drones

Article

Cooperative Truck–Drone Delivery Path Optimization under Urban Traffic Restriction

Ying-Ying Weng, Rong-Yu Wu and Yu-Jun Zheng *

School of Information Science and Technology, Hangzhou Normal University, Hangzhou 311121, China
* Correspondence: yujun.zheng@computer.org

Abstract: In the traditional express delivery sector, trucks are the most available and efficient transportation mode in urban areas. However, due to the pressures of traffic congestion and air pollution problems, many cities have implemented strict measures to restrict trucks' access to many zones during specified time periods, which has caused significant effects on the business of the industry. Due to their advantages, which include high speed, flexibility, and environmental friendliness, drones have great potential for being combined with trucks for efficient delivery in restricted traffic zones. In this paper, we propose a cooperative truck and drone delivery path optimization problem, in which a truck carrying cargo travels along the outer boundary of the restricted traffic zone to send and receive a drone, and the drone is responsible for delivering the cargo to customers. The objective of the problem is to minimize the completion time of all delivery tasks. To efficiently solve this problem, we propose a hybrid metaheuristic optimization algorithm to cooperatively optimize the outer path of the truck and the inner path of the drone. We conduct experiments on a set of test instances; the results demonstrate that the proposed algorithm exhibits a competitive performance compared to other selected popular optimization algorithms.

Keywords: traffic restriction; truck–drone cooperation; path optimization; metaheuristic; water wave optimization

Citation: Weng, Y.-Y.; Wu, R.-Y.; Zheng, Y.-J. Cooperative Truck–Drone Delivery Path Optimization under Urban Traffic Restriction. *Drones* **2023**, *7*, 59. https://doi.org/10.3390/drones7010059

Academic Editor: Carlos Tavares Calafate

Received: 28 November 2022
Revised: 10 January 2023
Accepted: 11 January 2023
Published: 14 January 2023

1. Introduction

With the rapid development of the social economy and urbanization, we are now facing increasing traffic congestion and air pollution problems, which have serious effects on the sustainable development of big cities. To tackle these problems, more and more cities have implemented traffic restriction measures, such as "tail number restriction methods" that limit car owners to driving only on alternate days [1] and truck restriction policies that restrict freight trucks from entering into specified zones during specified time periods [2]. These have effectively alleviated the pressures of congestion and pollution; however, inevitably, these measures have significantly affected the business of the logistics industry. In particular, the express delivery sector, which typically relies on the use of trucks to distribute a large number of parcels to customers (the majority of whom reside in urban traffic-restricted zones) has been affected.

In recent years, drones, or unmanned aerial vehicles (UAVs), have shown great potential in last-mile delivery, as they can reduce operational costs, air pollution, and congestion [3]. During the COVID-19 pandemic, drones appeared as an interesting solution to contactless delivery, especially for urgently needed medical supplies [4–7]. For instance, the SF express used cargo drones to transport tons of medical materials to Chinese cities, such as Wuhan and Ordos, when they were shut down in the spring of 2020 due to the outbreak of the pandemic [8]. Many food service companies also employed drones to deliver food to customers without human-to-human contact [9]. Because their advantages include high speed, flexibility, and environmental friendliness, drones can be considered an effective method for circumventing ground-based traffic restrictions in express delivery.

The main drawbacks of drones in delivery are the limited payload capacities and flying ranges, which prevent drones from independently performing larger-scale delivery tasks. In contrast, trucks have much larger payload capacities and mileages. Therefore, trucks and drones can cooperate to achieve complementary advantages by using trucks as moving depots from which drones can repeatedly take parcels and swap/recharge batteries to surmount their limitations [10–12].

In this paper, we study the problem of using a truck and a drone to cooperatively deliver parcels to a number of customers in a restricted traffic zone. Initially, the truck carries all of the parcels, the drone, and the replaceable batteries for the drone to a location on the outer boundary of the restricted zone. The truck cannot enter the restricted zone. The drone is responsible for delivering the parcels to the customers. Due to the payload and range limitations, the drone has to repeatedly return back to the truck to take parcels and, if needed, swap batteries; meanwhile, the truck can move along the boundary of the restricted zone to improve the delivery efficiency of the drone, as illustrated in Figure 1. The problem is to determine the outer path of the truck and the inner path of the drone, such that the delivery tasks can be completed as early as possible. To efficiently solve the problem, we propose a hybrid optimization algorithm, which adapts a main population-based metaheuristic to optimize the inner path (delivery sequence) of the drone and employs a sub-procedure to determine the truck–drone intersections based on convex relaxation. We conduct experiments on a set of test instances, the results of which demonstrate that the proposed algorithm exhibits a competitive performance compared to other selected popular optimization algorithms. The main contribution of this paper can be summarized as follows:

- We present the problem of cooperative truck–drone path optimization for delivering parcels to customers in restricted traffic zones.
- We propose a hybrid metaheuristic and convex relaxation optimization algorithm to efficiently solve the problem.
- We validate the effectiveness and efficiency of the proposed method on a variety of test instances.

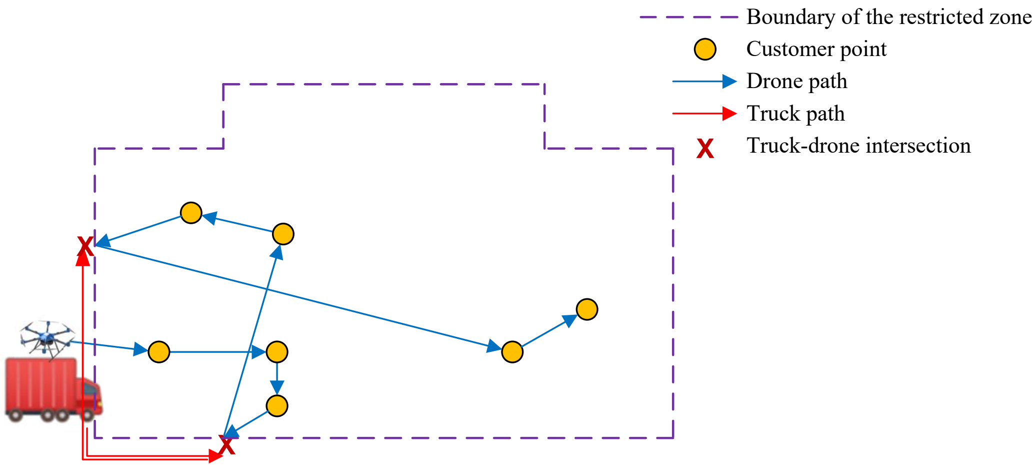


Figure 1. Illustration of the cooperative truck–drone delivery.

In the remainder of this paper, we will review the related works in Section 2, formulate the problem of cooperative truck–drone delivery path optimization in Section 3, propose the hybrid optimization algorithm in Section 4, and present the experimental results in Section 5. Finally, we conclude with discussions in Section 6.

2. Related Work

The application of drones in last-mile logistics has been practiced by many delivery companies. Amazon introduced its Prime Air drone delivery program in 2013 and conducted the first commercial drone delivery three years later [13]. Additionally, in 2016, DHL began to use a drone system to deliver consumer goods in the Bavarian community of Reit im Winkl [14], while UPS cooperated with Zipline (a UAV manufacturer) to deliver blood for lifesaving transfusions in Rwanda [15]. In China, SF obtained the first operating license for commercial drone deliveries in 2018 and has developed a number of UAV models to tackle issues related to the end delivery of special logistics operations, such as 'local specialty economy', medical cold-chain logistics, and emergency aid distribution [16]. In recent years, many other companies, including Google, FedEx, Russian Post, DPD, AMA, Alibaba, etc., have also reported the use of drones during delivery [17,18].

Due to the limitations of load capacity and the travel range of UAVs, a typical solution is to use drones in combination with ground vehicles (trucks), which has attracted recent attention from both the industry and academia. Based on the roles of drones and trucks in delivery, the research works can be divided into three classes. In the first class, drones and trucks have roughly equally important roles. Murray and Chu [19] presented mixed integer linear programming formulations for two problems, named the flying sidekick traveling salesman problem (FSTSP) and the parallel drone scheduling TSP (PDSTSP); they proposed two simple heuristic algorithms to solve problems of small and medium sizes. Murray and Raj [20] extended the FSTSP to deal with multiple drones, and they proposed a heuristic method that consists of solving a sequence of three subproblems. Ham [21] extended the PDSTSP to incorporate multiple trucks, drones, and hubs, where drones can also perform pick-up tasks; they modeled the problem as an unrelated parallel machine scheduling and employed constraint programming to solve it. Agatz et al. [22] proposed another formulation called the TSP with drone (TSP-D); they developed several route-first-cluster-second heuristics based on local search and dynamic programming. Ulmer and Thomas [23] studied a same-day delivery problem in which either a drone or a truck could be used to meet the customer demand, and dynamic programming is employed to decide the acceptance or rejection of randomly arriving customer orders. Wu et al. [24] considered a collaborative truck–drone routing problem for contactless parcel delivery in epidemic areas, in which each customer needs to be served exactly once by either a truck or a drone; they proposed an improved variable neighborhood descent combined with simulated annealing, tabu search, and K-means clustering to efficiently solve the problem. Zheng et al. [25] studied the problem of cooperatively using drones and police cars to search for and catch escaped criminals; they proposed an evolutionary algorithm for routing the drones and cars to minimize the expected time of capture. The method has been applied to a similar problem of the cooperation of drones and ground working units [26,27].

In the second class, drones play the primary role while trucks act as supporting units; e.g., trucks take parcels to a hub from which drones deliver parcels to customers. Carlsson and Song [28] studied a delivery system in which a drone provides delivery service to customers while making return trips to a truck that is itself moving. Wang and Sheu [29] proposed the VRP with drones (VRPD) with service hubs to which trucks supply items and from which drones deliver items, and the drones can land only on the hubs; they formulated the problem as mixed integer programming and developed a branch-and-price algorithm for the problem. Luo et al. [30] formulated a two-echelon cooperated routing problem for a truck and a drone that launches and lands on the truck; they proposed two heuristics: the first constructs a complete tour and then splits it by truck routes, while the second constructs the truck tour and assigns UAV flights to it. Karak and Abdelghany [31] presented a hybrid vehicle–drone routing problem for pick-up and delivery services, in which vehicles are used only as mobile depots for the drones; they proposed an extended Clarke and Wright algorithm to the problem. Salama and Srinivas [12] studied another problem of truck–drone cooperation that allows the truck to stop at non-customer locations for drone launch and recovery; they proposed a hybrid simulated annealing and variable neighborhood search algorithm for the problem. Bányai [11] presented four models, including first-mile/last-mile delivery by e-trucks, first-mile delivery by drones from e-trucks, last-mile delivery by e-trucks, and integrated first-mile/last-mile delivery by drones from e-trucks; they conducted a numerical analysis to show that the last model could lead to a significant reduction in energy consumption and virtual greenhouse gas emissions.

In the third class, trucks play the primary role; i.e., their routes have priorities that the drones have to follow. Savuran and Karakaya [32] defined a mobile depot VRP which routes a drone deployed on a mobile carrier to visit fixed targets; they proposed a genetic algorithm that has been adapted to satisfy the constraints of depot mobility and range while maximizing the number of targets visited by the UAV. The problem studied by Boysen et al. [33] fixes a sequence of truck stops and schedules the trips of a drone to determine its take-off and landing stops. These types of problems are categorized by [34] as the mothership and drone routing problem, which is solved by a set of heuristics embedded with a second-order cone program. In other cases, drones support the trucks in ways such as gathering information for the trucks or providing communication among the trucks [35]. Interested readers can refer to [36] for a survey on drone–truck combined operations. Compared with the existing studies in the literature, our study considers the problem of cooperatively planning the truck and drone paths to minimize the service time under the restricted traffic constraint, in which the truck can only move along the boundary of a restricted zone.

3. Problem Description

The problem under consideration is to cooperatively schedule a truck and a drone to provide delivery services to a set of customers in a restricted traffic zone. The input parameters of the problem are shown in Table 1. At the beginning, the truck carries all cargo and the drone to the initial location (coordinate) c_0. There are n customers that need to be served, the location (coordinate) of the i-th customer is c_i, and the weight of the cargo to be delivered to the i-th customer is w_i ($1 \leq i \leq n$). The maximum load of the drone is W (without loss of generality, it is assumed that $w_i \leq W$ for any i). Let B denote the outer boundary of the restricted traffic zone; the velocity of the truck along the boundary is v_r. The maximum distance of the drone (after being fully charged) is D. For the drone, its velocity changes with its load. The minimum velocity (when it is fully loaded) and the maximum velocity (when it is not loaded) are denoted as $v_u^{\min}$ and $v_u^{\max}$, respectively. When the load of the drone is w, its velocity is calculated as:

$$v_u(w) = v_u^{\min} + \frac{W - w}{W}(v_u^{\max} - v_u^{\min}) \tag{1}$$

This problem is to determine the delivery sequence of customers (i.e., the path of the drone), denoted by $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, and based on this, the path of the truck consisting of a set of locations (also called intersections) at which the truck receives the returned drone for battery replacement and cargo reloading and then re-sends the drone for subsequent delivery. Assume that the truck carries sufficient batteries for the drone to complete the delivery task, i.e., battery recharging is not required during the task.

Table 1. Input parameters of the considered problem.

Parameter	Description
n	Number of the customers
c_0	Initial location the truck and the drone
c_i	Location of the i-th customers
w_i	Weight of cargo to be delivered to the i-th customer
B	Outer boundary of the restricted traffic zone
v_r	Velocity of the truck
D	Maximum distance of the drone
W	Maximum load of the drone
$v_u^{\max}$	Maximum velocity of the drone
$v_u^{\min}$	Minimum velocity of the drone

The first customer c_{I_1} at which the drone needs to return back to the truck to replace the battery and reload cargo satisfies either one of the following conditions:

- According to the weight capacity limitation, the drone cannot load the cargo for the next customer:
$$\sum_{i=1}^{I_1} w_i \le W < \sum_{i=1}^{I_1+1} w_i \tag{2}$$
- According to the power and consequent distance limitation, the drone cannot fly to the next customer and then return back to the truck:
$$\left(\sum_{i=1}^{I_1} d(c_{x_{i-1}}, c_{x_i})\right) + d(c_{x_{I_1}}, s_{I_1}) \le D < \left(\sum_{i=1}^{I_1+1} d(c_{x_{i-1}}, c_{x_i})\right) + d(c_{x_{I_1+1}}, s_{I_1+1}) \tag{3}$$

where s_i denotes the optimal intersection of the truck and the drone on the outer boundary B to minimize the flight distance of the drone from the current customer x_i to the truck and then to the next customer x_{i+1}. The first intersection s_{I_1} is determined as follows so as to minimize the back-and-forth travel time of the drone:

$$\begin{aligned} s_{I_1} \quad = \quad & \min\{s : s \in B \wedge \left(d(c_0, s)/v_r \le t(x_{I_1}) + d(c_{x_{I_1}}, s)/v_u^{\max}\right) : \\ & d(c_{x_{I_1}}, s)/v_u^{\max} + d(s, c_{x_{I_1+1}})/v_{x_{I_1+1}}\} \end{aligned} \tag{4}$$

where $v_{x_{I_1+1}}$ is the velocity of the drone starting from the truck to the next round's first customer x_{I_1+1}, which is calculated based on the sum of the weights of customers $\{x_{I_1+1}, x_{I_1+2}, \ldots, x_{I_2}\}$ according to Equation (1).

Regardless, the intersection should be within the travel ranges of the truck and the drone. Specifically, if the boundary is convex, the intersection should not exceed the orthogonal projection of the current customer onto the boundary, as illustrated in Figure 2.

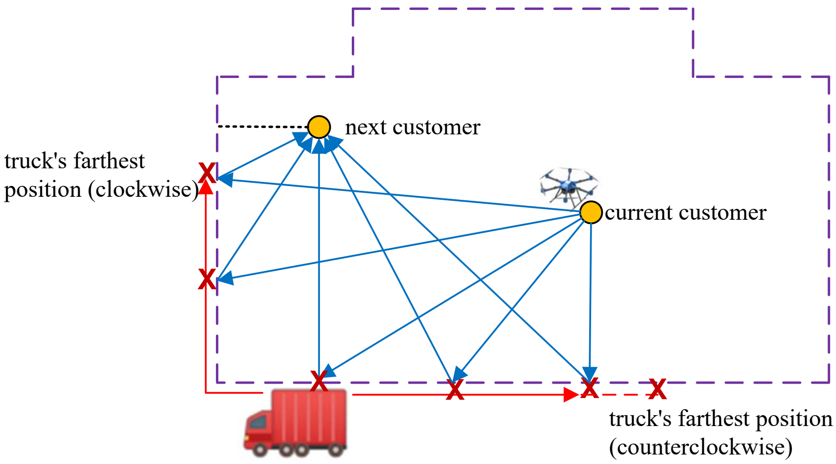


Figure 2. An illustration of the intersection of the truck and the drone. Six possible intersection points are drawn; however, the rightmost intersection is not considered, as it exceeds the orthogonal projection of the current customer onto the boundary.

Similarly, for each next k-th back-and-forth round, the customer c_{I_k} at which the drone needs to return back to the truck satisfies either one of the following conditions:

$$\sum_{i=I_{k-1}+1}^{I_k} w_i \le W < \sum_{i=I_{k-1}+1}^{I_k+1} w_i \tag{5}$$

$$\left(\sum_{i=I_{k-1}+1}^{I_k} d(c_{x_{i-1}}, c_{x_i})\right) + d(c_{x_{I_k}}, s_{I_k}) \le D < \left(\sum_{i=I_{k-1}+1}^{I_k+1} d(c_{x_{i-1}}, c_{x_i})\right) + d(c_{x_{I_k+1}}, s_{I_k+1}) \tag{6}$$

where s_{I_k} denotes the k-th intersection ($1 < k \le K$) of the truck and the drone on the outer boundary B, which is determined as follows:

$$\begin{aligned} s_{I_k} &= \min\{s : s \in B \wedge \big(d(s_{I_{k-1}}, s)/v_r \le t(x_{I_k}) - t(s_{I_{k-1}}) + d(c_{x_{I_k}}, s)/v_u^{\max}\big) : \\ &\qquad d(c_{x_{I_k}}, s)/v_u^{\max} + d(s, c_{x_{I_k+1}})/v_{x_{I_k+1}}\} \end{aligned} \tag{7}$$

where $v_{x_{I_k+1}}$ is the velocity of the drone starting from the truck to the $(k+1)$-th round's first customer x_{I_k+1}.

Based on the above conditions, we can obtain all of the customer locations $x_{I_1}, x_{I_2}, \ldots, x_{I_K}$ at which the drone needs to return back. These locations divide the delivery sequence into K subsequences. Consequently, we can iteratively calculate the velocity $v(x_i)$ of the drone flying to each customer c_{x_i} as well as the time $t(x_i)$ at which the drone arrives at c_i. First, $v(c_1)$ and $t(c_1)$ can be directly calculated as

$$v(x_1) = v_u^{\min} + \frac{W - \sum_{i=1}^{I_1} w_{x_i}}{W}(v_u^{\max} - v_u^{\min}) \tag{8}$$

$$t(x_1) = d(c_0, c_{x_1})/v(x_1) \tag{9}$$

Next, for each x_i that belongs to the k-th subsequence ($1 \le k \le K$), but is not the first location of a subsequence, $v(x_i)$ and $t(x_i)$ can be calculated based on the previous customer as

$$v(x_i) = v_u^{\min} + \frac{W - \sum_{i'=i}^{I_k} w_{x_i}}{W}(v_u^{\max} - v_u^{\min}) \tag{10}$$

$$t(x_i) = t(c_{i-1}) + d(c_{x_{i-1}}, c_{x_i})/v(x_i) \tag{11}$$

For x_i, which is the first location of the k-th subsequence ($1 \le k \le K$), i.e., $i = I_{k-1} + 1$, $v(x_i)$ and $t(x_i)$ can be calculated as

$$v(x_i) = v_u^{\min} + \frac{W - \sum_{i=I_{k-1}+1}^{I_k} w_{x_i}}{W}(v_u^{\max} - v_u^{\min}) \tag{12}$$

$$t(x_i) = t(x_{I_{k-1}}) + d(c_{x_{I_{k-1}}}, s_{I_{k-1}})/v_u^{\max} + d(s_{I_{k-1}}, c_i)/v(x_i) \tag{13}$$

The objective of the problem is to minimize the time at which the drone serves the last customer c_{x_n}, i.e., the completion time of all delivery tasks, and the problem can be formulated as follows (where u_{ij} are auxiliary variables for ensuring that $\{x_1, x_2, \ldots, x_n\}$ is a permutation of $\{1, 2, \ldots, n\}$):

$$\min \quad f(\mathbf{x}) = t(x_n) \tag{14}$$

$$\text{s.t.} \quad \text{Equations (1)–(12)}$$

$$x_i - x_j + n u_{ij} \le n - 1, \quad \forall 1 \le i \le n; 1 \le j \le n \tag{15}$$

$$1 \le x_i \le n, \quad \forall 1 \le i \le n \tag{16}$$

$$u_{ij} \in \{0, 1\}, \quad \forall 1 \le i \le n; 1 \le j \le n \tag{17}$$

4. A Hybrid Optimization Method for the Problem

In this section, we propose a hybrid metaheuristic optimization method for the problem. As shown in the flowchart in Figure 3, the method uses a main procedure that evolves a population of candidate solutions (i.e., customer delivery sequences) to search for an optimal (or near-optimal) drone path and employs a sub-procedure to optimize the truck–drone intersections for each main solution **x** in the population.

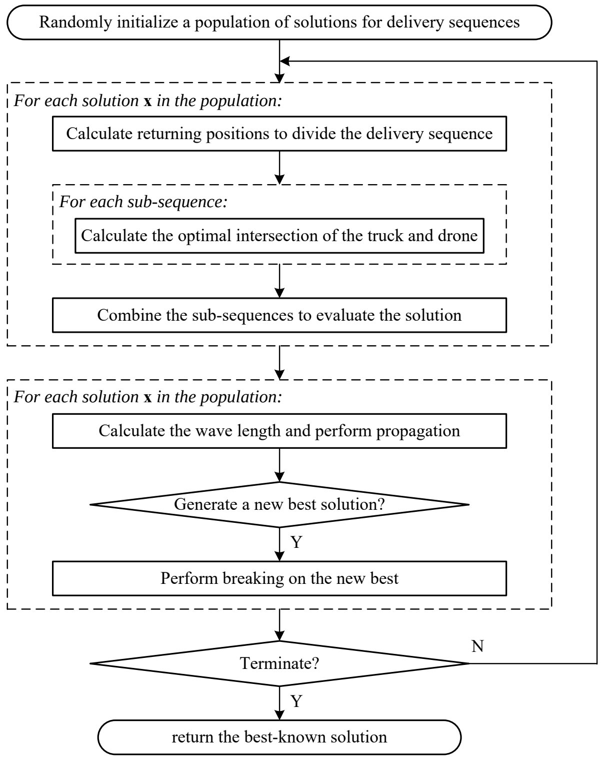


Figure 3. Flowchart of the hybrid metaheuristic optimization method.

4.1. Main Metaheuristic for Optimizing the Drone Path

The main problem for optimizing the drone path is a sequence optimization problem. The method randomly initializes a population of solutions, each of which is a random permutation $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ of the customer indices $\{1, 2, \ldots, n\}$. The method then evolves the solutions using the WWO metaheuristic, which has demonstrated performance advantages in sequence optimization compared to other popular metaheuristics, such as GA, PSO, and BBO.

The basic principle of WWO is to assign each solution $\mathbf{x}$ a wavelength $\lambda(\mathbf{x})$ that is inversely proportional to the solution fitness and make $\mathbf{x}$ search in a range proportional to $\lambda(\mathbf{x})$, such that high-fitness solutions search in small ranges, while low-fitness solutions search in large ranges to balance global and local searches. Initially, all wavelengths are set to 0.5; at each iteration, the wavelength of each solution $\mathbf{x}$ is updated according to its objective function value $f(\mathbf{x})$ as follows:

$$\lambda(\mathbf{x}') = \lambda(\mathbf{x})\alpha^{-(f(\mathbf{x})-f_{\min}+\epsilon)/(f_{\max}-f_{\min}+\epsilon)} \tag{18}$$

where α is a constant of 1.0026; $f_{\max}$ and $f_{\min}$ are the maximum and minimum objective function values in the population, respectively; and ϵ is a very small positive value to avoid division by zero.

At each iteration, each solution $\mathbf{x}$ performs a propagation operation, which, at each dimension i from 1 to n with a probability of $\lambda(\mathbf{x})$, chooses a random subsequence and reverses it. In this way, the expected number of subsequence reversals on $\mathbf{x}$ is $n\lambda(\mathbf{x})$; the better the solution, the smaller the $\lambda(\mathbf{x})$, and a small change in the solution is expected.

After propagation, if the updated solution is better than the original one, it replaces the original one in the solution. In particular, if the updated solution is better than any other solution searched before, i.e., it is a new best solution $\mathbf{x}^*$, a breaking operation is performed on $\mathbf{x}^*$ to produce a random number (between $[1, n]$) of neighboring solutions, each of which is obtained by randomly swapping two customer positions x_i and x_j in the solution. The best neighbor, if better than $\mathbf{x}^*$, will replace $\mathbf{x}^*$ in the population.

Compared to the basic WWO algorithm, here, we remove another operation called refraction, which replaces any solution that stays a predefined number of generations with a new solution moving towards the best-known solution; instead, we employ a population reduction policy from the simplified WWO (SimWWO), which reduces the population size N_P from an upper limit $N_P^{\max}$ to a lower limit $N_P^{\min}$ by iteratively removing the current worst solution:

$$N_P = N_P^{\max} - \frac{t}{t_{max}}(N_P^{\max} - N_P^{\min}) \tag{19}$$

where $N_P^{\max}$ and $N_P^{\min}$ are the maximum and minimum population sizes, respectively; t is the current generation number or the number of fitness evaluations (NFEs); and t_{max} is the maximum generation number or NFEs of the algorithm.

Algorithm 1 presents the pseudo-code of the SimWWO metaheuristic for optimizing the drone path.

Algorithm 1: Simplified WWO algorithm adapted to optimize the drone path for the problem.

```
Initialize a population of solutions of customer sequences;
while the stopping condition is not satisfied do
    for E do
        a
    end
    chsolution x in the population for i = 1 to n do
        Iteratively calculate the state of the drone at c_{x_i};
        if the drone should return back after visiting c_{x_i} then
            Call Algorithm 2 to calculate the intersection with the truck;
        end
    end
    Evaluate the objective function value of x;
    Let x* be the best-known solution;
    Calculate the wavelengths of the solutions according to Equation (18);
    foreach solution x in the population do
        Copy x to a new solution x';
        for i = 1 to n do
            if rand(0,1) < λ(x) then
                Select a random subsequence of x and reverse it;
            end
        end
        Evaluate the objective function value of x';
        if f(x') < f(x) then
            Set x to x';
            if f(x) < f(x*) then
                Set x* to x;
                Let K = rand(1,n);
                for i = 1 to n do
                    Generate a neighbor of x* by conducting a random swapping;
                    if the neighbor is better then
                        Set x* to the neighbor;
                    end
                end
            end
        end
    end
    Update the population N_P according to Equation (19) by reducing the worst solution;
end
return x*.
```

4.2. Sub-Procedure for Optimizing Truck–Drone Intersections

If the drone needs to return back after visiting a customer c_{x_i}, we search for an intersection of the truck and the drone that optimizes function (7) using the sub-procedure shown in Algorithm 2. The sub-procedure defines a recursive function to solve the non-convex problem by finding an optimal solution to the convex relaxation problem and, if the optimal solution is not a feasible solution to the original problem, branching the problem into subproblems (i.e., decompose the McCormick polyhedron into two polyhedra at the optimal solution point) and solving the subproblems recursively [37].

Algorithm 2: Algorithm for finding the intersection of the truck and the drone.

```
Calculate the two farthest positions that the truck can reach in the clockwise and
  counterclockwise directions, respectively;
Initialize the best-known solution B* as NULL and its objective value f* as +∞;
Denote the original problem as P;
Function OptimalSolution (P)
    Obtain the convex relaxation problem P' of P by McCormick envelopes [37];
    Removing all undesirable fractional solutions by cutting plane;
    if P' is unsolved, then return B*;
    Let S be the optimal solution to P';
    if S is a feasible solution to the original problem P, then
        if f* > f(S) then
            f* = f(S);
            B* = S;
        end
    else
        Branch P' into two subproblems, P1 and P2, i.e., decompose the McCormick
          polyhedron into two polyhedra at S;
        Let B1 = OptimalSolution (P1), B2 = OptimalSolution (P2);
        if f(B1) < f(B2) then
            B* = B1, f* = f(B1);
        else
            B* = B2, f* = f(B2);
        end
    end
    return B*;
end;
B* = OptimalSolution (P);
return the optimal solution B*.
```

5. Computational Experiments

We construct nine problem instances based on selected regions of the city of Hangzhou in Zhejiang Province, China. Table 2 gives the basic information about the test instances, including the number of customers, the areas and perimeters of the restricted traffic zones, the average distance between the customers, and the average weight of the demands of the customers. The areas of these regions are most suitable for one truck and one drone to efficiently perform delivery tasks (for larger areas, larger numbers of trucks and drones are expected). For these instances, we set the maximum velocity of the truck as $v_r = 30$ km/h (according to the average velocity on urban roads in common conditions), the velocities of the drone as $v_u^{\max} = 100$ km/h, $v_u^{\min} = 30$ km/h, the maximum distance of the drone as $D = 20$ km, and the maximum load of the drone as $W = 30$ kg (according to the technical performance of a typical type of drone for express delivery).

Table 2. Basic information about the nine problem instances.

Ins.	n	Area (km^2)	Perimeter (km)	Average Distance (km)	Average Weight (kg)
1	10	2.00	6	0.733	3.10
2	15	2.00	9	1.414	2.33
3	20	2.25	6	0.695	2.65
4	25	2.00	9	1.520	2.76
5	30	2.00	6	0.720	2.27
6	35	2.25	6	0.770	2.23
7	40	2.00	9	1.635	2.60
8	45	2.25	6	0.665	2.36
9	50	2.00	6	0.737	2.32

For each test instance, we compare the adapted SimWWO algorithm with the following six metaheuristic algorithms for permutation optimization:

- A GA that uses linear order crossover and shift-change mutation [38].
- A BBO algorithm [39] adapted to this problem using subsequence migration [40].
- A DE algorithm for permutation optimization based on floating-to-integer mapping [41].
- A discrete PSO adapted to this problem using subsequence learning [42].
- Enhanced BBO [43] that integrates local and global subsequence migration.
- Basic WWO [44] adapted to this problem using subsequence reverse propagation.

For each algorithm, we tune its control parameters to the nine problem instances, which results in the parameter setting, shown in Table 3, that exhibits the best average performance on the whole test set.

Table 3. Control parameters of the comparative algorithms.

Algorithm	Parameter Setting
GA	$N_P = 30$, crossover rate: 0.95, mutation rate: 0.2
BBO	$N_P = 30$, mutation rate: 0.1
DE	$N_P = 30$, crossover rate: 0.9, scale factor: 0.5
PSO	$N_P = 30$, maximum inertia weight: 0.9, minimum inertia weight: 0.4
EBO	$N_P = 30$, maximum maturity: 0.6, minimum maturity: 0.3
WWO	$N_P = 30$, maximum wave height: 12, maximum number of breaking waves: 12
SimWWO	$N_P^{\max} = 30$, $N_P^{\min} = 6$, maximum number of breaking waves: 12

To ensure a fair comparison, we set the termination condition as the maximum number of objective function evaluations (NFEs) that reaches $2000n$ for all algorithms. We run each algorithm 30 times for each instance. Table 4 presents the maximum, minimum, median, and standard deviation (std) of the objective values over the 30 runs of each algorithm for each instance, where the best median value on each instance is shown in bold; the last row gives the averaged median values of the nine instances. The best median value among the seven algorithms for each instance is shown in bold. A nonparametric Wilcoxon rank sum test is conducted to compare the results of the algorithms for each instance, and a superscript † before a median value indicates that there is a significant difference between the result of SimWWO and that of the corresponding comparative algorithm (at a confidence level of 95%). Moreover, the box plots in Figure 4a–i give the median, minimum, maximum, first quartile (25%) and third quartile (75%) of the objective values obtained by the algorithms for instances 1–9, respectively. For illustration, Figures 5–13 show the resulting routes obtained by the different algorithms on instances 1–9, respectively.

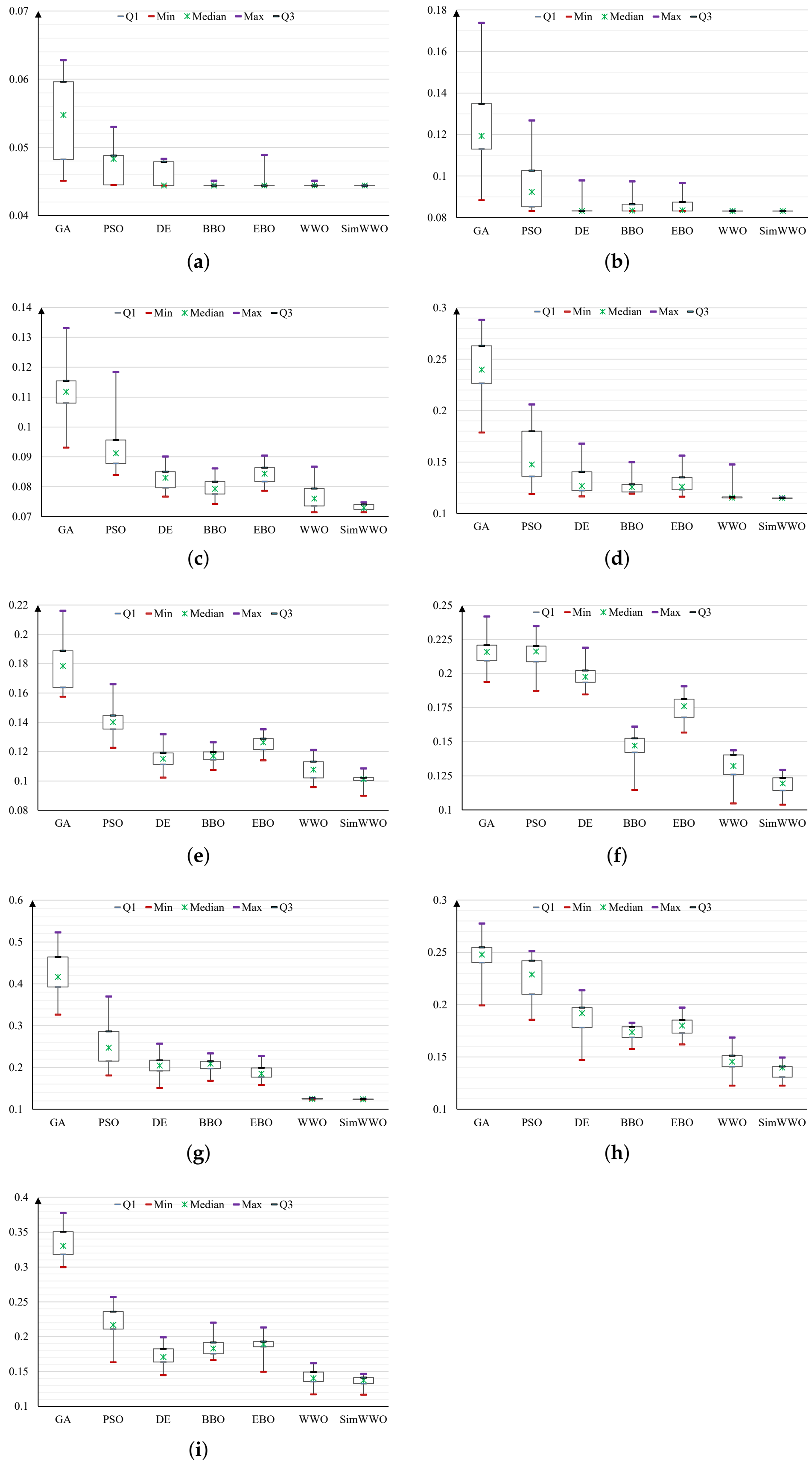
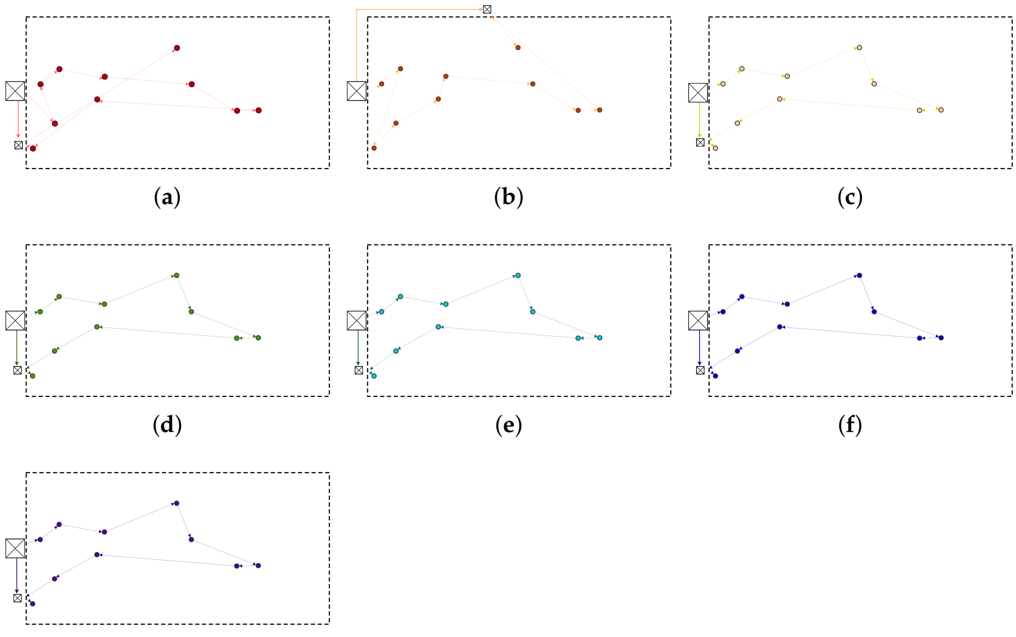


Figure 4. Box plots of the results of the comparative algorithms on the nine test instances. (**a**) Ins. 1. (**b**) Ins. 2. (**c**) Ins. 3. (**d**) Ins. 4. (**e**) Ins. 5. (**f**) Ins. 6. (**g**) Ins. 7. (**h**) Ins. 8. (**i**) Ins. 9.

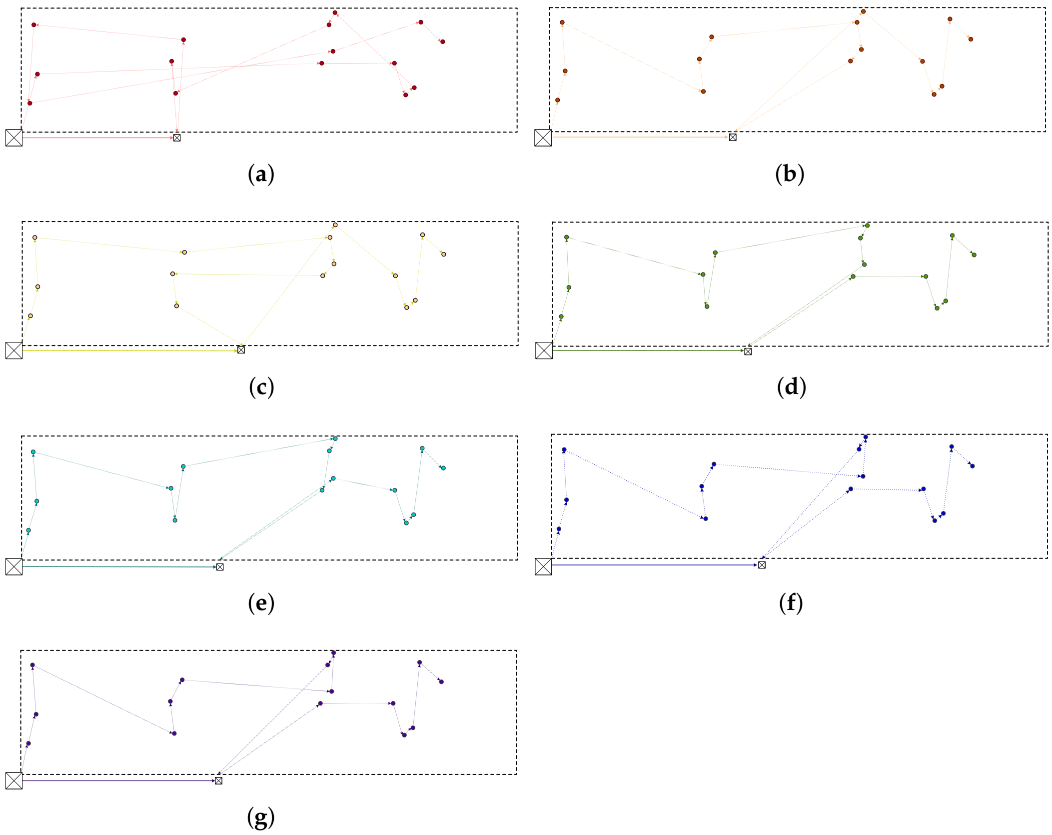


(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**)

Figure 5. Resulting drone routes (in dash lines) and truck routes (in solid lines) obtained by the nine algorithms on instance 1. (**a**) GA. (**b**) PSO. (**c**) DE. (**d**) BBO. (**e**) EBO. (**f**) WWO. (**g**) SimWWO.



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

(**c**) (**d**)

(**e**) (**f**)

(**g**)

Figure 6. Resulting drone routes (in dash lines) and truck routes (in solid lines) obtained by the nine algorithms on instance 2. (**a**) GA. (**b**) PSO. (**c**) DE. (**d**) BBO. (**e**) EBO. (**f**) WWO. (**g**) SimWWO.

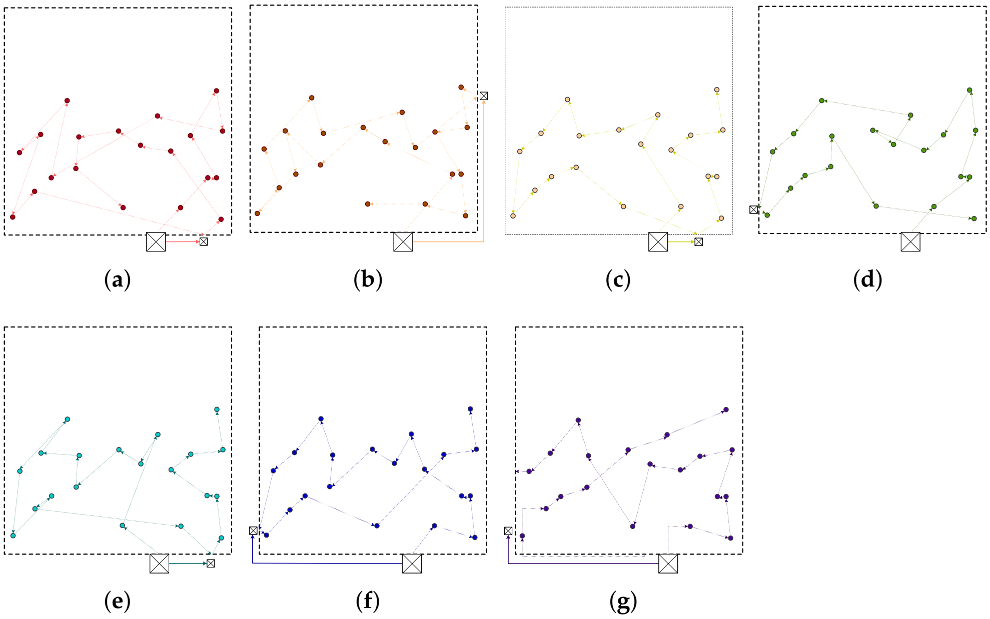


Figure 7. Resulting drone routes (in dash lines) and truck routes (in solid lines) obtained by the nine algorithms on instance 3. (**a**) GA. (**b**) PSO. (**c**) DE. (**d**) BBO. (**e**) EBO. (**f**) WWO. (**g**) SimWWO.

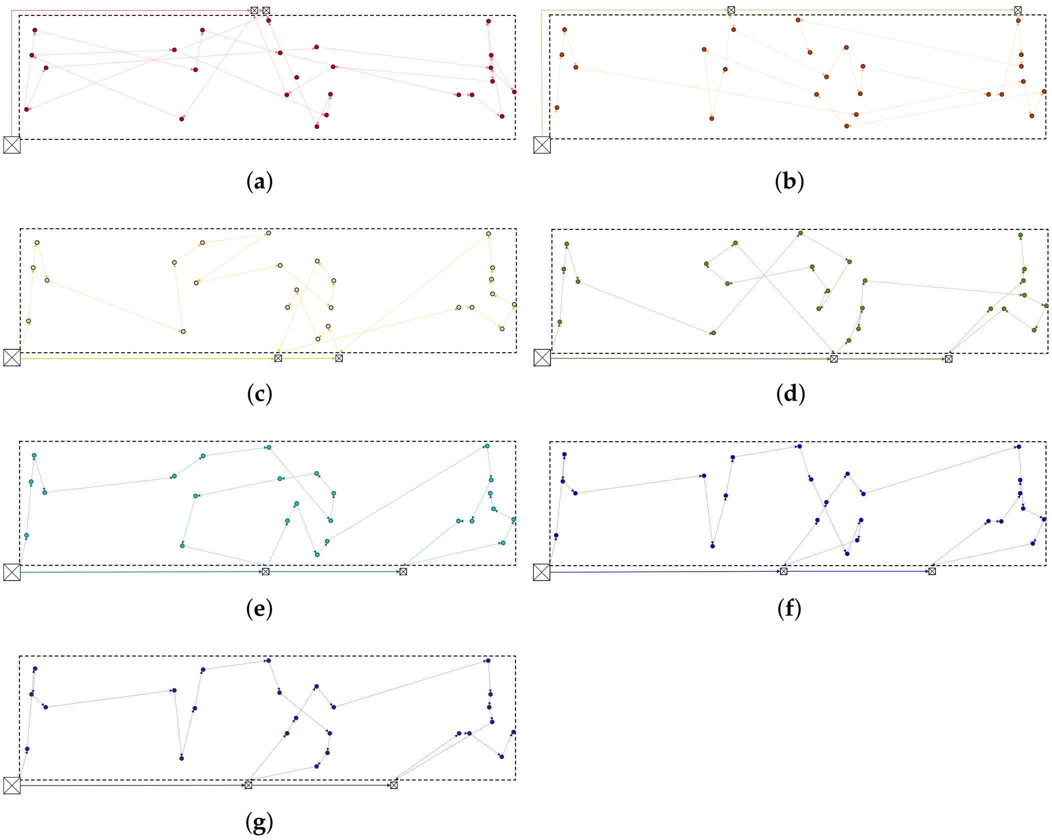


Figure 8. Resulting drone routes (in dash lines) and truck routes (in solid lines) obtained by the nine algorithms on instance 4. (**a**) GA. (**b**) PSO. (**c**) DE. (**d**) BBO. (**e**) EBO. (**f**) WWO. (**g**) SimWWO.

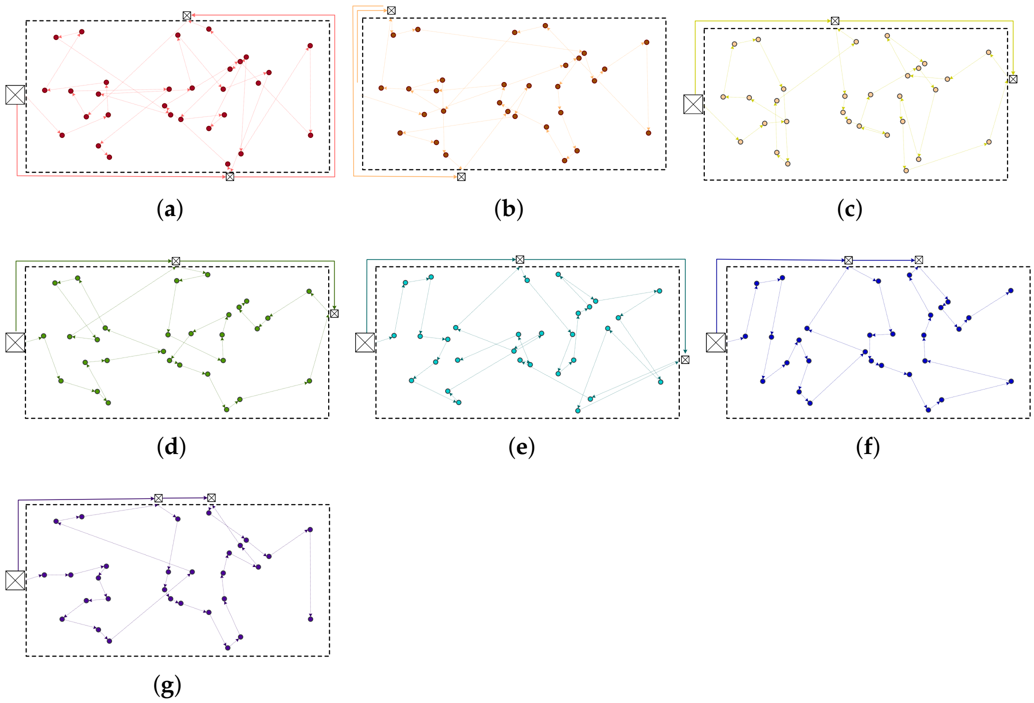


Figure 9. Resulting drone routes (in dash lines) and truck routes (in solid lines) obtained by the nine algorithms on instance 5. (**a**) GA. (**b**) PSO. (**c**) DE. (**d**) BBO. (**e**) EBO. (**f**) WWO. (**g**) SimWWO.

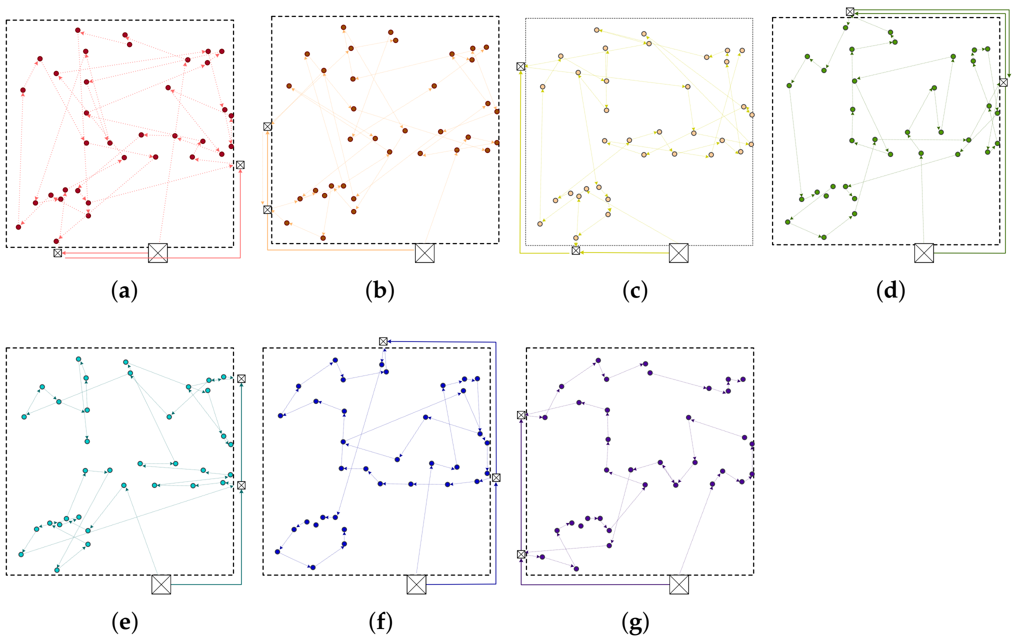


Figure 10. Resulting drone routes (in dash lines) and truck routes (in solid lines) obtained by the nine algorithms on instance 6. (**a**) GA. (**b**) PSO. (**c**) DE. (**d**) BBO. (**e**) EBO. (**f**) WWO. (**g**) SimWWO.

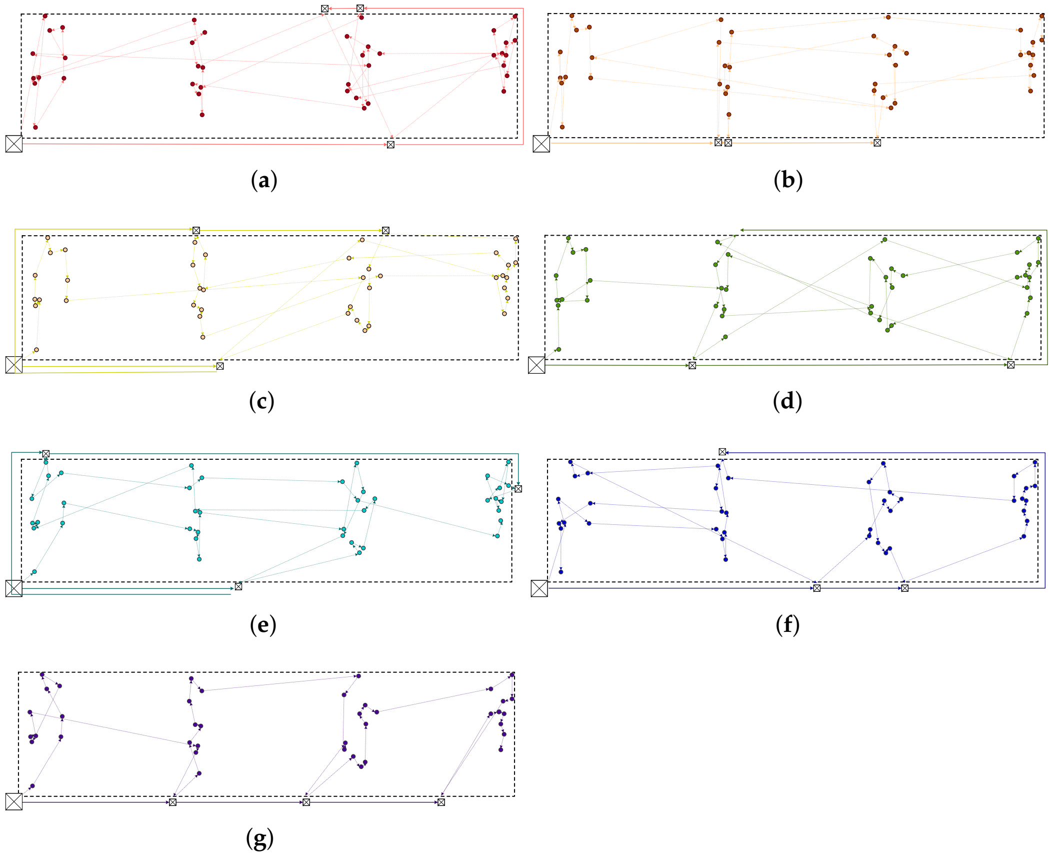


Figure 11. Resulting drone routes (in dash lines) and truck routes (in solid lines) obtained by the nine algorithms on instance 7. (**a**) GA. (**b**) PSO. (**c**) DE. (**d**) BBO. (**e**) EBO. (**f**) WWO. (**g**) SimWWO.

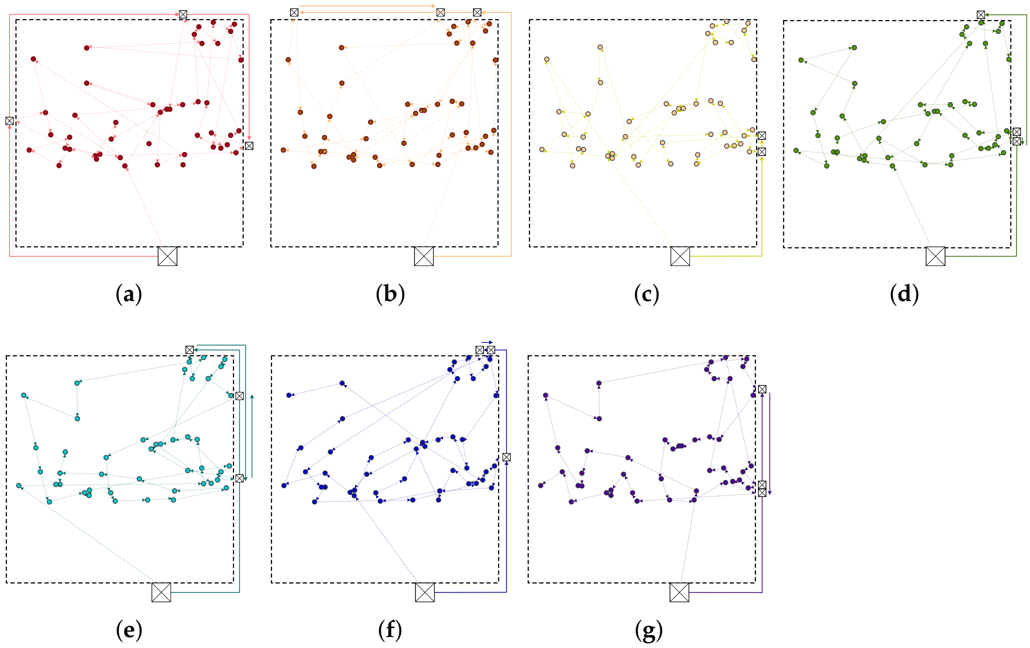


Figure 12. Resulting drone routes (in dash lines) and truck routes (in solid lines) obtained by the nine algorithms on instance 8. (**a**) GA. (**b**) PSO. (**c**) DE. (**d**) BBO. (**e**) EBO. (**f**) WWO. (**g**) SimWWO.

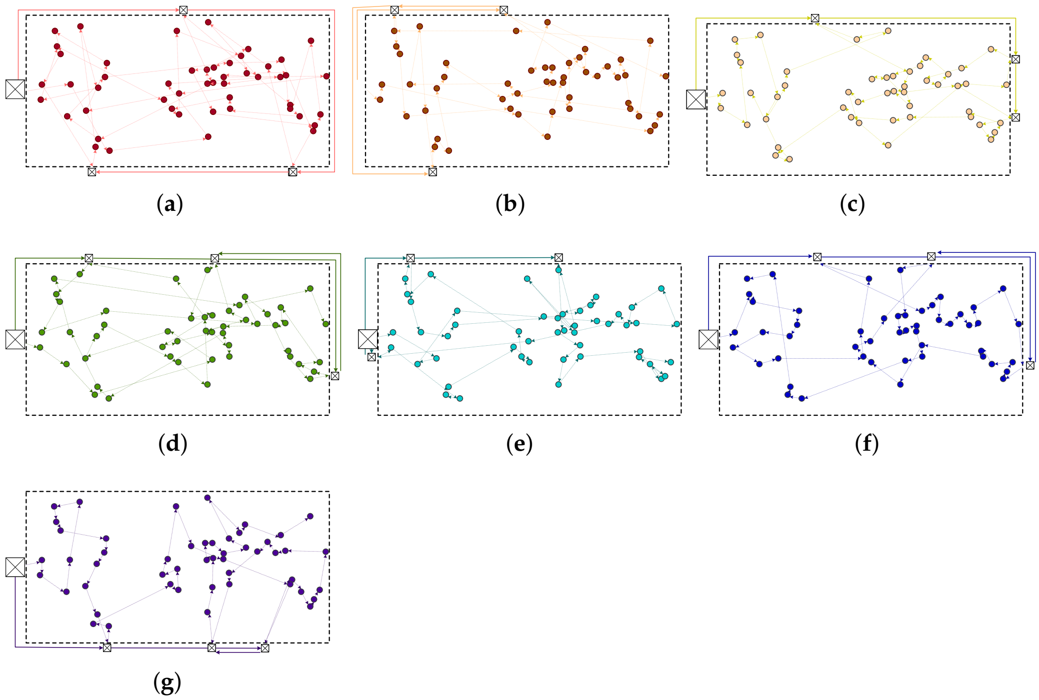


Figure 13. Resulting drone routes (in dash lines) and truck routes (in solid lines) obtained by the nine algorithms on instance 9. (**a**) GA. (**b**) PSO. (**c**) DE. (**d**) BBO. (**e**) EBO. (**f**) WWO. (**g**) SimWWO.

As we can observe from the results, the adapted SimWWO achieves the best median values among the seven algorithms for all instances. For instance 1, there are five algorithms (except GA and PSO) that obtain the same best median value; however, only SimWWO obtains the best median value in all 30 runs. For instance 2, DE, WWO, and SimWWO obtain the same best median value, and both WWO and SimWWO obtain the best median value in all 30 runs. For the remaining instances 3–9, SimWWO uniquely obtains the best median values among the seven algorithms. According to statistical tests, SimWWO obtains significantly better results than GA, PSO, and BBO for all nine instances and better results than DE, EBO, and WWO for eight instances. From the box plots, we can also observe that the deviation of the resulting objective function value of SimWWO among the 30 runs is the smallest among the comparative algorithms, which validates the robustness of SimWWO. Averaged over the nine instances, the median delivery time of the solution obtained by SimWWO is only 48.96% of that obtained by GA, 65.6% of that obtained by PSO, 77% of that obtained by DE, 80.6% of that obtained by BBO, and 78.45% of that obtained by EBO. In addition, Figure 14 presents the convergence curves of the algorithms for the nine instances, which shows that DE converges quickly for the first two small-size instances, but the convergence speed of SimWWO is the fastest for the remaining medium- and large-size instances. In summary, the experimental results demonstrate the performance advantages of SimWWO over the other comparative algorithms in solving the considered problem.

Table 4. Experimental results of the seven algorithms for the nine test instances.

Ins.	*n*	Metrics	GA	PSO	DE	BBO	EBO	WWO	SimWWO
1	10	median	† 0.0548	† 0.0483	† **0.0444**	† **0.0444**	**0.0444**	† **0.0444**	**0.0444**
		max	0.0628	0.0530	0.0483	0.0451	0.0489	0.0451	0.0444
		min	0.0451	0.0445	0.0444	0.0444	0.0444	0.0444	0.0444
		std	0.0056	0.0025	0.0017	0.0002	0.0008	0.0003	0.0000
2	15	median	† 0.1194	† 0.0920	**0.0832**	† 0.0834	† 0.0835	† **0.0832**	**0.0832**
		max	0.1738	0.1268	0.0979	0.0974	0.0967	0.0833	0.0832
		min	0.0884	0.0832	0.0832	0.0832	0.0832	0.0832	0.0832
		std	0.0198	0.0138	0.0028	0.0029	0.0041	0.0000	0.0000
3	20	median	† 0.1118	† 0.0912	† 0.0830	† 0.0793	† 0.0844	† 0.0760	**0.0730**
		max	0.1331	0.1184	0.0901	0.0861	0.0904	0.0867	0.0748
		min	0.0931	0.0839	0.0767	0.0742	0.0786	0.0714	0.0714
		std	0.0079	0.0073	0.0034	0.0032	0.0032	0.0038	0.0011
4	25	median	† 0.2399	† 0.1476	† 0.1268	† 0.1257	† 0.1259	0.1153	**0.1150**
		max	0.2882	0.2060	0.1677	0.1498	0.1562	0.1476	0.1155
		min	0.1787	0.1190	0.1166	0.1192	0.1163	0.1147	0.1147
		std	0.0294	0.0269	0.0119	0.0083	0.0111	0.0059	0.0002
5	30	median	† 0.1784	† 0.1401	† 0.1151	† 0.1171	† 0.1264	† 0.1077	**0.1012**
		max	0.2160	0.1661	0.1319	0.1264	0.1352	0.1212	0.1086
		min	0.1575	0.1226	0.1023	0.1075	0.1141	0.0958	0.0899
		std	0.0168	0.0101	0.0067	0.0045	0.0055	0.0068	0.0037
6	35	median	† 0.2159	† 0.2162	† 0.1976	† 0.1472	† 0.1761	† 0.1323	**0.1195**
		max	0.2418	0.2348	0.2190	0.1611	0.1907	0.1437	0.1294
		min	0.1939	0.1874	0.1847	0.1147	0.1567	0.1048	0.1039
		std	0.0117	0.0108	0.0084	0.0081	0.0092	0.0105	0.0072
7	40	median	† 0.4163	† 0.2473	† 0.2046	† 0.2090	† 0.1850	† 0.1247	**0.1238**
		max	0.5232	0.3696	0.2566	0.2337	0.2274	0.1268	0.1250
		min	0.3262	0.1807	0.1510	0.1682	0.1578	0.1232	0.1232
		std	0.0537	0.0498	0.0243	0.0174	0.0174	0.0009	0.0006
8	45	median	† 0.2479	† 0.2288	† 0.1919	† 0.1737	† 0.1800	† 0.1455	**0.1398**
		max	0.2775	0.2512	0.2138	0.1825	0.1972	0.1686	0.1495
		min	0.1993	0.1855	0.1470	0.1575	0.1620	0.1226	0.1226
		std	0.0169	0.0185	0.0156	0.0059	0.0095	0.0108	0.0067
9	50	median	† 0.3304	† 0.2169	† 0.1708	† 0.1832	† 0.1895	† 0.1406	**0.1376**
		max	0.3774	0.2569	0.1990	0.2200	0.2132	0.1620	0.1466
		min	0.2997	0.1633	0.1447	0.1664	0.1496	0.1171	0.1167
		std	0.0204	0.0224	0.0127	0.0119	0.0124	0.0111	0.0068
average (median)			0.2127	0.1587	0.1353	0.1292	0.1328	0.1077	**0.1041**

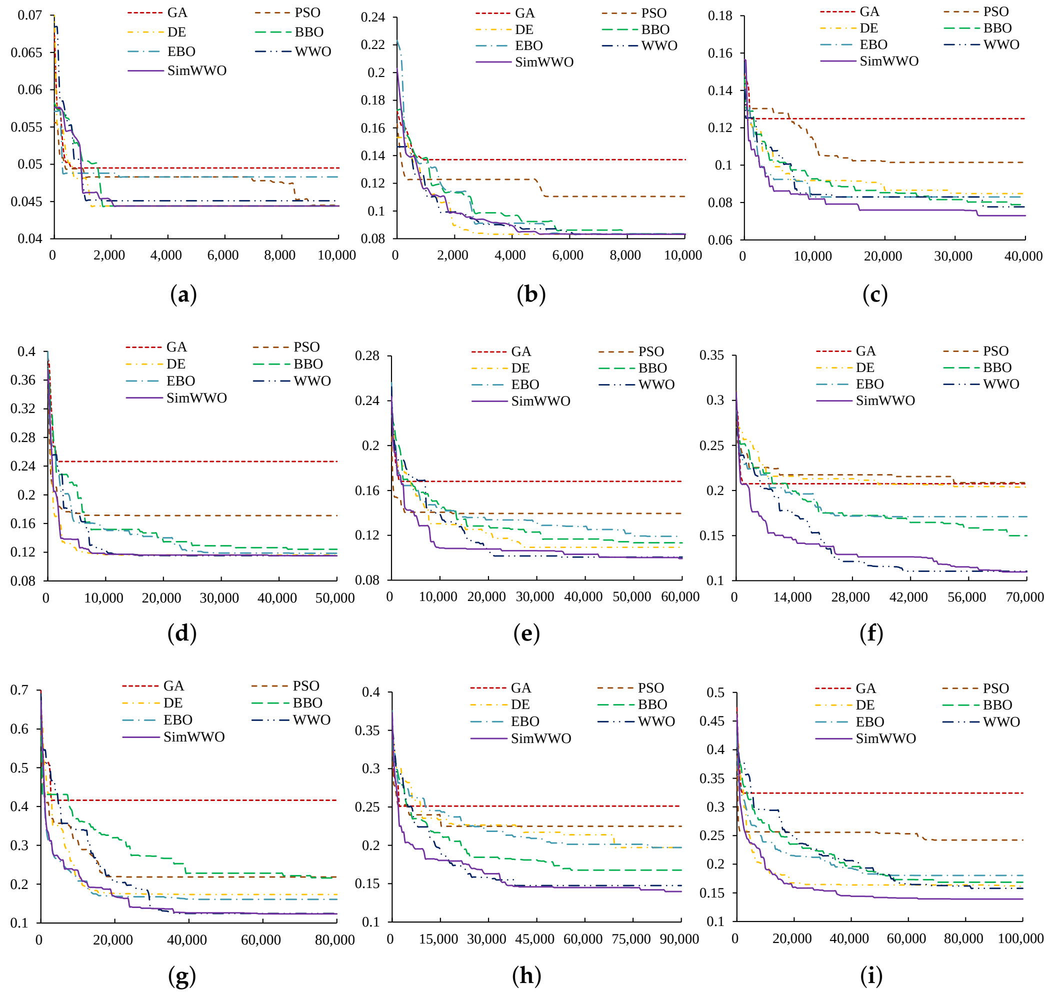


Figure 14. Convergence curves of the comparative algorithms for the test instances. (**a**) Ins. 1. (**b**) Ins. 2. (**c**) Ins. 3. (**d**) Ins. 4. (**e**) Ins. 5. (**f**) Ins. 6. (**g**) Ins. 7. (**h**) Ins. 8. (**i**) Ins. 9.

6. Conclusions

This paper studied the problem of cooperative truck–drone path planning for delivering parcels to customers in a restricted traffic zone that forbids the access of the truck. We propose an adapted SimWWO algorithm hybridizing convex relaxation to efficiently solve the problem. The experimental results demonstrate that the proposed algorithm always obtains the minimum median objective values (i.e., delivery completion time) for all nine test instances among seven comparative algorithms, and its averaged objective value over the nine instances is around 50∼80% of that obtained by the state-of-the-art algorithms. The current problem considers only one drone sent and received by the truck, which is normal in practice because the space inside a truck carrying many parcels is often limited and cannot accommodate more drones. However, for large-scale vehicles, multiple drones can be used to improve delivery efficiency. Therefore, our ongoing study is now considering a more complex scenario using multiple drones, for which we are developing more effective and efficient algorithms, such as multi-population swarm intelligence algorithms [45,46] and neural optimization with reinforcement learning [47–49].

Author Contributions: Conceptualization, Y.-Y.W.; methodology, Y.-J.Z.; software, Y.-Y.W.; investigation, R.-Y.W.; writing—original draft preparation, Y.-Y.W.; writing—review and editing, Y.-J.Z.; funding acquisition, Y.-J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (grant number 61872123) and the Natural Science Foundation of Zhejiang Province (grant number LR20F030002).

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets and program code used in this paper are available at https://www.compintell.cn/en/dataAndCode.html (accessed on 10 January 2023).

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

References

1. Zhang, L.; Long, R.; Chen, H. Do car restriction policies effectively promote the development of public transport? *World Dev.* **2019**, *119*, 100–110. [CrossRef]
2. Chen, X.; Wu, G.; Li, D. Efficiency measure on the truck restriction policy in China: A non-radial data envelopment model. *Transp. Res. Part A Policy Pract.* **2019**, *129*, 140–154. [CrossRef]
3. Lemardelé, C.; Estrada, M.; Pagès, L.; Bachofner, M. Potentialities of drones and ground autonomous delivery devices for last-mile logistics. *Transp. Res. Part E Logis. Transp. Rev.* **2021**, *149*, 102325. [CrossRef]
4. Sham, R.; Siau, C.S.; Tan, S.; Kiu, D.C.; Sabhi, H.; Thew, H.Z.; Selvachandran, G.; Quek, S.G.; Ahmad, N.; Ramli, M.H.M. Drone usage for medicine and vaccine delivery during the covid-19 pandemic: attitude of health care workers in rural medical centres. *Drones* **2022**, *6*, 109. [CrossRef]
5. Zheng, Y.J.; Chen, X.; Song, Q.; Yang, J.; Wang, L. Evolutionary optimization of COVID-19 vaccine distribution with evolutionary demands. *IEEE Trans. Evol. Comput.* **2022**, *in press*. [CrossRef]
6. Karatas, M.; Erişkin, L.; Bozkaya, E. Transportation and location planning during epidemics/pandemics: emerging problems and solution approaches. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25139–25156. [CrossRef]
7. Chen, X.; Yan, H.F.; Zheng, Y.J.; Karatas, M. Integration of machine learning prediction and heuristic optimization for mask delivery in COVID-19. *Swarm Evol. Comput.* **2023**, *76*, 101208. [CrossRef]
8. Li, T. A SWOT analysis of China's air cargo sector in the context of COVID-19 pandemic. *J. Air Transp. Manag.* **2020**, *88*, 101875. [CrossRef]
9. Hwang, J.; Kim, H. The effects of expected benefits on image, desire, and behavioral intentions in the field of drone food delivery services after the outbreak of COVID-19. *Sustainability* **2021**, *13*, 117. [CrossRef]
10. Hong, I.; Kuby, M.; Murray, A.T. A range-restricted recharging station coverage model for drone delivery service planning. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 198–212. [CrossRef]
11. Bányai, T. Impact of the integration of first-mile and last-mile drone-based operations from trucks on energy efficiency and the environment. *Drones* **2022**, *6*, 249. [CrossRef]
12. Salama, M.R.; Srinivas, S. Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites. *Transp. Res. Part E Logis. Transp. Rev.* **2022**, *164*, 102788. [CrossRef]
13. Wells, G.; Stevens, L. Amazon Conducts First Commercial Drone Delivery. Available online: https://www.wsj.com/articles/amazon-conducts-first-commercial-drone-delivery-1481725956 (accessed on 1 November 2022).
14. Burgess, M. DHL's Delivery Drone Can Make Drops Quicker than a Car. Available online: http://www.wired.co.uk/article/dhl-drone-delivery-germany/ (accessed on 1 November 2022).
15. Tilley, A. UPS experiments with drone delivery in partnership with Zipline. Available online: https://www.forbes.com/sites/aarontilley/2016/05/09/ups-experiments-with-drone-delivery-in-partnership-with-zipline/#58f1c1d145a4 (accessed on 1 November 2022).
16. SF Tech. SF UAV. Available online: https://www.sf-tech.com.cn/en/product/uav (accessed on 1 November 2022).
17. Poikonen, S.; Golden, B. Multi-visit drone routing problem. *Comput. Oper. Res.* **2020**, *113*, 104802. [CrossRef]
18. Benarbia, T.; Kyamakya, K. A literature review of drone-based package delivery logistics systems and their implementation feasibility. *Sustainability* **2022**, *14*, 360. [CrossRef]
19. 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]
20. Murray, C.C.; Raj, R. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transp. Res. Part C Emerg. Technol.* **2020**, *110*, 368–398. [CrossRef]
21. Ham, A.M. Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transp. Res. Part C Emerg. Technol.* **2018**, *91*, 1–14. [CrossRef]
22. Agatz, N.; Bouman, P.; Schmidt, M. Optimization approaches for the traveling salesman problem with drone. *Transp. Sci.* **2018**, *52*, 965–981. [CrossRef]
23. Ulmer, M.W.; Thomas, B.W. Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks* **2018**, *72*, 475–505. [CrossRef]
24. Wu, G.; Mao, N.; Luo, Q.; Xu, B.; Shi, J.; Suganthan, P.N. Collaborative truck-drone routing for contactless parcel delivery during the epidemic. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25077–25091. [CrossRef]

25. Zheng, Y.; Du, Y.; Ling, H.; Sheng, W.; Chen, S. Evolutionary collaborative human-UAV search for escaped criminals. *IEEE Trans. Evol. Comput.* **2020**, *24*, 217–231. [CrossRef]
26. Zheng, Y.J.; Du, Y.C.; Sheng, W.G.; Ling, H.F. Collaborative human-UAV search and rescue for missing tourists in nature reserves. *INFORMS J. Appl. Anal.* **2019**, *49*, 371–383. [CrossRef]
27. Zheng, Y.J.; Du, Y.C.; Su, Z.L.; Ling, H.F.; Zhang, M.X.; Chen, S.Y. Evolutionary human-UAV cooperation for transmission network restoration. *IEEE Trans. Ind. Inform.* **2021**, *17*, 1648–1657. [CrossRef]
28. Carlsson, J.G.; Song, S. Coordinated logistics with a truck and a drone. *Manag. Sci.* **2018**, *64*, 4052–4069. [CrossRef]
29. Wang, Z.; Sheu, J.B. Vehicle routing problem with drones. *Transp. Res. Part B Method.* **2019**, *122*, 350–364. [CrossRef]
30. Luo, Z.; Liu, Z.; Shi, J. A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle. *Sensors* **2017**, *17*, 1144. [CrossRef]
31. Karak, A.; Abdelghany, K. The hybrid vehicle-drone routing problem for pick-up and delivery services. *Transp. Res. Part C Emerg. Technol.* **2019**, *102*, 427–449. [CrossRef]
32. Savuran, H.; Karakaya, M. Efficient route planning for an unmanned air vehicle deployed on a moving carrier. *Soft Comput.* **2016**, *20*, 2905–2920. [CrossRef]
33. Boysen, N.; Briskorn, D.; Fedtke, S.; Schwerdfeger, S. Drone delivery from trucks: Drone scheduling for given truck routes. *Networks* **2018**, *72*, 506–527. [CrossRef]
34. Poikonen, S.; Golden, B. The mothership and drone routing problem. *INFORMS J. Comput.* **2020**, *32*, 249–262. [CrossRef]
35. Fawaz, W.; Atallah, R.; Assi, C.; Khabbaz, M. Unmanned aerial vehicles as store-carry-forward nodes for vehicular networks. *IEEE Access* **2017**, *5*, 23710–23718. [CrossRef]
36. Chung, S.H.; Sah, B.; Lee, J. Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Comput. Oper. Res.* **2020**, *123*, 105004. [CrossRef]
37. Castro, P.M. Tightening piecewise McCormick relaxations for bilinear problems. *Comput. Chem. Eng.* **2015**, *72*, 300–311. [CrossRef]
38. Wang, H.F.; Wu, K.Y. Hybrid genetic algorithm for optimization problems with permutation property. *Comput. Oper. Res.* **2004**, *31*, 2453–2471. [CrossRef]
39. Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [CrossRef]
40. Wang, X.; Duan, H. A hybrid biogeography-based optimization algorithm for job shop scheduling problem. *Comput. Ind. Eng.* **2014**, *73*, 96–114. [CrossRef]
41. Chakraborty, U.K.; Turvey, K.P. Floating-point to integer mapping schemes in differential evolution for permutation flow shop scheduling. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 183–204. [CrossRef]
42. Wang, X.; Tang, L. A discrete particle swarm optimization algorithm with self-adaptive diversity control for the permutation flowshop problem with blocking. *Appl. Soft Comput.* **2012**, *12*, 652–662. [CrossRef]
43. Zheng, Y.J.; Ling, H.F.; Xue, J.Y. Ecogeography-based optimization: Enhancing biogeography-based optimization with ecogeographic barriers and differentiations. *Comput. Oper. Res.* **2014**, *50*, 115–127. [CrossRef]
44. Zheng, Y.J. Water wave optimization: A new nature-inspired metaheuristic. *Comput. Oper. Res.* **2015**, *55*, 1–11. [CrossRef]
45. Han, H.; Lu, W.; Qiao, J. An adaptive multiobjective particle swarm optimization based on multiple adaptive methods. *IEEE Trans. Cybern.* **2017**, *47*, 2754–2767. [CrossRef] [PubMed]
46. Liu, W.; Niu, G.; Cao, Q.; Pun, M.O.; Chen, J. 3-D placement of UAVs based on SIR-measured PSO Algorithm. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]
47. Siddiqui, A.B.; Aqeel, I.; Alkhayyat, A.; Javed, U.; Kaleem, Z. Prioritized user association for sum-rate maximization in UAV-assisted emergency communication: A reinforcement learning approach. *Drones* **2022**, *6*, 45. [CrossRef]
48. Wu, G.; Fan, M.; Shi, J.; Feng, Y. Reinforcement learning based truck-and-drone coordinated delivery. *IEEE Trans. Artif. Intell.* **2021**, *in press*. [CrossRef]
49. Wu, C.X.; Liao, M.H.; Karatas, M.; Chen, S.Y.; Zheng, Y.J. Real-time neural network scheduling of emergency medical mask production during COVID-19. *Appl. Soft Comput.* **2020**, *97*, 106790. [CrossRef]