Hindawi
Scientific Programming
Volume 2020, Article ID 8887057, 16 pages
https://doi.org/10.1155/2020/8887057

Research Article

Model and Hybrid Algorithm of Collaborative Distribution System with Multiple Drones and a Truck

Min Lin, Jun-Yan Lyu, Jia-Jing Gao, and Ling-Yu Li

School of Management, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Ling-Yu Li; 2584687069@qq.com

Received 7 April 2020; Revised 12 May 2020; Accepted 19 August 2020; Published 1 September 2020

Academic Editor: Tingsong Wang

This paper studies a coordinated system for multidrone single-truck distribution, where a truck delivers goods to a group of customers along a closed ground path with the help of a number of drones. For each delivery, the truck departs from the distribution centre with drones and all goods needed and returns back to the centre after fulfilling the delivery tasks. That is, the truck assigns these delivery tasks to several of its drones, each of which is responsible for sending goods to a different subgroup of customers in the empty air space. This study provides a new mixed-integer programming model of the routing problem with this distribution system based on urban road network. Meanwhile, a hybrid genetic algorithm and a hybrid particle swarm algorithm are designed. Experimental results show that the performance of the hybrid algorithms is better than that of the corresponding basic algorithms.

1. Introduction

Drones or UAVs (Unmanned Aerial Vehicle) could be used to transport goods such as packages, food, and medicine. Compared with traditional delivery with trucks, delivery with drones has several advantages. Firstly, a drone is able to be unmanned without well-paid drivers so that human and operational costs can be reduced. Secondly, a drone has the ability of flying at a high speed in the empty airspace to avoid ground road congestion and then improve distribution efficiency effectually. Thirdly, drones are especially suitable for the areas or situations in which land transport is not much convenient, such as remote mountains and islands and transport for emergency supplies.

However, there are also some obvious drawbacks for drone delivery. The low load capacity of a drone limits the weight or size of the goods it carries. In addition, since drones are powered by batteries, they have a very short flight radius compared to ordinary fuel trucks. Both of these disadvantages require the drone to travel frequently between central warehouse and customer demand point, which means that, after each delivery, the drone needs to return back to the warehouse to take the next goods, as shown in Figure 1. However, although a normal fuel truck can travel longer distances and carry more goods, it may travel slowly and delivery efficiency may be affected easily due to the large weight itself and road traffic jams. In addition, truck drivers have to operate during the whole driving route so that high salary will be paid to drivers, which means high human cost. In contrast, drones are able to save this kind of cost by delivering automatically without being operated during the whole time. Besides, electric cost of drones is much cheaper than fuel cost of trucks, so truck expense is more in every mile, especially when customer points are very scattered.

A solution to the limited flight distance and load capacity of the drone is to allow the drone to collaborate with other types of delivery tools to complete the delivery tasks. Figure 2 shows a collaborative distribution system that uses a drone and a truck. The truck loads the goods for a group of customers at the distribution centre in advance, and it also needs to carry a drone. After leaving from the distribution centre, the truck is ought to move along a closed path on the ground. When the truck travels to a location near a customer demand point, it sends the drone carrying goods in the direction of this customer, while the truck continues to move on its intended path. The drone carries the goods along a

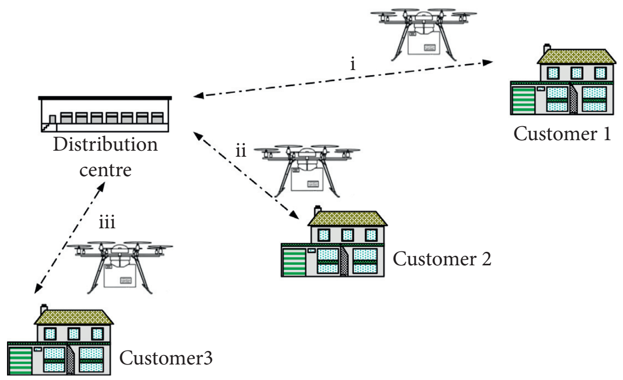


Figure 1: A distribution system that only uses a drone. (i) The drone delivers to Customer 1 from the distribution centre and returns to it. (ii) The drone delivers to Customer 2 from the distribution centre and then returns back. (iii) The drone delivers to another customer from the distribution centre and then returns back.

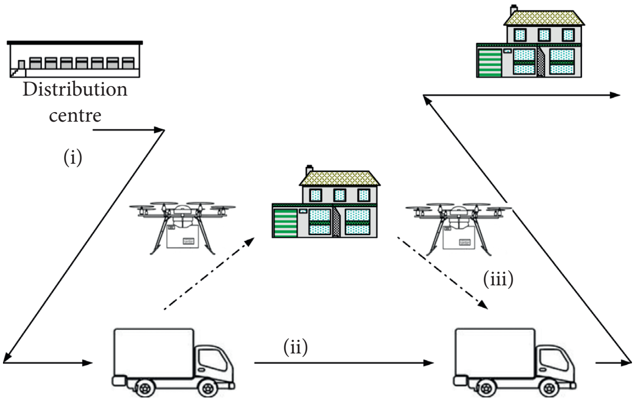


Figure 2: A collaborative distribution system that uses a drone and a truck. (i) The truck is loading goods from the distribution centre. (ii) When the truck approaches a customer, it sends a drone to deliver goods to the customer, and the truck continues to move. (iii) The drone returns to the truck and prepares the next customer's delivery.

straight line to the customer point, then returns back to the truck in another straight line after delivering, gets ready to pick-up new goods, and delivers them to the next customer. For every customer demand point, the drone performs the same actions as mentioned above so that the goods are able to be delivered to all the customers. At last, the truck returns to its original distribution centre.

At present, researchers have also begun to study the optimization of the collaborative distribution system where trucks and drones collaborate to complete delivery tasks, rather than simply relying on single delivery tools. Murray and Raj [1] studied a new extension of the "flying sidekick traveling salesman problem," where a truck works with multiple various drones to deliver packages [1]. They described the problem as a mixed-integer linear programming and proposed a method by using three subproblem sequences. Zhen et al. [2] modelled and solved a nonlinear MIP model for different instance sizes with two efficient algorithms, which are SOCP-based algorithm and dynamic linearization algorithm to handle nonconvex constraints and the model under different scales of route networks, respectively [2]. Bai et al. [3] studied the efficient routing problem for precedence-constrained package delivery for heterogeneous vehicles, where a truck and a microdrone are utilized to deliver packages to dispersed customers [3]. Trucks are restricted to the street network, while microdrones are available for the last-mile delivery with the restriction of their capacity and flight range. The goal of the problem is to minimize the delivery time in the priority constraint. The authors combined topological sorting technology and proposed several task distribution algorithms. Poikonen and Golden considered the k-multivisit drone routing problem, where every drone can take off from a truck and deliver one or more packages to customers [4]. Each drone can return to a truck to replace a battery or charge, take a new package, and then fly again to another customer demand point. de Freitas and Penna studied a flying sidekick traveling salesman problem, where the drone takes off from the truck, delivers the package to the customer, and returns to the truck in the third place [5]. When the drone flies, the truck delivers packages to other customers as long as the drone has enough batteries to hover in the air. Zhen imposed both a stochastic programming formulation that can cope with arbitrary probability distribution of operation time deviation and a robust formulation that is applicable to situation in which limited information about probability distributions is available [6]. Ha et al. studied the Traveling Salesman Problem with Drone (TSP-D) and proposed a hybrid genetic search with dynamic population management and adaptive diversity control based on a split algorithm to solve the completion time for the truck and drone [7]. Results show that the proposed algorithm outperforms existing methods in terms of solution quality and improves best known solutions found in the literature. Zhen et al. developed a column generation solution approach on a set partitioning-based reformulation of the original model [8]. Results show that this simple but practical solution approach can optimally solve the daily berth planning problem instances with reasonable and acceptable for the real-world applications. Liu presented a mixed-integer programming (MIP) model for on-demand meal delivery using drones and proposed an optimization-driven, progressive algorithm for online fleet dispatch operations [9]. Schermer et al. proposed an extension of the VRPD that is called the Vehicle Routing Problem with Drones and En Route Operations (VRPDERO), where drones may not only be launched and retrieved at vertices but also on some discrete points are located on each arc [10]. They also studied the Vehicle Routing Problem with Drones (VRPD), where given a fleet of trucks, each truck carries a given number of drones, and the objective consists in designing feasible routes and drone operations such that all customers are served and minimal make span is achieved [11]. Jeong et al. extended the previous vehicle routing models to the hybrid delivery systems by considering two important practical issues: the effect of parcel weight on drone energy consumption and restricted flying areas [12].

Zhen considered a combination of probabilistic and physics-based models for truck interruptions [13]. It enables to exactly evaluate link travel time, which then acts as the basis for proposing a mixed-integer programming model that minimizes the total expected travel time of moving containers. Wang et al. studied the traveling salesman problem with drone (TSP-D), which aims to find the coordinated routes of a drone and a truck to serve a list of customers. In practice, managers sometimes intend to attain a compromise between operational cost and completion time. Therefore, this article addresses a biobjective TSP-D considering both objectives [14]. Sacramento et al. published a paper where a mathematical model is formulated, defining a problem similar to the Flying Sidekick Traveling Salesman Problem, but for the capacitated multiple-truck case with time limit constraints and minimizing cost as objective function [15]. Karak and Abdelghany presented a mathematical formulation and efficient solution methodology for the hybrid vehicle-drone routing problem (HVDRP) for pick-up and delivery services. The problem is formulated as a mixed-integer program, which minimizes the vehicle and drone routing cost to serve all customers [16]. Javadi et al. extended the traveling Repairman Problem (TRP) by assuming a single truck which can stop at customer locations and launch drones multiple times for each stop location to serve customers. They also developed an efficient hybrid Tabu Search-Simulated Annealing algorithm to solve the problem [17]. Moshref-Javadi et al. presented a mathematical formulation and a heuristic solution approach for the optimal planning of delivery routes. They formulated the problem as a Mixed-Integer Linear Programming model and proposed an efficient Truck and Drone Routing Algorithm (TDRA) [18]. Gonzalez-R et al. proposed an iterated greedy heuristic based on the iterative process of destruction and reconstruction of solutions. This process is orchestrated by a global optimization scheme using a simulated annealing algorithm. The obtained results are quite promising even for large-size scenarios [19]. Chang and Lee focused on finding an effective delivery route for trucks carrying drones and proposed a new approach on a nonlinear programming model to find shift-weights that move the centres of clusters to make wider drone-delivery areas along shorter truck route after initial K-means clustering and TSP (Traveling Salesman Problem) modelling [20]. Carlsson and Song proved that the improvement in efficiency is related to the square root of the ratio of the speeds of the truck and the UAV by combining a theoretical analysis in the Euclidean plane with real-time numerical simulations on a road network [21]. Xia et al. studied how drones can be scheduled to monitor the sailing vessels in ECAs. They also modelled the dynamics of each sailing vessel using a real-time location function which allows to approximately represent the problem on a time-expanded network. Furthermore, they developed a Lagrangian relaxation-based method to obtain near-optimal solutions [22].

The contributions to this article are as follows. (1) This paper delved into the issue of collaborative distribution route planning for drones and trucks based on urban road networks. Although there is a lot of research on co-distribution in recent years, research on co-distribution path planning based on urban road networks is not enough, and our paper filled this gap. (2) This problem involves many decision variables, including determining the routes of trucks and multiple drones and distributing tasks between multiple drones. A number of constraints are also involved in the issue, such as the closeness of the route of the truck, the cargo capacity of the truck, the cruising ability of a drone, and the sequential relationship between all launch sites and landing sites for one drone. (3) For megacities such as Shanghai, London, and Paris, the urban road network is particularly complex and large with a particularly large number of customers. It is apparent that, with the increasing size of the problem and the growing number of decision variables, performance of the algorithm gets worse. Thus, we designed a hybrid algorithm that considers the efficiency and quality of the solution and achieves better computational results at a faster rate.

2. Problem

In this section, we describe a coordinated system for multidrone single-truck distribution and then illustrate the routing problem of the coordinated system on urban road network.

2.1. Multidrone Single-Truck Distribution System.

2.1. Multidrone Single-Truck Distribution System. A coordinated system for multidrone single-truck distribution means that a truck works with multiple drones to deliver to a set of determined customers. That is, the truck assigns these delivery tasks to several of its drones, each of which is responsible for sending goods to a different subgroup of customers in the empty air space. In particular, a truck loads goods and a number of drones from the distribution centre at first. After leaving the distribution centre, the truck travels along a closed ground path. When approaching a customer demand point, the truck releases one of the remaining drones on it and continues to move forward on its intended path, while the drone carries goods to the customer point along a straight line and returns to the truck along another straight line after delivery. When the truck approaches the next customer demand point, the truck releases another drone from the remaining drones, the truck goes on to move forward, and the drone executes the same operation as mentioned above before returning to the truck. At this point, some of previously released drones may still be on delivery to customers and have not yet returned to the truck. If there are no extra drones on the truck, the truck has to wait for a drone to come back and then launch it to another customer point.

2.2. Collaborative Distribution System on Urban Road Network. A city road network refers to a network of all roads in the city. It can be expressed as $G = (V, E)$, where V is a collection of all road intersections and E is a collection of the whole roads. In V, the number of intersections is $0, 1, 2, \ldots, n-1$, and n, where the intersection of number 0 represents the distribution centre. The coordinate of Intersection u is

$$\mu_u = \left(\mu_u^1, \mu_u^2\right) \in \mathfrak{R}, \quad u = 0, 1, 2, 3, \ldots, n. \qquad (1)$$

A directed line from Intersection u to Intersection v is defined as follows:

$$(u, v), \quad u, v = 0, 1, 2, 3, \ldots, n. \qquad (2)$$

Figure 3 is an example of an urban road network. This network is composed of 32 intersections, where the distribution centre is Intersection 0, and the other intersections are numbered as 1, 2, . . ., 30, and 31. The distribution centre 0 requires to deliver goods to six customers, a, b, c, d, e, and f, respectively. A truck will start from the distribution centre to deliver those six customers in turn and then return to the distribution centre. There are many options for this closed ground route driven by the truck. Here are two options:

$$\begin{aligned}&0 \longrightarrow 11 \longrightarrow 10 \longrightarrow 9 \longrightarrow 16 \longrightarrow 17 \longrightarrow 22 \longrightarrow 23 \longrightarrow\\ &24 \longrightarrow 18 \longrightarrow 0,\\ &0 \longrightarrow 18 \longrightarrow 13 \longrightarrow 12 \longrightarrow 11 \longrightarrow 10 \longrightarrow 9 \longrightarrow 16 \longrightarrow\\ &17 \longrightarrow 22 \longrightarrow 23 \longrightarrow 0.\end{aligned} \qquad (3)$$

Figure 4 shows the route of the truck and the flight route of multiple drones, where the black thick arrow represents the route of the truck, and the red and green light arrows represent the flight routes of Drone *1* and Drone *2*, respectively. In each delivery, Drone *1* moves in a red line to a customer and then returns back to the truck in another red line. The process is the same as Drone *2* while the colour of line is green. It is obvious to see that this collaborative distribution system has improved the delivery efficiency to a great extent.

The following is the procedure of collaborative distribution in Figure 4. The truck starts from the distribution centre and moves along a closed ground route indicated with the black thick arrow. First of all, it travels along the line 0⟶11⟶10⟶9. When the truck is gradually close to intersection 9, distance between the truck and Customer a decreases. The truck launches Drone *1* along with the red arrow to Customer a. After sending goods to Customer a, Drone *1* goes back along another red arrow to the truck which is moving forward and preparing the next delivery at the same time. The truck continues to drive along line 9⟶16⟶17. When traveling to Intersection 16, the truck is getting closer to Customer c, and it launches Drone *1* to deliver to Customer c once again. Similarly, the drone moves along the red arrow, performs the same set of actions as mentioned above, and finally returns to the truck which is moving forward after Customer c's delivery is completed. In the next stage, the truck continues to drive along the line 17⟶22⟶23, and when approaching Intersection *22*, the truck sends Drone *1* to Customer d along the red arrow. Noted that Drone *1* does not return to the truck when the truck is close to Intersection *23*, so Drone *2* is sent to Customer f along the green arrow. Then, the truck continues to drive along the line 23⟶24⟶18⟶0, and when it is near to Intersection 24, it sends Drone *2* along the green arrow to Customer e. Since Drone *2* has not returned to the truck when it is approaching to Intersection *18*, the truck needs to send Drone *1* instead along the red arrow to Customer b. At this moment, all customers have been delivered and the truck could return to the distribution centre.

2.3. Notational Conventions. Before discussing the mathematical model, it is necessary to illustrate some mathematical symbols involved. Here are some of the main mathematical symbols.

$\| * \|$: we use Euclidean distance to denote travel distance according to Carlsson [21] and Agatz [23], for any two points $A(x_1, y_1)$ and $B(x_2, y_2)$:

$$\|A - B\| = \sqrt{\left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2}. \qquad (4)$$

$\mathfrak{R}$: the Euclidean plane.

h: the number of customers to visit in the Euclidean plane.

I: the collection of the customers that need to be served, $I = \{1, 2, \ldots, h\}$.

i: customer number, $i \in I$.

p_i: the coordinates of the position of Customer i, $p_i = (p_i^1, p_i^2) \in \mathfrak{R}$.

g_i: the amount of demand for Customer i.

m: the total number of drones on a truck.

K: the collection of all the drones, $K = \{1, 2, \ldots, m\}$.

k: drone number, $k \in K$.

q: load weight of the truck.

$\wp$: closed transport route of the truck.

G: urban road network $G = (V, E)$.

n: the total number of the road intersections.

V: a collection of all road intersections, $V = \{0, 1, 2, 3, \ldots, n\}$.

(u, v): a directed line from Intersection u to Intersection v, $u, v \in V$, called a directed road.

$\varepsilon_{u,v}$: a binary indicating whether a directed line segment (u, v) is present in the urban road network $G = (V, E)$. Note that there is no directed line segment from any intersection u to itself, that is to say, $(u, u) \notin E$:

$$\varepsilon_{\mathrm{u,v}} = \begin{cases} 1, & \text{if } (u, v) \in E, u \neq v, \\ 0, & \text{if } (u, v) \notin E, u \neq v, \\ 0, & \text{if } u = v. \end{cases} \qquad (5)$$

E: a collection of all the directed roads, $E = \left\{(u, v) \mid \varepsilon_{u,v} = 1, \forall u, v \in V\right\}$.

μ_u: The coordinates of the road intersections, $\mu_u = (\mu_u^1, \mu_u^2) \in \mathfrak{R}$, $u \in V$.

ϕ_0: speed of the truck.

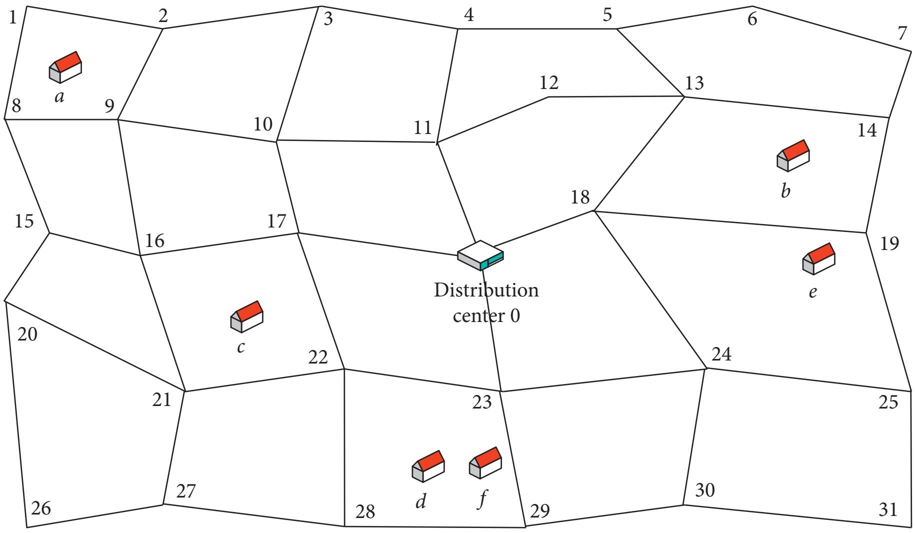


Figure 3: Example of an urban road network.

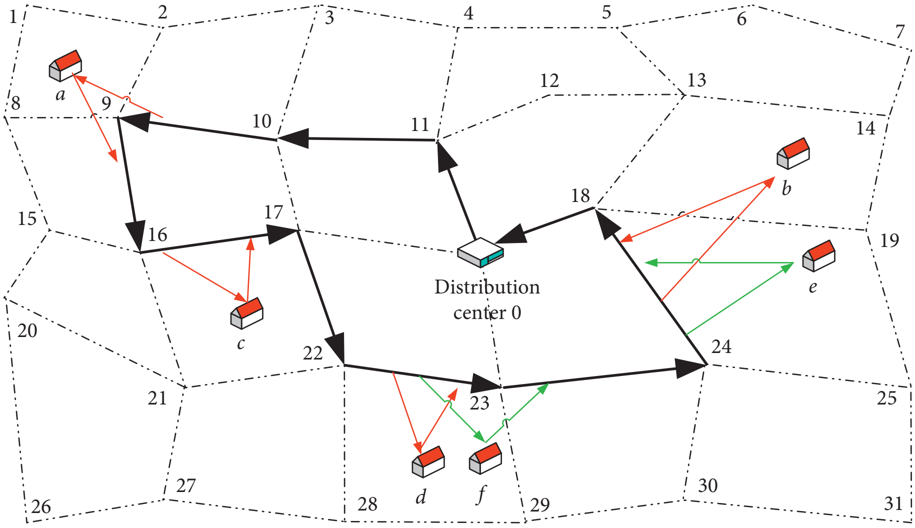


Figure 4: The movement route of a truck and drones.

φ_1: speed of each drone, $\varphi_0 < \varphi_1$.

τ_k: the battery life of the drone k, determining its maximum endurance.

2.4. Decision Variables. The decision variables for the routing problem of collaborative distribution system are listed as follows.

$\omega_{u,v}$ binary equals one if the directed line segment (u, v) belongs to a transport closed route $\wp$, while $u, v = 0, 1, \ldots, n$; otherwise, equals zero:

$$\omega_{u,v} = \begin{cases} 1, & \text{if } (u, v) \in \wp, u \neq v, \\ 0, & \text{if } (u, v) \notin \wp, u \neq v, \\ 0, & \text{if } u = v. \end{cases} \tag{6}$$

$\theta_{i,k}$ binary equals one if Customer i is delivered by Drone k; otherwise, equals zero.

x_i "Launch site" of the drone released from the truck to Customer i.

y_i "Landing place" of the drone which returns onto the truck from Customer i.

Note that each Customer i corresponds to a Launching Point x_i and a Landing Point y_i for the drone. The drone starts from Launching Site x_i, sends goods to Customer i along a straight line, and then returns from Customer i along another straight line to Landing Site y_i on the truck. The details are shown in Figure 5, where the thick black line indicates the route of the truck, the blue dotted lines indicate the flight path of the first drone, the pink dotted lines indicate the flight path of the second drone, and the green one indicates the flight path of the third drone. x_{i-2} and y_{i-2} are, respectively, the launching point and landing point of the first drone for delivery of the customer point $i-2$; x_{i-1} and y_{i-1} are that of customer point $i-1$; x_i and y_i are the launching point and landing point of the second drone for

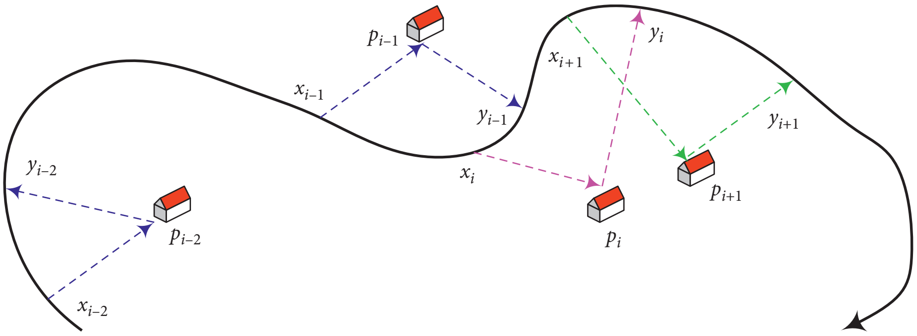


Figure 5: Launching sites and landing sites of drones.

delivery of the customer point i; x_{i+1} and y_{i+1} are the launching point and landing point of the third drone for delivery of customer point $i+1$.

2.5. Routing Problem of the Collaborative Distribution System. The routing problem of the coordinated system for multi-drone single-truck distribution can be defined as follows. Given a city road network $G=(V,E)$, the distribution centre is numbered as 0 while each customer is numbered as $i\,(i=1,2,\ldots,h)$; the location coordinates and demand quantity of all the customers are known. In addition, some assumptions are mentioned as follows to complete our problem:

(i) Only one truck is provided, but is equipped with a certain number of drones

(ii) A customer demand point can only be delivered by only one drone, while a drone is able to deliver to none or more customer demand points

(iii) The weight of cargo for every customer is no more than the capacity of a drone

(iv) All drones have the same speed

(v) Each drone can only transport only one good each time

(vi) The truck is just a platform for drones to launch and land, while itself does not visit customer demand points

(vii) Drones could land on or fly off the truck no matter if the truck is motionless or locomotor

(viii) The number of drones on each truck is fixed and the load weight of the truck excludes the total weight of drones on it, which means the weight of drones is negligible

Our problem aims to find out two sorts of routes in the shortest distribution time: one is the closed transport route $\wp$ in which starting and terminal points are both distribution centre in the whole delivery of the truck, and the other is flight path of each drone on the truck, which means that launching and landing points of all drones are ought to be decided in this problem.

3. Mathematical Model

Some of relevant intermediate variables will be illustrated before recommending the mathematical model.

3.1. Intermediate Variables Related to the Closed Route of the Truck. We define a symbol δ_u that indicates the mileage of the truck from the distribution centre to Intersection u. Obviously, $\delta_0 = 0$.

If the intersection u is not on the transport closed route $\wp$, then there is no intersection v that makes $\omega_{v,u}=1$. Set $\delta_u=-1$ when $u\notin\wp$.

If the intersection u is on the transport closed route $\wp$, then there is only one intersection v that makes $\omega_{v,u}=1$; in other words, $\exists v$, $(v,u)\in\wp$. Because the length of the directed line segment (u,v) is

$$\left\|\mu_u-\mu_v\right\| = \sqrt{\left(\mu_u^1-\mu_v^1\right)^2+\left(\mu_u^2-\mu_v^2\right)^2}, \tag{7}$$

so

$$\delta_u = \sum_{v=0}^{n}\omega_{v,u}\times\left(\delta_v+\sqrt{\left(\mu_u^1-\mu_v^1\right)^2+\left(\mu_u^2-\mu_v^2\right)^2}\right). \tag{8}$$

In a word,

$$\delta_u = \begin{cases} 0, & \text{if } u=0, \\ \displaystyle\sum_{v=0}^{n}\omega_{v,u}\times\left(\delta_v+\sqrt{\left(\mu_u^1-\mu_v^1\right)^2+\left(\mu_u^2-\mu_v^2\right)^2}\right), & \text{if } u\neq 0, \forall u\in\wp, \\ -1, & \text{if } u\neq 0, \forall\, u\notin\wp. \end{cases} \tag{9}$$

3.2. Intermediate Variables Related to Launching and Landing Sites of Drones. In order to simplify the mathematical model, the following decision variables are utilized directly to substitute launching site x_i and landing site y_i.

$\beta_{i,u,v}$: a binary indicating whether the launching point x_i is located on the directed line segment (u, v):

$$\beta_{i,u,v} = \begin{cases} 1, & \text{if } x_i \in (u, v), \\ 0, & \text{if } x_i \notin (u, v). \end{cases} \tag{10}$$

α_i: a decimal indicating dividing the distance between x_i and its start point u by the length of line segment (u, v), so $\alpha_i \in [0, 1)$:

$$\alpha_i = \frac{\left\| x_i - \mu_u \right\|}{\left\| \mu_v - \mu_u \right\|}. \tag{11}$$

$\beta^*_{i,u,v}$: an integer indicating the line segment of $\wp$ in which landing point y_i belongs to

$$\beta^*_{i,u,v} = \begin{cases} 1, & \text{if } y_i \in (u, v), \\ 0, & \text{if } y_i \notin (u, v). \end{cases} \tag{12}$$

α^*_i : a decimal indicating dividing the distance between y_i and its start point u by the length of line segment (u, v), so $\alpha^*_i \in [0, 1)$:

$$\alpha^*_i = \frac{\left\| y_i - \mu_u \right\|}{\left\| \mu_v - \mu_u \right\|}. \tag{13}$$

Depending on the decision variables above, coordinates of the launching point x_i and landing point y_i could be calculated as follows:

χ_i^1: the x-coordinate of the launching point: $\chi_i^1 = \sum_{u=1}^n \sum_{v=1}^n \beta_{i,u,v} \times (\mu_u^1 + \alpha_i \times (\mu_v^1 - \mu_u^1)))$

χ_i^2: the y-coordinate of the launching point: $\chi_i^2 = \sum_{u=1}^n \sum_{v=1}^n \beta_{i,u,v} \times (\mu_u^2 + \alpha_i \times (\mu_v^2 - \mu_u^2))$

γ_i^1: the x-coordinate of the landing point: $y_i^1 = \sum_{u=1}^n \sum_{v=1}^n \beta^*_{i,u,v} \times (\mu_u^1 + \alpha^*_i \times (\mu_v^1 - \mu_u^1))$

γ_i^2: the y-coordinate of the landing point: $y_i^2 = \sum_{u=1}^n \sum_{v=1}^n \beta^*_{i,u,v} \times (\mu_u^2 + \alpha^*_i \times (\mu_v^2 - \mu_u^2))$

Define a symbol ξ_i that indicates the mileage of the truck from the distribution centre to launching point χ_i:

$$\xi_i = \sum_{u=1}^{n} \sum_{v=1}^{n} \beta_{i,u,v} \times \left(\delta_u + \sqrt{\left(\chi_i^1 - \mu_u^1\right)^2 + \left(\chi_i^2 - \mu_u^2\right)^2} \right). \tag{14}$$

That is,

$$\xi_i = \sum_{u=0}^{n} \sum_{v=0}^{n} \beta_{i,u,v} \times \left(\delta_u + \alpha_i \times \left\| \mu_v - \mu_u \right\| \right). \tag{15}$$

Define a symbol ψ_i that indicates the mileage of the truck from the distribution centre to landing point γ_i:

$$\psi_i = \sum_{u=1}^{n} \sum_{v=1}^{n} \beta^*_{i,u,v} \times \left(\delta_u + \sqrt{\left(y_i^1 - \mu_u^1\right)^2 + \left(y_i^2 - \mu_u^2\right)^2} \right). \tag{16}$$

That is,

$$\psi_i = \sum_{u=0}^{n} \sum_{v=0}^{n} \beta^*_{i,u,v} \times \left(\delta_u + \alpha^*_i \times \left\| \mu_v - \mu_u \right\| \right). \tag{17}$$

The flight mileage of the drone when delivering to Customer i can also be calculated as follows:

$$\begin{aligned} \left\| \chi_i - p_i \right\| + \left\| p_i - \gamma_i \right\| = {} & \sqrt{\left(\chi_i^1 - p_i^1)\right)^2 + \left(\chi_i^2 - p_i^2\right)^2} \\ & + \sqrt{\left(p_i^1 - \gamma_i^1\right)^2 + \left(p_i^2 - \gamma_i^2\right)^2}. \end{aligned} \tag{18}$$

3.3. Mathematical Model and Its Description.

$$\begin{aligned} \text{Minimizer} R = {} & \sum_{i=1}^{h} \left| \frac{1}{\phi_0} \left(\psi_i - \xi_i \right) - \frac{1}{\phi_1} \left(\left\| \chi_i - p_i \right\| + \left\| p_i - \gamma_i \right\| \right) \right| \\ & + \frac{1}{\phi_0} \sum_{u=0}^{n} \sum_{v=0}^{n} \omega_{u,v} \times \left\| \mu_u - \mu_v \right\|, \end{aligned} \tag{19}$$

subject to

$$\omega_{u,u} = 0, \quad \forall u \in V, \tag{20}$$

$$\omega_{u,v} \le \varepsilon_{u,v}, \quad \forall u, v \in V, \tag{21}$$

$$\sum_{u=0}^{n} \omega_{u,v} \le 1, \quad \forall v \in V, \tag{22}$$

$$\sum_{v=0}^{n} \omega_{u,v} \le 1, \quad \forall u \in V \tag{23}$$

$$\sum_{v=0}^{n} \omega_{u,v} = \sum_{t=0}^{n} \omega_{t,u}, \forall u \in V, \tag{24}$$

$$\delta_0 = 0, \tag{25}$$

$$\delta_u = \sum_{v=0}^{n} \omega_{v,u} \times \left(\delta_v + \left\| \mu_u - \mu_v \right\| \right), \quad \forall u \in V, u \ne 0, \tag{26}$$

$$\beta_{i,u,v} \le \omega_{u,v}, \quad \forall u, v \in V, \forall i \in I, \tag{27}$$

$$\beta^*_{i,u,v} \le \omega_{u,v}, \quad \forall u, v \in V, \forall i \in I, \tag{28}$$

$$\sum_{u=1}^{n}\sum_{v=1}^{n}\beta_{i,u,v}=1, \quad \forall i\in I, \tag{29}$$

$$\sum_{u=1}^{n}\sum_{v=1}^{n}\beta_{i,u,v}^{*}=1, \quad \forall i\in I, \tag{30}$$

$$\chi_i^1=\sum_{u=1}^{n}\sum_{v=1}^{n}\beta_{i,u,v}\times\left(\mu_u^1+\alpha_i\times\left(\mu_v^1-\mu_u^1\right)\right), \quad \forall i\in I, \tag{31}$$

$$\chi_i^2=\sum_{u=1}^{n}\sum_{v=1}^{n}\beta_{i,u,v}\times\left(\mu_u^2+\alpha_i\times\left(\mu_v^2-\mu_u^2\right)\right), \quad \forall i\in I, \tag{32}$$

$$y_i^1=\sum_{u=1}^{n}\sum_{v=1}^{n}\beta_{i,u,v}\times\left(\mu_u^1+\alpha_i^*\times\left(\mu_v^1-\mu_u^1\right)\right), \quad \forall i\in I, \tag{33}$$

$$y_i^2=\sum_{u=1}^{n}\sum_{v=1}^{n}\beta_{i,u,v}^{*}\times\left(\mu_u^2+\alpha_i^*\times\left(\mu_v^2-\mu_u^2\right)\right), \quad \forall i\in I, \tag{34}$$

$$\xi_i=\sum_{u=0}^{n}\sum_{v=0}^{n}\beta_{i,u,v}\times\left(\delta_u+\alpha_i\times\left\|\mu_v-\mu_u\right\|\right), \quad \forall i\in I, \tag{35}$$

$$\psi_i=\sum_{u=0}^{n}\sum_{v=0}^{n}\beta_{i,u,v}^{*}\times\left(\delta_u+\alpha_i^*\times\left\|\mu_v-\mu_u\right\|\right), \quad \forall i\in I, \tag{36}$$

$$\xi_i\le\psi_i, \quad \forall i\in I, \tag{37}$$

$$\sum_{k=1}^{m}\theta_{i,k}=1, \quad \forall i\in I, \tag{38}$$

$$\sum_{i=1}^{h}\theta_{i,k}\times\frac{1}{\phi_1}\left(\left\|\chi_i-p_i\right\|+\left\|p_i-\gamma_i\right\|\right)\le\tau_k, \quad \forall k\in K, \tag{39}$$

$$\left(\xi_i-\xi_j\right)\times\left(\xi_i-\psi_j\right)\times\theta_{i,k}\times\theta_{j,k}\ge 0, \quad \forall i,j\in I,\forall k\in K, \tag{40}$$

$$\left(\psi_i-\xi_j\right)\times\left(\psi_i-\psi_j\right)\times\theta_{i,k}\times\theta_{j,k}\ge 0, \quad \forall i,j\in I,\forall k\in K, \tag{41}$$

$$\sum_{i=1}^{h}g_i\le q. \tag{42}$$

Here, $\omega_{u,v}$, α_i, α_i^*, $\beta_{i,u,v}$, $\beta_{i,u,v}^*$, and $\theta_{i,k}$ are decision variables with $0\le\alpha_i<1$ and $0\le\alpha_i^*<1$. In particular, $\omega_{u,v}$, $\beta_{i,u,v}$, $\beta_{i,u,v}^*$, or $\theta_{i,k}$ is an integer for 0 or 1, while α_i or α_i^* is a decimal between 0 and 1, that is, $\alpha_i\in[0,1)$ and $\alpha_i^*\in[0,1)$.

The objective function (19) minimizes the total delivery time. We do not simply add the truck time and drone time to obtain the whole completion time during delivery tasks. This is because these two vehicles need to be synchronized at the drone's landing point, which means that the truck have to stop and wait for the drone if it arrives to the landing point early, while the drone is also ought to wait until the truck comes if the drone arrives to the landing point early.

In function (19), the first term $\sum_{i=1}^{h}|(1/\phi_0)(\psi_i-\xi_i)-(1/\phi_1)(\|\chi_i-p_i\|+\|p_i-\gamma_i\|)|$ represents the total time which the truck spends to wait for the drone or the drone spends to wait for the truck in all pairs of launching point χ_j and landing point γ_j. In this term, $(1/\phi_0)(\psi_i-\xi_i))$ shows the time which the truck takes from a launch point to the corresponding landing point, and $(1/\phi_1)(\|\chi_i-p_i\|+\|p_i-\gamma_i\|)$ indicates the total time that the drone takes to leave the truck from the launching point, reach the customer demand point, and then return to the truck at the landing point. The second term $(1/\phi_0)\sum_{u=0}^{n}\sum_{v=0}^{n}\omega_{u,v}\times\|\mu_u-\mu_v\|$ represents the total time of the truck on the closed loop when the drone and the truck are fully synchronized for each delivery. In other words, flight time of the drone is completely equal to the time of the truck for each pair of take-off points and landing points.

Constraint (20) indicates that there is no directed line segment from the intersection to itself. Constraint (21) says if $\varepsilon_{u,v}=0$, that is to say $(u,v)\notin E$, then $\omega_{u,v}=0$. Otherwise, if $\varepsilon_{u,v}=1$, then $\omega_{u,v}=0$ or $\omega_{u,v}=1$. Constraint (22)shows that the number of input lines of any Intersection v cannot exceed 1. Constraint (23) ensures that the quantity of input lines of any Intersection u is 1 at most. These two constraints are used to guarantee that any line segment is passed by the truck only once. Constraint (24) means that an output or input line segment of an intersection is allocated to at most one input or output line segment or not. These constraints enable a closed loop of the truck, and there is no need for it to travel through all intersections in the road network. Instead, the truck just requires to go through part of the intersections, starting from and returning back to the distribution centre. Constraint (25) indicates that the total distance of the truck driving from the distribution centre to the distribution centre is 0. Constraint (26) represents the total distance of the truck driving from the distribution centre to the intersection u, $\forall u\in V, u\neq 0$. Constraint (27) says if $\omega_{u,v}=0$, that is to say $(u,v)\notin\wp$, then $\beta_{i,u,v}=0$. Otherwise, if $\omega_{u,v}=1$, then $\beta_{i,u,v}=0$ or $\beta_{i,u,v}=1$. Constraint (28) says if $\omega_{u,v}=0$, that is to say $(u,v)\notin\wp$, then $\beta_{i,u,v}^*=0$. Otherwise, if $\omega_{u,v}=1$, then $\beta_{i,u,v}^*=0$ or $\beta_{i,u,v}^*=1$. Constraint (29) indicates that a launching point can only be on one directed line segment. Constraint (30) indicates that a landing point can only be on one directed line segment. Constraints (31) and (32) represent the x-coordinate and y-coordinate of the launching point, respectively, while Constraints (33) and (34) are the x-coordinate and y-coordinate of the landing point, respectively. Constraints (35) and (36) show the mileage of the truck from distribution centre to launching point and to landing point separately. Constraint (37) imposes that, for the drone of each customer, the launching point corresponded on the truck trajectory is earlier than the landing point. Constraint (38) guarantees that a customer demand point is allocated to only one drone, while a drone could deliver to none or some customer demand points. Constraint (39) states the relationship between endurance mileage of a drone and its battery life. Constraints (40) and (41) ensure that for any Customer i and Customer j which are delivered by the same drone, both launching point χ_i and landing point γ_i, are either earlier than launching point χ_j or later than landing point γ_j, which means that χ_i and γ_i are not able to locate between χ_j and γ_j on the truck trajectory. Constraint (42) imposes that the gross weight of goods loading on the truck is no more than the maximum carrying weight q.

Since our problem is a new extension of the Vehicle Routing Problem (VRP) which has been shown to be NP-hard, the problem in this paper is also NP-hard and the model is nonlinear. In addition, CPLEX is always used to solve small-scale vehicle routing problem, but not to large-scale vehicle routing problem due to the fact of low solution efficiency. Therefore, we utilize heuristic algorithm in this paper for large-scale problem.

4. Algorithm

With the increasing number of decision variables, problem scale, and search space of genetic algorithm (GA) or particle swarm optimization (PSO) for a complex problem, performances of algorithm is impacted negatively due to the slow convergence rate of algorithm and large amount of iterations.

In order to improve the convergence speed, this paper designs a hybrid genetic algorithm (h-GA) and a hybrid particle swarm optimization (h-PSO). The basic idea of these two hybrid algorithms is to decompose a problem into a master problem and a child problem and just encode decision variables in the master problem as a chromosome or a particle instead of encoding them all. Decision variables of the child problem are solved by a nonlinear programming algorithm or a heuristic method. This way greatly reduces the search space of genetic algorithm or particle swarm algorithm and then enhances the convergence rate to a great extent.

4.1. Hybrid Genetic Algorithm. Hybrid genetic algorithm breaks down this problem into a master problem and a child problem. The master problem is to device the best closed transport route, which the truck along will start from and return to the distribution centre after delivery tasks are completed.

According to the closed route solved by the master problem, the child problem is utilized for every customer to plan a best flight route, which the drone flies along from the truck to the customer and returns back to the moving truck after present delivering. As the demand and coordinate location of each customer is known, this child problem is equivalent to finding out the best launching point and landing point of a drone for each customer on the known closed transport route of the truck.

4.1.1. Genetic Algorithm of the Master Problem. In this section, the master problem is solved by genetic algorithm. Figure 6 shows the flowchart of this algorithm.

The main steps of the genetic algorithm are encoding and decoding of chromosomes and the fitness function. Please refer to relevant documents for details of other steps.

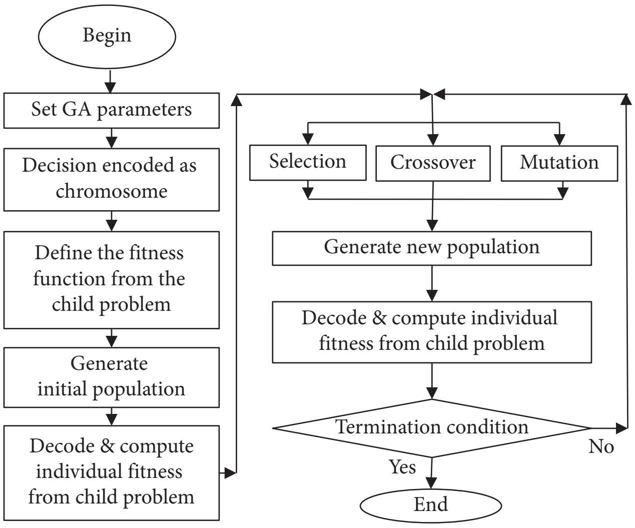

Begin
Set GA parameters
Decision encoded as chromosome
Define the fitness function from the child problem
Generate initial population
Decode & compute individual fitness from child problem
Selection
Crossover
Mutation
Generate new population
Decode & compute individual fitness from child problem
Termination condition
No
Yes
End

Figure 6: Genetic algorithm of the master problem.

(1) Encoding and Decoding of Chromosomes of the Master Problem. Because the master problem is simply solving a best closed transport line of the truck, the only decision variable needs to be encoded into chromosomes is $\omega_{\mathrm{u,v}}$, where $u, v \in \{0, \ldots, n\}$. Here, we adopt natural number encoding for $\omega_{\mathrm{u,v}}$. On the contrary, other decision variables such as α_{i}, $\beta_{i,u,v}$, α_i^*, $\beta_{i,u,v}^*$, and $\theta_{\mathrm{i,k}}$ do not need to be encoded into chromosomes, while will be solved by the child problem. As shown in Figure 7, genetic chromosomes just contain $\omega_{u,v}$, which is either 0 or 1. In this way, search space of the genetic algorithm is drastically reduced, and decision variables in the master problem can be solved in short time. For instance, if the closed loop of the truck is as

$$0 \longrightarrow 11 \longrightarrow 10 \longrightarrow 9 \longrightarrow 16 \longrightarrow 17 \longrightarrow 22 \longrightarrow 23 \longrightarrow 24 \longrightarrow 18 \longrightarrow 0, \tag{43}$$

then, in the chromosomes, the values of genes $\omega_{0,11}$, $\omega_{11,10}$, $\omega_{10,9}$, $\omega_{9,16}$, $\omega_{16,17}$, $\omega_{17,22}$, $\omega_{22,23}$, $\omega_{23,24}$, $\omega_{24,18}$, and $\omega_{18,0}$ are 1, and the value of the rest genes are 0.

For a chromosome, if the line segments of its related genes with value 1 can form a closed loop from distribution centre and back to it, the chromosome corresponds to a feasible solution of the master problem.

(2) The Fitness Function of the Master Problem. The fitness function shows whether the solution to the master problem is good or not, and the degree is depended on the following factors. The first factor is to see if this solution to the master problem satisfies the constraints; the second one is to see if the solution to its related child problem meets the constraints; and the third one is to see whether the target value of its child problem is good or not. Specifically, the solution to the master problem requires to be a closure of the path of the truck.

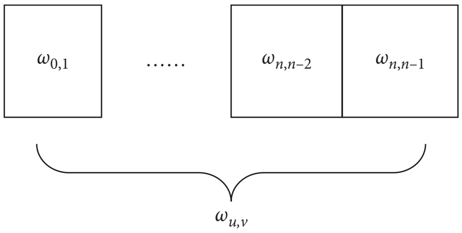


Figure 7: Chromosome of the master problem.

Then, the best flight route of the drone for each customer and target value of child problem are figured out based on the solution of the master problem.

The solution in child problem needs to consider the following conditions: (1) sequence of departure points; (2) landing points of the drone; (3) maximum endurance of the drone; (4) total weight of demand goods, which is at most the maximum load of the truck.

(3) Random Generation of Initial Populations. Each gene is randomly generated within its range of values so as to get a random chromosome. Repeat the above operation until a certain number of chromosomes are produced.

(4) Replication of Chromosomes and Selection of the Elite. Copy a certain number of chromosomes into the next generation with a roulette method. And for each generation of chromosomes, choose the best fitness individuals depending on a certain selection rate to replace the lowest one for the next generation.

(5) Crossover Operation. In the last generation, select two individuals randomly as fathers, randomly identify an intersection, and then perform the crossover operation.

(6) Mutation Operation. Randomly select mutation points in the chromosome, and randomly generate a value for the genes at these mutation points.

(7) Construct New Population. Production of individuals in a new population comes in three approaches: (1) produce new individuals by crossover operation; (2) generate them by mutation operation; (3) select 10% of the optimal fitness individuals of the original population.

(8) Termination Condition. If the number of iterations in the algorithm reaches the specified maximum, terminate the algorithm and output the optimal solutions. Otherwise, return to Step 4 and continue to perform the genetic operations above.

4.1.2. Heuristic Method of the Child Problem. This child problem enables to figure out the best take-off point and landing point for each customer demand point on a known closed transport line of the truck. Since the child problem is nonlinear programming, it can be solved by a heuristic method. Figure 8 shows the details of the heuristic.

(1) Determine the order of assigned customer points.

First, the child problem needs to determine the order of allocated customer points which the drone deliveries. If the total number of customers is n, the amount of all possible delivery orders is $n!$. The calculated amount will be enormous if all possible delivery orders are solved with the child problem. Thus, a construction method is used here to get a better delivery order in a shorter time.

Given the closed transport route $\wp$ of the truck, solution of this construction method is as follows. For each customer demand point p_i, calculate point $\perp_i$ which is the closest to p_i on Route $\wp$. Then, according to the sequence of these points $\perp$ on Route $\wp$, determine the order of assigned customer points. Specifically, if $\perp_i$ is earlier than $\perp_j$ on Route $\wp$, then corresponding demand point p_i in the delivery order is arranged before p_j. On the contrary, if $\perp_i$ is later than $\perp_j$, p_i is arranged after p_j.

(2) Given the truck route and delivery order as above, plan the drone route

When the truck closed route $\wp$ and customer delivery order are solved, this child problem is transformed into a convex optimization problem for the take-off point and landing point of the drone and can also be solved by a structured method as follows.

When solving customer's delivery order following the above step, point $\perp_i$ which is the closest to each point p_i on Route $\wp$ is calculated. The corresponding decision variables for the delivery of a drone to the customer p_i are x_i and y_i. The truck and a drone are ought to avoid waiting for each other at a landing point in order to guarantee time effect. In other words, the best case is a drone and the truck arrives at the landing point at the same time. Thus, the total delivery time of the drone is equal to that of the truck:

$$\frac{1}{\phi_0}\left(\psi_i - \xi_i\right) = \frac{1}{\phi_1}\left(\left\|\chi_i - p_i\right\| + \left\|p_i - \gamma_i\right\|\right). \tag{44}$$

Assuming that take-off point x_i and landing point y_i are symmetric about the line segment from customer point p_i to point $\perp_i$, the flying distance of the drone from x_i to p_i is equal to that from p_i to y_i, and the driving distance of the truck from x_i to $\perp_i$ is equal to that from $\perp_i$ to y_i. So, an approximation can be derived:

$$\frac{1}{\phi_0}\left\|\chi_i - \perp_i\right\| \approx \frac{1}{\phi_1}\left\|\chi_i - p_i\right\|. \tag{45}$$

Assume that the line segment from customer point p_i to point $\perp_i$ is perpendicular to the line segment from x_i to y_i. The approximate solution of the take-off point x_i and landing point y_i can be figured out according to the Pythagorean theorem, respectively.

(3) Solve decision variable $\theta_{i,k}$ according to the order of delivery and the drone route.

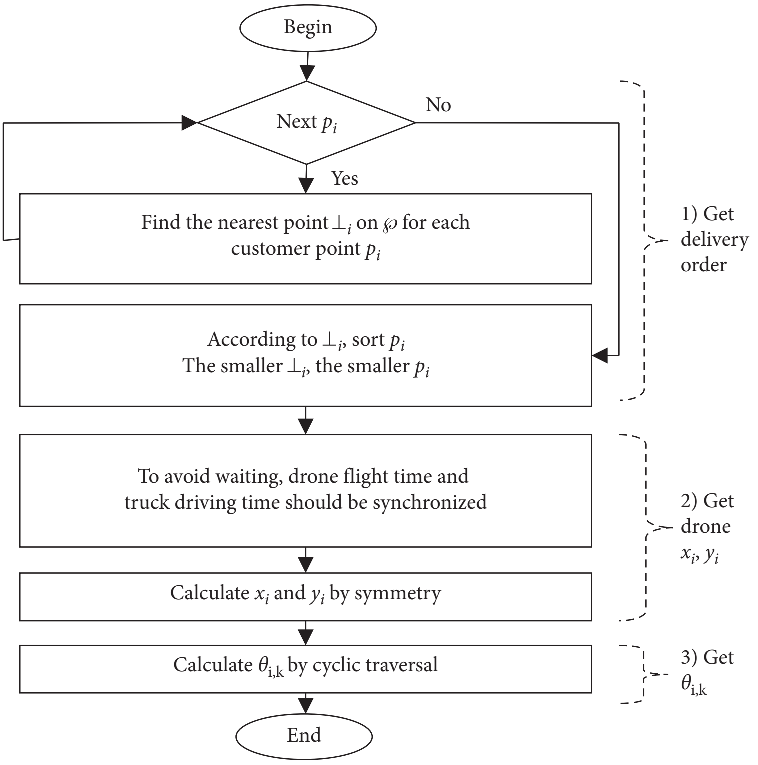

Begin
Next p_i
No
Yes
Find the nearest point $\perp_i$ on $\wp$ for each customer point p_i
According to $\perp_i$, sort p_i
The smaller $\perp_i$, the smaller p_i
1) Get delivery order
To avoid waiting, drone flight time and truck driving time should be synchronized
Calculate x_i and y_i by symmetry
2) Get drone x_i, y_i
Calculate $\theta_{i,k}$ by cyclic traversal
3) Get $\theta_{i,k}$
End

Figure 8: Heuristic algorithm of the child problem.

After the delivery order and drone route are solved by the above steps 1 and 2, decision variables $\theta_{i,k}$ can be solved in order by the sequence loop traversal method as follows. For each drone k, choose each unvisited customer demand point in turn according to the solved delivery order. If a customer point is the ith one for Drone k to visit successfully, then set $\theta_{i,k} = 1$. Otherwise, if launching point of the ith customer point is before the landing point of the $(i-1)$th customer point, set $\theta_{i,k} = 0$ and turn to consider the next customer point until all points are delivered.

4.2. Hybrid Particle Swarm Optimization. Similar to the hybrid genetic algorithm, the hybrid particle swarm algorithm here also divides the routing problem into a master problem and a child problem.

4.2.1. Particle Warm Optimization of the Master Problem. In this section, the master problem is solved by particle swarm optimization, the flowchart of which is shown in Figure 9.

The major steps of particle swarm optimization are the expression of particles and the evaluation function. For detailed information on its other steps, please refer to relevant documents.

(1) Expression of the particle of the master problem.

The key to implement the particle swarm optimization is to find a suitable expression of the particle. Here, the expression only contains decision variable $\omega_{u,v}$, where $u, v \in \{0, \ldots, n\}$, and the value of $\omega_{u,v}$ is either 0 or 1.

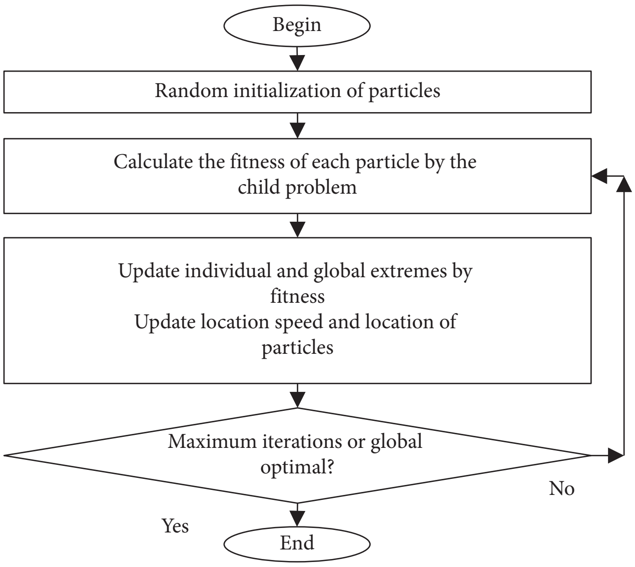

Begin
Random initialization of particles
Calculate the fitness of each particle by the child problem
Update individual and global extremes by fitness
Update location speed and location of particles
Maximum iterations or global optimal?
No
Yes
End

Figure 9: Particle swarm optimization of the master problem.

(2) Evaluation function of the master problem.

The evaluation function indicates whether a solution to the master problem is good or not. Criteria depend on the following factors: (1) this solution to the master problem satisfies the constraints; (2) this solution to its child problem satisfies corresponding constraints; (3) the target value of its corresponding child problem is good.

The implementation steps of the particle swarm algorithm are as follows:

(I) Initialize the particle swarm.

(1) Divide the particle group into numbers of overlapping subgroups
(2) Randomly generate each particle X, where $\omega_{u,v}$ is set either 0 or 1 randomly
(3) Randomly generate each velocity vector V, each dimension of which is set with a real number between $-V\max_d$ and $V\max_d$ randomly
(4) Evaluate all particles with evaluation functions
(5) Set the initial evaluation value as optimal solution P_i for individual history and then search optimal solution P_l and P_g in the subgroups and total group, respectively

(II) Repeat the following steps until satisfying the termination condition or reaching the maximum number of iterations.

(1) For each particle X, the latest velocity vector V is calculated according to the following speed update formula:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times \text{rand}(\cdot) \times [p_{id}(t) - x_{id}(t)] + c_2 \times \text{rand}(\cdot) \times [p_{gd}(t) - x_{id}(t)]. \quad (46)$$

Calculate the latest position vector according to the following position update formula:

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1). \quad (47)$$

If velocity vector V and position vector X exceed its range, then the boundary values are evaluated.

(2) Evaluate all particles with evaluation functions.
(3) If current evaluation of a certain particle is better than the past best evaluation, it will be set as the optimal evaluation value and the current location will be set as the optimal position of the last particle.
(4) Find the optimal solution in the current subgroup and total group. If it is better than the historical best one, update P_l and P_g. If all the individuals in a subgroup are unsolvable, or multiple individuals in the subgroup have the same best values, then randomly take one of them as the current optimal solution in the subgroup.

4.2.2. Heuristic Method of the Child Problem. This child problem is equivalent to figuring out the best take-off point and the best landing point for each customer demand point on a known closed transport line of the truck. This is a nonlinear programming problem, which can be solved by using the heuristic method, as shown in Figure 8.

5. Experiment

In this section, we conduct numerical experiments using a PC (Intel i5, 2.90 GHz, 8 GB of Memory) in Windows 10 to validate the effectiveness of our models and efficiency of our solution methods. The algorithms mentioned above are implemented by C#.

5.1. Experimental Configuration. Figure 10 is the outcome of a program which is used in this experiment to generate a road network randomly, where small red dots represent customer points. Apart from producing any number of customer demand points at any position coordinates, this program enables to generate any size of road network randomly, which is denoted by numbers of line segments. Take some customer points, for example, Figure 11 shows the distribution route calculated by the algorithm, where blue lines and green lines present the flying routes of the first drone and the second drone, respectively, black lines mean the driving route truck, while small red dots represent customer demand points and the blue pentagon in the centre means distribution centre.

In order to test universality of results, we use same input parameters in this experiment with any number of demand points in a road network which is in the size of large, medium, and small, respectively. Besides the distribution centre which is noted as serial number 0, position coordinate and quantity of goods for each customer are generated in a random way.

5.2. Results and Discussion. Figure 12 shows the comparison of four algorithms in different scales, where the horizontal coordinate represents the number of iterations and the vertical coordinate represents current best fitness. Higher fitness means being closer to the optimal solution. In each graph, blue line shows the result of basic genetic algorithm (GA), orange line is that of hybrid genetic algorithm (h-GA), while grey line and yellow line present the result of basic particle swarm algorithm (PSO) and the hybrid particle swarm algorithm (h-PSO), respectively.

It is obvious to see from Figure 12 that, in experiments (a), (b), (e), and (f), the current optimal fitness of h-PSO exceeds all other three algorithms after about 70 iterations. In (c), (g), and (h), the current optimal fitness of h-PSO exceeds all other three algorithms after about 200 iterations. In (d), the current best fitness of h-PSO performs better after approximately 800 iterations.

However, from experiment (a), (b), and (d), we can observe that the current optimal fitness of h-GA is basically a little better than GA and much worse than PSO, while it is better than GA and PSO after about 500 iterations. The situation that current optimal fitness of h-GA is better than GA and PSO in experiments (c) and (g) happens after around 800 iterations, and the same case is shown in experiments (e) and (f) after about 200 iterations and 400 iterations, respectively. In (h), h-GA is basically a little better than GA, and a little worse than PSO.

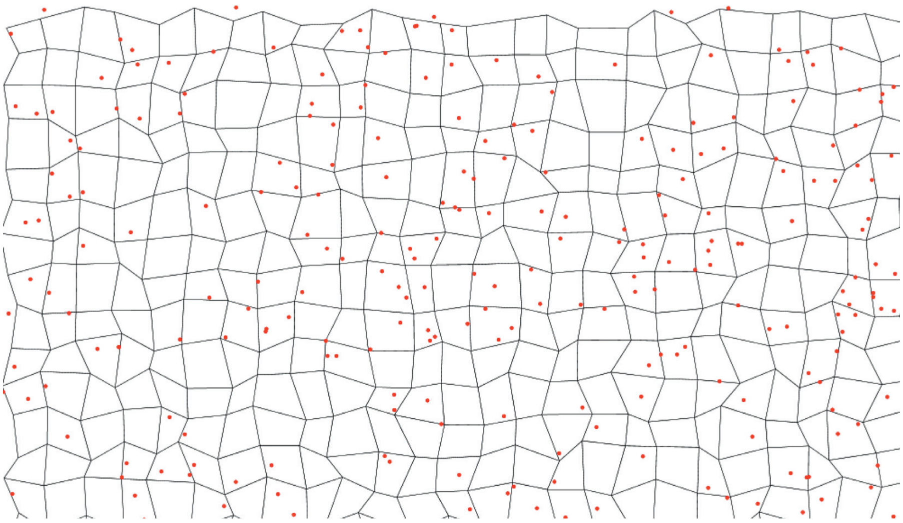


Figure 10: Road network in the algorithm experiment.

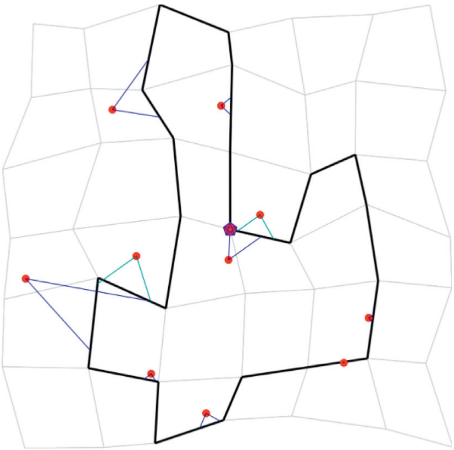
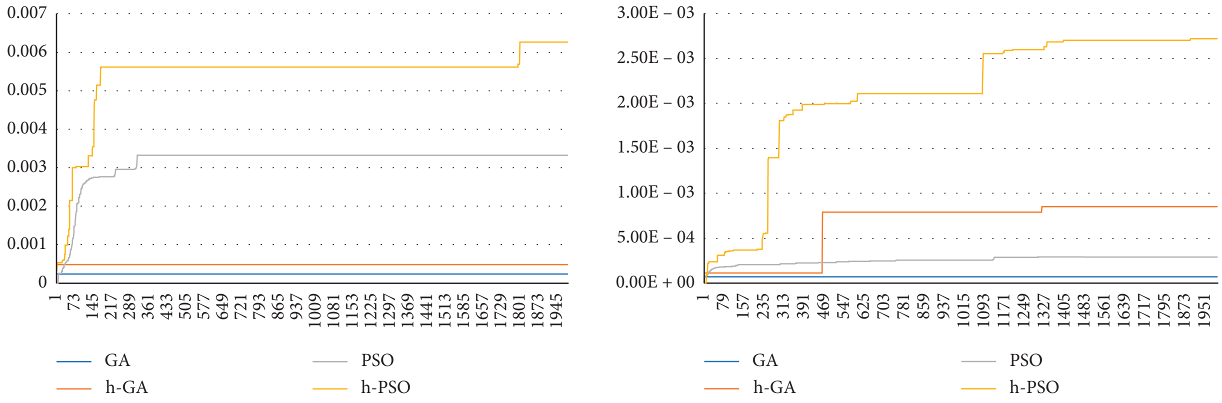


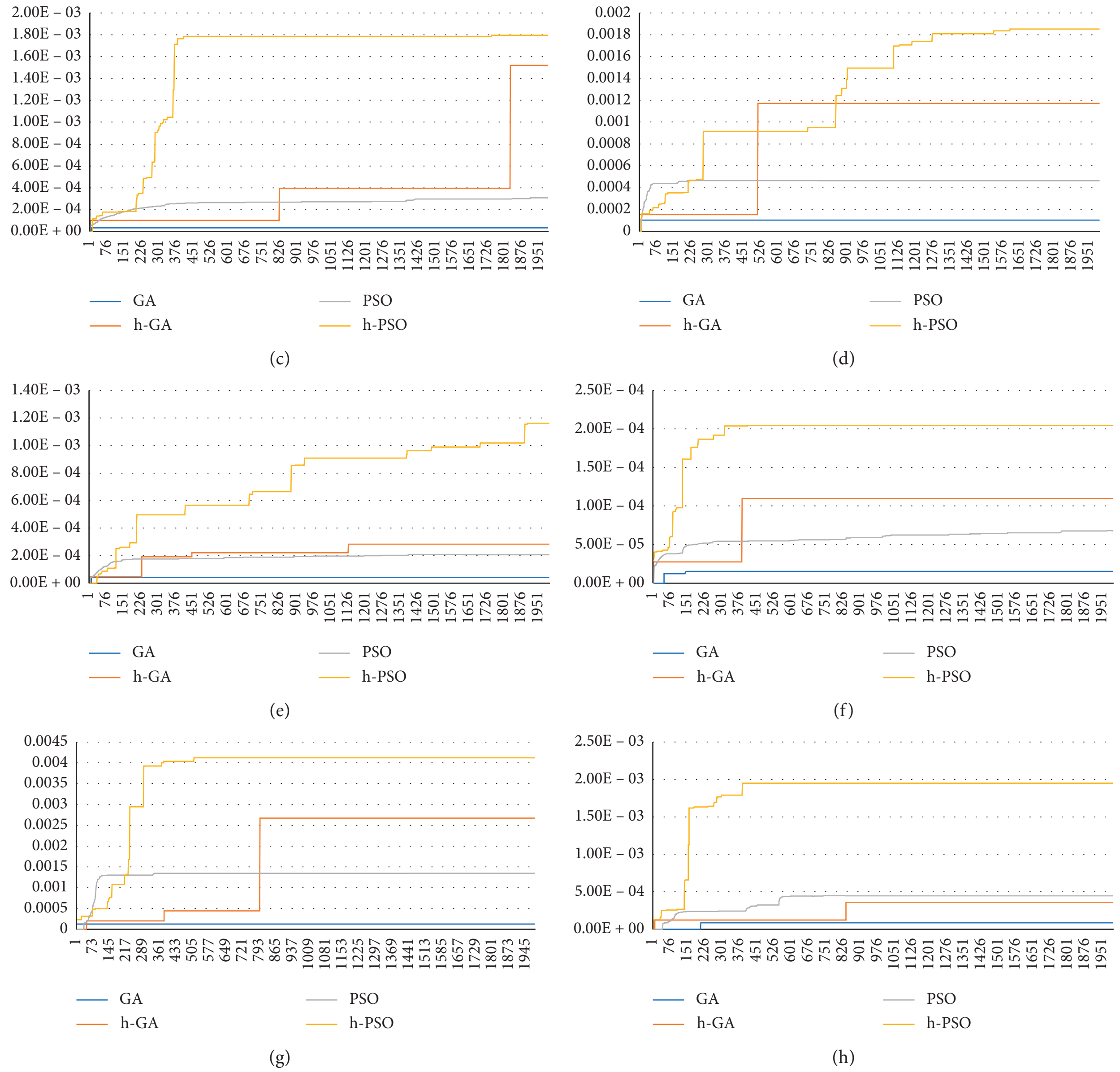
Figure 11: Distribution route calculated by the algorithm.



(a)

(b)

Figure 12: Continued.



(c)

(d)

(e)

(f)

(g)

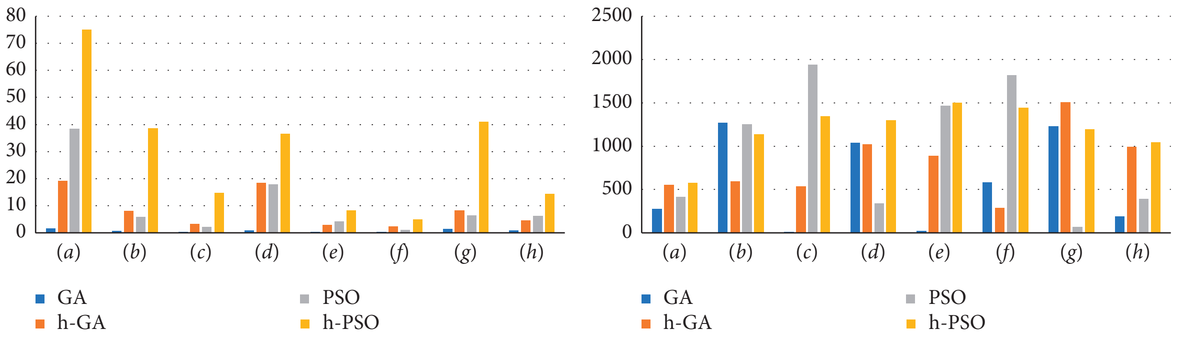
(h)

Figure 12: Algorithm comparison experiment. (a) Network size is 100 and customer number is 20. (b) Network size is 100 and customer number is 60. (c) Network size is 100 and customer number is 100. (d) Network size is 400 and customer number is 20. (e) Network size is 400 and customer number is 60. (f) Network size is 400 and customer number is 100. (g) Network size is 900 and customer number is 10. (h) Network size is 900 and customer number is 20.

Figures 13(a) and 13(b) are the comparison results of the average convergence fitness and average convergence iteration number of the four algorithms, respectively, which are the results of taking the average value after each algorithm is run for 50 times, respectively. In Figure 13, experiments (*a*), (*b*), and (*c*) are realized with the scale of 100 networks, while the number of customers is 20, 60, and 100, respectively. In experiments (*d*), (*e*), and (*f*), network size was replaced with 400, and customer number is also 20, 60, and 100, respectively. Network size in experiments (*g*) and (*h*) is 900. The number of customers in experiment (*g*) is 10 and that in experiment (*h*) is 20.

It can be seen from Figure 13(a) that, from experiments (*a*) to (*h*), the average convergence fitness of h-PSO is better than the other three algorithms. In (*a*) and (*h*), the average convergence fitness of h-GA is better than GA and worse than PSO, while in (*b*), (*c*), (*d*), (*f*), and (*g*), the average convergence fitness of h-GA is better than both GA and PSO.

From Figure 13(b), we can see that, in experiment (*a*), when average convergence iteration number is about 500, the average convergence fitness of *h* = PSO is around 70 while that of h-GA and PSO are no more than 20 and 40, respectively. Besides, GA only achieves an average convergence fitness which is less than 2 when the average convergence iteration number is about 250. Therefore, we can conclude that the solving efficiency of h-PSO is the best, and that of PSO is the second, which is better than h-GA and GA.



(a)

(b)

Figure 13: (a) Comparison of average convergence fitness between hybrid algorithms and basic algorithms. (b) Comparison of average convergence iterations between hybrid algorithms and basic algorithms.

In experiment (*b*), the solution efficiency of h-PSO is the best (the fitness is 40 after about 1000 iterations), that of h-GA is the second (the fitness is below 10 after about 500 iterations), that of PSO is the third (the fitness is below 10 after about 1000 iterations), and the efficiency of GA is the worst (the fitness is below 2 after about 1000 iterations). Experimental results are similar in (*c*)–(*h*). In short, the solving efficiency of these four algorithms could be sorted from the best to the worst as follows: h-PSO, PSO, h-GA, and GA.

Moreover, compared with basic algorithms, the corresponding hybrid algorithms are able to achieve better fitness with fewer iterations and thus achieve a better balance between solution efficiency and solution quality. In addition, the hybrid particle swarm optimization (h-PSO) is the best among all algorithms in terms of solution efficiency and quality.

6. Conclusions

This paper studies a coordinated system for multidrone single-truck distribution, where a truck carrying goods and a number of drones, starting from the distribution centre, will deliver goods to a group of customers along a closed ground path with the help of multiple drones and return to the distribution centre after completing the delivery tasks. In short, the truck assigns delivery tasks to drones, each of which is responsible for different subgroups to send goods to customers. Our study presents a new mixed-integer programming model for the routing problem of this distribution system based on an urban road network. After that, the hybrid genetic algorithm and hybrid particle swarm algorithm are designed. Experimental results show that effectiveness of the hybrid algorithms is better than that of corresponding basic algorithms for the routing problem. Our collaborative distribution model is suitable for the road networks of various cities because trucks are allowed to pass on this kind of network. Besides, in some areas such as remote mountains and islands, road for trucks is usually far away from customer points, which leads to the distance between trucks and customer points being very long for drones to delivery under the restrictions of electric quantity and flight radius of drones. To solve this problem, better battery with bigger battery capacity could be changed, and part of the parameter in our model could be recomposed for different circumstances.

Some future research directions are suggested as follows. First, consider other more additional constraints that exist in reality, such as goods, must be delivered within the time specified by the customer. Second, add uncertainties in collaborative distribution route planning, including urban road conditions with a high degree of randomness such as vehicle breakdowns and traffic accidents, occasional drone malfunction, and uncertain customer demand. Third, consider better types of algorithms. Fourth, combine the simulation technique and the optimization algorithm to build a simulation optimization model to solve the synergistic distribution path problem.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] C. C. Murray and R. Raj, “The multiple flying sidekicks traveling salesman problem: parcel delivery with multiple drones,” *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 368–398, 2020.

[2] L. Zhen, Y. Hu, S. Wang, G. Laporte, and Y. Wu, “Fleet deployment and demand fulfillment for container shipping liners,” *Transportation Research Part B: Methodological*, vol. 120, pp. 15–32, 2019.

[3] X. Bai, M. Cao, W. Yan, and S. S. Ge, “Efficient routing for precedence-constrained package delivery for heterogeneous vehicles,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 248–260, 2020.

[4] S. Poikonen and B. Golden, “Multi-visit drone routing problem,” *Computers & Operations Research*, vol. 113, Article ID 104802, 2020.

[5] J. C. de Freitas and P. H. V. Penna, "A variable neighborhood search for flying sidekick traveling salesman problem," *International Transactions in Operational Research*, vol. 27, no. 1, pp. 267–290, 2020.

[6] L. Zhen, "Tactical berth allocation under uncertainty," *European Journal of Operational Research*, vol. 247, no. 3, pp. 928–944, 2015.

[7] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "A hybrid genetic algorithm for the traveling salesman problem with drone," *Journal of Heuristics*, vol. 26, no. 2, pp. 219–247, 2019.

[8] L. Zhen, Z. Liang, D. Zhuge, L. H. Lee, and E. P. Chew, "Daily berth planning in a tidal port with channel flow control," *Transportation Research Part B: Methodological*, vol. 106, pp. 193–217, 2017.

[9] Y. Liu, "An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones," *Computers & Operations Research*, vol. 111, pp. 1–20, 2019.

[10] D. Schermer, M. Moeini, and O. Wendt, "A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations," *Computers & Operations Research*, vol. 109, pp. 134–158, 2019.

[11] D. Schermer, M. Moeini, and O. Wendt, "A matheuristic for the vehicle routing problem with drones and its variants," *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 166–204, 2019.

[12] H. Y. Jeong, B. D. Song, and S. Lee, "Truck-drone hybrid delivery routing: payload-energy dependency and No-Fly zones," *International Journal of Production Economics*, vol. 214, pp. 220–233, 2019.

[13] L. Zhen, "Modeling of yard congestion and optimization of yard template in container ports," *Transportation Research Part B*, vol. 90, pp. 83–104, 2016.

[14] K. Wang, B. Yuan, M. Zhao, and Y. Lu, "Cooperative route planning for the drone and truck in delivery services: a bi-objective optimisation approach," *Journal of the Operational Research Society*, 2019.

[15] D. Sacramento, D. Pisinger, and S. Ropke, "An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 289–315, 2019.

[16] A. Karak and K. Abdelghany, "The hybrid vehicle-drone routing problem for pick-up and delivery services," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 427–449, 2019.

[17] M. Moshref-Javadi, S. Lee, and M. Winkenbach, "Design and evaluation of a multi-trip delivery model with truck and drones," *Transportation Research Part E: Logistics and Transportation Review*, vol. 136, 2020.

[18] M. Moshref-Javadi, A. Hemmati, and M. Winkenbach, "A truck and drones model for last-mile delivery: a mathematical model and heuristic approach," *Applied Mathematical Modelling*, vol. 80, pp. 290–318, 2020.

[19] P. L. Gonzalez-R, D. Canca, J. L. Andrade-Pineda, M. Calle, and J. M. Leon-Blanco, "Truck-drone team logistics: a heuristic approach to multi-drop route planning," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 657–680, 2020.

[20] Y. S. Chang and H. J. Lee, "Optimal delivery routing with wider drone-delivery areas along a shorter truck-route," *Expert Systems with Applications*, vol. 104, pp. 307–317, 2018.

[21] J. G. Carlsson and S. Song, "Coordinated logistics with a truck and a drone," *Management Science*, vol. 64, no. 9, pp. 4052–4069, 2018.

[22] J. Xia, K. Wang, and S. Wang, "Drone scheduling to monitor vessels in emission control areas," *Transportation Research Part B: Methodological*, vol. 119, pp. 174–196, 2019.

[23] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transportation Science*, vol. 52, no. 4, pp. 965–981, 2018.