41.76901	123.5652	17.5	10:00–11:00
7	41.83439	123.3972	1.1	9:00–10:00	18	41.75193	123.5579	3.5	10:00–11:00
8	41.80242	123.3434	13.5	10:00–11:00	19	41.69756	123.4184	2.6	9:00–10:00
9	41.7789	123.4403	15.5	8:00–9:00	20	41.75746	123.5306	23.5	9:00–10:00
10	41.87804	123.4808	0.5	9:00–10:00	-	-	-	-	-

First, when entering the initial planning stage, we use the SACHOA algorithm to plan the delivery route for the initial static customers; the results are shown in Table 6.

Table 6. Initial planning stage delivery scheme.

Vehicle	Route	Vehicle	Route
Truck 1	0→4→3→17→15→20→0	Drone 1	3→6→2→5→17; 17→18→15
Truck 2	0→14→16→0	Drone 2	14→7→13→16; 16→10→0
Truck 3	0→9→11→1→8→12→19→0	-	-

Before determining the delivery route for the dynamic planning stage, we use the k-means clustering method to cluster all customers. The clustered customer set is shown in Table 7. Based on the dynamic degree, only areas 1, 2, and 3 are considered for delivery route changes in the dynamic planning stage, while delivery routes for other regions remain the same.

Table 7. K-means clustering results.

Number	The Radiation Customer Sets	Dynamic Degree
1	7, 10, 13, 16, 26, 28, 29	0.43
2	9, 11, 14, 21, 22, 27, 30	0.57
3	1, 8, 12, 19, 20, 23, 24, 25	0.63
4	2, 3, 4, 5, 6, 15, 17, 18	0.00

According to the actual situation, the dynamic service time window also opens when the distribution center starts the same-day delivery service, that is, it will enter the dynamic planning stage. The update interval is set to 60 min, resulting in three dynamic time windows [8:00–9:00], [9:00–10:00], and [10:00–11:00] on the given day. Table 8 displays the relevant dynamic customer information received during these periods. According to the information in Table 8, both [9:00–10:00] and [10:00–11:00] have dynamic demands, so two delivery route adjustments are required. At the beginning of each period, starting from the location of the current vehicle service customer, the routes for both the customers whose deliveries have not yet been made and the dynamic customers in the previous period are re-planned. It is worth noting that the SACHOA algorithm is more suitable for handling cases with 100 or fewer customers. In the end, the company used 4 trucks and 3 drones to provide delivery services to all customers, at a total delivery cost of $232.1869. Figure 11 shows the delivery routes of all vehicles. Of these, the solid lines represent the delivery routes for the trucks and the dotted lines represent the delivery routes for the drones. It can be observed from Figure 11 that when there is a new demand, the vehicle will continue to serve new customers without returning to the distribution center, which improves its utilization rate. At the same time, the new delivery route tries to keep close to the delivery

route of the original customers as far as possible to avoid reducing the level of service offered to customers.

Table 8. Information for dynamic customers.

Number	Latitude	Longitude	Demand	Time Window	Customer Type
19	41.69756	123.4184	2.6	9:00–10:00	a
20	41.75746	123.5306	23.5	9:00–10:00	a
21	41.83218	123.432	1.6	10:00–11:00	b
22	41.79236	123.4909	21.3	10:00–11:00	b
23	41.77769	123.3386	2.5	9:00–10:00	b
24	41.77384	123.3875	1.6	9:00–10:00	b
25	41.80065	123.3116	2.2	9:00–10:00	b
26	41.89783	123.4372	2.6	9:00–10:00	b
27	41.81732	123.4693	35	10:00–11:00	b
28	41.83438	123.3824	17.5	9:00–10:00	b
29	41.89038	123.4706	1.5	9:00–10:00	b
30	41.80003	123.4408	19.5	10:00–11:00	b

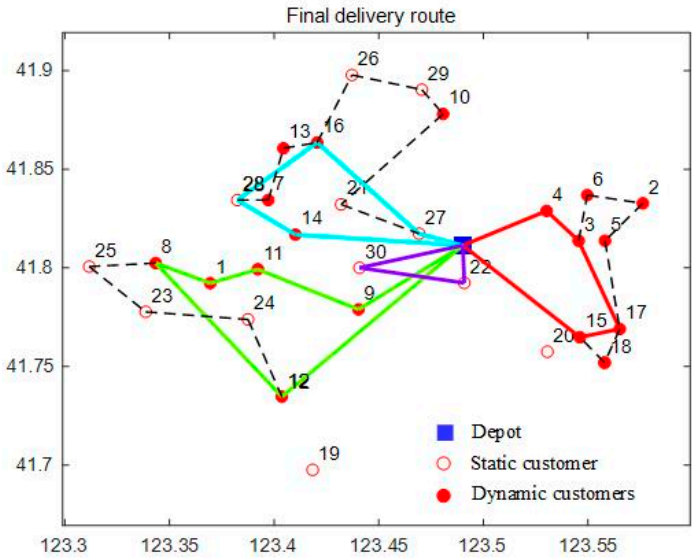


Figure 11. Optimized delivery route.

5.3. Sensitivity Analysis

Because the type of dynamic demand will directly affect the change in delivery cost, this section only considers the relationship between newly added customers and delivery cost. We change all the dynamic demands of the actual cases to newly added customers and set the dynamic degrees to 0, 0.2, 0.4, 0.6, 0.8, and 1.0, in turn, and run the algorithm 10 times under each gradient to get the smallest objective function value. The changes in delivery costs with the dynamic degree are shown in Figure 12. The results indicate that, under the condition that the total number of customers remains constant, the delivery cost will increase with the increase in the dynamic degree. This is due to the fact that when the number of dynamic demand customers increases, the number of driving routes will increase and the company will need to arrange additional vehicles to provide delivery services for the additional customers.

It is important to note that there are several limitations in our case study that need to be highlighted to ensure a clearer understanding of the applicability and limitations of the study. First, the experimental results of the actual case study may be node-number-limited. The number of nodes in this instance is 30, which may be relatively small. This means that further experiments using larger groups are still needed. Second, to simplify the research, we have not considered traffic congestion. Future research should consider traffic

congestion factors to gain a more comprehensive understanding of the actual situation of the cases.

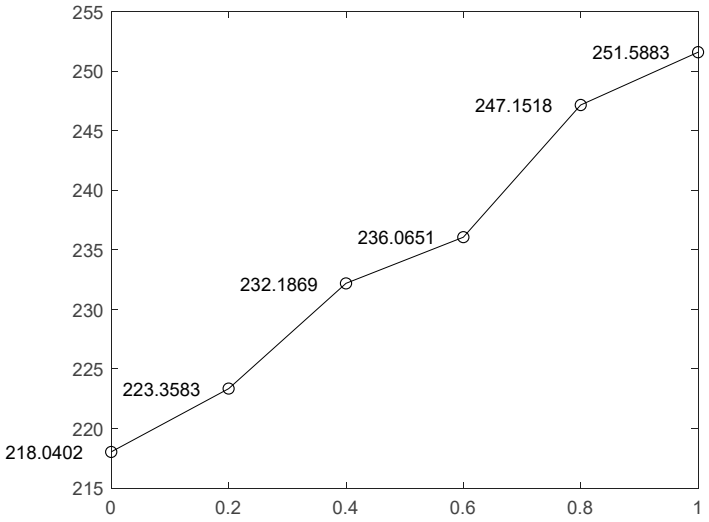


Figure 12. Changes in delivery costs under different dynamics.

6. Conclusions

This study aimed to solve the practical problems of untimely dynamic delivery network responses and high delivery costs caused by the dynamics of customer orders and the strictness of time constraints during actual delivery. This study establishes a two-stage optimization model based on different demand response strategies with the goal of minimizing the total delivery cost and proposes the simulated annealing chimp optimization algorithm (SACHOA) with a sine–cosine operator to solve this problem. Using test examples and case analysis, the following conclusions are drawn:

- The number of customers served by trucks is positively correlated with the delivery cost, and the more concentrated their delivery locations are, the smaller the delivery cost is.
- Integrating sine–cosine operators and simulated annealing mechanisms can effectively improve the performance of the algorithm. In addition, SACHOA exhibits certain stability and can effectively solve the VRPDTW problem.
- Response strategies based on different demands have certain practical effectiveness.
- The delivery cost will increase with the increase in the dynamic degree under the condition that the total number of customers remains unchanged.

This study expands on the theory of VRPD and aims to promote research in the field of the joint distribution of trucks and drones. Through the proposed model and algorithm, we provide logistics companies with an effective tool for optimizing dynamic distribution networks, which is expected to reduce distribution costs and improve distribution efficiency. Ultimately, the research conclusions can provide useful references for relevant decision-makers and support future distribution strategies and decision-making.

Future research can extend our lines of investigation in several directions. First, this research can be extended to three-dimensional space, where drone obstacle avoidance and collision-free path problems are introduced to better represent the real delivery situation encountered by drones. In addition, there may be more complex factors in practical applications, such as traffic congestion, which will require further research. Finally, there is space for further improvement of the efficiency of the algorithm, and comparison and integration with other optimization algorithms can also be considered.

Author Contributions: Conceptualization, Y.L. (Yanqiu Liu) and J.H.; methodology, J.H.; software, J.H.; validation, Y.L. (Yanqiu Liu) and J.H.; formal analysis, Y.L. (Yanqiu Liu); investigation, J.H. and Y.L. (Yan Li); resources, Y.L. (Yanqiu Liu); data curation, J.H. and Y.L. (Yan Li); writing—original draft preparation, J.H. and Y.L. (Yan Li); writing—review and editing, Y.L. (Yanqiu Liu); visualization, J.H.; supervision, Y.L. (Yanqiu Liu); project administration, Y.L. (Yanqiu Liu); funding acquisition, Y.L. (Yanqiu Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 70431003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data in this article can be obtained from the VRP Website (dorronsoro.es accessed on 7 November 2023).

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive comments.

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

References

1. Boysen, N.; Fedtke, S.; Schwerdfeger, S. Last-mile delivery concepts: A survey from an operational research perspective. *OR Spectr.* **2021**, *43*, 1–58. [CrossRef]
2. Li, H.Q.; Wang, H.T.; Chen, J.; Bai, M. Two-echelon vehicle routing problem with time windows and mobile satellites. *Transp. Res. Part B Methodol.* **2020**, *138*, 179–201. [CrossRef]
3. Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [CrossRef]
4. Uchoa, E.; Pecin, D.; Pessoa, A.; Marcus, P.; Thibaut, V.; Anand, S. New benchmark instances for the Capacitated Vehicle Routing Problem. *Eur. J. Oper. Res.* **2017**, *257*, 845–858. [CrossRef]
5. Marinakis, Y.; Marinaki, M.; Migdalas, A. A mulit-adaptive particle swarm optimization for the vehicle routing problem with time windows. *Inf. Sci.* **2019**, *481*, 311–329. [CrossRef]
6. Lai, D.S.W.; Demirag, O.C.; Leung, J.M.Y. A tabu search heuristic for the heterogenerous vehicle routing problem on a multigraph. *Transp. Res. Part C Emerg. Technol.* **2016**, *86*, 32–52. [CrossRef]
7. Brenner, H.O.R.; Eduardo, C.X.; Miyazawa, F.K.; Pedro, A.; Eduardo, C.; Maria, J.S. Recent dynamic vehicle routing problems: A survey. *Comput. Ind. Eng.* **2021**, *160*, 107604.
8. Wang, F.; Liao, F.; Li, Y.; Yan, X.S.; Xu, C. An ensemble learning base multi-objective evolutionary algorithm for the dynamic vehicle routing problem with time windows. *Comput. Ind. Eng.* **2021**, *154*, 107131. [CrossRef]
9. Larsen, A. *The Dynamic Vehicle Routing Problem*; Technical University of Denmark: Kogens Lyngby, Denmark, 2000.
10. Hong, L.X. An improved LNS algorithm for real-time vehicle routing problem with time windows. *Comput. Oper. Res.* **2012**, *39*, 151–163. [CrossRef]
11. Chen, S.F.; Chen, R.; Wang, G.G.; Gao, J.; Sangaiah, A.K. An adaptive large neighborhood search heuristic for dynamic vehicle routing problems. *Comput. Electr. Eng.* **2018**, *67*, 596–607. [CrossRef]
12. Yao, C.; Chen, S.; Yang, Z. Online Distributed Routing Problem of Electric Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16330–16341. [CrossRef]
13. Wang, X.X.; Xu, Q.; Shen, X.P. EV Charging Path Distribution Solution Based on Intelligent Network Connection. *Mathematics* **2023**, *11*, 28793. [CrossRef]
14. Dayarian, I.; Savelsbergh, M.; Clarke, J.P. Same-day delivery with drone resupply. *Transp. Sci.* **2020**, *54*, 229–249. [CrossRef]
15. Ulmer, M.; Thomas, B. Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks* **2018**, *72*, 475–505. [CrossRef]
16. Otto, A.; Agatz, N.; Campbell, J.; Golden, B.; Pesch, E. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks* **2018**, *72*, 411–458. [CrossRef]
17. Macrina, G.; Pugliese, L.D.P.; Guerriero, F.; Laporte, G. Drone-aided routing: A literature review. *Transp. Res. Part C Emerg. Technol.* **2020**, *120*, 102762. [CrossRef]
18. 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]
19. Ha, Q.M.; Deville, Y.; Pham, Q.D.; Hà, M.H. On the min-cost traveling salesman problem with drone. *Transp. Res. Part C Emerg. Technol.* **2018**, *86*, 597–621. [CrossRef]
20. Sarab, A.; Taghreed, A.; Sara, A.; Yara, A.; Lujain, A.; Haifa, A. Optimization of Truck-Drone Parcel Delivery Using Metaheuristics. *Appl. Sci.* **2021**, *11*, 6443.
21. Tong, B.; Wang, J.W.; Wang, X.; Zhou, F.H.; Mao, X.H.; Zheng, W.L. Optimal Route Planning for Truck–Drone Delivery Using Variable Neighborhood Tabu Search Algorithm. *Appl. Sci.* **2022**, *12*, 529. [CrossRef]

22. Chang, Y.S.; Lee, H.J. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. *Expert Syst. Appl.* **2018**, *104*, 307–317. [CrossRef]
23. Yurek, E.E.; Ozmutlu, H.C. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transp. Res. Part C Emerg. Technol.* **2018**, *91*, 249–262. [CrossRef]
24. Kim, S.; Moon, I. Traveling salesman problem with a drone station. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 42–52. [CrossRef]
25. Wang, K.; Yuan, B.; Zhao, M.; Lu, Y.W. Cooperative route planning for the drone and truck in delivery services: A bi-objective optimisation approach. *J. Oper. Res. Soc.* **2019**, *71*, 1657–1674. [CrossRef]
26. Boccia, M.; Masone, A.; Sforza, A.; Sterle, C. A column-and-row generation approach for the flying sidekick travelling salesman problem. *Transp. Res. Part C Emerg. Technol.* **2021**, *124*, 102913. [CrossRef]
27. Roberti, R.; Ruthmair, M. Exact methods for the traveling salesman problem with drone. *Transp. Sci.* **2021**, *55*, 275–552. [CrossRef]
28. Mahmoudinazlou, S.; Kwon, C. A hybrid genetic algorithm with type-aware chromosomes for traveling salesman problems with drone. neural and evolutionary computing. *arXiv* **2023**, arXiv:2303.00614.
29. Ha, Q.M.; Deville, Y.; Pham, Q.D.; Hà, M.H. A hybrid genetic algorithm for the traveling salesman problem with drone. *J. Heuristics* **2020**, *26*, 219–247. [CrossRef]
30. Kitjacharoenchai, P.; Ventresca, M.; Moshref-Javadi, M.; Lee, S.; Tanchoco, J.M.A.; Brunese, P.A. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Comput. Ind. Eng.* **2019**, *129*, 14–30. [CrossRef]
31. Raj, R.; Murray, C. The multiple flying sidekicks traveling salesman problem with variable drone speeds. *Transp. Res. Part C Emerg. Technol.* **2020**, *120*, 102813. [CrossRef]
32. Peng, K.; Du, J.; Lu, F.; Sun, Q.G.; Dong, Y.; Zhou, P.; Hu, M.L. A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery. *IEEE Access* **2019**, *7*, 49191–49200. [CrossRef]
33. Freitas, J.C.D.; Penna, P.H.V. A variable neighborhood search for flying sidekick traveling salesman problem. *Int. Trans. Oper. Res.* **2020**, *27*, 267–290. [CrossRef]
34. Dell'Amico, M.; Montemanni, R.; Novellani, S. Modeling the flying sidekick traveling salesman problem with multiple drones. *Networks* **2021**, *78*, 303–327. [CrossRef]
35. Cavani, S.; Iori, M.; Roberti, R. Exact methods for the traveling salesman problem with multiple drones. *Transp. Res. Part C Emerg. Technol.* **2021**, *130*, 103280. [CrossRef]
36. Tiniç, G.O.; Karasan, O.E.; Kara, B.Y.; Campbell, J.F.; Ozel, A. Exact solution approaches for the minimum total cost traveling salesman problem with multiple drones. *Transp. Res. Part B Methodol.* **2023**, *168*, 81–123. [CrossRef]
37. Gonzalez-R, P.L.; Canca, D.; Andrade-Pineda, J.L.; Calle, M.; Leon-Blanco, J.M. Truck-drone team logistics: A heuristic approach to multi-drop route planning. *Transp. Res. Part C Emerg. Technol.* **2020**, *114*, 657–680. [CrossRef]
38. Luo, Z.; Poon, M.; Zhang, Z.; Liu, Z.; Lim, A. The multi-visit traveling salesman problem with multi-drones. *Transp. Res. Part C Emerg. Technol.* **2021**, *128*, 103172. [CrossRef]
39. Mara, S.T.W.; Rifai, A.P.; Sopha, B.M. An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops. *Expert Syst. Appl.* **2022**, *205*, 117647. [CrossRef]
40. Mahmoudi, B.; Eshghi, K. Energy-constrained multi-visit tsp with multiple drones considering non-customer rendezvous locations. *Expert Syst. Appl.* **2022**, *210*, 118479. [CrossRef]
41. Wang, X.; Poikonen, S.; Golden, B. The vehicle routing problem with drones: Several worst-case results. *Optim. Lett.* **2017**, *11*, 679–697. [CrossRef]
42. Masmoudi, M.A.; Mancini, S.; Baldacci, R.; Kuo, Y.H. Vehicle routing problems with drones equipped with multipackage payload compartments. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *164*, 102757. [CrossRef]
43. Wang, Z.; Sheu, J.B. Vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **2019**, *122*, 350–364. [CrossRef]
44. 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]
45. Euchi, J.; Sadok, A. Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones. *Phys. Commun.* **2021**, *44*, 101236. [CrossRef]
46. Lei, D.; Cui, Z.; Li, M. A dynamical artificial bee colony for vehicle routing problem with drones. *Eng. Appl. Artif. Intell.* **2022**, *107*, 104510. [CrossRef]
47. Schermer, D.; Moeini, M.; Wendt, O. A matheuristic for the vehicle routing problem with drones and its variants. *Transp. Res. Part C Emerg. Technol.* **2019**, *106*, 166–204. [CrossRef]
48. Kuo, R.; Edbert, E.; Zulvia, F.E.; Lu, S.H. Applying nsga-ii to vehicle routing problem with drones considering makespan and carbon emission. *Expert Syst. Appl.* **2023**, *221*, 119777. [CrossRef]
49. Meng, S.; Guo, X.; Li, D.; Liu, G.Q. The multi-visit drone routing problem for pickup and delivery services. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *169*, 102990. [CrossRef]
50. Kuo, R.; Lu, S.H.; Lai, P.Y.; Mara, S.T.W. Vehicle routing problem with drones considering time windows. *Expert Syst. Appl.* **2022**, *191*, 116264. [CrossRef]

51. Khishe, M.; Mosavi, M. Chimp optimization algorithm. *Expert Syst. Appl.* **2020**, *149*, 113338. [CrossRef]
52. Campbell, J.F.; Sweeney, D.C., II; Zhang, J. Strategic design for delivery with linked transportation assets: Trucks and drones. In *Technical Report Midwest Transportation Center*; Institute for Transportation Iowa State University: Ames, IA, USA, 2018.

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.