drones

MDPI

Article

Delivery Route Scheduling of Heterogeneous Robotic System with Customers Satisfaction by Using Multi-Objective Artificial Bee Colony Algorithm

Zhihuan Chen [1,2], Shangxuan Hou [1,2], Zuao Wang [1,2], Yang Chen [1,2,*], Mian Hu [1,2] and Rana Muhammad Adnan Ikram [3,*]

[1] Engineering Research Center for Metallurgical Automation and Measurement Technology of Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China; czh@wust.edu.cn (Z.C.); 201904134022@wust.edu.cn (S.H.); chenzhiheng@wust.edu.cn (Z.W.); humian@wust.edu.cn (M.H.)

[2] Institute of Robotics and Intelligent Systems, Wuhan University of Science and Technology, Wuhan 430081, China

[3] School of Architecture and Urban Planning, Guangzhou University, Guangzhou 510006, China

* Correspondence: chenyag@wust.edu.cn (Y.C.); rana@gzhu.edu.cn (R.M.A.I.)

Abstract: This study addresses the route scheduling problem for the heterogeneous robotic delivery system (HRDS) that perform delivery tasks in an urban environment. The HRDS comprises two distinct types of vehicles: an unmanned ground vehicle (UGV), which is constrained by road networks, and an unmanned aerial vehicle (UAV), which is capable of traversing terrain but has limitations in terms of energy and payload. The problem is formulated as an optimal route scheduling problem in a road network, where the goal is to find the route with minimum delivery cost and maximum customer satisfaction (CS) enabling the UAV to deliver packages to customers. We propose a new method of route scheduling based on an improved artificial bee colony algorithm (ABC) and the non-dominated sorting genetic algorithm II (NSGA-II) that provides the optimal delivery route. The effectiveness and superiority of the method we proposed are demonstrated by comparison in simulations. Moreover, the physical experiments further validate the practicality of the model and method.

Keywords: heterogeneous robotic delivery system; customer satisfaction; route scheduling; multi-objective optimization; artificial bee colony algorithm

check for updates

Citation: Chen, Z.; Hou, S.; Wang, Z.; Chen, Y.; Hu, M.; Ikram, R.M.A. Delivery Route Scheduling of Heterogeneous Robotic System with Customers Satisfaction by Using Multi-Objective Artificial Bee Colony Algorithm. *Drones* **2024**, *8*, 519. https://doi.org/10.3390/drones8100519

Academic Editor: Pablo Rodríguez-Gonzálvez

Received: 28 August 2024
Revised: 18 September 2024
Accepted: 20 September 2024
Published: 24 September 2024

1. Introduction

The rapid development of e-commerce has brought many opportunities and challenges to the express delivery industry. With the improvement of delivery efficiency and service quality, market competition has become increasingly intense. In this context, the application of mobile robots in the express delivery field is attracting considerable critical attention [1]. Especially unmanned ground vehicles (UGV) and unmanned aerial vehicles (UAV) have been widely used for parcel delivery [2]. UAVs benefit from their aerial mode of transportation, enabling them to overcome barriers of time and space [3]. UGVs excel at large-scale delivery tasks due to their excellent endurance and carrying capacity [4]. However, UAVs are usually limited in application scenarios because of energy and load constraints, while UGVs are generally limited in movement due to the restrictions of ground obstacles [5]. To address the limitations of individual robots, many fields [6,7] have begun exploring UAV-aided UGV solutions to achieve better results. Therefore, a novel system composed of UGVs and UAVs has been conceptualized, called the Heterogeneous Robotic Delivery System (HRDS) [8]. This system not only has the merit of a wide range for delivery but also has the characteristics of flexible distribution.

One of the most critical tasks in HRDS is to schedule routes, i.e., assigning the best route for UAVs and UGVs. Many researchers have devoted their efforts to this field. Murray and Chu [9] proposed HRDS and developed models such as the Flying Sidekick Traveling Salesman Problem (FSTSP) and Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP). Agatz et al. [10] studied the hybrid delivery model of 'last-mile' delivery using HRDS and demonstrated that HRDS reduces costs compared to the UGV-only model. Kitjacharoenchai [11] proposed a new routing model that considers a synchronized HRDS operation by allowing multiple UAVs to fly from a UGV, serve one or multiple customers, and return to the same UGV for a battery swap and package retrieval. Wang et al. [12] addressed UAV/UGV scheduling by solving fixed and discrete charging set problems and introducing additional tasks for UGVs beyond recharging UAVs, using graph-based methods. These studies promote the application of HRDS. However, the aforementioned researches simplify customer locations into a two-dimensional model, thereby enabling both UGV and UAV to complete deliveries. Nevertheless, in specific scenarios, such as for residents in high-rise buildings or remote mountainous areas, only the UAV can carry out the delivery tasks. Therefore, this paper focuses on these special environments, aiming to enable the UAV to handle deliveries to all customers. Meanwhile, most existing studies assume that UGVs can move freely without any restrictions, but in the practical delivery process, UGVs usually only can drive on predefined road network routes. This is the first difficulty of our study that should be solved.

On the other hand, as global markets gradually become more competitive and service demand increases continuously, customers' requirements for the timeliness of express delivery are constantly increasing. Therefore, the customer satisfaction (CS) becomes more and more important. Nowadays, CS is considered in many engineering fields, such as hotel services, postal delivery, and school bus routing [13,14]. In these domains, each customer has an expected time window for the service, and if the customer is served within that time window, the degree of CS is 1; if not, the degree of CS is 0. The delivery officer must consider the customer's time window for a better CS assessment. However, it is interesting to note that there is no unified standard for calculation methods of customer satisfaction since the definitions of CS in different industrial fields are various. For instance, Xu et al. [15] introduced the fuzzy set theory and employed a fuzzy membership function to describe customer satisfaction. Zhang [16] proposed an improved fuzzy due-time window to quantify customer satisfaction. Yan and Xiao [17] introduced a convex customer satisfaction function to characterize customers' sensitivity to time. Zhang et al. [18] proposed enhancing customer time satisfaction by integrating real-world factors, such as soft time windows and delivery preparation times, into the route optimization model. In most of the above studies, the preference information of customers is represented as a convex fuzzy number with respect to the satisfaction for service time, because the simple concept of time windows does not model customer satisfaction very well [19]. Inspired by the above standard, and considering the practical demands of express delivery, a novel evaluation standard for delivery CS is proposed for HRDS in this study. A good CS can improve the customer stickiness of express companies, but it brings about an increase in delivery costs caused by longer routes that are unwanted. Therefore, we consider both CS and delivery costs simultaneously in this study. Decision makers can select delivery solutions according to the actual needs of the express company; when the company is in the development stage it can focus on the CS, and when the company is in the stable stage can focus on the delivery costs. But how to balance CS and delivery costs is the second difficulty to be addressed in this study.

To address the aforementioned difficulties, a novel heterogeneous robotic delivery system is proposed in this study. In this system, a temporary docking point strategy is introduced to tackle the first difficulty and the multi-objective optimization idea is introduced for solving the second difficulty. Therefore, a route scheduling model for a multi-objective heterogeneous robotic delivery system (MOHRDS) with the objectives of maximizing customer satisfaction and minimizing delivery cost is constructed in this study. In this model, the UGV carries the UAV starting from the depot, driving on the road network routes and releasing/recovering the UAV to collaborate on delivery tasks. In terms of model solving, the artificial bee colony algorithm (ABC) is improved in this study to solve the route of MOHRDS and balance the CS and delivery costs. The contributions of this paper are the following:

(1) Different from prevent research [10–12], which mainly focus on one constraint of path planning, either the flight time limitation of UAV or the road network of UGV, in this study, four constraints such as ground road network, UAV load, UAV endurance and time window are considered, a novel customer satisfaction standard is proposed to evaluate the delivery service quality, and the MOHRDS route scheduling model based on customer satisfaction and delivery cost is established. These are more closely to the real cases.

(2) Based on the route scheduling model of MOHRDS, the ABC algorithm is subjected to multiple improvements and multi-objective optimization. The accuracy of the proposed model construction, the feasibility of the solutions, and the superiority of the improved ABC are demonstrated by comparing it with various algorithms. The method is validated through physical experiments, which demonstrate its practical applicability and feasibility in real-world scenarios.

The rest of this paper is organized as follows: Section 2 formally defines the problem. In Section 3, we present our algorithmic approach. The simulation results and the effectiveness of the algorithm are analyzed in Section 4. Section 5 shows the validation on the physical platform. Finally, Section 6 presents the conclusions.

2. Route Scheduling of the Multi-Objective Heterogeneous Robotic Delivery System

The route scheduling problem of MOHRDS includes two key issues. One is to find the delivery route covering all customers and determine the releasing/recovering points for the UGV and the UAV. The other is to optimize delivery costs and customer satisfaction simultaneously. The following problem description and delivery CS standard are given prior to formulating the route scheduling problem of MOHRDS.

2.1. Problem Description

As shown in Figure 1, the MOHRDS visits all customers $b_i \in R^{2\times 1}$ $i = 1, 2, \ldots, M$ starts from a predefined depot S. Owing to the limited load and endurance of the UAV, the resupply of the UGV is required in every delivery sortie of the UAV. In other words, after the UAV completes its delivery tasks of b_i to b_j, it will return and land on the UGV to upload parcels, then move together with the UGV and replenish energy on it. Next, the UAV can take off at an appropriate docking point to deliver to the remaining customers. This progress is repeated until M customers are all delivered. Finally, the UGV carries the UAV back to S.

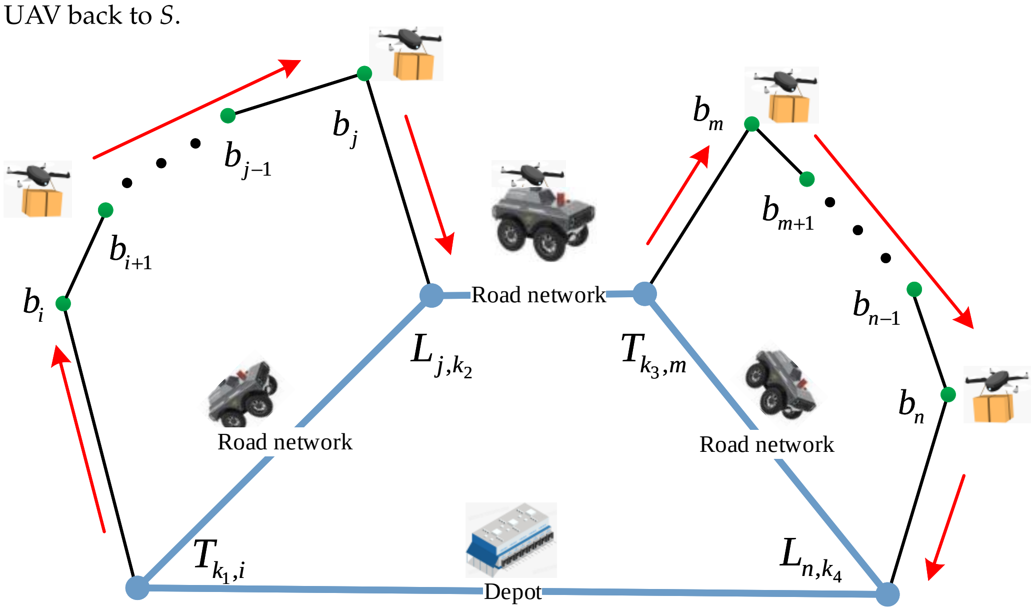


Figure 1. Heterogeneous Robotic Delivery System.

In addition, during the UAV delivery process, the UGV is driving on the road network. Considering the practical situation, the UGV cannot stop at any point of the road to release or recover the UAV, but it can release or recover it at some fixed points. Therefore, as shown in Figure 2, the road network is cut appropriately based on these fixed points in this study. A temporary docking point strategy is proposed for MOHRDS in this section. Figure 2 shows an undirected topological graph $G = (V, E)$ which based on this strategy, where the road network is cut according to the accuracy φ, the cut points are stored in $V = [v_1, v_2, \ldots v_n]$ and the edges of the predefined road network are stored in $E = \{e_{i,j}, \forall i, j\}$. The adjacency matrix $l \in \{0,1\}^{n \times n}$ is then introduced, where $l_{i,j} = 1$ indicates the presence of an undirected edge $e_{i,j}$ connecting vertices v_i and v_j, otherwise, $l_{i,j} = 0$. Finally, the weight $W_{i,j}$ is defined as the Euclidean distance $d_{i,j}$ between the vertex v_i and v_j.

$$W_{ij} = \begin{cases} d_{ij}, l_{ij} = 1 \\ \infty, l_{ij} = 0 \end{cases} \tag{1}$$

As shown in Figure 2, the temporary docking point strategy denotes v_i, $i = 1 \ldots n$ as the point for the UGV to release or recover the UAV. $F_{i,j} \in \{0,1\}^{m \times m}$ is defined as the UAV sorties, where $F_{i,j} = 1$ indicates the UAV continuously visits customers i to j in the delivery sequence P. v_k is defined as the UAV's release point $T_{k,i}$, which is the closest point in $V = [v_1, v_2, \ldots v_n]$ to customer i when $F_{i,j} = 1$. Similarly, $v_{k'}$ is defined as the UAV's recovery point $L_{j,k'}$, which is the closest point in $V = [v_1, v_2, \ldots v_n]$ to customer j when $F_{i,j} = 1$. The temporary docking point strategy ensures delivery safety while making full use of the UAV's endurance.

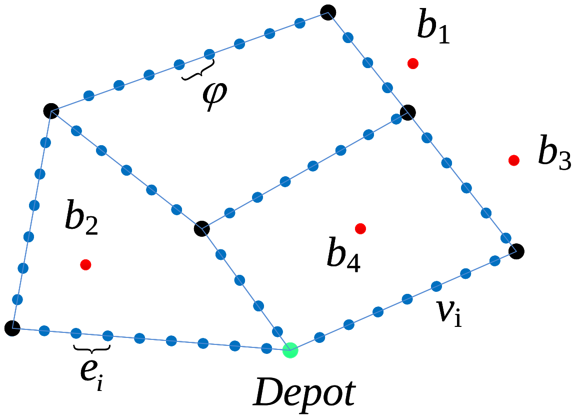


Figure 2. Temporary docking points strategy.

2.2. Definition of a New Delivery Customers Satisfaction

In the new global economy, customer satisfaction is highly valued as it is crucial to understand customer perceptions and evaluations. In the field of express delivery, customer satisfaction is related to whether the customers are served within their expected delivery time windows. However, the simple concept of time windows does not represent the CS well [20]. Therefore, this study is a deeper investigation of the previous time windows. According to expectancy theory, as the delivery time approaches the customer's expected timeframe, satisfaction increases rapidly, reflecting the characteristics of a convex function [21]. Conversely, in accordance with the economic principle of diminishing marginal utility, as the delivery time approaches the customer's final tolerance limit, their tolerance for delays decreases significantly, resulting in a faster decline in satisfaction. This aligns with the behavior of a concave function [22]. Based on these insights, a new customer satisfaction (CS) model for delivery time is introduced in this section. The membership function of CS is illustrated in Figure 3.

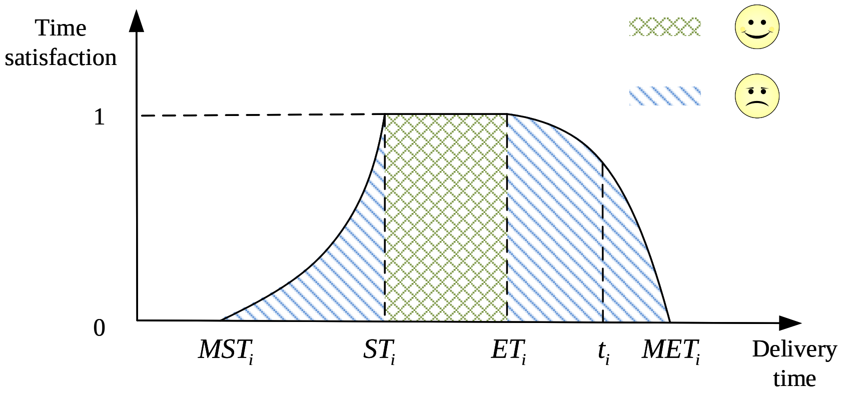


Figure 3. Customer satisfaction function.

This membership function reflects the customers' satisfaction and changes the classical hard time window $[ST_i, ET_i]$ to the quadruple $[MST_i, ST_i, ET_i, MET_i]$. The vector $[ST_i, ET_i]$ explains the soft time window and that its violation causes customer dissatisfaction; vector $[MST_i, MET_i]$ explains the hard time window and that its violation makes the problem infeasible. Of course, time windows may sometimes be violated for economic, operational, or even environmental causes [23]. The CS function of customer b_i is defined by the membership function as follows:

$$\delta(i) = \begin{cases} 0 & , t_i < MST_i \\ \left(\frac{t_i - MST_i}{ST_i - MST_i}\right)^2 & , MST_i \le t_i < ST_i \\ 1 & , T_i \le t_i < ET_i \\ 1 - \left(\frac{t_i - ET_i}{MET_i - ET_i}\right)^2 & , ET_i \le t_i \le MET_i \\ 0 & , t_i > MET_i \end{cases} \tag{2}$$

where t_i is the time for the UAV to arrive at customer b_i, MST_i is the earliest delivery time acceptable to the customer, ST_i is the preferred start time for delivery, ET_i is the time at which the customer starts to feel dissatisfied, and MET_i is the final delivery time that the customer can tolerate.

The CS function is a piecewise-linear function in that the function shape is subjected to the tightness of the time between two adjacent points of the vector. So, if a customer is served at its desired time window $[ST_i, ET_i]$, the grade of satisfaction is 1; otherwise, the degree of satisfaction gradually decreases along with the increase in the difference between the UAV's arrival time and desired times.

2.3. Route Scheduling Model for MOHRDS

Following the preceding discussions, to optimize both CS and delivery costs under road network constraints, a route scheduling model for a Multi-objective Heterogeneous Robotic Delivery System (MOHRDS) is constructed in this section. In order to facilitate the modeling process, it is stipulated that both the UAV and the UGV travel at a constant speed. Furthermore, the time required for the UAV to take off and land vertically, as well as the time spent serving each customer, can be considered negligible. The model encompasses two objective functions: minimizing travel distance and maximizing customer satisfaction. Additionally, various constraints are considered, such as the UAV′s limited endurance and payload capacity, time window restrictions, and ground traffic conditions. The corresponding symbols of MOHRDS are shown in Table 1. These symbols represent different variables and parameters used to describe various aspects of the problem.

Table 1. Symbol explanation.

Symbols	Description
E [1]	UAV endurance
Q	Maximum payload of the UAV
v_u	Flight speed of the UAV
v_c	Drive speed of the UGV
M	Number of customers
N	Number of temporary docking points
B	Set of customers $B = \{1, 2, \ldots, M\}$
A	The set A, consisting of customers B and temporary docking points V, in the undirected graph G.
P	The sequence P of UAV delivering services to customers.
p_i	The i-th customer in the delivery sequence set P.
q_i	The weight of the i-th customer's demand $q_i \in R,\ \ i = 1, 2, \ldots, M$.
$\lVert v_k - p_i \rVert$	The flying distance between the customer point p_i and the temporary docking point v_k for the UAV.
$D_{i,j}$	The flying distance between the customer points b_i and b_j for the unmanned aerial vehicle (UAV) $D_{i,j} \in R,\ i = 1, 2, \ldots, M;\ j = 1, 2, \ldots, M$.
$\Delta_{k,k'}$	The driving distance between the vehicle's temporary stopping points is v_k and v'_k. $\Delta_{k,k'} \in R,\ k = 1, 2, \ldots, N;\ k' = 1, 2, \ldots, N$

[1] E is the endurance limit of the UAV, and this constraint ensures the UAV's flight distance doesn't exceed its endurance based on Euclidean estimates.

Furthermore, which customer is served, which route each vehicle is taken, and which temporary docking points are utilized are determined by the decision variables. Therefore, there are five decision variables in the proposed model. The first decision variable in Equation (3) represents the service status of the customer. The second decision variable in Equation (4) represents the path of the UAV. The last three decision variables in Equations (5)–(7) represent the usage of temporary docking points.

$$R_{x,y} = \begin{cases} 1, \text{ if the } x\text{-th customer point is visited on the } y\text{-th delivery in sequence } P \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

$$F_{i,j} = \begin{cases} 1, \text{ if the UAV continuously access customer points } p_i \text{ and } p_j \\ 0, \text{ otherwise} \end{cases} \tag{4}$$

$$T_{k,i} = \begin{cases} 1, \text{ if the UAV takes off from temporary docking point } k \\ \quad \text{to visit customer } p_i \\ 0, \text{ otherwise} \end{cases} \tag{5}$$

$$L_{j,k} = \begin{cases} 1, \text{ if the UAV flies from customer } p_j \text{ to temporary docking point } k \\ \quad \text{to land on the UGV} \\ 0, \text{ otherwise} \end{cases} \tag{6}$$

$$C_{i,j} = \begin{cases} 1, \text{ if the UGV travels from point } i \text{ to point } j \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

The mathematical model for MOHRDS route scheduling proposed in this study is constructed as follows:

$$\begin{cases} \min f_1 = \sum\limits_{i \in M} \sum\limits_{j \in M} F_{i,j}\big(D_{i,j} + a + b\big) + \sum\limits_{i \in N} \sum\limits_{j \in N} C_{i,j} \Delta_{i,j} \\ \quad a = \sum\limits_{k \in N} T_{k,i} \lVert v_k - p_i \rVert \\ \quad b = \sum\limits_{k' \in N} L_{j,k'} \lVert v_{k'} - p_j \rVert \end{cases} \tag{8}$$

$$\max f_2 = \sum_{i \in M} \delta(i) \tag{9}$$

Subject to

$$\textstyle\sum_{i=1}^{N} T_{i,j} \leq 1, \quad \forall j \in M \tag{10}$$

$$\textstyle\sum_{j=1}^{N} L_{i,j} \leq 1, \quad \forall i \in M \tag{11}$$

$$\textstyle\sum_{x=1}^{M} R_{x,y} = 1, \quad \forall y \in M \tag{12}$$

$$\textstyle\sum_{y=1}^{M} R_{x,y} = 1, \quad \forall x \in M \tag{13}$$

$$\textstyle\sum_{i=1}^{m} \sum_{j=m}^{M} F_{i,j} = 1, \quad \forall m \in M \tag{14}$$

$$\textstyle\sum_{k=1}^{N} T_{k,i} - \sum_{j=1}^{M} F_{i,j} = 0, \quad \forall i \in M \tag{15}$$

$$\textstyle\sum_{i=1}^{M} F_{i,j} - \sum_{k=1}^{N} L_{j,k} = 0, \quad \forall j \in M \tag{16}$$

$$\textstyle\sum_{i=1}^{n} \left(L_{j,i} - T_{i,j}\right) \leq 0, \quad \forall n \in N,\, \forall j \in M \tag{17}$$

$$\textstyle\sum_{i=1}^{n} \left(T_{i,j+1} - L_{j,i}\right) \leq 0, \quad \forall n \in N,\, \forall j \in M-1 \tag{18}$$

$$\left(\textstyle\sum_{k=1}^{N} T_{k,i}a + \sum_{k'=1}^{N} L_{j,k'}b + D_{i,j}\right) F_{i,j} \leq E \tag{19}$$

$$F_{i,j} \textstyle\sum_{a=i}^{j} q_a \leq Q, \quad \forall i \in M,\, \forall j \in M \tag{20}$$

$$\begin{gathered} \tfrac{1}{v_u}\left(\textstyle\sum_{k=1}^{N} T_{k,i}a + \sum_{k'=1}^{N} L_{j,k'}b + D_{i,j}\right) \geq \tfrac{\Delta_{k,k'}}{v_c} \\ a = \left\|v_k - p_i\right\|,\ b = \left\|v_{k'} - p_j\right\|, \forall i \in M,\, \forall j \in M \end{gathered} \tag{21}$$

$$\textstyle\sum_{j=1}^{N} C_{i,j} = 1, \quad i = S \tag{22}$$

$$\textstyle\sum_{j=1}^{N} C_{i,j} = \sum_{k=1}^{N} C_{k,l} \quad i = S, l = S \tag{23}$$

$$\begin{gathered} t_i = \begin{cases} t_{i-1} + D_{k',k}/v_c & \text{if } D_{k',k} = 1 \\ t_{i-1} + \left\|v_k - p_i\right\|/v_u & \text{if } T_{k,i} = 1 \text{ or } L_{j,k'} = 1 \\ t_{i-1} + D_{i,j}/v_u & \text{if } F_{i,j} = 1 \end{cases} \\ t_0 = 0, \quad k, k' \in [S, N], \quad i, j \in M \end{gathered} \tag{24}$$

where Equations (8) and (9) are the objective functions, Equations (10) and (11) are the constraints between the UGV and the UAV, Equations (12)–(20) are the constraints of the UAV, and Equations (21) –(23) are the constraints of the UGV, the specific meaning is as shown in Table 2.

In the above model, the constraints of the road network, the UAV energy and load are considered, and CS and the delivery distance of MOHRDS are taken as the optimization objective of the route scheduling model simultaneously. Therefore, the delivery scheme obtained by solving the model takes into account customer satisfaction and delivery costs, which is more in line with the actual needs of express companies.

Table 2. Equation meaning.

Equations	Meanings
Equation (8)	Minimizing the total distance traveled by UGV and UAV
Equation (9)	Maximizing the customer satisfaction
Equation (10)	Indicate that each temporary docking point allows at most one takeoff for the UAV
Equation (11)	Indicate that each temporary docking point allows at most one landing for the UAV
Equation (12)	States that the UAV can only visit one customer at a time
Equation (13)	Ensures that each customer is visited only once
Equation (14)	Allows the UAV to serve multiple customers within one flight
Equation (15)	Stipulates that the UAV starts the delivery immediately after takeoff
Equation (16)	Requires the UAV to return to the UGV for replenishment immediately after completing the delivery task
Equation (17)	Ensures that the takeoff point for each flight of the UAV is before the landing point
Equation (18)	Indicates that the UAV can only take off for the next flight after completing replenishment from the previous flight
Equation (19)	Restricts the UAV's distance traveled within its endurance limit during the delivery task
Equation (20)	Limits the weight of packages carried by the UAV to its capacity
Equation (21)	Ensures that the UGV does not arrive at the landing point later than the UAV to facilitate timely recovery and replenishment
Equation (22)	States that each UGV is dispatched at most once
Equation (23)	Represents the movement of UGV carrying UAV from the depot, and they return to the depot after completing their tasks
Equation (24)	Defines the calculation of the UAV's arrival time at each customer

3. Modified Artificial Bee Colony Algorithm for the Problem

The artificial bee colony algorithm (ABC) is recognized as a probabilistic algorithm used to seek approximate optimal solutions [24]. It has been successfully applied in various fields, including the Traveling Salesman Problem (TSP) [25], image processing [26], logistics and path planning [27]. Notably, the convergence of the ABC algorithm has been rigorously proven [28]. Due to its effectiveness and proven convergence, the ABC algorithm is chosen as the route scheduling method for MOHRDS in this study. Two critical modifications of ABC are proposed in this section: discretization and multi-objective optimization. Regarding discretization, the original position update formula is replaced with a neighborhood search strategy to improve the convergence accuracy of the algorithm. In addition, a local search strategy based on rolling optimization is introduced to prevent the algorithm from falling into a local optimum. On the other hand, inspired by the Non-dominated Sorting Genetic Algorithm II (NSGA-II), a fast non-dominated sorting of solution individuals is performed to generate non-dominated solutions. A food source evaluation strategy is designed based on dominance strength and crowding distance to improve the diversity of optimal Pareto frontier solutions.

3.1. Overview of the ABC

ABC is a biological optimization method that simulates the intelligent search behavior of a swarm of bees. The algorithm finds the optimal food source through the collaborative efforts of three types of bee colonies. Like most bionic algorithms, the ABC consists of two processes: global exploration and local exploitation. The initialization phase and the scout bee phase perform random initialization of bees for global exploration with the following formula:

$$x_i^j = x^{j\min} + R \times (x^{j\max} - x^{j\min}) \tag{25}$$

Equation (25) represents the process of random initialization for the j-th dimension of the i-th individual, where R is a random number between $[0,1]$, $x^{j\min}$ and $x^{j\max}$ represent the minimum and maximum values of the j-th dimension.

The local exploitation of the ABC mainly occurs during the employed bee phase and the onlooker bee phase, where the exploitation process is achieved through information exchange with a randomly selected individual. The formula is as follows:

$$x_i^{j*} = x_i^j + R \times (x_i^j - x_k^j) \tag{26}$$

Equation (26) represents the local exploitation process of the j-th dimension of the i-th individual, where $k \neq i, R \in [-1, 1]$.

Furthermore, in the roulette wheel selection of ABC, it is necessary to evaluate the quality of each food source in advance. The fitness value is calculated as Equation (27):

$$fit_i = \begin{cases} 1/(1 + obj_i), & obj_i \geq 0 \\ 1 + |obj_i|, & obj_i < 0 \end{cases} \tag{27}$$

where obj_i is the objective function value of the i-th individual.

The onlooker bee is chosen based on the fitness value of food sources. The higher the fitness value, the greater the probability of being selected. Therefore, this selection method can be formulated as Equation (28):

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{28}$$

where p_i represents the selection probability, fit_i is the fitness value of the candidate solution X_i, and SN stands for the number of food sources.

3.2. Discretization of the ABC

Due to the discrete nature of the MOHRDS route scheduling problem, traditional ABC cannot search in the discrete solution space, thus ABC is discretized in this study. The following three steps are performed during the discretization process to ensure that ABC can search efficiently in the discrete solution space: (1) encoding and decoding, (2) position updating strategy, and (3) local search strategy, which is specified in Section 3.2.1, Section 3.2.2 and Section 3.2.3, respectively.

3.2.1. Encoding and Decoding

The encoding and decoding method of solutions is crucial in algorithm design, as it determines how the solution space is represented and how efficiently the algorithm can search. In this section, an integer encoding method is adopted to encode the solutions, based on the delivery sequence of customers, as illustrated in Figure 4. The continuous variables are converted into discrete variables by the encoding method to search in the discrete solution space. Meanwhile, the decoding rules are designed based on the proposed temporary docking point strategy to improve the solution efficiency of the algorithm. It can transform the solution into the corresponding transport route, as shown in Figure 5. The calculation process of the decoding method is as follows:

1. The UGV departs from depot S, carrying the UAV, and selects the temporary stopping point v_k closest to the first customer b_i in the decoded individual as the launch point $T_{k,i}$ for the UAV. It then checks if the constraints are satisfied. If the conditions are not met, a penalty is applied.
2. Iterate through the remaining customers until the current customer b_j does not meet the endurance or payload constraints. Then, return to the previous customer b_{j-1} and find the closest temporary stopping point v'_k to it as the recovery point $L_{j-1,k'}$ for the UAV. It checks if the constraints are satisfied. If the conditions are not met, a penalty is applied.
3. Remove the customers before b_j from the solution individual. If there are still unvisited customers remaining in the solution individual, go back to Step 1. Otherwise, proceed to Step 4.
4. At this point, all customers have been visited, indicating that all the missions of the UAV have been determined. The UGV, carrying the UAV, returns to depot S. As shown in Figure 5, both UGV and UAV paths are recorded. The total distance traveled by UGV and UAV is calculated based on Equations (8) and (9), and the overall customer satisfaction is evaluated.

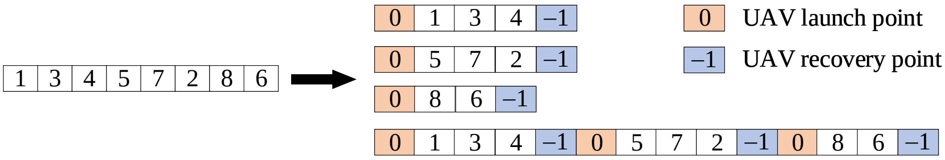


Figure 4. Encoding and Decoding.

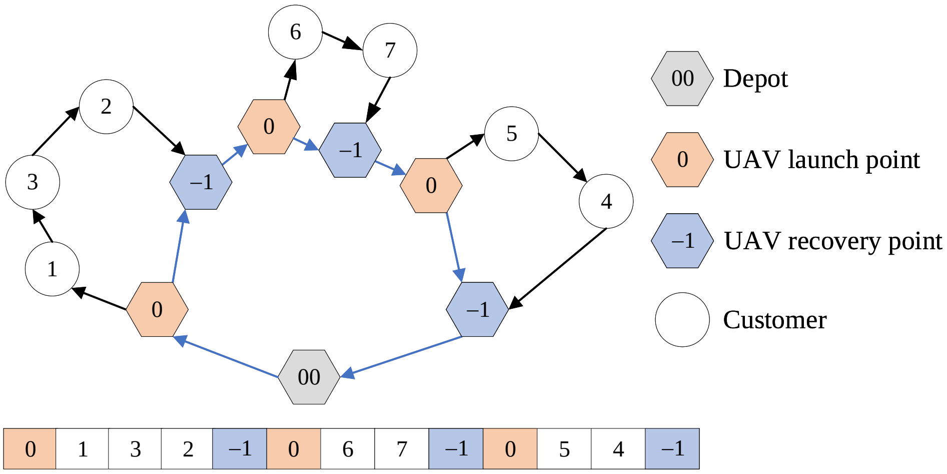


Figure 5. Corresponding rules for routes.

3.2.2. Position Updating Strategy

The path planning problem addressed in this study is of a highly intricate nature, encompassing the calculation of customer delivery sequences, the routing of UGV across the road network, and the selection of UAV and UGV releasing/recovering points. In order to enhance the efficacy of the ABC algorithm for this challenge, this section proposes a novel position update strategy. Furthermore, aiming at the defect of the ABC which is easy to fall into local optimum due to the lack of communication among the same kind of bees, a variety of neighborhood search operators in the position update phase of the ABC is introduced in this study to explore the solution space as much as possible [29–32]. Figure 6 illustrates all the neighborhood search operators used in ABC, and the specific are as follows:

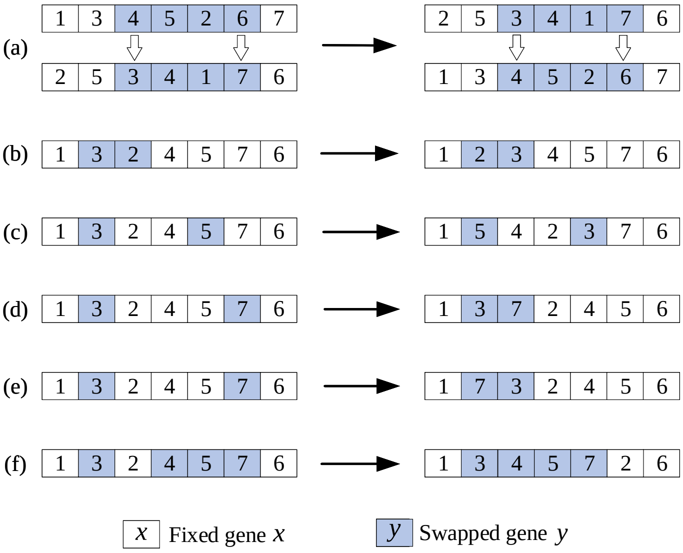


Figure 6. Neighborhood search operators. (**a**) PMX. (**b**) RCEM. (**c**) RCRM. (**d**) RCTM (+1). (**e**) RCTM (−1). (**f**) RCTM (−n).

- The Partial-Mapped Crossover (PMX) operator is an efficient and widely used crossover operator, especially for permutation optimization problems. The PMX operator cross-combines the gene information of the two parent individuals during the crossover process to generate a new offspring individual, which retains the chromosome structure of the parent individual and ensures that no gene will be lost in the offspring. PMX is introduced to avoid premature convergence to local optimal solutions in this section.
- Random Customer Exchange Mutation (RCEM) is a mutation operator with strong local search capability. RCEM can effectively explore the local solution space by exchanging elements within the neighborhood of the solution, thus helping to find the locally optimal solution. Due to the lack of convergence of the initial position update strategy, RCEM is introduced in this section to improve the local search capability of the algorithm.
- Random Customer Reversal Mutation (RCRM) is a mutation operator with strong disturbance. By inverting a part of the current solution, this operator is able to break the local structure of the original solution and thus explore new solutions in the search space. Due to the poor perturbation of the initial location update strategy, RCRM is introduced in this section to improve the diversity of the population.
- Random Customer Transfer Mutation (RCTM) is a mutation operator with simple strategy and good exploration ability. By inserting elements, RCTM can change the structure of the solution and increase the diversity of solutions to avoid falling into a local optimal solution. RCTM is introduced in this section to improve the possibility of finding the global optimal solution.

3.2.3. Local Search Strategy

The basic ABC algorithm suffers from the issue of strong exploration capability but weak exploitation capability [33]. Furthermore, the solution space under consideration in this paper is subject to a number of complex constraints, which serves to compound the challenge of searching with the ABC algorithm. In order to address this issue, a local search strategy is introduced to the ABC algorithm in this section, aiming to enhance its local exploitation capabilities and improve overall search efficiency.

The rolling optimization method is introduced in the local search strategy to reduce the range of solution space, as depicted in Figure 7.

Furthermore, to improve the local exploitation capability of the algorithm, six local search operators are designed in this section [20,34–36]. As shown in Figures 8 and 9, these operators are designed for searching within and between UAV sorties, respectively.

(i) Search operators within UAV sorties:

2-opt: Randomly select two customers, b_i and b_j, within the UAV sortie $F_{i,j}$, and reverse the order of the customers between b_i and b_j to obtain a new offspring solution.

Exchange: Randomly select two customers, b_i and b_j, within the UAV sortie $F_{i,j}$, and swap their positions to obtain a new offspring solution.

Insert: Randomly select two customers, b_i and b_j, within the UAV sortie $F_{i,j}$, and insert customer b_j before customer b_i, resulting in a new offspring solution.

(ii) Search operators between UAV sorties:

±insert: Randomly select two customers b_i and b_j from two UAV sorties $F^1_{i,j}$ and $F^2_{i,j}$. Insert customer b_j before (or after) customer b_i, resulting in a new offspring solution.

Swap: Randomly select two customers b_i and b_j from two UAV sorties $F^1_{i,j}$ and $F^2_{i,j}$. Swap the positions of customer b_i and customer b_j, resulting in a new offspring solution.

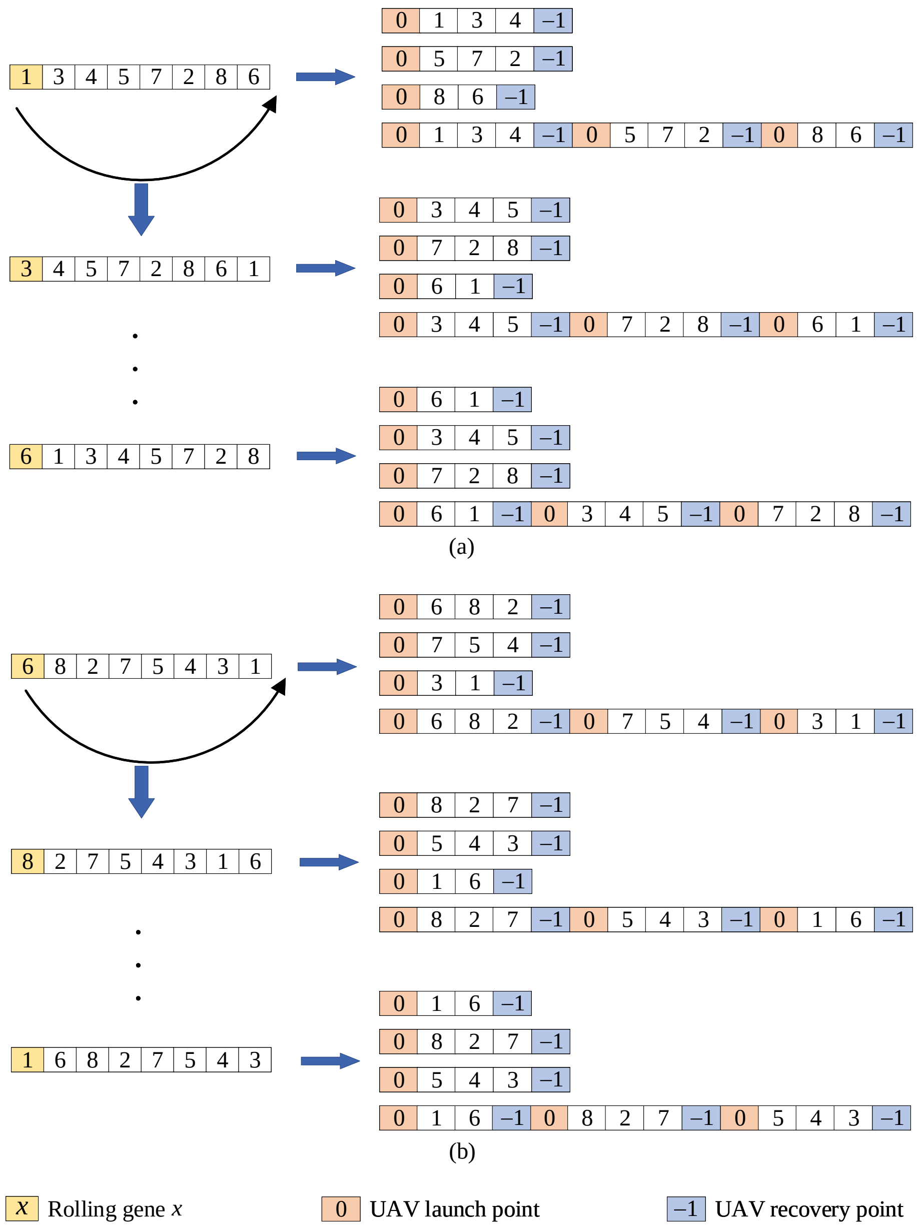


Figure 7. Rolling optimization strategy. (**a**) Forward Rolling Search. (**b**) Reverse Rolling Search.

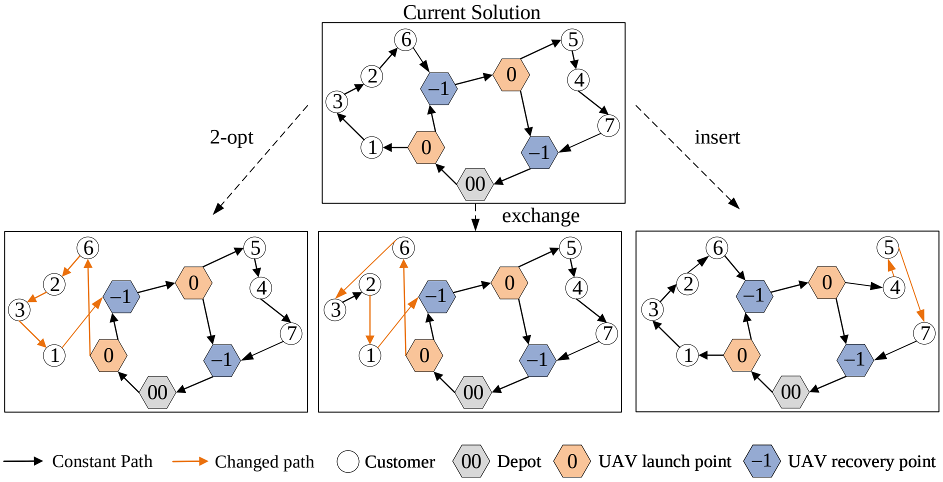


Figure 8. Local search operators within UAV sorties.

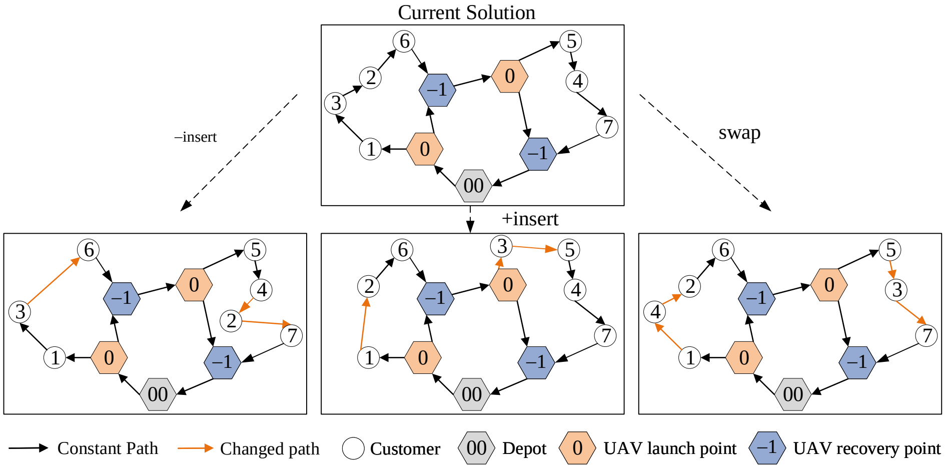


Figure 9. Local search operators between UAV sorties.

3.3. *Multi-Objective Improvement of the ABC*

Most of the previous studies weighted multiple objectives into one optimization objective. However, when the delivery requirements of express companies for HRDS change, the weights of different goals need to be adjusted again. It cannot meet the real-time demand of express delivery. Therefore, it is better to support multi-alternative non-inferior solutions for MOHRDS route scheduling problems. But there are two problems that should be solved to extend ABC to multi-objective ABC: (1) how to generate non-dominated solutions in the process of population evolution and constitute the optimal Pareto front; (2) how to choose the better solution in the onlooker bee phase. In this section, the fast non-dominated sorting method and a new food source evaluation strategy are introduced to solve these problems, and the specific processes are given in Section 3.3.1 and Section 3.3.2, respectively.

3.3.1. Acceptance Criteria

The traditional greedy-based acceptance criterion in ABC cannot generate non-dominated solutions, so in order to obtain the optimal Pareto front, a new acceptance criterion based on the external archiving method and the fast non-dominated sorting method is proposed in this section [37].

In this acceptance criterion, as shown in Figure 10, it is first assumed that K offspring solutions are generated by neighborhood search on parent solutions, and these offspring solutions are added to the external archive EA. Then, the fast non-dominated sorting method is used to sort between each parent solution α and its generated offspring solutions to obtain a set of mutually non-dominated solutions. The solution with the largest crowding distance in the Pareto front is selected as the new solution to improve the diversity of solutions, and compared with the dominance strength of α to judge whether it is necessary to update. The value of the *limit* is recorded according to the updated times. In addition, in order to enhance the convergence speed of the algorithm, after all the solutions in the population P are updated, the fast non-dominated sorting method is used in the mixed population of P and EA, and the solution set equal to the number of solutions in P is selected as the updated population P_{new}.

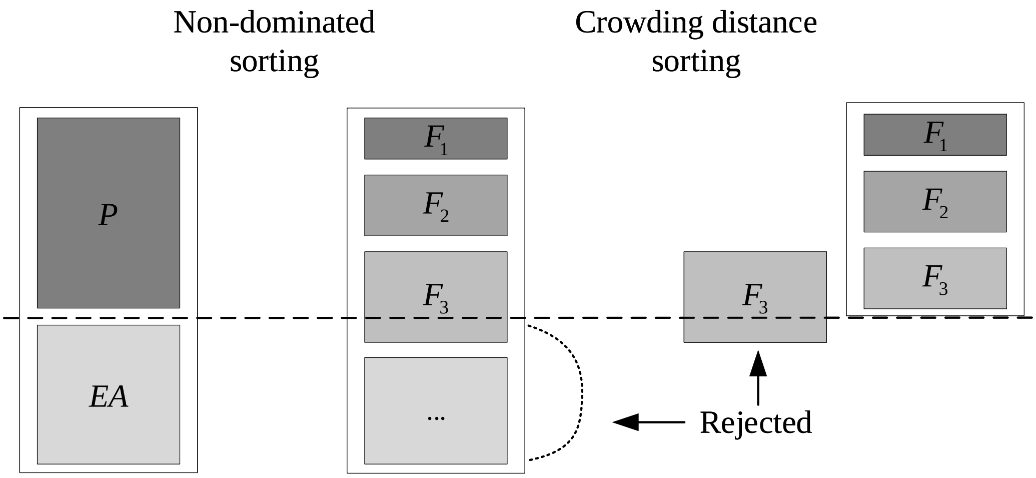


Figure 10. Acceptance criteria.

3.3.2. Food Source Evaluation Strategy

Multi-objective optimization problems do not have a fixed evaluation criterion regarding the merit of the solution due to the competition between the objective functions. This poses a challenge for the selection of onlooker bees in ABC. As shown in Figure 11, an evaluation strategy based on dominance strength and crowding distance is proposed in this section to solve the selection problem of the onlooker bee phase and improve the diversity of solutions [38]. Firstly, the dominance relationship is defined by Equation (29), the population is partitioned into several mutually disjoint subsets.

$$\begin{cases} S_1 = \{a \in S | \forall : b \in S,\ s.t : a \succ b\} \\ S_2 = \{a \in S - S_1 | \forall : b \in S - S_1,\ s.t : a \succ b\} \\ \quad \vdots \\ S_i = \{a \in S - \bigcup_{j=1}^{i-1} S_j | \forall : b \in S - \bigcup_{j=1}^{i-1} S_j,\ s.t : a \succ b\} \end{cases} \tag{29}$$

Then, for each solution θ in the set $S_l,\ l \in [1, i]$, its dominance strength value is defined as follows:

$$strength(\theta) = (1 - l/i) + 1/i,\ \forall \theta \in S_l \tag{30}$$

The elements in the first layer have the highest dominance strength value and then decrease with $1/i$ successively to ensure the superiority of non-dominated solutions.

In addition, the probabilities of finding new solutions near different solutions in the same dominance level are different. To increase the diversity of solutions and improve the population distribution, the crowding distance is introduced in this section when calculating selection probabilities. First, the boundary value $nObj \times 1.0001$ for the crowding distance is defined as the maximum value, where $nObj$ represents the number of optimization objectives. The value of the crowding distance needs to be controlled within the range of $(0, 1/i)$ to ensure the dominant strength is given higher priority when selecting individuals. The calculation is shown in Equation (31):

$$d(\theta) = \frac{\left(\sum_{k=1}^{nObj} \frac{f_k(\theta - 1) - f_k(\theta + 1)}{f_k^{\max} - f_k^{\min}} \right)}{nObj \times 1.0001 \times i} \tag{31}$$

where $f_k(\theta)$ denotes the k-th objective function value of the θ-th individual, $f_k^{\max}$ represents the maximum value of the k-th objective function, and $f_k^{\min}$ represents the minimum value of the k-th objective function.

After defining the two evaluation criteria, dominance strength and crowding distance, the fitness value of an individual is calculated according to Equation (32).

$$fitness(\theta) = strenth(\theta) + d(\theta) \tag{32}$$

Finally, during the onlooker bee phase, the roulette wheel selection method was used to select individuals in the population. Based on the obtained fitness values, the probability of an individual being selected is defined as follows:

$$P(\theta) = \frac{fitness(\theta)}{\sum_{i=1}^{|S|} fitness(\theta_i)} \quad (\theta,\ \theta_i \in S) \tag{33}$$

The above definition not only ensures that the selection probability of a dominated individual is smaller than the selection probability of all individuals dominating it, but also satisfies that the sparser the distribution of other individuals around a non-dominated individual, the higher the selection probability.

In summary, the flowchart of the proposed NSABC (Non-dominated Sorting ABC) algorithm is illustrated in Figure 12.

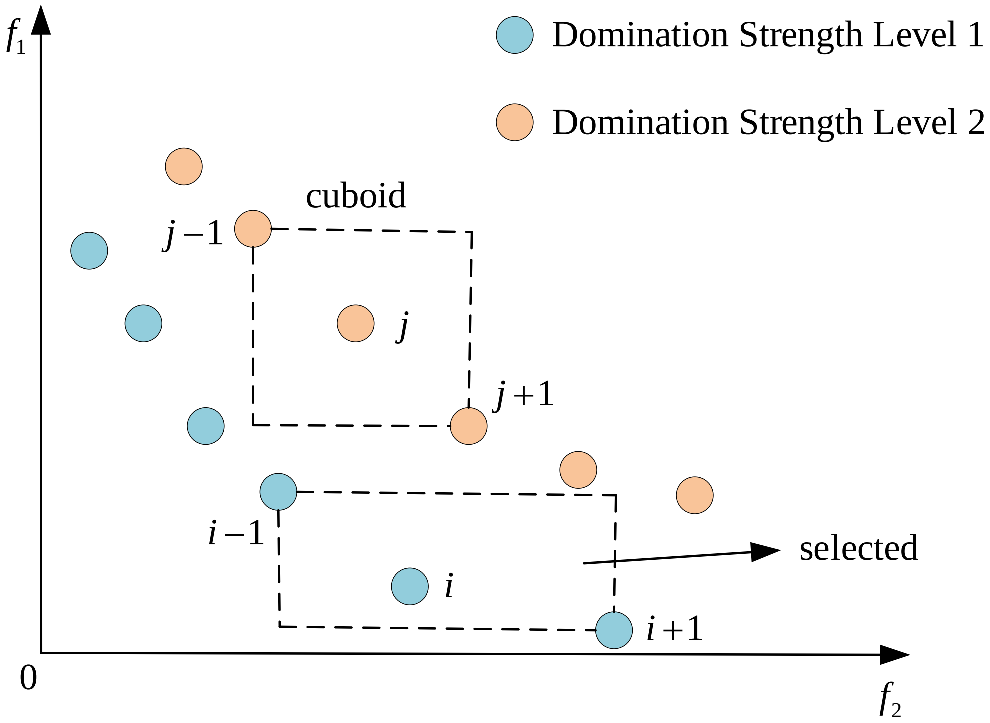


Figure 11. Food source evaluation strategy.

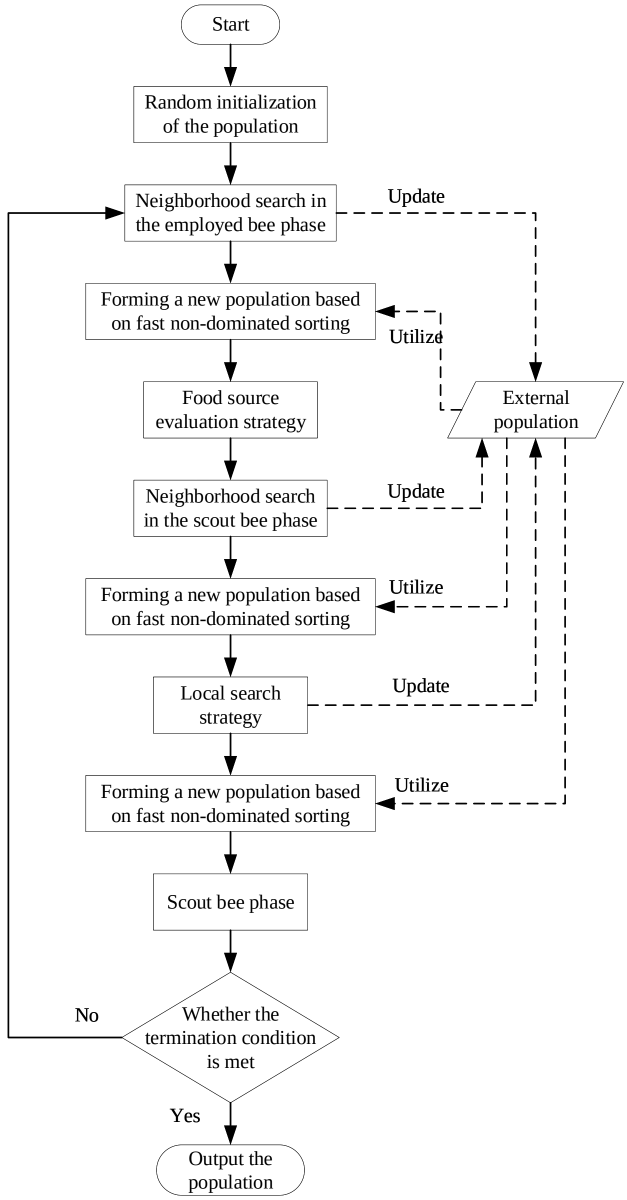

Start
Random initialization of the population
Neighborhood search in the employed bee phase
Update
Forming a new population based on fast non-dominated sorting
Utilize
Food source evaluation strategy
External population
Neighborhood search in the scout bee phase
Update
Forming a new population based on fast non-dominated sorting
Utilize
Local search strategy
Update
Forming a new population based on fast non-dominated sorting
Utilize
Scout bee phase
No
Whether the termination condition is met
Yes
Output the population

Figure 12. Flowchart of NSABC.

4. Simulations Validation

This section presents an evaluation study to assess the performance of the proposed NSABC algorithm. We conducted a simulation analysis on a computer equipped with an R9-5000 CPU and 16 GB of RAM using MATLAB R2022a.

4.1. Prerequisite Knowledge

4.1.1. Algorithms for Comparison

To verify the superiority of the proposed NSABC algorithm, this section compares it with the NLNSABC (No Local-search NSABC) algorithm and the NSGA-II. NLNSABC is a simplified version of the proposed NSABC in this study, which lacks the local search strategy. The NSGA-II is a classical multi-objective heuristic algorithm widely used in various fields, such as the TSP and the Vehicle Routing Problem (VRP) [39,40].

4.1.2. Algorithm Performance Metrics

For multi-objective optimization problems, decision-makers aim to achieve high accuracy, good diversity, and even distribution in the optimal Pareto front. Therefore, to quantitatively analyze the role of the proposed NSABC algorithm in MOHRDS route scheduling, three performance metrics are of particular interest to us [41].

(i) Quality Metric (QM): Mixing the optimal Pareto front solutions obtained by different algorithms, identifying the non-dominated solutions among them, and calculating the proportion of non-dominated solutions for each algorithm.

(ii) Spacing Metric (SM): The calculation method for this performance metric is as follows:

$$SM = \frac{\sum_{t=1}^{N-1} |d_{mean} - d_i|}{(N-1) \times d_{mean}} \tag{34}$$

where d_i represents the Euclidean distance between the obtained non-dominated solution i and its adjacent non-dominated solution $(i+1)$, and d_{mean} is the average value of all d_i. This metric measures the uniformity of the obtained non-dominated solutions.

(iii) Edge Convergence (EC): The calculation method for this performance metric is as follows:

$$EC = 0.01 \times cost + cs \tag{35}$$

where *cost* is the minimum travel distance in the optimal Pareto front and *cs* is the reciprocal of the maximum customer satisfaction in the optimal Pareto front. This performance metric measures the convergence of the boundaries of the multi-objective algorithm. Undoubtedly, a smaller value indicates better convergence of the algorithm.

Additionally, the concept of Pareto optimality is only the first step in solving multi-objective optimization problems. Subsequent decisions need to be made regarding the selection of a delivery plan. To provide decision-makers with more detailed reference solutions, this section defines the calculation of an optimal compromise solution in addition to the minimum delivery cost and maximum customer satisfaction solutions:

$$f(i) = \frac{cost(i)}{avs(i)} \tag{36}$$

where $f(i)$ is the ratio of delivery cost $cost(i)$ to average customer satisfaction $avs(i)$ of the i-th solution in the optimal Pareto front.

4.2. Simulation Analysis

In this section, data selection and extension were conducted based on the actual parcel delivery situation in a 15 km × 13 km area of Wuhan city, as depicted in Figure 13. The settings of the relevant parameters used are presented in Table 3, with the UAV speed derived from the modified DJI S300 and the UGV speed derived from the custom-built Ackermann vehicle (see Figure 14). The evaluation study consists of three experiments to validate the performance of the proposed NSABC algorithm that simulates a simple road network, a complex road network and tight time windows. The simple road network is a polygon surrounded by 15 path segments, consisting of 12 customers to be served and 110 temporary docking points, as depicted in Figure 15a. MOHRDS starts from the depot, performs package delivery tasks and returns to the depot after servicing all customers. The complex road network consists of 51 path segments, 26 customers to be served and 192 temporary docking points, shown in Figure 15b. MOHRDS starts from the depot to perform delivery tasks, with thousands of possible routes for MOHRDS to service all customers and return to the depot. Meanwhile, this section defines a method for calculating the degree of overlap between customer time windows to represent better the possibility of satisfying customers' demands under conflicting requirements. It conducts simulation analysis on time windows with different degrees of overlap.

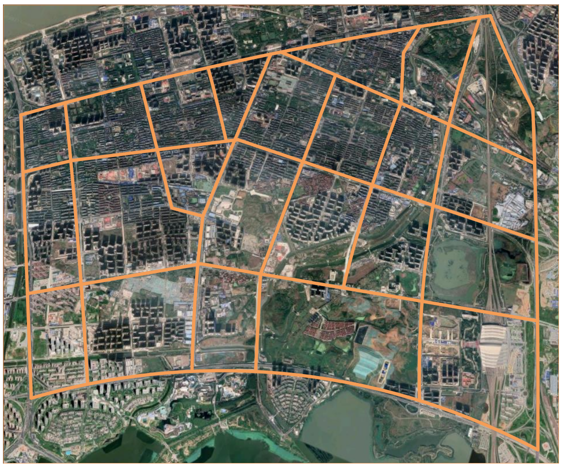


Figure 13. Satellite map of an area of Wuhan.

Table 3. Setting of parameters.

Description	Value	Description	Value
Iteration count	$iter_{max} = 100$	The speed of the UAV	$v_a = 60\,\text{km/h}$
Population size	$N_{pop} = 100$	The speed of the UGV	$v_g = 50\,\text{km/h}$
Number of employed bees	$N_{emp} = 100$	The endurance of the UAV	$E = 20\,\text{km}$
Number of onlooker bees	$N_{onl} = 100$	The payload capacity of the UAV	$Q = 20\,\text{kg}$
Number of stagnation iterations	$limit = 60$		

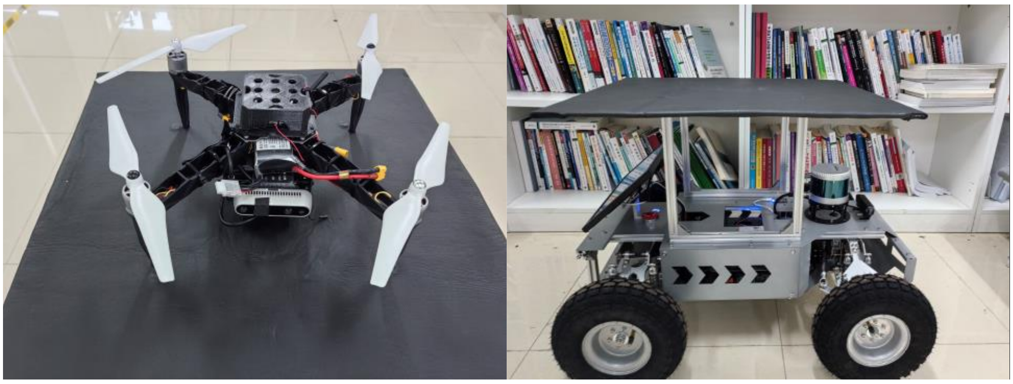
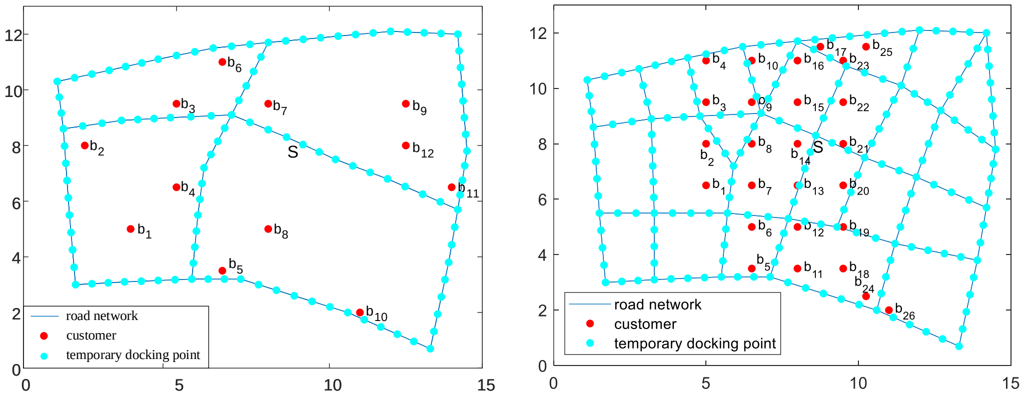


Figure 14. The actual UAV and UGV.



(**a**) (**b**)

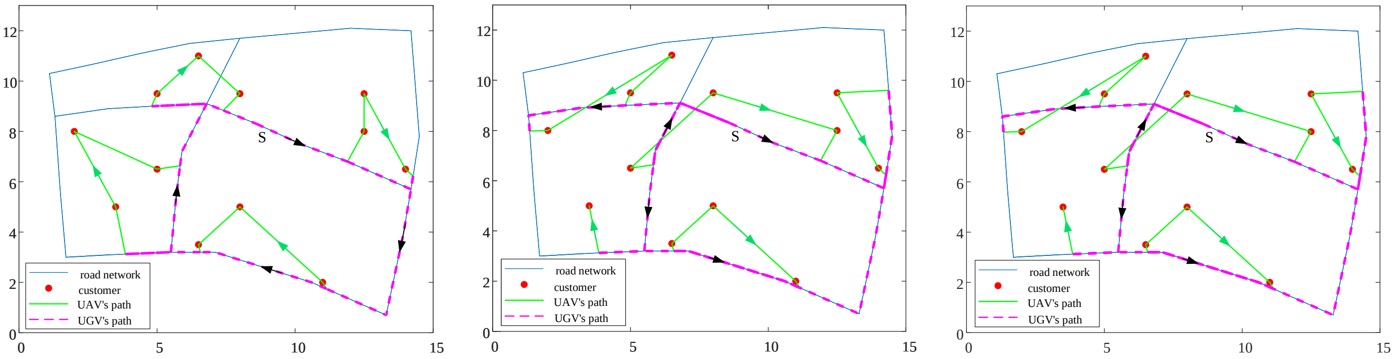
Figure 15. Topological map. (**a**) The simple topological map. (**b**) The complex topological map.

4.2.1. Simple Road Network

The simple road network is shown in Figure 15a and the customer demands and locations are provided in Table 4. The problem is solved by NSABC proposed in this study, and a set of mutually non-dominated delivery schemes are obtained. In order to provide a reference for the decision maker, as shown in Figure 16, three reference solutions are provided: the maximum customer satisfaction solution, the shortest delivery distance solution, and the optimal compromise solution. The MOHRDS delivery paths are shown in Table 5.

Table 4. The information of customers with simple road network restriction.

No.	Location	Demand (kg)	Time Windows	No.	Location	Demand (kg)	Time Windows
1	(3.5, 5.0)	1	[77, 79, 83, 85]	7	(8.0, 9.5)	2	[25, 27, 31, 33]
2	(2.0, 8.0)	0.5	[8, 10, 14, 16]	8	(8.0, 5.0)	1	[62, 64, 68, 70]
3	(5.0, 9.5)	1	[1, 3, 7, 9]	9	(12.5, 9.5)	0.5	[38, 40, 44, 46]
4	(5.0, 6.5)	1	[20, 22, 26, 28]	10	(11.0, 2.0)	1	[66, 68, 72, 74]
5	(6.5, 3.5)	2	[60, 62, 66, 68]	11	(14.0, 6.5)	1.5	[35, 37, 41, 43]
6	(6.5, 11.0)	1.5	[3, 5, 9, 11]	12	(12.5, 8.0)	0.5	[29, 31, 35, 37]



(**a**) (**b**) (**c**)

Figure 16. Delivery routes of MOHRDS under simple road network restriction. (**a**) The minimum delivery cost. (**b**) The maximum customer satisfaction. (**c**) The optimal compromise solution.

Table 5. The MOHRDS delivery routes with simple road network restriction.

Solution	Route	Delivery Distance	Customer Satisfaction
the shortest delivery distance solution	**UGV route:** $S \rightarrow v_{17} \rightarrow v_{85} \rightarrow v_{45} \rightarrow v_{36} \rightarrow v_{32} \rightarrow v_{58} \rightarrow v_{13} \rightarrow v_{72} \rightarrow S$ **UAV route:** $b_{12} \rightarrow b_9 \rightarrow b_{11} \rightarrow b_{10} \rightarrow b_8 \rightarrow b_5 \rightarrow b_1 \rightarrow b_2 \rightarrow b_4 \rightarrow b_3 \rightarrow b_6 \rightarrow b_7$	64.7335 km	0%
the maximum customer satisfaction solution	**UGV route:** $S \rightarrow v_{13} \rightarrow v_{56} \rightarrow v_{58} \rightarrow v_{17} \rightarrow v_{85} \rightarrow v_{104} \rightarrow v_{36} \rightarrow v_{45} \rightarrow v_{32} \rightarrow S$ **UAV route:** $b_3 \rightarrow b_6 \rightarrow b_2 \rightarrow b_4 \rightarrow b_7 \rightarrow b_{12} \rightarrow b_{11} \rightarrow b_9 \rightarrow b_5 \rightarrow b_8 \rightarrow b_{10} \rightarrow b_1$	102.683 km	100%
the optimal compromise solution	**UGV route:** $S \rightarrow v_{13} \rightarrow v_{56} \rightarrow v_{58} \rightarrow v_{17} \rightarrow v_{85} \rightarrow v_{104} \rightarrow v_{36} \rightarrow v_{45} \rightarrow v_{32} \rightarrow S$ **UAV route:** $b_3 \rightarrow b_6 \rightarrow b_2 \rightarrow b_4 \rightarrow b_7 \rightarrow b_{12} \rightarrow b_{11} \rightarrow b_9 \rightarrow b_5 \rightarrow b_8 \rightarrow b_{10} \rightarrow b_1$	102.683 km	100%

Interestingly, the path solution of the optimal compromise solution is the same as the path with maximum customer satisfaction. When the express company needs to improve market competitiveness, the decision maker can focus on the maximum customer satisfaction scheme; when the express company needs to reduce operating costs, the decision maker can focus on the shortest route scheme; when the express company needs to maintain stability, the decision maker can focus on the optimal compromise scheme.

Furthermore, to verify the superiority of the NSABC algorithm, this section compares NSABC with NLNSABC and NSGA-II. Figure 17a shows the optimal Pareto fronts obtained by these three algorithms. The encoding and decoding strategies proposed in this study enable all three algorithms to solve the problem. However, it is difficult to compare their performance solely based on the Pareto fronts visually. Therefore, to validate the optimization performance and stability of NSABC, this section conducts ten repetitions for each experimental group and calculates the QM and SM values for each repetition, as shown in Table 6.

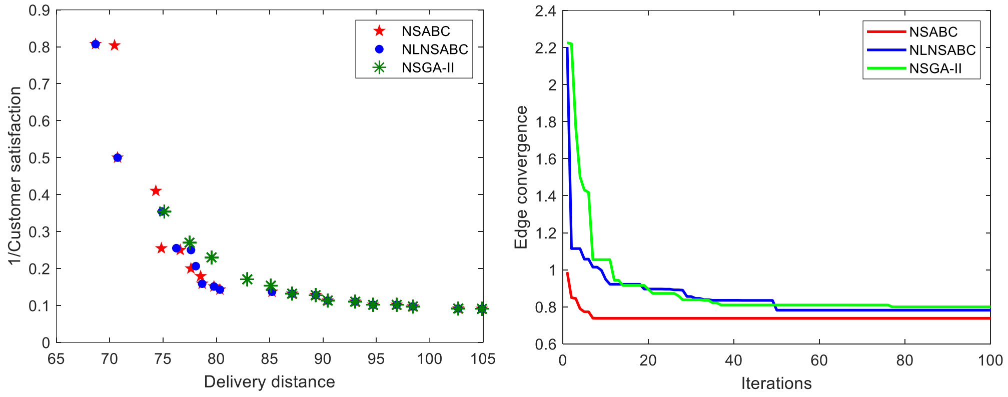
Table 6. The comparison of NSABC algorithm, NLNSABC algorithm and NSGAII algorithm under different constraints.

(a) Comparison of Pareto solutions obtained using three algorithms under simple road network restriction											
	Trials	1	2	3	4	5	6	7	8	9	10
QM	NSABC	**73.077**	**92.308**	**72.000**	**72.000**	**69.231**	**96.154**	**68.000**	**100**	**70.370**	**96.296**
	NLNSABC	50.000	61.379	44.000	44.000	42.308	65.385	48.000	73.077	48.148	66.667
	NSGAII	65.385	19.231	48.000	48.000	50.000	3.8462	48.000	0	14.815	3.7037
SM	NSABC	**1.1139**	1.1139	**1.1139**	1.1139	**1.1139**	**1.1139**	1.0821	**1.1139**	**1.1139**	**1.1139**
	NLNSABC	1.2640	1.2198	1.1610	**1.0710**	1.1141	1.5484	**0.9714**	1.3183	1.2944	1.1733
	NSGAII	1.6137	**0.6254**	1.4464	1.3882	1.6709	1.8512	1.9879	1.3452	1.2942	1.5175
(b) Comparison of Pareto solutions obtained using three algorithms under complex road network restriction											
	Trials	1	2	3	4	5	6	7	8	9	10
QM	NSABC	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	NLNSABC	0	0	0	0	0	0	0	0	0	0
	NSGAII	0	0	0	0	0	0	0	0	0	0
SM	NSABC	**0.1724**	**0.3456**	**0.2075**	**0.3492**	**0.1585**	0.2984	**0.3563**	**0.3353**	0.2209	**0.2979**
	NLNSABC	0.4985	1.0222	0.5119	0.5677	0.6818	**0.2849**	0.6529	0.4092	**0.2107**	1.5201
	NSGAII	4.3103	2.1266	0.2478	1.0276	1.8850	1.6967	0.8098	2.2704	1.1836	2.2978
(c) Comparison of Pareto solutions obtained using three algorithms under tight time windows restriction											
	Trials	1	2	3	4	5	6	7	8	9	10
QM	NSABC	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	NLNSABC	0	0	0	0	0	0	0	0	0	0
	NSGAII	0	0	0	0	0	0	0	0	0	0
SM	NSABC	0.9292	0.7881	0.7652	0.8047	1.0028	0.8733	0.7128	0.9702	0.9994	0.8728
	NLNSABC	**0.2448**	**0.2881**	**0.3538**	1.3584	0.9174	**0.7730**	0.9271	**0.6960**	0.7450	0.8794
	NSGAII	1.2276	0.5357	0.5901	**0.6555**	**0.6481**	0.9137	**0.5154**	1.0406	**0.6354**	**0.6296**

Abbreviation: QM: Quality Metric (%); SM: Spacing Metric; NSABC: Non-dominated Sorting Artificial Bee Colony Algorithm; NLNSABC: No Local-search NSABC; NSGAII: Non-dominated Sorting Genetic Algorithm.

Table 6(a) shows that in a simple road network, compared to NLNSABC and NSGA-II, NSABC exhibits superior performance in both the quality and distribution of the obtained non-dominated solutions. It should be noted that although NLNSABC and NSGA-II achieve smaller SM values in the second, fourth, and seventh experiments, their Quality Metric falls short compared to NSABC. Furthermore, as shown in Figure 17b, the convergence of the EC values for the three algorithms is presented. The NSABC converges after eight iterations with an optimal EC value of 0.7382. The NLNSABC algorithm converges after

50 iterations with an optimal EC value of 0.7821. The NSGA-II algorithm converges after 77 iterations with an optimal EC value of 0.8000. The results indicate that the NSABC algorithm achieves high solution efficiency and fast convergence. It can be attributed to considering both dominance strength and crowding distance when calculating the selection probabilities and the utilization of local search strategy before the scout bees start working, enabling the swarm to discover the optimal delivery solutions more easily.



(**a**) (**b**)

Figure 17. Simple road network simulation results. (**a**) Pareto front. (**b**) The EC curve.

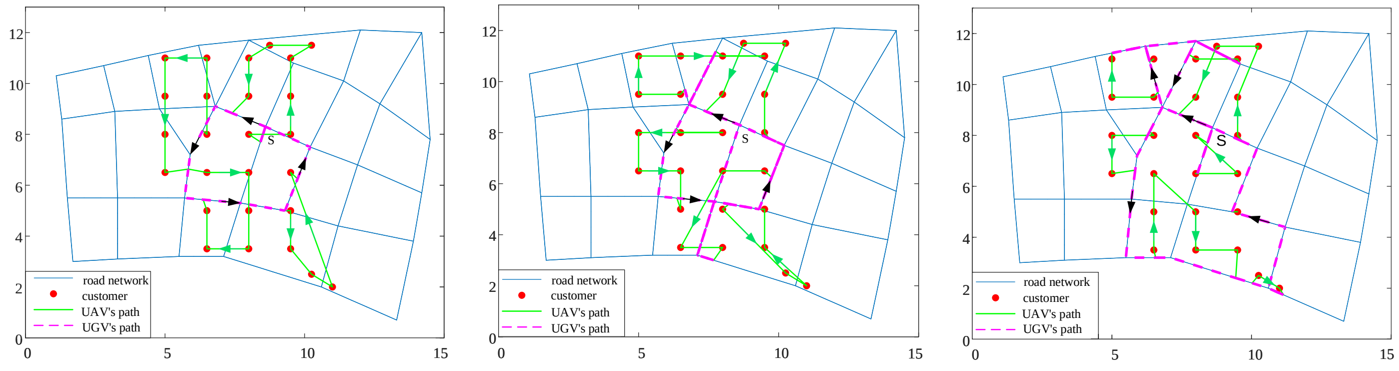
4.2.2. Complex Road Network

To validate NSABC's ability to solve complex problems, the road network and customers are expanded in this section to form a complex road network as shown in Figure 15b. The customer demand, locations, and other information are provided in Table 7. The complex problem is solved by NSABC and three reference solutions are provided as shown in Figure 18. The MOHRDS delivery routes are shown in Table 8. Similar to the previous section, decision-makers can make choices based on the actual solutions.

Table 7. The information of customers with complex road network restrictions.

No.	Location	Demand (kg)	Time Windows	No.	Location	Demand (kg)	Time Windows
1	(5.0, 6.5)	1	[12, 18, 28, 40]	14	(8.0, 8.0)	1.5	[0, 15, 30, 40]
2	(5.0, 8.0)	0.5	[15, 17, 21, 23]	15	(8.0, 9.5)	0.5	[46, 48, 52, 54]
3	(5.0, 9.5)	1	[0, 2, 6, 8]	16	(8.0, 11.0)	0.5	[8, 10, 14, 16]
4	(5.0, 11.0)	1	[2, 4, 8, 19]	17	(8.75, 11.5)	1	[44, 46, 50, 52]
5	(6.5, 3.5)	2	[30, 43, 53, 65]	18	(9.5, 3.5)	1	[29, 31, 35, 37]
6	(6.5, 5.0)	1.5	[0, 16, 40, 50]	19	(9.5, 5.0)	2	[24, 32, 43, 55]
7	(6.5, 6.5)	1	[15, 20, 26, 30]	20	(9.5, 6.5)	2	[53, 55, 59, 61]
8	(6.5, 8.0)	0.5	[6, 12, 28, 30]	21	(9.5, 8.0)	1.5	[39, 41, 45, 47]
9	(6.5, 9.5)	2	[0, 1, 7, 20]	22	(9.5, 9.5)	1.5	[40, 42, 46, 48]
10	(6.5, 11.0)	1	[3, 5, 9, 11]	23	(9.5, 11.0)	1	[6, 8, 12, 14]
11	(8.0, 3.5)	1	[20, 39, 67, 70]	24	(10.25, 2.5)	1	[30, 32, 36, 38]
12	(8.0, 5.0)	1	[5, 20, 35, 60]	25	(10.25, 11.5)	2	[32, 34, 48, 60]
13	(8.0, 6.5)	2	[54, 56, 60, 62]	26	(11.0, 2.0)	0.5	[0, 20, 40, 70]

Additionally, Figure 19a shows the optimal Pareto fronts obtained by the three algorithms in the complex road network environment, demonstrating that the Pareto front obtained by NSABC is superior to the other algorithms. To accurately illustrate the effectiveness of the proposed algorithm, each experiment in this section is repeated ten times, the QM and SM values for each experiment are calculated and presented in Table 6(b).

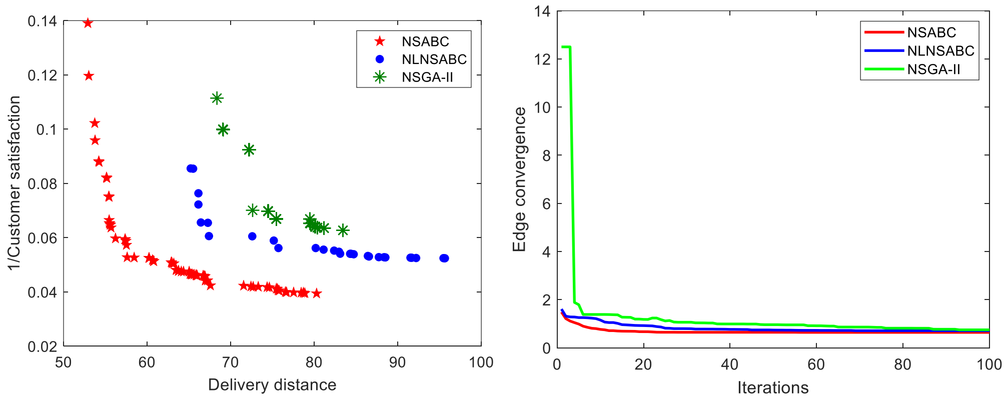


(**a**) (**b**) (**c**)

Figure 18. Delivery routes of MOHRDS under complex road network restriction. (**a**) The minimum delivery cost. (**b**) The maximum customer satisfaction. (**c**) The optimal compromise solution.

Table 8. The MOHRDS delivery routes with complex road network restriction.

Solution	Route	Delivery Distance	Customer Satisfaction
the shortest delivery distance solution	**UGV route:** $S \rightarrow v_{92} \rightarrow v_{126} \rightarrow v_{111} \rightarrow v_{86} \rightarrow v_{86} \rightarrow v_{83} \rightarrow v_{11} \rightarrow v_{96} \rightarrow S$ **UAV route:** $b_{14} \rightarrow b_{21} \rightarrow b_{22} \rightarrow b_{23} \rightarrow b_{25} \rightarrow b_{17} \rightarrow b_{16} \rightarrow b_{15} \rightarrow b_{8} \rightarrow b_{9} \rightarrow b_{10} \rightarrow b_{4} \rightarrow b_{3} \rightarrow b_{2} \rightarrow b_{1} \rightarrow b_{7} \rightarrow b_{13} \rightarrow b_{12} \rightarrow b_{11} \rightarrow b_{5} \rightarrow b_{6} \rightarrow b_{19} \rightarrow b_{18} \rightarrow b_{24} \rightarrow b_{26} \rightarrow b_{20}$	52.3691 km	35.5%
the maximum customer satisfaction solution	**UGV route:** $S \rightarrow v_{128} \rightarrow v_{134} \rightarrow v_{111} \rightarrow v_{83} \rightarrow v_{87} \rightarrow v_{11} \rightarrow v_{136} \rightarrow v_{126} \rightarrow v_{96} \rightarrow v_{50} \rightarrow S$ **UAV route:** $b_{9} \rightarrow b_{3} \rightarrow b_{4} \rightarrow b_{10} \rightarrow b_{23} \rightarrow b_{16} \rightarrow b_{2} \rightarrow b_{8} \rightarrow b_{14} \rightarrow b_{7} \rightarrow b_{1} \rightarrow b_{6} \rightarrow b_{12} \rightarrow b_{18} \rightarrow b_{24} \rightarrow b_{26} \rightarrow b_{19} \rightarrow b_{21} \rightarrow b_{22} \rightarrow b_{25} \rightarrow b_{17} \rightarrow b_{15} \rightarrow b_{20} \rightarrow b_{13} \rightarrow b_{5} \rightarrow b_{1}$	81.0334 km	100%
the optimal compromise solution	**UGV route:** $S \rightarrow v_{128} \rightarrow v_{159} \rightarrow v_{130} \rightarrow v_{28} \rightarrow v_{134} \rightarrow v_{111} \rightarrow v_{86} \rightarrow v_{46} \rightarrow v_{53} \rightarrow v_{54} \rightarrow v_{58} \rightarrow v_{11} \rightarrow v_{136} \rightarrow v_{126} \rightarrow v_{90} \rightarrow v_{92} \rightarrow S$ **UAV route:** $b_{9} \rightarrow b_{3} \rightarrow b_{4} \rightarrow b_{10} \rightarrow b_{23} \rightarrow b_{16} \rightarrow b_{8} \rightarrow b_{2} \rightarrow b_{1} \rightarrow b_{5} \rightarrow b_{6} \rightarrow b_{7} \rightarrow b_{12} \rightarrow b_{11} \rightarrow b_{18} \rightarrow b_{24} \rightarrow b_{26} \rightarrow b_{19} \rightarrow b_{21} \rightarrow b_{22} \rightarrow b_{25} \rightarrow b_{17} \rightarrow b_{15} \rightarrow b_{13} \rightarrow b_{20} \rightarrow b_{14}$	67.5776 km	90.7%



(**a**) (**b**)

Figure 19. Complex road network simulation results. (**a**) Pareto front. (**b**) The EC curve.

Table 6(b) shows the value of QM and SM is about the NSABC, the NLNSABC and the NSGA-II. In this study, the acceptance criterion enables NSABC to generate Pareto solutions, while the neighborhood search strategy and the local search strategy enhance the exploration space and convergence of NSABC, and obtain better QM values compared to other algorithms. The introduction of crowding distance makes the optimal Pareto front of NSABC have good diversity thus enhancing its SM value. As shown in Figure 19b, the convergence of the EC values for the three algorithms is presented. The NSABC converges at iteration 57, achieving the optimal EC value of 0.6394. The NLNSABC algorithm converges at iteration 92, with the optimal EC value of 0.7065. The NSGA-II algorithm converges at iteration 93, with the optimal EC value of 0.7463. These results demonstrate that the NSABC algorithm exhibits superior convergence speed and convergence performance in complex road network environments. Furthermore, the effectiveness of the proposed local search strategy in complex problems is also demonstrated.

4.2.3. Tight Time Windows

In practical package delivery processes, customers have varying widths of time windows. Some time windows are very tight, while others have minimal constraints. For this reason, this section introduces an overlap index to measure the tightness of customer-related time windows [42]. The formula for calculating the index value is as follows:

$$Overlap\ Index = \frac{\sum_{i=2}^{n}\sum_{j=1}^{m}(\beta_i - \alpha_i) \cap (\beta_j - \alpha_j)}{n-1} \tag{37}$$

where α_i is the earliest time a customer can accept a service, while β_i is the latest time a customer can be serviced. The higher the index, the tighter the time windows, resulting in more constraining sets to achieve customer satisfaction.

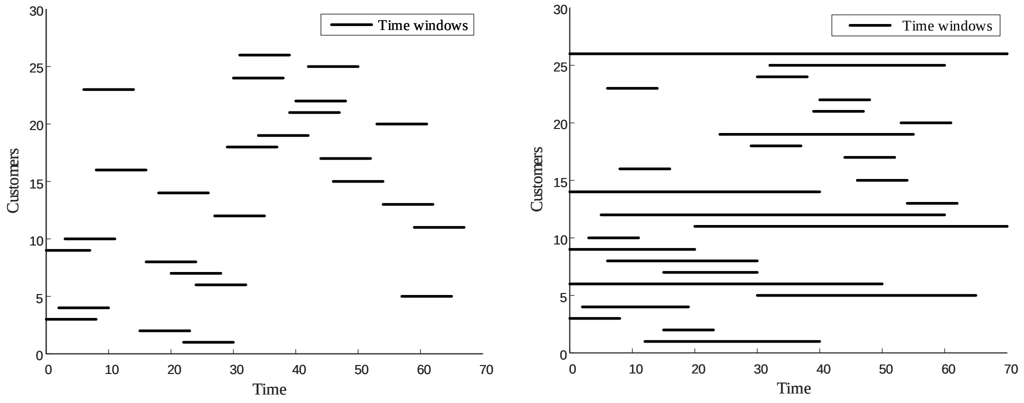
To validate the potential of NSABC in satisfying conflicting customer demands, this section extends the original customer time windows to a set of customers with significantly conflicting satisfaction requirements. The location and time information of the customers are provided in Table 9, and the overlap degree of the time windows for the two datasets is illustrated in Figure 20. Figure 20a represents the tight time windows corresponding to this section's dataset, while Figure 20b shows the lenient time windows from the dataset in Section 4.2.2. As shown in Figure 21, the problem is solved by NSABC, which provides decision-makers with three reference solutions. The MOHRDS delivery paths are shown in Table 10.

Table 9. The information of customers with tight time windows restriction.

No.	Location	Demand (kg)	Time Windows	No.	Location	Demand (kg)	Time Windows
1	$(5.0, 6.5)$	1	$[22, 24, 28, 30]$	14	$(8.0, 8.0)$	1.5	$[18, 20, 24, 26]$
2	$(5.0, 8.0)$	0.5	$[15, 17, 21, 23]$	15	$(8.0, 9.5)$	0.5	$[46, 48, 52, 54]$
3	$(5.0, 9.5)$	1	$[0, 2, 6, 8]$	16	$(8.0, 11.0)$	0.5	$[8, 10, 14, 16]$
4	$(5.0, 11.0)$	1	$[2, 4, 8, 10]$	17	$(8.75, 11.5)$	1	$[44, 46, 50, 52]$
5	$(6.5, 3.5)$	2	$[57, 59, 63, 65]$	18	$(9.5, 3.5)$	1	$[29, 31, 35, 37]$
6	$(6.5, 5.0)$	1.5	$[24, 26, 30, 32]$	19	$(9.5, 5.0)$	2	$[34, 36, 40, 42]$
7	$(6.5, 6.5)$	1	$[20, 22, 26, 28]$	20	$(9.5, 6.5)$	2	$[53, 55, 59, 61]$
8	$(6.5, 8.0)$	0.5	$[16, 18, 22, 24]$	21	$(9.5, 8.0)$	1.5	$[39, 41, 45, 47]$
9	$(6.5, 9.5)$	2	$[0, 1, 5, 7]$	22	$(9.5, 9.5)$	1.5	$[40, 42, 46, 48]$
10	$(6.5, 11.0)$	1	$[3, 5, 9, 11]$	23	$(9.5, 11.0)$	1	$[6, 8, 12, 14]$
11	$(8.0, 3.5)$	1	$[59, 61, 65, 67]$	24	$(10.25, 2.5)$	1	$[30, 32, 36, 38]$
12	$(8.0, 5.0)$	1	$[27, 29, 33, 35]$	25	$(10.25, 11.5)$	2	$[42, 44, 48, 50]$
13	$(8.0, 6.5)$	2	$[54, 56, 60, 62]$	26	$(11.0, 2.0)$	0.5	$[31, 33, 37, 39]$

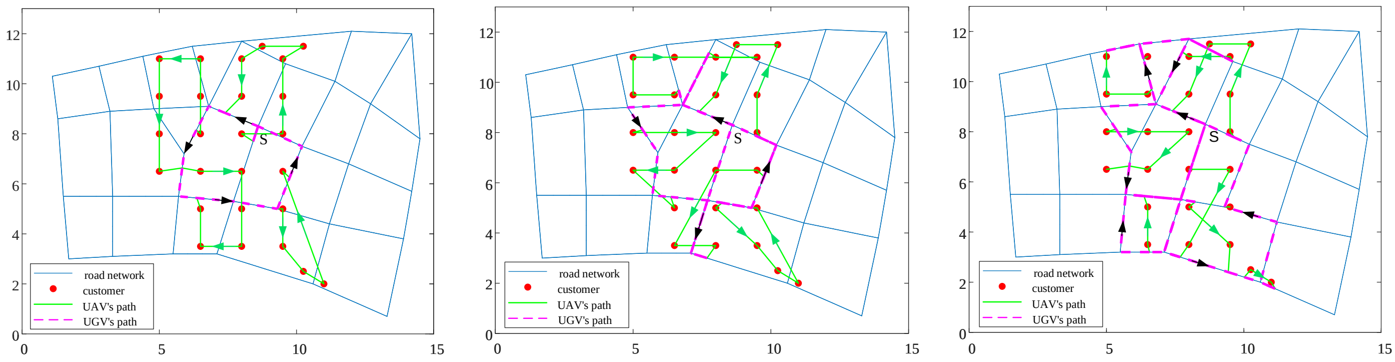
In addition, the optimal Pareto fronts solved by the three algorithms are depicted in Figure 22a, and it is evident that NSABC maintains a dominant position in terms of the optimal Pareto front compared to the other two algorithms.

Table 6(c) shows that the NSABC algorithm still achieves superior non-dominated solution quality compared to NLNSABC and NSGA-II when dealing with highly conflicting time windows. NLNSABC and NSGA-II fail to converge to the optimal Pareto front obtained by NSABC. Although they exhibit better performance in terms of SM, the quality gap of the optimal Pareto front remains irreparable. As evident from the EC curves in Figure 22b, the NSABC algorithm converges at iteration 77 with the optimal EC value of 0.5675, while NLNSABC converges at iteration 90 with the optimal EC value of 0.7138, and NSGA-II converges at iteration 93 with the optimal EC value of 0.8458. These results indicate that the NSABC algorithm maintains efficiency and quality in solving the problem even with significant conflicts in customer time windows.



(**a**) (**b**)

Figure 20. Time windows with different levels of overlap. (**a**) Tight time windows. (**b**) Loose time windows.



(**a**) (**b**) (**c**)

Figure 21. Delivery routes of MOHRDS under tight time windows restriction. (**a**) The minimum delivery cost. (**b**) The maximum customer satisfaction. (**c**) The optimal compromise solution.

Table 10. The MOHRDS delivery routes with tight time windows constraint.

Solution	Route	Delivery Distance	Customer Satisfaction
the shortest delivery distance solution	**UGV route:** $S \rightarrow v_{92} \rightarrow v_{126} \rightarrow v_{111} \rightarrow v_{86} \rightarrow v_{86} \rightarrow v_{83} \rightarrow v_{11} \rightarrow v_{96} \rightarrow S$ **UAV route:** $b_{14} \rightarrow b_{21} \rightarrow b_{22} \rightarrow b_{23} \rightarrow b_{25} \rightarrow b_{17} \rightarrow b_{16} \rightarrow b_{15} \rightarrow b_{8} \rightarrow b_{9} \rightarrow b_{10} \rightarrow b_{4} \rightarrow b_{3} \rightarrow b_{2} \rightarrow b_{1} \rightarrow b_{7} \rightarrow b_{13} \rightarrow b_{12} \rightarrow b_{11} \rightarrow b_{5} \rightarrow b_{6} \rightarrow b_{19} \rightarrow b_{18} \rightarrow b_{24} \rightarrow b_{26} \rightarrow b_{20}$	52.3691 km	13.9%
the maximum customer satisfaction solution	**UGV route:** $S \rightarrow v_{128} \rightarrow v_{134} \rightarrow v_{108} \rightarrow v_{83} \rightarrow v_{87} \rightarrow v_{11} \rightarrow v_{136} \rightarrow v_{126} \rightarrow v_{96} \rightarrow v_{50} \rightarrow S$ **UAV route:** $b_{9} \rightarrow b_{3} \rightarrow b_{4} \rightarrow b_{10} \rightarrow b_{23} \rightarrow b_{16} \rightarrow b_{2} \rightarrow b_{8} \rightarrow b_{14} \rightarrow b_{7} \rightarrow b_{1} \rightarrow b_{6} \rightarrow b_{12} \rightarrow b_{18} \rightarrow b_{24} \rightarrow b_{26} \rightarrow b_{19} \rightarrow b_{21} \rightarrow b_{22} \rightarrow b_{25} \rightarrow b_{17} \rightarrow b_{15} \rightarrow b_{20} \rightarrow b_{13} \rightarrow b_{5} \rightarrow b_{1}$	81.0334 km	100%
the optimal compromise solution	**UGV route:** $S \rightarrow v_{128} \rightarrow v_{159} \rightarrow v_{130} \rightarrow v_{28} \rightarrow v_{134} \rightarrow v_{108} \rightarrow v_{86} \rightarrow v_{87} \rightarrow v_{53} \rightarrow v_{54} \rightarrow v_{58} \rightarrow v_{11} \rightarrow v_{136} \rightarrow v_{126} \rightarrow v_{90} \rightarrow v_{50} \rightarrow v_{46} \rightarrow v_{83} \rightarrow S$ **UAV route:** $b_{9} \rightarrow b_{3} \rightarrow b_{4} \rightarrow b_{10} \rightarrow b_{23} \rightarrow b_{16} \rightarrow b_{2} \rightarrow b_{8} \rightarrow b_{14} \rightarrow b_{7} \rightarrow b_{1} \rightarrow b_{12} \rightarrow b_{18} \rightarrow b_{24} \rightarrow b_{26} \rightarrow b_{19} \rightarrow b_{21} \rightarrow b_{22} \rightarrow b_{25} \rightarrow b_{17} \rightarrow b_{15} \rightarrow b_{13} \rightarrow b_{20} \rightarrow b_{11} \rightarrow b_{5} \rightarrow b_{6}$	75.0424 km	94.8%

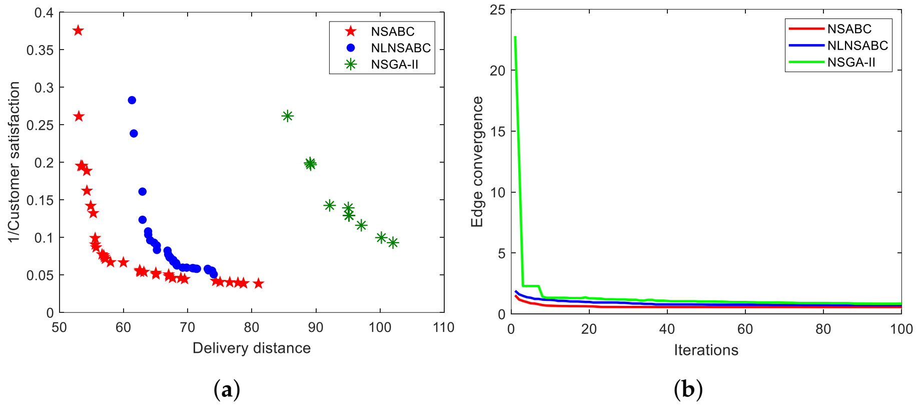


Figure 22. Tight time windows simulation results. (**a**) Pareto front. (**b**) The EC curve.

4.3. The Computational Complexity of NSABC

This section presents an analysis of the complexity of the NSABC algorithm, with a particular focus on its performance across different input sizes. This approach enabled the evaluation of the algorithm's efficiency, performance, and potential advantages in a real-world scenario.

To ensure the reliability of the results, each set of experiments was repeated ten times. The results of the experiments are presented in Table 11. The data presented in the table illustrate the running time of the NSABC algorithm under two distinct road network scales: a simple road network comprising 110 temporary docking points and 12 customer points, and a complex road network comprising 192 temporary docking points and 26 points. The two types of road networks represent disparate input scales, thereby enabling a comparison of the algorithm's performance in simple versus complex environments.

Table 11. Running time of NSABC across different input sizes.

	Simple Road Network	Complex Road Network	Increased Running Time
Average running time (s)	542.903	626.249	15.3519%
Shortest running time (s)	452.237	546.979	20.9496%
Longest running time (s)	644.638	757.295	17.4760%

It can be observed that as NSABC progresses from handling simple road networks to more intricate ones, the running time increases by approximately 15% to 20%. This increase is considerably less pronounced than the expansion of the input size, which suggests that NSABC remains an efficient solution when dealing with large-scale data. When considering the context of package delivery, the low growth rate of NSABC implies that it is well-suited to handling an increasing number of customers while maintaining high operational efficiency in scenarios involving complex road networks and large customer bases.

5. Physical Experiments to Prove the Concept of the Model

To further validate the effectiveness and practicality of the above MOHRDS route scheduling method, this section will conduct physical experiments as a proof of concept based on the numerical simulation experiments in Section 4 for verification and analysis.

5.1. Setup of the Physical Experiment

In this section, physical experiments on HRDS route scheduling are conducted using the scenario described in Section 4.2.2. The real environment used in the physical experiment is shown in Figure 23, with dimensions of 3 m × 5 m. To enhance UAV positioning accuracy, replace GPS with the NOKOV optical 3D motion capture system, which provides highly accurate position and posture measurements [43]. The experimental equipment is shown in Figure 24. In addition, in order to observe the HRDS movement trajectory more intuitively, this study has scaled down the complex road network used in the simulation experiments and simulated the experimental ground road network using tape on the floor. The 3D scene of the physical experimental platform is depicted in Figure 25.

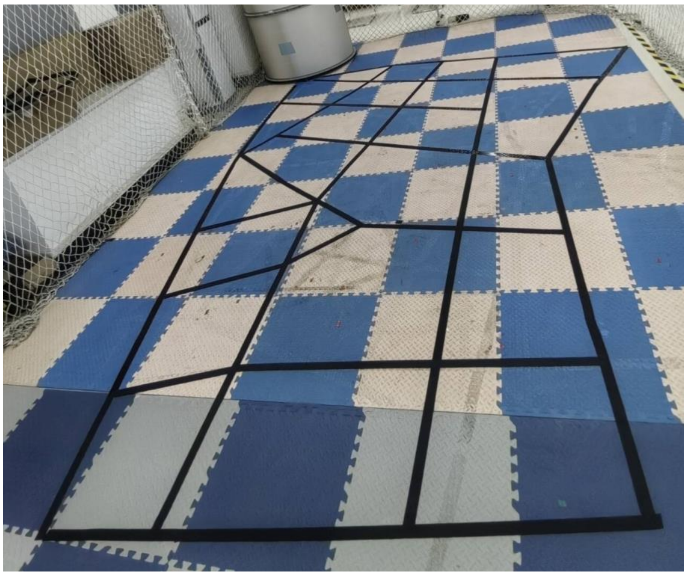


Figure 23. The experimental scenario.

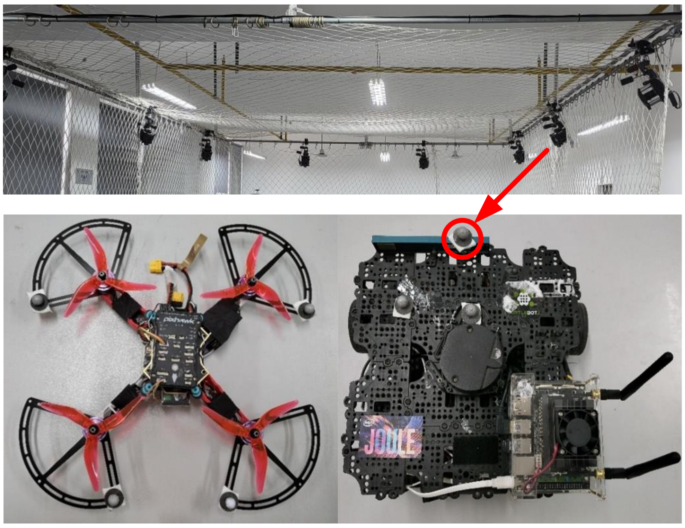


Figure 24. Experimental equipment.

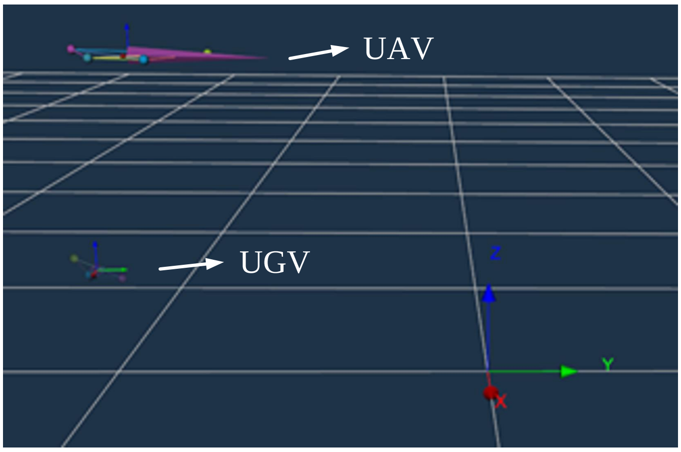


Figure 25. The 3D scene.

5.2. Experimental Results and Analysis

Physical experiments in this section are conducted using Figure 18a as an example. Figure 26a shows a physical shot of UAV and UGV separation during a mission: the UAV carries the package away from the UGV to perform delivery tasks, while the UGV moves to the next temporary stop point to provide replenishment. Figure 26b shows a physical shot of the UAV and UGV during a convergence. At this moment, the UAV is heading to the temporary stop point where the UGV has arrived early to perform replenishment operations. Once the replenishment is complete, the UAV will continue on to the next customer point, while the UGV will proceed to the next temporary stopping point. This cooperation between the UAV and the UGV ensures smooth delivery operations. More visualization results can be found at https://www.bilibili.com/video/BV1QYehenEZA/?vd_source=18395f0af87f302242f1ac8a71ac41fe; accessed on 25 July 2024.

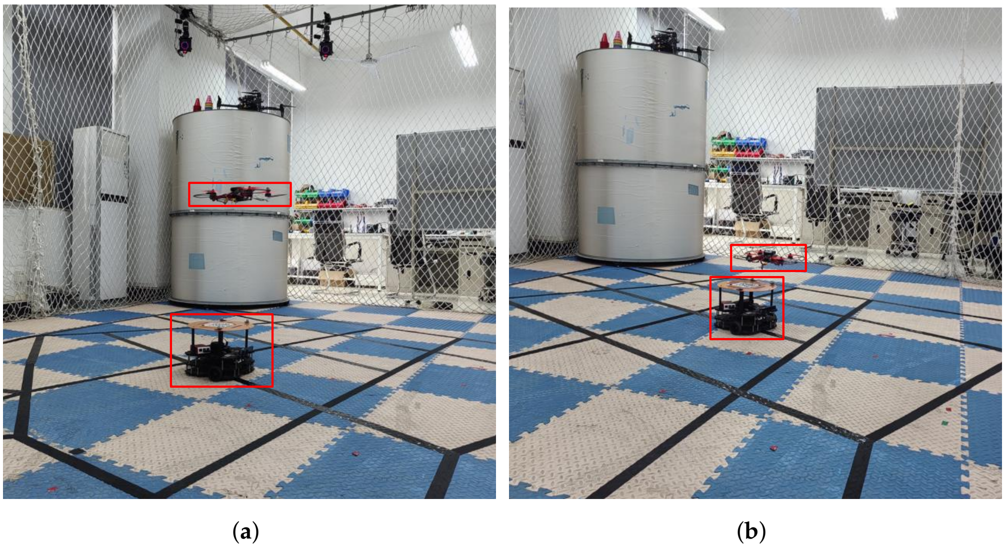

(**a**) (**b**)

Figure 26. The Collaboration Process between UAV and UGV. (**a**) Separation of the UAV and the UGV. (**b**) Convergence of the UAV and the UGV.

This study employs the 3D coordinates of various points captured by the high-speed camera in the motion capture environment to ascertain the actual flight trajectory of four UAV flights and the driving trajectory of the UGV in the road network during the physical experiment. Figure 27 illustrates that the discrepancy between the actual flight trajectory of the UAV and the simulated path is minimal, and the UAV largely follows the simulated delivery path. The primary discrepancy between the actual flight trajectory of the UAV and the simulated path is that when the distance between the UAV and the target customer point is less than 10 cm, it is considered to have reached the customer point. Therefore, the deviation is within an acceptable range. Additionally, the deviation between the trajectory of the UGV and the road network is also minimal, without exceeding the road network range. Furthermore, the arc-shaped trajectory depicted in the figure is generated when the UAV and UGV change direction. The observation results indicate that the actual paths of the UAV and UGV are largely consistent with the simulated paths, thereby corroborating the reliability of the simulation experiment and attesting to the efficacy and viability of the model and algorithm proposed in this study.

Figure 28 illustrates the 3D coordinate variation curves of HRDS, with the vertical axis representing the corresponding axis position changes and the horizontal axis representing the running time. From the graph, it can be observed that the flight control system of the UAV is relatively stable, without significant jitter. Moreover, the 3D coordinate transformation curve of the UGV during the delivery process exhibits significant jitter in the z-axis direction, which can be attributed to the UGV's bouncing caused by uneven ground. Throughout the entire HRDS delivery task, the UAV conducted a total of four landing replenishment operations and successfully completed the delivery task. These results indicate a high degree of precision in the path planning experiment, with minimal motion errors. This demonstrates the effectiveness and practicality of the algorithm designed in this study.

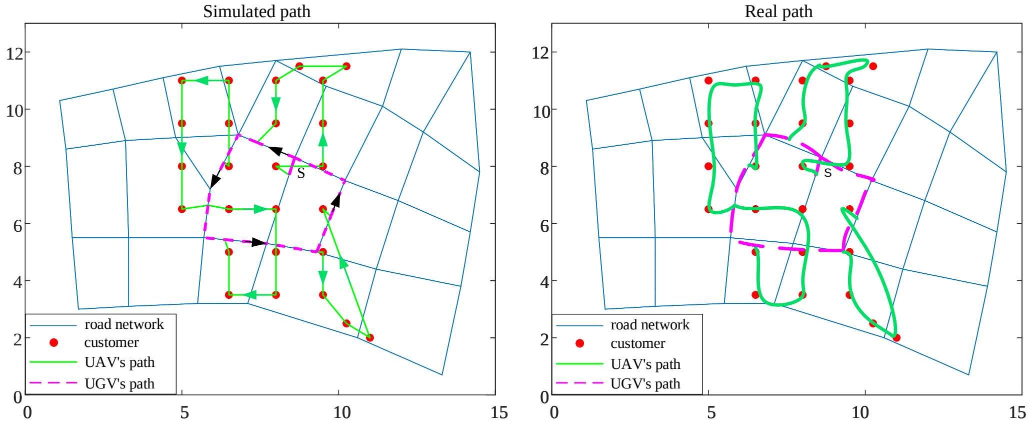


Figure 27. Comparison between simulated and real paths.

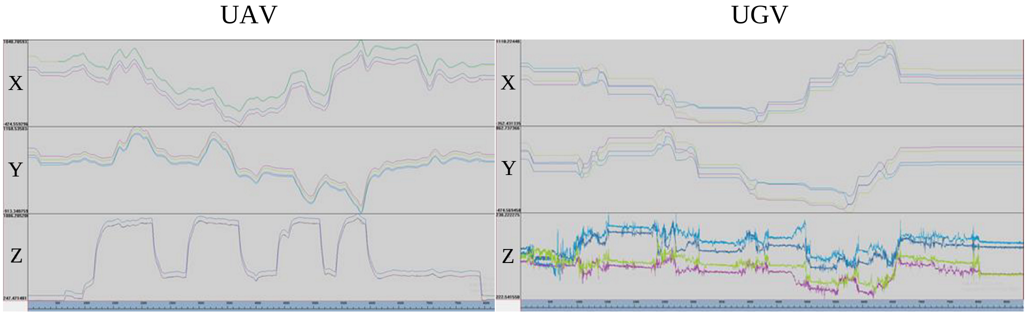


Figure 28. The coordinate change curve of UAV and UGV.

Table 12 presents a comparison between the simulated and actual motion path lengths of both the UAV and UGV. The results indicate that while there is some discrepancy between the actual path lengths of the UAV and UGV compared to the simulated experiment results, the error is generally around 10%, which remains within an acceptable range. In conclusion, the experimental error is relatively minor, largely aligning with the specifications of the HRDS delivery task path planning model described in this study. This outcome validates the accuracy of the model constructed in this study.

Table 12. UAV and UGV simulation and physical experimentation errors.

Experimental Data	Simulation Experiment (Equal Scaling)	Physical Experiment	Error
The first flight distance of UAV (m)	2.1432	1.9394	9.5096%
The second flight distance of UAV (m)	2.0112	2.2067	3.0532%
The third flight distance of UAV (m)	1.7267	1.9083	10.522%
The fourth flight distance of UAV (m)	1.7897	1.9701	10.082%
The distance of UGV (m)	2.8031	3.0532	8.9213%

6. Conclusions

This study investigates a multi-objective heterogeneous robotic delivery system that considers constraints including UAV energy, payload capacity, ground road network, customer waiting time, and rendezvous time. The objectives of this study are to maximize customer satisfaction while minimizing delivery costs. A mathematical model for the route scheduling of MOHRDS is established, and the ABC is improved based on fast non-dominated sorting strategies to obtain the optimal set of delivery routes. Simulation experiments demonstrate that this method exhibits strong optimization capabilities and high computational efficiency and can obtain the optimal Pareto front for MOHRDS. Physical experiments further validate the feasibility and effectiveness of the proposed method.

The optimal Pareto front obtained in this study can provide various delivery plans for decision-makers and enhance the market competitiveness of express companies. However, the delivery system in this study is comprised of a single UAV and a single UGV. In order to expand the coverage of the delivery system and enhance its flexibility, future efforts will focus on exploring the coordinated delivery of multiple UAVs and multiple UGVs. On the other hand, the weight of the package can significantly affect the energy consumption of the UAV, which in turn, has a major impact on the planned route [44]. Consequently, future experiments will consider this as a crucial factor in achieving more realistic MOHRDS path planning.

Author Contributions: Conceptualization, Z.C.; Methodology, Z.C. and Z.W.; Software, S.H.; Investigation, Z.C.; Resources, Z.C., Y.C. and R.M.A.I.; Data curation, S.H.; Writing—original draft preparation, Z.W.; Writing—review and editing, S.H., Y.C., M.H. and R.M.A.I.; Visualization, S.H.; Funding acquisition, Z.C., Y.C., M.H. and R.M.A.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under Grant Nos. 62173262, 62073250, 62203339, and 62003249.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Rodrigues, G.; Caldas, R.; Araujo, G.; de Moraes, V.; Rodrigues, G.; Pelliccione, P. An architecture for mission coordination of heterogeneous robots. *J. Syst. Softw.* **2022**, *191*, 111363. [CrossRef]
2. Chen, Y.; Chen, M.; Chen, Z.; Cheng, L.; Yang, Y.; Li, H. Delivery path planning of heterogeneous robot system under road network constraints. *Comput. Electr. Eng.* **2021**, *92*, 107197. [CrossRef]
3. Kumar, S.P.; Jat, D.; Sahni, R.K.; Jyoti, B.; Kumar, M.; Subeesh, A.; Parmar, B.S.; Mehta, C. Measurement of droplets characteristics of UAV based spraying system using imaging techniques and prediction by GWO-ANN model. *Measurement* **2024**, *234*, 114759. [CrossRef]
4. Farrag, T.A.; Askr, H.; Elhosseini, M.A.; Hassanien, A.E.; Farag, M.A. Intelligent Parcel Delivery Scheduling Using Truck-Drones to Cut down Time and Cost. *Drones* **2024**, *8*, 477. [CrossRef]
5. Mourelo Ferrandez, S.; Harbison, T.; Webwer, T.; Sturges, R.; Rich, R. Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *J. Ind. Eng. Manag.* **2016**, *9*, 374–388. [CrossRef]
6. He, Y.; Wang, D.; Huang, F.; Zhang, R.; Min, L. Aerial-Ground Integrated Vehicular Networks: A UAV-Vehicle Collaboration Perspective. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 5154–5169. [CrossRef]
7. Niu, G.; Wu, L.; Gao, Y.; Pun, M.O. Unmanned aerial vehicle (UAV)-assisted path planning for unmanned ground vehicles (UGVs) via disciplined convex-concave programming. *IEEE Trans. Veh. Technol.* **2022**, *71*, 6996–7007. [CrossRef]
8. Bacheti, V.P.; Brandão, A.S.; Sarcinelli-Filho, M. A path-following controller for a uav-ugv formation performing the final step of last-mile-delivery. *IEEE Access* **2021**, *9*, 142218–142231. [CrossRef]
9. Murray, C.C.; Chu, A.G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transp. Res. Part Emerg. Technol.* **2015**, *54*, 86–109. [CrossRef]
10. Agatz, N.; Bouman, P.; Schmidt, M. Optimization approaches for the traveling salesman problem with drone. *Transp. Sci.* **2018**, *52*, 965–981. [CrossRef]
11. Kitjacharoenchai, P.; Min, B.C.; Lee, S. Two echelon vehicle routing problem with drones in last mile delivery. *Int. J. Prod. Econ.* **2020**, *225*, 107598. [CrossRef]
12. Wang, Q.; Chen, H.; Qiao, L.; Tian, J.; Su, Y. Path planning for UAV/UGV collaborative systems in intelligent manufacturing. *IET Intell. Transp. Syst.* **2020**, *14*, 1475–1483. [CrossRef]
13. Rajesh, S.; Abd Algani, Y.M.; Al Ansari, M.S.; Balachander, B.; Raj, R.; Muda, I.; Bala, B.K.; Balaji, S. Detection of features from the internet of things customer attitudes in the hotel industry using a deep neural network model. *Meas. Sens.* **2022**, *22*, 100384. [CrossRef]
14. Omar Ali, S.R.; Abd Hakim Amir, S.N. Service quality and customer satisfaction: Experience of customers in postal service. *J. Intelek* **2020**, *15*, 67–75. [CrossRef]
15. Xu, Z.; Elomri, A.; Pokharel, S.; Mutlu, F. A model for capacitated green vehicle routing problem with the time-varying vehicle speed and soft time windows. *Comput. Ind. Eng.* **2019**, *137*, 106011. [CrossRef]

16. Zhang, J.; Wang, W.; Zhao, Y.; Cattani, C. Multiobjective quantum evolutionary algorithm for the vehicle routing problem with customer satisfaction. *Math. Probl. Eng.* **2012**, *2012*. [CrossRef]
17. Yan, X.; Xiao, B.; Zhao, Z. Multi-objective vehicle routing problem with simultaneous pick-up and delivery considering customer satisfaction. In Proceedings of the 2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE), Hangzhou, China, 20–21 April 2019; pp. 93–97. [CrossRef]
18. Zhang, Y.; Yuan, C.; Wu, J. Vehicle routing optimization of instant distribution routing based on customer satisfaction. *Information* **2020**, *11*, 36. [CrossRef]
19. Ghannadpour, S.F.; Noori, S.; Tavakkoli-Moghaddam, R. Multiobjective dynamic vehicle routing problem with fuzzy travel times and customers' satisfaction in supply chain management. *IEEE Trans. Eng. Manag.* **2013**, *60*, 777–790. [CrossRef]
20. Zhang, H.; Zhang, Q.; Ma, L.; Zhang, Z.; Liu, Y. A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows. *Inf. Sci.* **2019**, *490*, 166–190. [CrossRef]
21. Hong, C.; Choi, E.K.C.; Joung, H.W.D. Determinants of customer purchase intention toward online food delivery services: The moderating role of usage frequency. *J. Hosp. Tour. Manag.* **2023**, *54*, 76–87. [CrossRef]
22. Lebedev, D.; Goulart, P.; Margellos, K. A dynamic programming framework for optimal delivery time slot pricing. *Eur. J. Oper. Res.* **2021**, *292*, 456–468. [CrossRef]
23. Afshar-Bakeshloo, M.; Mehrabi, A.; Safari, H.; Maleki, M.; Jolai, F. A green vehicle routing problem with customer satisfaction criteria. *J. Ind. Eng. Int.* **2016**, *12*, 529–544. [CrossRef]
24. Karaboga, D.; Basturk, B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In Proceedings of the International Fuzzy Systems Association World Congress, Cancun, Mexico, 18–21 June 2007; pp. 789–798. [CrossRef]
25. Choong, S.S.; Wong, L.P.; Lim, C.P. An artificial bee colony algorithm with a modified choice function for the traveling salesman problem. *Swarm Evol. Comput.* **2019**, *44*, 622–635. [CrossRef]
26. Öztürk, Ş.; Ahmad, R.; Akhtar, N. Variants of Artificial Bee Colony algorithm and its applications in medical image processing. *Appl. Soft Comput.* **2020**, *97*, 106799. [CrossRef]
27. Xu, F.; Li, H.; Pun, C.M.; Hu, H.; Li, Y.; Song, Y.; Gao, H. A new global best guided artificial bee colony algorithm with application in robot path planning. *Appl. Soft Comput.* **2020**, *88*, 106037. [CrossRef]
28. Jiang, Y.; Qian, H.; Chu, Y.; Liu, J.; Jiang, Z.; Dong, F.; Jia, L. Convergence analysis of ABC algorithm based on difference model. *Appl. Soft Comput.* **2023**, *146*, 110627. [CrossRef]
29. Xiao, S.; Wang, H.; Wang, W.; Huang, Z.; Zhou, X.; Xu, M. Artificial bee colony algorithm based on adaptive neighborhood search and Gaussian perturbation. *Appl. Soft Comput.* **2021**, *100*, 106955. [CrossRef]
30. Wang, H.; Wang, W.; Xiao, S.; Cui, Z.; Xu, M.; Zhou, X. Improving artificial bee colony algorithm using a new neighborhood selection mechanism. *Inf. Sci.* **2020**, *527*, 227–240. [CrossRef]
31. Ye, T.; Wang, W.; Wang, H.; Cui, Z.; Wang, Y.; Zhao, J.; Hu, M. Artificial bee colony algorithm with efficient search strategy based on random neighborhood structure. *Knowl.-Based Syst.* **2022**, *241*, 108306. [CrossRef]
32. Xiao, W.S.; Li, G.X.; Liu, C.; Tan, L.P. A novel chaotic and neighborhood search-based artificial bee colony algorithm for solving optimization problems. *Sci. Rep.* **2023**, *13*, 20496. [CrossRef]
33. Zhou, X.; Tan, G.; Wang, H.; Ma, Y.; Wu, S. Artificial bee colony algorithm based on multi-neighbor guidance. *Expert Syst. Appl.* **2024**, *259*, 125283. [CrossRef]
34. Hakli, H.; Kiran, M.S. An improved artificial bee colony algorithm for balancing local and global search behaviors in continuous optimization. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 2051–2076. [CrossRef]
35. Dumez, D.; Lehuédé, F.; Péton, O. A large neighborhood search approach to the vehicle routing problem with delivery options. *Transp. Res. Part Methodol.* **2021**, *144*, 103–132. [CrossRef]
36. Masmoudi, M.A.; Mancini, S.; Baldacci, R.; Kuo, Y.H. Vehicle routing problems with drones equipped with multi-package payload compartments. *Transp. Res. Part Logist. Transp. Rev.* **2022**, *164*, 102757. [CrossRef]
37. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
38. Zitzler, E.; Laumanns, M.; Thiele, L. SPEA2: Improving the strength Pareto evolutionary algorithm. *Tik Rep.* **2001**, *103*. [CrossRef]
39. Srivastava, G.; Singh, A.; Mallipeddi, R. NSGA-II with objective-specific variation operators for multiobjective vehicle routing problem with time windows. *Expert Syst. Appl.* **2021**, *176*, 114779. [CrossRef]
40. Shuai, Y.; Yunfeng, S.; Kai, Z. An effective method for solving multiple travelling salesman problem based on NSGA-II. *Syst. Sci. Control. Eng.* **2019**, *7*, 108–116. [CrossRef]
41. Chen, Z.; Wu, H.; Chen, Y.; Cheng, L.; Zhang, B. Patrol robot path planning in nuclear power plant using an interval multi-objective particle swarm optimization algorithm. *Appl. Soft Comput.* **2022**, *116*, 108192. [CrossRef]
42. Elgharably, N.; Easa, S.; Nassef, A.; El Damatty, A. Stochastic multi-objective vehicle routing model in green environment with customer satisfaction. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1337–1355. [CrossRef]

43. Sheng, B.; Chen, L.; Cheng, J.; Zhang, Y.; Hua, Z.; Tao, J. A markless 3D human motion data acquisition method based on the binocular stereo vision and lightweight open pose algorithm. *Measurement* **2024**, *225*, 113908. [CrossRef]
44. Saponi, M.; Borboni, A.; Adamini, R.; Faglia, R.; Amici, C. Embedded payload solutions in UAVs for medium and small package delivery. *Machines* **2022**, *10*, 737. [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.