

Logistics in the Sky: A Two-Phase Optimization Approach for the Drone Package Pickup and Delivery System

Fangyu Hong^{ID}, Guohua Wu^{ID}, Member, IEEE, Qizhang Luo^{ID}, Huan Liu^{ID}, Xiaoping Fang, and Witold Pedrycz^{ID}, Life Fellow, IEEE

Abstract—The application of drones in last-mile distribution has been a contentious research topic in recent years. Existing urban distribution modes mostly depend on trucks. This paper proposes a novel package pickup and delivery mode and system wherein multiple drones collaborate with automatic devices. The proposed mode uses free areas on top of residential buildings to set automatic devices as delivery and pickup points of packages, and employs drones to transport packages between buildings and depots. The integrated scheduling problem of package drop-pickup considering m -drones, m -depots, and m -customers is crucial for the system. Therefore, we propose a simulated-annealing-based two-phase optimization (SATO) approach to solve this problem. In the first phase, tasks are allocated to depots for serving, such that the initial problem is decomposed into multiple single-depot scheduling problems with m -drone. In the second phase, considering the drone capability and task demand constraints, we generated a route-planning scheme for drones in each depot. Concurrently, an improved variable neighborhood descent (IVND) algorithm was designed in the first phase to reallocate tasks, and a local search (LS) algorithm was proposed to search for high-quality solutions in the second phase. Finally, extensive experiments and comparative studies were conducted to verify the effectiveness of the proposed approach.

Index Terms—Last-mile distribution, drop-pickup, drone, two-phase optimization approach, SATO-IVND.

I. INTRODUCTION

THE last link in logistics transportation is last-mile package delivery [1], wherein logistics enterprises need to communicate with customers directly and deliver packages to them. The rapid development of e-commerce has driven

Manuscript received 12 October 2021; revised 11 October 2022 and 27 February 2023; accepted 24 April 2023. Date of publication 9 May 2023; date of current version 30 August 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62073341, in part by the Natural Science Fund for Distinguished Young Scholars of Hunan Province under Grant 2019JJ20026, and in part by the China Scholarship Council under Grant 202006370285. The Associate Editor for this article was C. G. Panayiotou. (*Corresponding author: Guohua Wu.*)

Fangyu Hong, Guohua Wu, Qizhang Luo, Huan Liu, and Xiaoping Fang are with the School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China (e-mail: fangyuhong@csu.edu.cn; guohuawu@csu.edu.cn; qz_luo@csu.edu.cn; liuhuan1095@csu.edu.cn; fangxp@csu.edu.cn).

Witold Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada, also with the Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia, and also with the Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland (e-mail: wpedrycz@ualberta.ca).

Digital Object Identifier 10.1109/TITS.2023.3271430

the development of the express industry and promoted continuous changes in the package delivery mode. Furthermore, low-altitude airspaces are gradually being opened in many countries. For example, China has approved the operation of 20 civil unmanned aerial system (UAS) test bases, including two test areas of drone logistics and three urban test areas of drone delivery [2]. This policy has greatly promoted the development of drone delivery systems. As new delivery tools, drones have been increasingly considered in last-mile delivery because of their flexibility, low cost, and high reliability. Because traffic congestion occurs frequently in urban environments, classical delivery tools may be inefficient during rush hour. In contrast, drones have obvious advantages in urban package delivery systems. First, they are not restricted by road conditions and can deliver packages without drivers; therefore, packages can be carried by drones throughout the day. In addition, drones can reach areas inaccessible to trucks, such as residential areas.

The common settings of last-mile delivery include couriers, delivery stations, and intelligent express cabinets. With the development of e-commerce businesses and expansion of cities, the shortcomings of these models are becoming increasingly evident. Couriers deliver or pick up packages door-to-door. However, delivery or pickup by couriers is less efficient and incurs a high human cost. Delivery by delivery stations or intelligent express cabinets is more efficient when dealing with packages. Customers can visit sites to pick up or deliver their packages. However, setting the locations of delivery stations and intelligent express cabinets is usually restricted by limited space in urban areas. Additionally, customers who live far from a delivery station or an intelligent express cabinet need to spend considerable time and resources to pick up packages.

To address the abovementioned problems in last-mile delivery, many scholars have begun to design new distribution modes wherein the application of drones has attracted much attention. Amazon started trying to deliver small packages using drones in 2013 [4]. Subsequently, several other companies began similar research. DHL is trying to deliver medicine using drones to people living in remote areas [5]. Many Chinese enterprises, such as Meituan, Jingdong, and Shunfeng, are also committed to research on providing just-in-time delivery services using drones [6]. Shunfeng launched a pilot of local emergency delivery services using drones [7]. The intelligent



Fig. 1. The drone community delivery station developed by Meituan. [3].

power station can provide support for the process of drone takeoff and landing and automatically replace batteries for drones [8]. In addition, the drone community delivery station developed by Meituan was used in certain scenarios [9]. As shown in Fig. 1, after the drone delivers a package to the delivery station, the customer can pick up the package by scanning a specific code.

Compared to traditional delivery modes, delivery with drones can effectively ease the pressure of manual delivery. As drones are flexible and easy to operate, delivery with drones is almost unlimited by traffic conditions. This mode of delivery can deliver packages successfully and can potentially bring about better service quality for last-mile delivery, even in a crowded city. Nevertheless, owing to the current power restriction in long-distance delivery, single-drone delivery is limited to certain constraints, such as the flying range [10] and load-capacity constraints [11]. Therefore, the limited battery capacity of drones is a current challenge in delivering many packages in a large area. To address this issue, scholars have developed two common drone-delivery settings.

In the first setting, a drone is only used to deliver packages. Dorling et al. [12] proposed that optimizing the battery and payload weights could reduce the cost of drone delivery. Song et al. [13] proposed persistent UAV delivery schedules, wherein drones are allowed to share multiple stations to replenish their consumables. This mode may pose security risks in urban environments when drones with delivery packages take off and land from customers.

The other setting is combined truck-drone delivery. In 2015, Murray and Chu [14] proposed a delivery mode that used a truck and drone working cooperatively to deliver packages. Phan et al. [15] proposed the traveling salesman problem with multiple drones (TSP- mD) and employed trucks and drones to deliver packages in parallel. However, there are still many challenges in this mode. For example, it is difficult to arrange a parking stop for trucks and security risks may exist when drones take off and land from trucks or customers.

In contrast to the abovementioned, we propose a novel last-mile delivery mode and system. We named this system the

drone package pickup and delivery system, which makes good use of the free areas at the top of residential buildings. In our system, we set automatic devices at the top of buildings to deposit packages and use drones to deliver packages between buildings and depots. When a package is transported to the correct location, the automatic device can identify the unique barcode of the package and deposit it. In addition, the device informs customers about picking up their packages. To make it more convenient for customers to send packages, customers can make appointments for delivery services at home or at a location of their choice. After customers place their packages on the automatic device, the system arranges drones to pick up and transport the packages. As the entire process is carried out on roofs, it is not limited by urban conditions, and it is more secure for drones to take off and land. Furthermore, as the entire delivery process is almost completed by drones and automation devices, it can reduce the human cost of delivery, improve customer satisfaction, and promote intelligent distribution in last-mile delivery. Therefore, the entire system is efficient and robust.

Integrated scheduling with m -drones, m -depots, and m -customers is crucial for the proposed drone package pickup and delivery system. This problem has been proven an NP-hard problem [16]. Exact algorithms can obtain the optimal solution to the problem at the cost of high computational complexity and considerable running time [17]. In contrast, heuristic [18], [19] and metaheuristic algorithms [20] are well-known and promising alternatives for solving large-scale problems. Currently, package delivery with drones is considered in relevant research, whereas package pickup is scarcely mentioned. In addition, for the integrated scheduling problem considering drop-pickup, a few algorithms can find high-quality solutions within a reasonable time.

Hence, we designed a simulated-annealing-based two-phase optimization (SATO) approach to solve this problem. This approach divides the initial problem into a task-allocation phase for m -depots and a route-planning phase for each depot. In the task-allocation phase, we generated a task-allocation scheme for each depot such that the original problem was decomposed into route-planning problems. In the route-planning phase, we planned the pickup and delivery routes of drones based on the task-allocation scheme. Route planning of drones must meet the drone-capacity and task-demand constraints. These two phases were executed iteratively and interactively until the predefined stopping criteria were satisfied. Under this framework, in the task-allocation phase, we employed the k -means algorithm to generate an initial task-allocation scheme and proposed an improved variable neighborhood descent (IVND) algorithm to reallocate tasks. In the route-planning phase, the effect of the payload on the drone fly range was considered, and heuristic rules were used to plan the route for each depot. In addition, a local search (LS) algorithm was designed to exploit the route planning better. Consequently, an iterative two-phase optimization method, named SATO-IVND, was proposed for the integrated scheduling problem with m -drones, m -depots, and m -customers. The proposed two-phase approach (SATO) can effectively reduce

the complexity of the original problem, whereas the IVND and LS can explore and exploit a satisfactory solution.

The main contributions of this study are highlighted as follows:

- We propose a novel drone package pickup and delivery mode and system to realize agile and efficient last-mile delivery. Because of the limitations of space in urban environments and security risks during the takeoff and landing of drones, automatic devices were placed in the free areas on the top of residential buildings. We used automatic devices as the delivery and pickup points of packages and drones to transport packages between buildings and depots.
- To address the scheduling and routing problems in the proposed system, we developed a two-phase optimization approach. In the first phase, we generated a task-allocation scheme for each depot. In the second phase, the route planning of drones was generated for every depot according to the task-allocation scheme generated in the first phase. These two phases were executed iteratively and interactively until the predefined stopping criteria were satisfied. This approach can reduce computational complexity and solve problems efficiently.
- We proposed an IVND algorithm to assist task allocation in the first phase and used an LS algorithm to design route planning in the second phase. The IVND algorithm was designed to generate a task-allocation scheme considering drone-capability and task-demand constraints. The LS algorithm was used to search for a satisfactory scheme based on the current best scheme generated by the IVND. It can effectively determine high-quality scheduling schemes.
- We conducted extensive experiments to validate the efficiency of the proposed SATO-IVND algorithm. The experimental results show that the SATO-IVND algorithm could obtain near-optimal solutions within a 0.95% gap from the optimal solutions obtained by CPLEX. Furthermore, the SATO-IVND algorithm is superior to the other six heuristic and metaheuristic algorithms with respect to the solution quality and computing overhead, particularly for large-scale problems. In addition, under a realistic scenario with 80 tasks, the proposed SATO-IVND algorithm generated a high-quality scheduling scheme, demonstrating its effectiveness and practicability.

The remainder of this paper is organized as follows. In Section II, we review the related work. Section III describes the construction of the drone package delivery and pickup model. Section IV describes the proposed two-phase optimization approach. Section V introduces the computational experiments and presents the results. Section VI presents conclusions and future study trends.

II. RELATED STUDIES

With the maturity of technology, drones have been widely used in various traffic fields, such as package delivery, traffic-data acquisition, and traffic surveillance [21], [22]. Among these, package delivery with drones has been a contentious

research topic in recent years. In addition, with the increase in online shopping, reverse logistics caused by returning and changing items lacks scientific solutions. Thus, our research mainly focused on the application of drones in last-mile package delivery and pickup.

To the best of our knowledge, research on the application of drones in package delivery mostly focuses on two modes: the independent use of drones or “trucks and drones” [23]. For example, Mathew et al. [24] established a cooperative package delivery mode with a truck and drone, wherein customers could be served by the drone. In addition, as an auxiliary, the truck could provide package transportation, charging, and other services for the drone. This study was later extended by Karak and Abdelghany to consider delivering and picking up packages by drones and a truck [25]. Wang et al. [26] addressed a setting wherein customers could be served by trucks or drones. In their design, trucks could also serve as depots and landing platforms for drones. In contrast to previous studies, Dayarian et al. [27] introduced a new package delivery mode wherein trucks were used to deliver packages to customers and were resupplied by drones.

The integrated scheduling of m -drones, m -depots, and m -customers is a type of combinatorial optimization problem that satisfies certain constraints, such as drop-pickup, drone capacity, and task demand. Similar to solving common integrated scheduling problems of m -drones, task allocation and route planning are the two main parts to deal with the initial problem [28]. Task allocation determines which depot the tasks will be assigned to, whereas route planning determines the order of task access. Previous work studies have attempted to solve the integrated scheduling problems of m -drones as a whole, which makes it difficult to generate high-quality scheduling schemes in a reasonable time. To solve this issue, some studies have focused on innovation of the scheduling framework when addressing complex scheduling problems. For example, Deng et al. [29] proposed a two-phase coordinated planning approach for heterogeneous Earth-observation resources, which included area target decomposition and task-allocation phases. Liu et al. [30] proposed a divide-and-conquer framework to solve multi-drone task-scheduling problems, which included task-allocation and single-drone scheduling phases.

Search algorithms for task allocation mainly include heuristic and metaheuristic algorithms such as clustering algorithms [31], distributed algorithms based on market mechanism [32], randomized greedy-algorithms [33], genetic algorithms (GAs) [34]. Currently, distributed algorithms based on market mechanism are widely used in task allocation. For example, Lee et al. [32] proposed a decentralized auction algorithm for task allocation. In each round of the auction, the robots bid for the ideal task, and the decision system determines the ownership of the task. This algorithm exhibits high communication requirements, and the communication cost may be slightly high. Metaheuristic algorithms, such as GA, require continuous iterative calculations to address the task-allocation problem. Randomized greedy and clustering algorithms are often invoked in the initial task-allocation phase.

TABLE I
THE ARTICLES RELATED TO DRONE DELIVERY

Relation	Drop-Pick	Scalability	Depot	Jobs	Solution	Objective	payload & Energy-consumption
Murray and Chu(2015) [14]	drop	1-truck-1-drone	1	20	heuristics	Min-time	×
Agatz et al. (2018) [35]	drop	1-truck-1-drone	1	100	heuristics	Min-time	×
Song et al. (2018) [13]	drop	k-drone-s-service-station	2	30	heuristics	Max-tasks & Min-travel-distance	✓
Ham(2018) [17]	drop-pick	m-truck-d-drone	2	100	CP	Min-time	×
Murray & Raj (2020) [36]	drop	1-truck-m-drone	1	100	heuristics	Min-time	×
Salama and Srinivas(2020) [10]	drop	1-truck-m-drone	m	50	K-means	min-travel-distance & min-fixed-cost	✓
Dayarian et.al.(2020) [27]	drop	1-truck-1-drone	1	60	heuristics	max-orders/min-travel-distance&time	×
Liu et al.(2021) [37]	drop	1-truck-1-drone	1	100	hybrid heuristic	min-travel-distance	✓
Liu et al.(2021) [30]	×	m-drone	1	400	hybrid heuristic	max-task-profit	×
this paper	drop-pick	m-drone	m	200	SATO-IVND	min-travel-distance & min-launch-sorties	✓

The key to randomized greedy-algorithms is the choice of a greedy strategy, whereas the clustering algorithm can generate task-allocation schemes considering the geographic locations of the tasks.

Search algorithms for route planning mainly include exact algorithms [38] and heuristic and metaheuristic algorithms [39]. Exact algorithms can determine the optimal solution for small-scale problems. However, exact algorithms may not be suitable for solving large-scale problems owing to their high computational complexity [40]. Heuristic and metaheuristic algorithms are more suitable for finding satisfactory solutions to large-scale problems. Metaheuristic algorithms can be applied to different problems, but the optimization effect can be unstable without sophisticated design. Heuristic algorithms are generally oriented toward specific problems, but they depend on specific heuristic rules. Extensive studies have been conducted to improve the efficiency of algorithms. Liu et al. [30] applied a tabu list to simulated annealing (SA) algorithm, which prohibited the circulation or repetition of solutions in a short time, and experiments demonstrated that it could effectively reduce the solving time. Peng et al. [41] proposed a hybrid GA to resolve the traveling-salesman problem with multiple drones. Gao et al. [42] used deep reinforcement learning to train the destroy and repair operators of a large neighborhood search (LNS) algorithm, and its good performance was proven experimentally. Kitjacharoenchai et al. [43] proposed an LNS to find promising routes for trucks and drones. Li et al. [44] proposed an LNS algorithm for scheduling trucks and drones. Fernandez et al. [45] adopted k -means and GA to optimize the route planning of trucks and drones.

Table I presents an overview of the literature on drone deliveries. Based on the analyses of the above existing research progress, the following conclusions can be drawn:

(1) The mode of combined truck-drone delivery is a popular research topic. However, combined truck-drone delivery is more suitable in rural areas, whereas in an urban environment, it may be difficult to arrange stops for trucks and have security risks during drone takeoff and landing.

(2) Most models in the literature have various assumptions and limitations. For example, drones can deliver packages without picking up packages simultaneously. In addition, the effect of the payload on the energy consumption of drones has rarely been considered in the literature.

(3) Effective and efficient heuristic algorithms and problem-specific operators are essential to solve the integrated scheduling of m -drones, m -depots, and m -customers constrained by drop-pickup.

In summary, we propose a novel package pickup and delivery mode and system. We constructed a mixed-integer linear programming (MILP) model for the integrated scheduling problem of m -drones, m -depots, and m -customers considering drop-pickup and the effect of payload. Subsequently, we designed the SATO-IVND algorithm to solve the original problem. The proposed system shows good performance in dealing with large-scale task-allocation and route-planning problems.

III. MODEL OF THE DRONE PACKAGE DELIVERY AND PICKUP

A. Problem Assumptions

The drone package pickup and delivery system for urban last-mile distribution proposed in this study consisted of m -drones, m -depots, m -customers. We assumed the following: (1) Multiple drone stations existed in a city. (2) Each station had multiple drones. (3) Each drone station can be regarded as a depot.

The roof of an urban residential building was used as the delivery/pickup point for drones. Each delivery/pickup point was regarded as a customer. Additionally, each customer could perform many package delivery and pickup tasks. In addition, each customer could visit multiple times. Automatic devices were placed on the roof to store, load, and unload the packages. The drone could load or unload packages on the roof and then move to the next location. In addition, the drone could automatically replace its battery on the top of the residential building, and the replaced battery could be charged on the roof

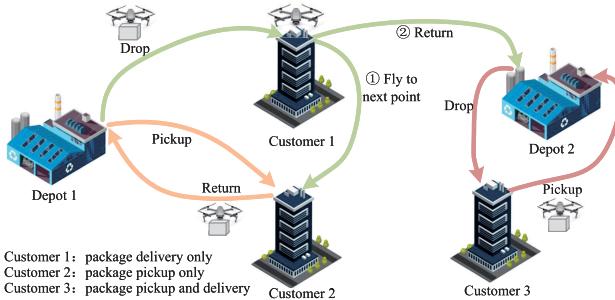


Fig. 2. The drone package pickup and delivery mode and system.

to improve the transportation efficiency and enlarge the drone flying range.

As shown in Fig. 2, the system had multiple customers, depots, and drones. The green line indicates that the drone can deliver packages from the depot to the customer and drop packages; subsequently, the drone can return to the depot or fly to the next customer. The red line indicates that the drone can deliver a package from the depot to the customer and drop the package; if the customer needs a package pickup service, the drone will pick up the package from the customer and return to the depot. The orange line indicates that the drone can fly to the customer to pick up the package and return it to the depot. The problem can be regarded as an integrated scheduling problem of m -drones, m -depots, and m -customers considering drop-pickup.

During the drone delivery process, drones can encounter some obstacles in an urban environment. For obstacle avoidance in drones, we can refer to the method of Liu et al. [46], who proposed an autonomous path-planning method. When a drone encounters an obstacle, the method generates two paths between two points based on the tangent intersection and target-guidance strategy. Finally, one of the paths can be selected according to heuristic rules (such as distance and obstacle avoidance conditions).

The assumptions made in this study are as follows:

- (1) The packages transported by the drones were packed in special boxes of uniform size.
- (2) The charging times of the drones were not considered.
- (3) The drone would fly directly to the customer without a detour.
- (4) The drone could carry only one package each time.
- (5) The drones could fly at a constant speed without considering the energy consumption of takeoff and landing.

B. Parameters

Basic element description: All authentic tasks belonged to set $C = \{1, 2, \dots, c\}$. To distinguish three different task types: package pickup only, package delivery only, and package pickup and delivery simultaneously; C was divided into three sets: $DROP$, $PICKUP$, PD . $DROP$ denotes the set of tasks that require only the package delivery service. $PICKUP$ denotes the set of tasks that require only the package pickup service. PD denotes the set of tasks that require the package pickup and delivery services concurrently. Each task in the PD included a pickup service and delivery service. We assumed that

TABLE II
DESCRIPTION OF MAIN NOTATIONS

Notations	Description
B	A set of depots, $B = 1, 2, \dots, b$
U	A set of drones, $U = 1, 2, \dots, u$
C	A set of tasks, $C = 1, 2, \dots, c$
m	Index of depots
k	Index of drones
T_i	Pickup service of task i
T_{i+c}	Delivery service of task i
$C_{i,j}$	The payload weight of drones flying from point i to point j
C_{max}	Maximum capacity of drones
$h_{i,j}$	Maximum flying range of drones carrying payload $C_{i,j}$
h	Maximum flying range of drones
$d_{i,j}$	The distance between point i, j

package pickup and delivery must be done simultaneously for tasks in PD . These two services often have different payloads, which are related to the value of the maximum drone flight range. To distinguish between the two different services in the same task, we factorized each authentic task $i \in C$ into two virtual tasks: $i \in C^{pick}$ and $i + c \in C^{drop}$. The virtual task $i + c \in C^{drop}$ was used to represent whether authentic task i required a delivery service. Task $i \in C^{pick}$ was used to represent whether authentic task i required a pick service. The primary notations used in this study are listed in Table II.

We defined a binary variable $x_{i,j}^{m,k}$, which denotes whether drone k in depot m flies from point i to point j . If drone k in depot m flies from point i to point j , $x_{i,j}^{m,k} = 1$. Otherwise, $x_{i,j}^{m,k} = 0$.

We defined an integer variable T_i to denote the task type of task i . If task i required the package delivery service only, then $T_i = 0$ and $T_{i+c} = -1$; if task i required the package pickup service only, $T_i = 1$ and $T_{i+c} = 0$. Furthermore, $T_i = 0 / T_{i+c} = 0$ indicates that task i did not require package pickup/delivery.

We defined $\beta(C_{i,j})$ as the payload penalty factor of drones when they fly from point i to point j . Based on the literature [13], $\beta(C_{i,j})$ can be formulated as follows:

$$\beta(C_{i,j}) = \frac{\beta_{max} - 1}{C_{max}} C_{i,j} + 1 \quad (1)$$

where β_{max} is the maximum of $\beta(C_{i,j})$, and $\beta(C_{i,j})$ increases linearly with increasing payload weight. When the drone was empty, $\beta(C_{i,j}) = 1$. Conversely, $\beta(C_{i,j}) = \beta_{max}$ when the drone was fully loaded.

Experiments have shown that the flight time of a UAV decreases linearly with an increase in payload [47], [48]. According to assumption (5), when drones fly at a constant speed without considering the energy consumption of drone takeoff and landing, the flight range of drones is equal to the flight time multiplied by the drone flight speed. Therefore, the flight range of drones decreases linearly with an increase in payload. We defined a variable $h_{i,j}$ to denote the maximum

flying range of drones carrying payload $C_{i,j}$. When the drone power was constant, the flying range of drone $h_{i,j}$ decreased linearly with an increase in payload weight $C_{i,j}$. Here, $h_{i,j}$ can be formulated as follows:

$$h_{i,j} = \frac{h}{\beta(C_{i,j})} \quad (2)$$

when the drone payload is zero, $h_{i,j}$ is equal to the maximum flying range h .

C. Model

The original problem considered in this study was the integrated scheduling of m -drones, m -depots, and m -customers constrained by drop-pickup, flying range, and payload. This can be seen as an m -drone parallel scheduling traveling salesman problem (m D-PSTSP). To find a high-quality solution, we built a model to minimize the total cost of drones, mainly including the flight costs of drones and the number of drone launch sorties [49].

The drone package delivery and pickup model can be formulated as follows:

$$\text{Minimize } f = \alpha \sum_{m \in B} \sum_{k \in U} \sum_{i \in B \cup C} \sum_{j \in B \cup C, j \neq i} d_{i,j} \cdot x_{i,j}^{m,k} + \rho \sum_{m \in B} \sum_{k \in U} \sum_{i \in B} \sum_{j \in C} x_{i,j}^{m,k} \quad (3)$$

$$x_{i,j}^{m,k} \in \{0, 1\}, \forall m \in B, k \in U, i, j \in B \cup C \quad (4)$$

$$T_i \in \{0, 1\}, \forall i \in C^{\text{pick}} \quad (5)$$

$$T_{i+c} \in \{0, -1\}, \forall i + c \in C^{\text{drop}} \quad (6)$$

$$T_i + T_{i+c} = -1, \forall i \in \text{DROP} \quad (7)$$

$$T_i + T_{i+c} = 1, \forall i \in \text{PICKUP} \quad (8)$$

$$T_i + T_{i+c} = 0, \forall i \in \text{PD}, T_i \neq T_{i+c} \quad (9)$$

$$\sum_{j \in B \cup C} x_{m,j}^{m,k} = 1, \forall m \in B, k \in U \quad (10)$$

$$\sum_{i \in B \cup C} x_{i,m}^{m,k} = 1, \forall m \in B, k \in U \quad (11)$$

$$\sum_{m \in B} \sum_{k \in U} x_{i,j}^{m,k} = 1, \forall i \in \text{DROP}, j \in \text{PICKUP} \cup B \quad (12)$$

$$\sum_{m \in B} \sum_{k \in U} \sum_{j \in B} x_{i,j}^{m,k} = 1, \forall i \in \text{PICKUP} \cup \text{PD} \quad (13)$$

$$\sum_{m \in B} \sum_{k \in U} \sum_{i \in B} x_{i,j}^{m,k} = 1, \forall j \in \text{DROP} \cup \text{PD} \quad (14)$$

$$x_{i,j}^{m,k} \cdot d_{i,j} \leq h_{i,j}, \forall i, j \in B \cup C, i \neq j, m \in B, k \in U \quad (15)$$

$$\sum_{m \in B} \sum_{k \in U} \sum_{j \in B \cup C, j \neq i} x_{i,j}^{m,k} = 1, \forall i \in C \quad (16)$$

$$\sum_{m \in B} \sum_{k \in U} \sum_{i \in B \cup C, i \neq j} x_{i,j}^{m,k} = 1, \forall j \in C \quad (17)$$

$$\sum_{m \in B} \sum_{k \in U} x_{i,i}^{m,k} = 0, \forall i \in C \quad (18)$$

In our model, the objective function f (Eq. (3)) aimed to minimize the total cost of drones. The first part of f minimized the travel cost of drones, as the longer the drone flies, the more energy it consumes. The second part of f minimized the number of drone launch sorties. Additionally, α and ρ ($\alpha, \rho \in [0, 1], \alpha + \rho = 1$) were the weight coefficients of the two parts [39].

The domain of decision variables are prescribed in Eq. (4). Eq. (5)-Eq. (9) prescribes the type of task i . Eq. (10)-Eq. (11) implies that the drone must start from the depot and finally return to the depot after completing all its scheduled tasks. Eq. (12)-Eq. (14) prescribes constraints on tasks. Here, Eq. (12) indicates that after the drone completes the delivery task, it can choose to go to the next task with the demand for package pickup or return to the nearest depot. Furthermore, Eq. (13) indicates that the drone returns to the nearest depot after completing the package pickup task, because the drone can only carry one package at a time, and Eq. (14) indicates that drones must complete package delivery from depots. Eq. (15) represents the maximum flying range constraint, which indicates that the maximum fly range of drones carrying a payload should be higher than the distance between points i and j . Eq. (16)-Eq. (17) indicates that each task should be completed by one drone only. Eq. (18) indicates that drones cannot go from task i to task i .

IV. TWO-PHASE OPTIMIZATION APPROACH

The integrated scheduling of m -drones, m -depots, and m -customers constrained by drop-pickup, flying range, and payload is a complex combinatorial optimization problem. This section presents a two-phase optimization method designed to solve this type of problem, thereby improving the efficiency of obtaining a high-quality solution.

A. Algorithm Framework

With an increasing number of tasks, the complexity of the combinatorial optimization problem increases sharply. Traditional heuristic and metaheuristic algorithms cannot determine satisfactory solutions in a reasonable time. Thus, we designed the SATO-IVND algorithm to resolve the original problem. The SATO approach decomposes the original integrated scheduling problem into two phases: a task-allocation phase for m -depots and a route-planning phase for each depot. In the first phase, the original m -depots integrated scheduling problem is transformed into multiple single depot route-planning problem, and in the second phase, multiple route-planning schemes for each depot are found to complete all tasks. The two stages were performed iteratively and interactively to obtain a high-quality solution. The pseudocode for the SATO-IVND is provided in Algorithm 1.

In the first phase, we divided all tasks into three types: package pickup only, package delivery only, and package pickup and drop concurrently. After the classification of all tasks, the k -means algorithm was used to generate an initial task-allocation scheme considering the geographical location of the tasks. Subsequently, the integrated scheduling

Algorithm 1 SATO-IVND

Input: task information; distance between each pair of locations; iteration gap L

Output: route planning of all depots S ;

- 1 Initialize a task allocation scheme $G \leftarrow G_1, G_2, \dots, G_m$ for all depots by k -means algorithm;
- 2 Generate a route planning scheme S_0 based on G by ERPA;
- 3 Calculate initial the objective function value $f(S_0)$;
- 4 Let $S \leftarrow S_0$, $f(S) \leftarrow f(S_0)$;
- 5 **while** stopping criteria 1 is not satisfied **do**
- 6 **while** stopping criteria 2 is not satisfied **do**
- 7 Generate a new allocation scheme G' from S by IVND;
- 8 Generate a new route planning scheme S_0 based on G' ;
- 9 Calculate the objective function value $f(S_0)$;
- 10 **if** the conditions of Metropolis for accepting the new scheme are met **then**
- 11 $S \leftarrow S_0$, $f(S) \leftarrow f(S_0)$;
- 12 Generate a new scheduling scheme S_0 from S by LS;
- 13 Calculate the objective function value $f(S_0)$;
- 14 **if** $f(S_0)$ is superior to $f(S)$ **then**
- 15 $S \leftarrow S_0$, $f(S) \leftarrow f(S_0)$;

problem of multiple depots was transformed into a single-depot scheduling problem. Based on the derived initial task-allocation scheme, an elitist-based route planning algorithm (ERPA) was used to generate the initial route planning for each depot, considering all constraints.

In the optimization phase, we designed an IVND algorithm to adjust the allocation scheme. Simultaneously, six special neighborhood search operators (that is, 2-exchange, 3-exchange, 30%-exchange, relocation, other-relocation, and 10%-relocation) were designed in the IVND. The 2-exchange, 3-exchange, and 30%-exchange operators were used to adjust the allocation scheme in a depot. The new solution was accepted as the current solution if it satisfied the conditions of the Metropolis principles [12] are as follows:

$$P = \begin{cases} 1, & \text{if } df < 0, \\ \exp\left(-\left(\frac{df}{t}\right)\right), & \text{otherwise} \end{cases} \quad (19)$$

where $df = f(S_0) - f(S)$, $f(S_0)$ is the objective function value of the new solution, and $f(S)$ is the objective function value of the previous solution. In addition, P denotes the probability of accepting a new solution. If $df < 0$ we accept the new solution with probability 1; otherwise, we accept the new solution with probability $\exp\left(-\left(\frac{df}{t}\right)\right)$.

Additionally, the LS algorithm was used to refine the current scheme. It is noteworthy that the elitist mechanism decides whether to accept the new solution generated by the LS. The entire scheduling scheme was formed from the scheduling scheme of each depot.

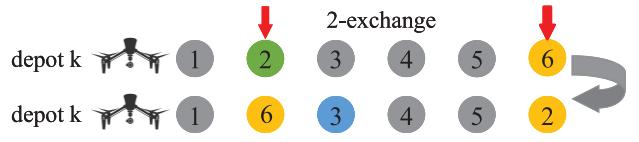


Fig. 3. 2-exchange.

The task-assignment and route-planning processes were executed until the termination condition of the algorithm was satisfied. There were two loops in the algorithm. The inner loop used the IVND algorithm to generate a new allocation scheme that satisfied various constraints. The inner loop was executed until the maximum number of iterations was satisfied. The external loop involved finding a promising scheduling scheme based on the results generated by the inner loop. The external loop was executed until the current temperature reached the lowest temperature of the SA algorithm.

B. Task Allocation

1) *Initial Task Allocation:* The k -means algorithm is one of the commonly used algorithms in the field of unsupervised learning. In particular, the k -means algorithm aims to divide the data points into multiple clusters such that the sum of squares of the distance between the sample points in each cluster and the cluster center is the smallest. To solve the original problem, considering the geographical position of each task, the k -means algorithm is used to assign tasks to different depots according to the distance between the depots and the tasks. Subsequently, each depot obtained a task-allocation scheme containing many tasks. In this manner, the original m -depot scheduling problem was partitioned into multiple single-depot scheduling problems. Specifically, the result of the initial task allocation was disordered and could not be handed over directly to the drones for execution.

2) *Neighborhood Search Operators in IVND:* Based on this allocation scheme, we proposed an IVND algorithm to obtain a satisfactory solution. Six neighborhood search operators were designed in the IVND to reallocate the task. Problem-specific operators include 2-exchange, 3-exchange, 30%-exchange, relocation, other-relocation, and 10%-relocation. The description of problem-specific operators is as follows:

a. 2-exchange: exchanges two tasks assigned to the same depot. More specifically, the operation of 2-exchange is as follows. First, we randomly choose depot k from all depots. Subsequently, two tasks i and j allocated to depot k are selected randomly ($i, j \in \text{PICK UP}$). Exchange the positions of tasks i and j . This operator is illustrated graphically in Fig. 3.

b. 3-exchange: exchanges three tasks assigned the same depot. The operator is similar to 2-exchange.

c. 30%-exchange: used where the 2-exchange and 3-exchange operators are not suitable for handling large-scale problems. We designed a 30%-exchange operator to manage allocating for large-scale tasks. The operator is similar to the 2-exchange operator: selects 30% of the tasks assigned to a depot randomly and then exchanges their positions randomly.

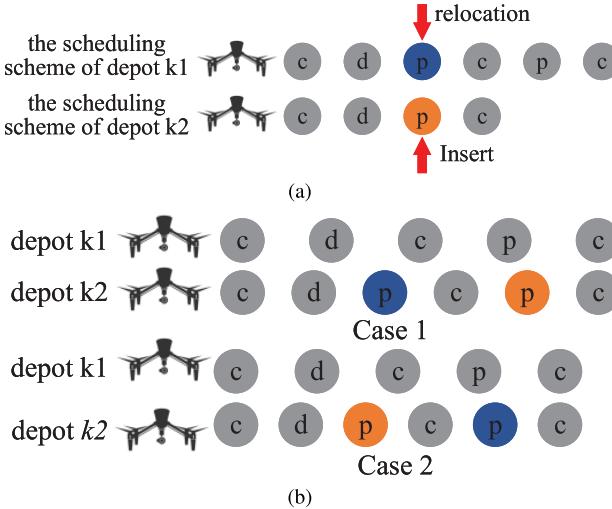


Fig. 4. Relocation. (a) Before relocation. (b) After relocation.

d). Relocation: selects task i from all tasks assigned to depot $k1$ randomly and swaps it to depot $k2$ for serving ($i \in PICKUP$). The relocation operation is as follows: Select task i from depot $k1$. An insertion point is then randomly selected in the existing allocation scheme of depot $k2$. Finally, task i is inserted into the position of the insertion point according to various constraints. This operator is illustrated graphically in Fig. 4.

5) Other-relocation: selects task i ($i \in PICKUP \cup PD$) farthest from depot $k1$ and relocates it to depot $k2$ for serving.

6) 10%-relocation: we designed a 10%-relocation operator to handle the scheduling of large-scale tasks and selected 10% of tasks assigned to depot $k1$ based on the distance between tasks and depot $k1$. We then relocated these tasks to depot $k2$ according to the constraints.

3) *Improved Variable Neighborhood Search Algorithm (IVND) for Task Reallocation:* Variable neighborhood search algorithm is an improved meta-heuristic optimization algorithm that can be used to solve combinatorial optimization problems. Here, we design an IVND algorithm to refine the allocation scheme. When the iteration time L was not reached, we inputted an initial allocation scheme G into the algorithm. Subsequently, one of the aforementioned six neighborhood search operators was chosen randomly to refine the allocation scheme G . A new solution G' could be obtained for each cycle. Subsequently, a route planning scheme S_0 was generated based on the result of the task-reallocation scheme. In addition, we calculated the objective function value $f(S_0)$. According to the Metropolis principles, we updated the value of the current local optimal solution S . The above steps were repeated until the maximum number of iterations L was reached. The pseudocode of IVND is presented in Algorithm 2.

C. Route Planning

1) *Initial Route Planning:* The initial task-allocation scheme of each depot was disordered such that it could not be directly executed by drones. Therefore, we designed an ERPA to generate an initial route-planning scheme based on the result

Algorithm 2 IVND

```

Input: allocation scheme  $G$ ; current objective function
       value  $f(S)$ ;
Output:  $G'$ ;  $S$ ;
1  $i = 1$ ;
2 while  $i \leq L$  do
3   Generate new allocation scheme  $G'$  from  $G$  via one
      of the neighborhood search operators;
4   Generate new route planning  $S_0$ ;
5   Calculate the cost of  $S_0 \rightarrow f(S_0)$ ;
6    $df = f(S_0) - f(S)$ ;
7   Generate a random number  $\varepsilon \leftarrow rand(0, 1)$ 
8   if  $df < 0$  then
9      $S \leftarrow S_0, G \leftarrow G', f(s) \leftarrow f(S_0)$ ;
10  else if  $exp(-df/t) \geq \varepsilon$  then
11     $S \leftarrow S_0, G \leftarrow G', f(s) \leftarrow f(S_0)$ ;
12   $i = i + 1$ ;

```

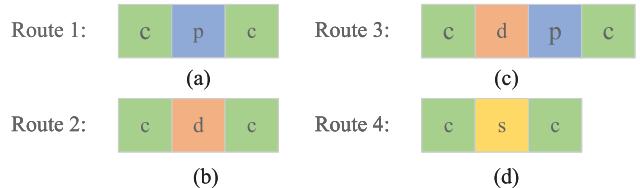


Fig. 5. Possible route of drones (c: depot, d: task which need package delivery only, p: task which need package pickup only, s: task which need package delivery and pickup).

of initial task-allocation scheme. Previously, we differentiated the tasks into three types and divided them into three sets: $DROP$, $PICKUP$, and PD .

Because a drone can deliver one package each time, there are several rules for generating route planning for drones. First, tasks that belong to the $DROP$ and PD sets would be completed using one drone to load the package from the depot and drop them to the exact location. Second, tasks that belong to the $PICKUP$ set could be assigned to the drone that has been assigned a task (the task must belong to the $DROP$ set) or a new drone for completion.

According to the abovementioned rules, the route-planning scheme of drones has the following rules: First, the drone must start from a depot and finally return to the depot after completing all its tasks. Second, after the drone completes a task in the $DROP$, it can choose to return to the depot directly or go to the next task with the demand for package pickup. Finally, because the drone can carry only one package each time, it must return to the depot after picking up the package. Thus, there are four possible routes for drones (Fig. 5).

(1) The drone starts from the depot to complete the task of picking up the package and then returns to the depot (Fig. 5(a)).

(2) The drone starts from a depot to complete the task of delivering the package and then returns to the depot (Fig. 5(b)).

(3) The drone starts from the depot to complete the task of dropping the package, completes the task of picking up the package, and finally, it returns to the depot (see Fig. 5(c)).

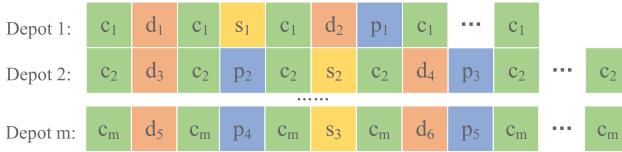


Fig. 6. A route planning scheme of depots.

(4) If task i has the requirement of both package delivery and pickup, the drone starts from the depot for dropping and reloading packages. Subsequently, the drone returns to the depot with the package (Fig. 5(d)).

In this study, each depot and task had a unique index to distinguish between depots and different types of tasks. We indexed tasks first. We then indexed the depots based on the index of the tasks. For example, if we assume that the number of tasks is c and the number of depots is b , then the indexes of the depots are $c+1, c+2, \dots, c+m$. As shown in Fig. 6, the route planning of each depot can be freely combined using the above drone routes. The route-planning scheme of each depot was combined to form a complete scheduling scheme.

Algorithm 3 Elitist-Based Route Planning Algorithm (ERPA)

```

Input: task allocation schemes of depot  $m$   $G_m$ ; the
       number of depots  $b$ ; the number of tasks  $c$ 
Output: the initial route planning  $S_0$ ;
1 for  $m = 1 : b$  do
2   Rearrange sequences of tasks in  $G_m \rightarrow P_m$ ;
3   for  $j = 1 : \text{the number of tasks in } P_m$  do
4     if task  $j \in DROP \cup PD$  then
5       Arrange a drone to complete task  $j$ ;
6     else if task  $j \in PICKUP$  then
7       if task  $i \in DROP$  then
8         Generate a random number
9          $\varepsilon \leftarrow rand(0, 1)$ ;
10        if  $\varepsilon \leq 0.5$  then
11          Arrange a drone to complete task  $j$ ;
12        else
13          Assign task  $j$  to the drone which have
             been arranged to complete task  $i$ ;
14
15 Merge the schemes of each depot  $\rightarrow S_0$ ;

```

The ERPA was proposed to obtain an initial route-planning scheme for depots. First, for each depot k , we rearranged the sequences of tasks based on the task-allocation scheme G_k and obtained an initial scheduled task set P_k . For each task in set P_k , if task $j \in DROP \cup PD$, an unassigned drone would be used to complete task j . Otherwise, the task would be randomly assigned to an unassigned drone or a drone that has been assigned task i (task i must belong to the $DROP$ set). In addition, drones must start from depots and return to the depots after completing the tasks. We then obtained multiple drone routes, as shown in Fig. 5. The routes of all

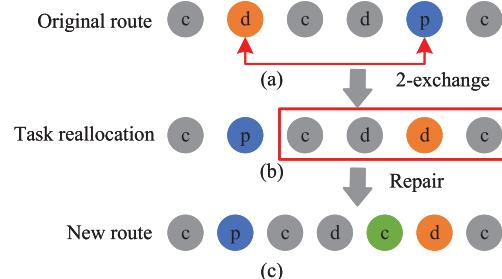


Fig. 7. The operation of repairing.

drones of each depot were connected to form the route plan of each depot. As the route-planning scheme can be regarded as a scheduling sub-scheme, we obtained the overall scheduling scheme for all depots by merging all sub-schemes. To ensure the quality of the initial solution, we constructed multiple initial solutions using the above method and then selected the best solution as S_0 . The pseudocode of ERPA is presented in Algorithm 3.

2) Adjustment of Route Planning Scheme:

a) *Repair solutions:* Task reallocation may render the original route-planning scheme infeasible. For example, Fig. 7(a) shows an initial route-planning scheme for drones. However, after task reallocation by a 2-exchange operator, the original route-planning scheme is shown in Fig. 7(b), and the scheme is obviously infeasible for drones to perform, because a drone cannot complete two tasks that belong to the $DROP$ set at once. Therefore, after task reallocation, the route-planning scheme must be modified when the sub-route of the scheme does not conform to the four route types described in Fig. 5. In addition, the route-planning scheme should be modified to represent a combination of multiple drone routes, as shown in Fig. 5. Here, Fig. 7(c) shows the modified route-planning scheme.

b) *Local search algorithm:* To overcome the shortcomings of the IVND algorithm and obtain a promising scheduling scheme, we designed an LS algorithm. The LS refined the current scheduling scheme, and we obtained a better scheduling scheme. The operator is illustrated graphically in Fig. 8, wherein two tasks that were originally assigned to two drones are selected and reassigned to one drone to serve.

Specifically, the process of LS is as follows: we randomly select depot k , where s_{0k} is the current route-planning scheme of depot k . First, according to the task type of tasks, we add tasks in s_{0k} to sets $s_{0k} - pick$, $s_{0k} - drop$, and $s_{0k} - pd$. When sets $s_{0k} - pick$ and $s_{0k} - drop$ are not empty, for tasks in set $s_{0k} - pick$, if $drone_{ik}$ is arranged to complete task i only, then we add task i to set $temp1$. A similar operation is repeated for set $s_{0k} - drop$. Finally, we select tasks i and j from sets $temp1$ and $temp2$, respectively, and reassign the above two tasks to one drone for completion. The pseudocode of LS is presented as Algorithm 4.

V. COMPUTATIONAL EXPERIMENTS WITH OTHER HEURISTIC ALGORITHMS

We evaluated the effectiveness of the SATO-IVND algorithm by comparing it with CPLEX and six other heuristic and

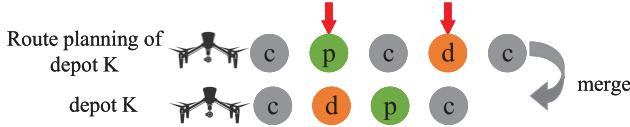


Fig. 8. The operation of LS.

Algorithm 4 local Search Algorithm(LS)

Input: route planning S_{0k} of the depot k ;
Output: S_k ;

- 1 Classify tasks in $S_{0k} \rightarrow$ sets:
 $S_{0k} - pick, S_{0k} - drop, S_{0k} - pd$;
- 2 Initialize the unscheduled sets $temp1 \leftarrow \emptyset, temp2 \leftarrow \emptyset$;
- 3 **if** $S_{0k} - pick$ and $S_{0k} - drop \neq 0$ **then**
- 4 Calculate the number of tasks in $S_{0k} - pick \rightarrow y1$;
- 5 **for** $i = 1 : y1$ **do**
- 6 **if** $drone_{ik}$ complete task i only **then**
- 7 Add i to $temp1$;
- 8 Calculate the number of tasks in $S_{0k} - DROP \rightarrow y2$;
- 9 **for** $i = 1 : y2$ **do**
- 10 **if** $drone_{jk}$ complete task j only **then**
- 11 Add j to $temp2$;
- 12 Select $m \in temp1, n \in temp2$ randomly;
- 13 Reassign task i, j to one drone to complete $\rightarrow S_k$;

meta-heuristic algorithms. In addition, we tested the SATO-IVND algorithm in real scenarios to prove its effectiveness. The proposed algorithm and the other six heuristics/meta-heuristic algorithms are coded in Python and run on a personal computer with a Core i5-8400 2.80 GHz CPU, 8 GB memory, and Windows 10 operating system. CPLEX 12.9 is chosen as the solver in MATLAB.

A. Experimental Setup

Because a few studies have focused on drop-pickup task scheduling, to the best of our knowledge, there is currently no public benchmark for this problem. Therefore, we generated twenty-two instances (C1-C22) for different tasks and depots, as presented in Table IV to Table VI. Each instance comprised five parts: they are the number of tasks (*Num-C*), number of depots (*Num-D*), type of tasks, coordinates of tasks, and weight of packages. Among them, The coordinates of nine instances (C1-C9) were developed from the well-known instances proposed by Solomon [50]. The coordinates of thirteen instances (C10-C22) were randomly distributed in a 50 km × 50 km area. The number of tasks and depots was set according to Reference [51]. The coordinates of the depots in each instance were at the center of the cluster and were calculated by k -means. We assumed that 50%, 30%, and 20% of tasks required delivery, pickup, or delivery and pickup simultaneously services, respectively. The weight of each package was a random number between 1 kg and 8 kg.

TABLE III
THE PARAMETER OF SATO-IVND

parameters	value
Initial temperature T_0	1000
Terminating temperature T_{end}	1e-07
Cooling rate q	0.93
Maximum number of iterations L	20
Flying range (km)	30

Suppose that all drones are homogenous, the relevant parameter settings are presented in Table III. The parameters of the SATO-IVND were determined from relevant literature [52] or via the trial-and-error method [53]. The weight coefficients α and ρ were set to 0.7 and 0.3. Numerous experiments were conducted in the simulation and realistic scenarios to explore the performance of the proposed SATO-IVND.

B. Comparison With CPLEX

To evaluate the optimization performance of the proposed SATO-IVND algorithm, we compared it with CPLEX on instances C1-C9. This allowed us to determine the gap between the solutions obtained by SATO-IVND and the optimal solutions obtained by CPLEX. Each instance (C1-C9) was solved using SATO-IVND 10 times to obtain reliable results. To compare the performance of CPLEX with SATO-IVND on the same instances, we calculated the gap values using the following formula:

$$gap = \frac{S_i - S_j}{S_j} \quad (20)$$

where S_i is the objective value of the solution obtained by the comparison algorithm i , S_j is the objective value of the solution obtained by the baseline algorithm j . Here, the baseline algorithm is CPLEX, and the comparison algorithm is SATO-IVND. The number of tasks in different instances was denoted as *Num-C*. The number of depots in different instances was denoted as *Num-D*. All results are presented in Table IV. C.V in Table IV is the coefficient of variation of cost, which is the ratio of the standard deviation to the mean of the cost for each instance.

The results presented in Table IV demonstrate the computational efficiency and effectiveness of the proposed SATO-IVND algorithm in comparison to CPLEX. As shown, while CPLEX was able to find the optimal solution for some problem instances, its computational time significantly increased with an increase in task scale and was deemed impractical for solving large-scale problems, with a runtime of more than 23 hours required to find the optimal solution for C9. In contrast, SATO-IVND produced near-optimal solutions with a gap of less than 0.95% compared to CPLEX's optimal solutions for each instance, all within a much shorter time. For instance, SATO-IVND was able to find a promising solution for C9 with a gap of only 0.45% compared to CPLEX, consuming only 18 seconds. It is worth noting that SATO-IVND found the optimal solution for small-scale problems

TABLE IV
COMPARISON WITH CPLEX

Instance	Num-C	Num-D	CPLEX		SATO-IVND			Gap(%)
			Cost	Time(s)	Cost	Time(s)	C.V (%)	
C1	10	2	31.73	1.96	31.73	1.98	0.00	0.00
C2	30	3	244.37	61.86	244.47	5.37	0.05	0.04
C3	40	4	284.83	335.94	286.40	7.08	0.80	0.55
C4	50	5	333.51	1272.05	336.39	8.94	0.73	0.86
C5	60	6	349.81	4044.96	353.12	10.74	0.65	0.95
C6	70	7	472.48	10732.29	474.99	12.63	0.50	0.53
C7	80	8	454.88	23163.60	457.02	13.95	0.57	0.47
C8	90	9	476.50	48049.96	479.11	16.64	0.52	0.55
C9	100	10	539.58	85943.05	542.01	18.04	0.48	0.45

TABLE V
THE RESULTS OF EACH INSTANCE GENERATED BY SATO-IVND AND OTHER SIX ALGORITHMS

Instance	Num-C	Num-D	SATO-IVND		no-IVND-LS		Rand-Allocate		no-LS		TSAM		Switch-PSO		ALNS	
			Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time
C10	40	5	57.2	6.9	66.5	0.1	59.2	7.0	58.5	6.7	69.7	10.4	60.6	34.5	58.4	18.6
C11	60	5	93.0	10.3	117.3	0.1	103.3	10.5	106.9	10.1	120.0	16.2	135.4	52.5	105.5	27.3
C12	80	5	117.9	13.2	146.7	0.1	126.3	13.6	128.5	12.8	149.3	19.9	149.8	68.4	132.3	34.7
C13	100	5	189.1	17.5	227.8	0.1	213.2	17.7	202.8	17.7	238.6	26.1	216.0	89.3	218.8	45.4
C14	150	5	215.6	26.7	275.1	0.2	261.9	26.3	261.5	26.9	281.7	40.4	472.0	132.5	256.8	70.0
C15	200	5	323.7	36.9	417.7	0.3	329.3	38	364.1	36.9	429.1	57.1	714.1	189.4	375.8	98.5
C16	40	2	82.7	6.6	103.1	0.1	82.8	6.8	90.5	6.7	109.1	10.3	94.6	32.9	93.7	17.6
C17	40	4	73.3	6.6	83.1	0.1	77.4	6.9	74.3	6.5	87.5	10.2	78.3	34.1	76.3	18.1
C18	60	3	117.2	10.5	156.5	0.1	119.8	10.4	129.9	10.3	166.4	16.1	136.7	52.8	128.3	27.3
C19	80	4	146.4	13.3	178.2	0.1	153.5	13.4	153.5	13.0	181.4	20.1	188.7	67.0	163.1	34.1
C20	100	10	100.3	17.5	118.1	0.1	114.2	17.7	106.0	17.1	112.9	27.1	130.5	91.5	119.7	46.1
C21	150	7	188.6	26.5	230.5	0.2	211.8	26.4	203.3	25.6	242.0	40.1	411.6	136.6	214.9	68.1
C22	200	10	222.1	36.9	278.8	0.3	259.0	37.7	243.1	35.8	285.8	57.6	574.8	187.3	266.1	98.9

such as C1. Furthermore, the robustness of SATO-IVND is evident from the low coefficient of variation (C.V) values shown in Table IV, indicating that SATO-IVND is well-suited for solving the proposed integrated-scheduling problem, especially for large-scale instances.

C. Comparison With Other Algorithms

Due to the excessive computational requirements of CPLEX in solving large-scale instances, it is not a suitable comparison algorithm to evaluate the performance of the SATO-IVND algorithm for larger instances. As there is no optimization algorithm directly applicable to the proposed problem, we selected TSAM [20], Switch-PSO [54], and ALNS [55], [56] and modified them as comparison algorithms to evaluate the performance of the SATO-IVND algorithm. ALNS is a well-known heuristic algorithm that we modified it by incorporating worst-remove and greedy-repair operators into it. Shift factor, swap factor and tube factor are incorporated into

Switch-PSO. The TSAM, Switch-PSO and ALNS were used to verify the performance of the proposed two-phase optimization approach (SATO-IVND) in solving the original problem.

Additionally, to determine the effectiveness of the various components of the SATO-IVND algorithm, we introduced three variants of SATO-IVND as competitors in our experiments: Rand-Allocate, which is SATO-IVND that randomly generates initial task-allocation schemes; no-IVND-LS, which is SATO-IVND without the improved variable neighborhood descent algorithm and local search algorithm; and no-LS, which is SATO-IVND without the local search algorithm. These competitors were included to provide a thorough analysis of the performance of the proposed SATO-IVND algorithm.

The SATO-IVND and the abovementioned six comparative algorithms were run 10 times to solve the thirteen instances (C10-C22). The results were compared in terms of cost and running-time. The average cost and time-consumption of the 10 experimental results are presented in Table V and Fig. 9.

TABLE VI
GAP AND C.V OF ALGORITHMS

Instance	Gap (%)						C.V(%)	
	no-IVND-LS	Rand-Allocate	no_LS	TSAM	Switch-PSO	ALNS	Rand-Allocate	SATO-IVND
C10	16.28	3.40	2.27	21.87	5.90	2.14	6.69	4.26
C11	26.13	11.02	14.89	29.04	45.58	13.43	7.85	0.75
C12	24.43	7.12	9.01	26.65	27.01	12.16	0.77	0.23
C13	20.52	12.75	7.27	26.21	14.25	15.74	3.84	0.03
C14	27.60	21.44	21.26	30.63	118.90	19.11	1.11	0.13
C15	29.05	1.73	12.50	32.57	120.62	16.12	4.15	0.82
C16	24.59	0.13	9.41	31.94	14.38	13.30	0.46	0.44
C17	13.25	5.47	1.30	19.27	6.79	3.97	2.80	1.31
C18	33.59	2.25	10.90	42.00	16.69	9.48	6.79	0.29
C19	21.72	4.87	4.88	23.89	28.88	11.42	5.99	0.10
C20	17.67	13.77	5.68	12.49	30.07	19.34	0.16	1.07
C21	22.24	12.35	7.79	28.35	118.31	13.99	4.71	0.06
C22	25.52	16.61	9.44	28.67	158.80	19.80	5.72	0.34
Min	13.25	0.13	1.30	12.49	5.90	2.14	0.16	0.03
Max	33.59	21.44	21.26	42.00	158.80	19.80	7.85	4.26
Average	23.28	8.69	8.97	27.20	54.32	13.08	3.93	0.76

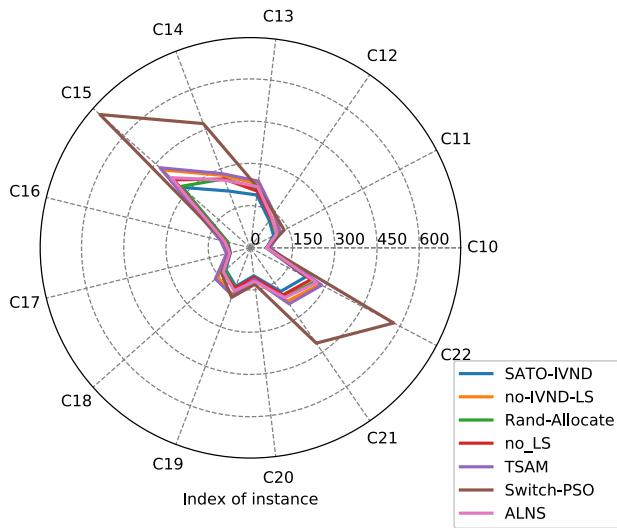


Fig. 9. Results of experimental instances generated by different algorithms.

To compare the results obtained by the other six algorithms with that obtained by the SATO-IVND further for the same instance, we used gap values to describe the differences between SATO-IVND and the other algorithms, wherein SATO-IVND was the baseline algorithm. The gap values of cost between SATO-IVND and the other six algorithms are presented in Table VI and Fig. 10.

Regarding solution quality, as observed from Fig. 9 and Fig. 10, the proposed SATO-IVND is significantly superior to the other six heuristic algorithms for all thirteen instances. Specifically, SATO-IVND, Rand-Allocate and no-LS, which are based on the two-phase optimization approach, consistently

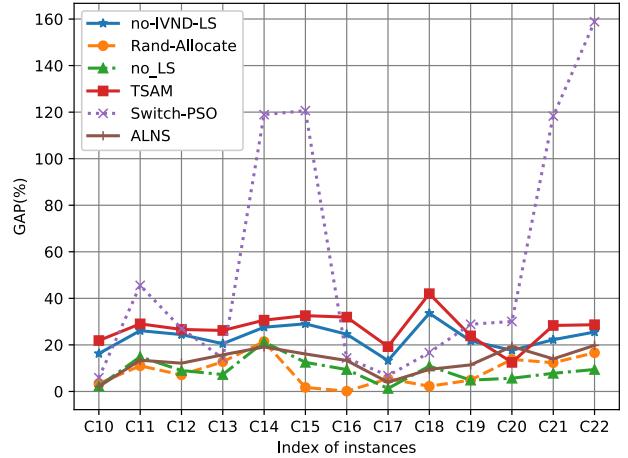


Fig. 10. The gap values of algorithms.

achieved higher quality solutions than the TSMA, Switch-PSO, and ALNS algorithms. Notably, gap values tended to increase with increasing number of tasks. Although SATO-IVND achieved results comparable to the other algorithms when the size of the tasks was small, it exhibited visible performance in solving large-scale instances.

These observations can be attributed to multiple factors. First, although TSAM, Switch-PSO and ALNS are state of the art heuristic algorithms, they are not suitable for solving the proposed problem, which need problem-specific modifications to improve their performances. However, SATO-IVND is different from these algorithms in both problem-solving framework and the exploration-exploitation strategy. On the one hand, SATO-IVND is a two-phase optimization approach which splits the original problem (integrated scheduling of

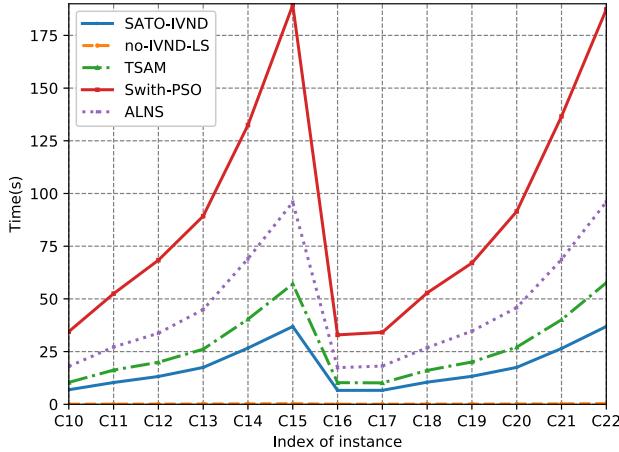


Fig. 11. The running time of algorithms for instance C10-C22.

multiple drones, multiple depots and multiple customers) into multiple single-depot scheduling problems. On the other hand, SATO-IVND relocates the worst tasks in some depots and searches for the local optimal solution of the current optimal solution at each iteration.

Second, thanks to the problem-specific operators (30%-exchange and 10%-relocation) in IVND, SATO-IVND can efficiently obtain high-quality solutions in large-scale instances. Third, although Rand-Allocate and SATO-IVND used the same two-phase optimization framework and problem search strategy, but SATO-IVND significantly outperformed Rand-Allocate. This is because the quality of the initial solution would affect the performance of our proposed algorithm. Concurrently, SATO-IVND has better performance than no-LS because the LS algorithm has strong local search capability.

In terms of time consumption, Fig. 11 shows that the no-IVND-LS algorithm had the shortest running time among the seven algorithms, followed by the SATO-IVND, Rand-Allocate and no-LS. This is because no-IVND-LS uses k -means algorithm and ERPA algorithm (in Section IV-C) to construct solutions quickly without iteratively improving them using IVND and LS. However, the quality of solutions obtained by no-IVND-LS was consistently lower than that of SATO-IVND. Therefore, no-IVND-LS may be appropriate for situations that are highly time sensitive but do not require high solution quality. Moreover, the running-time of SATO-IVND, Rand-Allocate and no-LS was similar, with solutions obtained by SATO-IVND significantly outperforming those of Rand-Allocate and no-LS. This highlights the effectiveness of the initial task-allocation mechanism and the LS algorithm in improving solution quality in a comparable amount of time. Furthermore, TSAM, Switch-PSO and ALNS cost much more computing time than the other algorithms. Finally, the running time of all algorithms increased rapidly with an increasing number of tasks, wherein Switch-PSO was the most time-consuming because of its population search mechanism.

To visually compare the robustness of SATO-IVND and the other six algorithms, instance C17 was selected to illustrate the solutions obtained by running the seven algorithms for

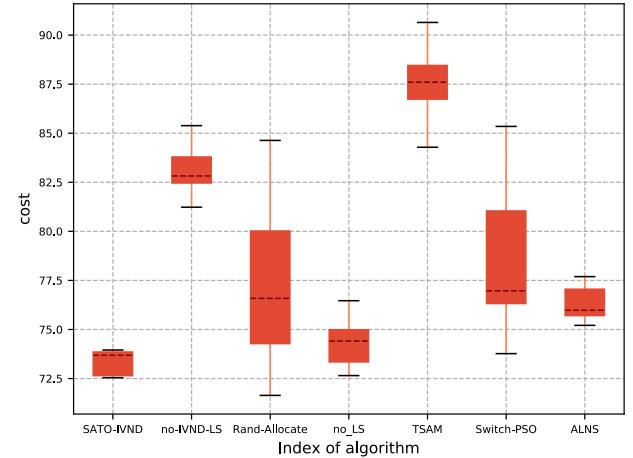


Fig. 12. Results of the experimental instance C17 generated by different algorithms.

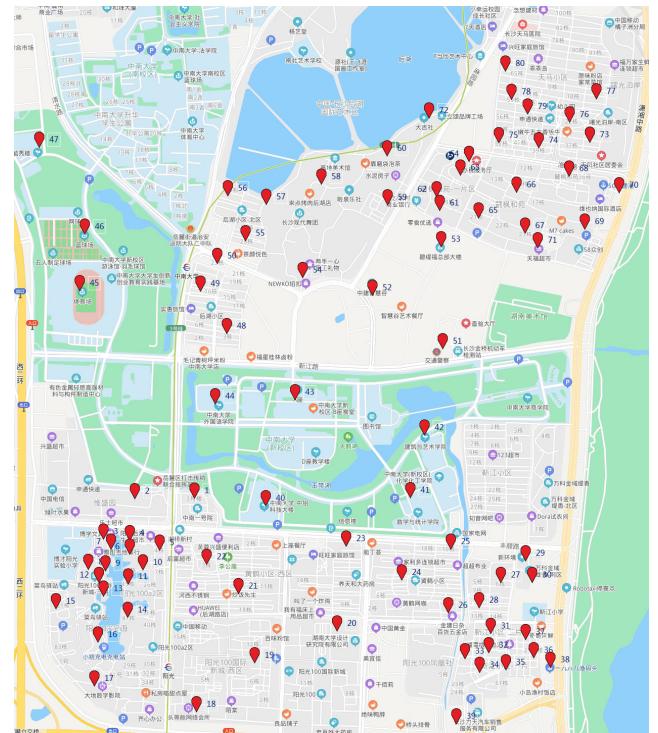


Fig. 13. Illustration of 80 tasks in Changsha city (red dot: the location of tasks).

ten times, respectively. Fig. 12 shows that the SATO-IVND is obviously superior to the other six algorithms in terms of robustness and optimization performance. Simultaneously, the solutions obtained by Rand-Allocate was relatively scattered. Although Rand-Allocate and no-LS could occasionally obtain high-quality solutions, they tended to converge to the local optima easily. Additionally, to further analyze the performance of the algorithm, the proposed SATO-IVND and Rand-Allocate was quantified using the coefficient of variation of cost (C.V.). The results are presented in Table VI. As observed from Table VI, the SATO-IVND was significantly better than that of the Rand-Allocate in all instances except C20, indicating that the initial task-allocation mechanism contributes to the robustness of SATO-IVND.

TABLE VII
SCHEDULING SCHEME GENERATED BY SATO-IVND

Depot NO.	Scheduling Scheme
01	01→44→48→01→50→01→56→55→01→57→54→01→49→43→01→47→45→01→46→01
02	02→36→31→02→24→02→27→02→20→23→02→32→28→02→33→34→02→38→35→02→37→30→02→29→02→39→02→42→25→02→26→02→41→02
03	03→52→51→03→59→03→64→65→03→72→63→03→60→03→62→03→53→03→58→03→61→03
04	04→12→8→04→22→04→17→15→04→9→11→04→13→04→2→10→04→16→14→04→7→3→04→18→04→19→04→4→04→40→5→04→1→04→6→04→21→04
05	05→67→71→05→69→70→05→73→05→76→05→78→75→05→80→05→79→74→05→77→68→05→66→05

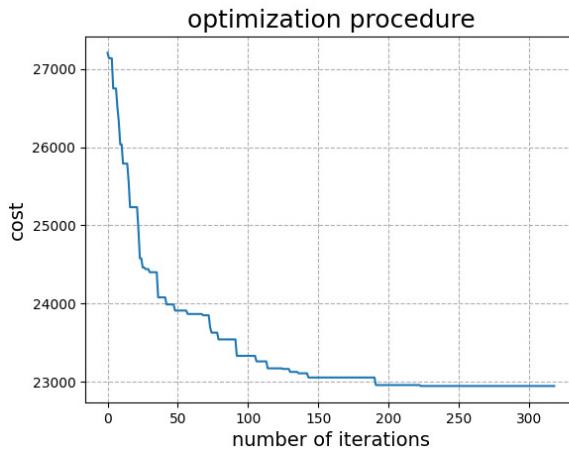


Fig. 14. Convergence curve of SATO-IVND.

In summary, SATO-IVND strikes a balance between solution quality and computational time. Concurrently, SATO-IVND has highly robustness for solving large-scale problems, which has good application prospects in solving integrated scheduling problems considering m -drones, m -depots, and m -customers constrained by drop-pickup. In contrast, the no-IVND-LS is suitable for in time-sensitive scenarios at the expense of solution quality.

D. Experiments in a Realistic Scenario

To further verify the effectiveness of the SATO-IVND in solving the integrated scheduling problem considering m -drones, m -depots, and m -customers constrained by drop-pickup, the following experiments were conducted in a realistic scenario. In this study, the related region was in the Yuelu District, Changsha City, Hunan Province. We selected 80 tasks completed by the drones, as illustrated in Fig. 13. In addition, we set up five depots in this area. The weight of each package was a random number within 1 kg-8 kg.

The scheduling scheme generated by the SATO-IVND is presented in Table VII, and the cost convergence curve for the SATO-IVND is shown in Fig. 14. The index of the depot is denoted as Depot NO. The running time was 12.91 s, and the cost of the satisfactory solution was reduced by 15.8% compared with that of the initial solution.

Here, Fig. 14 shows that the cost of the scheduling scheme generated by the SATO-IVND decreased rapidly within 50 iterations, indicating that the SATO-IVND had strong optimization ability in a short time. The Metropolis principles, IVND, and LS were applied for avoiding premature convergence of SATO-IVND. Therefore, the algorithm converged to the promising solution in 225 iterations.

VI. CONCLUSION AND FUTURE WORK

In this study, we proposed a novel drone package pickup and delivery mode and system to address last-mile distribution issues. The integrated scheduling problem of package drop-pickup considering m -drones, m -depots, and m -customers is crucial for the system. To solve this problem, we proposed a two-phase optimization approach, SATO-IVND. The proposed approach decomposed the integrated scheduling problem into two phases: task-allocation and route-planning. In the task-allocation phase, the k -means algorithm was used to generate the initial allocation scheme, while the IVND algorithm was designed to reallocate tasks using six problem-specific operators. In the route-planning phase, the ERPA algorithm was proposed to generate the initial route-planning scheme, and the LS algorithm was designed to refine the current scheduling scheme. This approach aimed to obtain a high-quality scheduling scheme through repeated iterations.

To demonstrate the effectiveness and robustness of the proposed SATO-IVND algorithm, numerous comparative experiments were conducted. The comparison experiments with CPLEX showed that the proposed SATO-IVND algorithm not only obtained the optimal solution for some small-scale problem instances (*C1*) but also generated high-quality solutions for some large-scale instances (e.g., *C9*, with gap below 0.45%) with significant computational time savings, which reflects a trade-off between solution quality and computational time. The experimental results of the SATO-IVND and other six competitors verified the superiority of the SATO-IVND. Besides, experiments on a Realistic Scenario tested the practicability of the SATO-IVND.

In the grand scheme of things, it is evident that the proposed drone package pickup and delivery mode and system has great potential to reduce last-mile operations costs and externalities due to its high degree of automation. These can help enterprises pursue efficient and environmentally friendly delivery. In comparison to traditional delivery methods, the proposed mode and system has promising application prospects for last-mile distribution in densely populated urban areas since it does not occupy ground space of communities and is safer for drones to take-off and land. Furthermore, the proposed SATO-IVND algorithm can be used to solve large-scale integrated scheduling problems in realistic scenarios owing to its promising performance and robustness.

However, several issues warrant further research. For instance, some constraints of drones, such as one drone serving multiple customers [21], may need to be relaxed. Additionally, the limitations of the SATO-IVND algorithm must be addressed to ensure a more comprehensive study. For example, the algorithm cannot handle dynamic disturbances,

such as inserting orders dynamically. Moreover, in the case of emergency, the SATO-IVND algorithm cannot adjust solutions in real-time, which should be considered in future research.

REFERENCES

- [1] T. Aized and J.-S. Srai, "Hierarchical modelling of last Mile logistic distribution system," *Int. J. Adv. Manuf. Technol.*, vol. 70, nos. 5–8, pp. 1053–1061, 2014.
- [2] CCA Network. (Aug. 2022). *China Civil UAS Development International Forum*. [Online]. Available: <http://www.caacnews.com.cn/thpd/202208/20220825.html>
- [3] D. Hao. (Jul. 2021). *Meituan Self-Developed Drones Has Been Used to Deliver Food*. [Online]. Available: <https://tech.ifeng.com/c/87mnWs梓ZOH>
- [4] A. Meola. (Jul. 2017). *Shop Online and Get Your Items Delivery by a Drone Delivery Service: The Future Amazon and Domino's Have Envisioned for US*. [Online]. Available: <https://www.businessinsider.com/delivery-drones-market-service-2017-7>
- [5] A. Hern. (Sep. 2014). *DHL Launches First Commercial Drone 'Parcelcopter' Delivery Service*. [Online]. Available: <https://www.theguardian.com/technology/2014/sep/25/german-dhl-launches-first-commercial-drone-delivery-service>
- [6] K Research Institute. (Aug. 2020). *Unmanned Delivery Field Research Report*. [Online]. Available: <http://www.199it.com/archives/1072522.html>
- [7] S Technology. (Sep. 2022). *SF UAV*. [Online]. Available: <https://www.sftech.com.cn/product/uav>
- [8] Y. Weilong, "A intelligent power exchange station and a method of intelligent power exchange for inspection drone," China, Patent CN113511102 B, Jul. 2021.
- [9] D UAV. (Mar. 2022). *How to Realize Drone Delivery*. [Online]. Available: <http://www.woiwrj.com/wurenjibaike/48044/>
- [10] M. Salama and S. Srinivas, "Joint optimization of customer location clustering and drone-based routing for last-mile deliveries," *Transp. Res. C, Emerg. Technol.*, vol. 114, pp. 620–642, May 2020.
- [11] Y. Choi and P. M. Schonfeld, "A comparison of optimized deliveries by drone and truck," *Transp. Planning Technol.*, vol. 44, no. 3, pp. 319–336, Apr. 2021.
- [12] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.
- [13] B. D. Song, K. Park, and J. Kim, "Persistent UAV delivery logistics: MILP formulation and efficient heuristic," *Comput. Ind. Eng.*, vol. 120, pp. 418–428, Jun. 2018.
- [14] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 86–109, May 2015.
- [15] P. A. Tu, N. T. Dat, and P. Q. Dung, "Traveling salesman problem with multiple drones," in *Proc. 9th Int. Symp. Inf. Commun. Technol. SoICT*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 46–53.
- [16] T. Shima, S. J. Rasmussen, A. G. Sparks, and K. M. Passino, "Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms," *Comput. Oper. Res.*, vol. 33, no. 11, pp. 3252–3269, Nov. 2006.
- [17] A. M. Ham, "Integrated scheduling of m -truck, m -drone, and m -depot constrained by time-window, drop-pickup, and m -visit using constraint programming," *Transp. Res. C, Emerg. Technol.*, vol. 91, pp. 1–14, Jun. 2018.
- [18] Y.-H. Jia et al., "A dynamic logistic dispatching system with set-based particle swarm optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1607–1621, Sep. 2018.
- [19] X. Wang, T.-M. Choi, Z. Li, and S. Shao, "An effective local search algorithm for the multidepot cumulative capacitated vehicle routing problem," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 12, pp. 4948–4958, Dec. 2020.
- [20] Y. Ancele, M. H. Hà, C. Lersteau, D. B. Matellini, and T. T. Nguyen, "Toward a more flexible VRP with pickup and delivery allowing consolidations," *Transp. Res. C, Emerg. Technol.*, vol. 128, pp. 1–21, Jul. 2021.
- [21] S. H. Chung, B. Sah, and J. Lee, "Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions," *Comput. Oper. Res.*, vol. 123, Nov. 2020, Art. no. 105004.
- [22] D. N. Das, R. Sewani, J. Wang, and M. K. Tiwari, "Synchronized truck and drone routing in package delivery logistics," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5772–5782, Sep. 2021.
- [23] X. Ren, H. Hui, Y. Shaowei, S. Feng, and L. Gongqian, "Review on vehicle-UAV combined delivery problem," *Control Decis.*, vol. 36, no. 10, pp. 2313–2327, Oct. 2021.
- [24] N. Mathew, S. L. Smith, and S. L. Waslander, "Planning paths for package delivery in heterogeneous multirobot teams," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1298–1308, Oct. 2015.
- [25] A. Karak and K. Abdelghany, "The hybrid vehicle-drone routing problem for pick-up and delivery services," *Transp. Res. C, Emerg. Technol.*, vol. 102, pp. 427–449, May 2019.
- [26] C. Wang and H. Lan, "An expressway based TSP model for vehicle delivery service coordinated with truck + UAV," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Bari, Italy, Oct. 2019, pp. 307–311.
- [27] I. Dayarian, M. Savelsbergh, and J. P. Clarke, "Same-day delivery with drone resupply," *Transp. Sci.*, vol. 54, no. 1, pp. 229–249, Jan. 2020.
- [28] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, "A clustering-based coverage path planning method for autonomous heterogeneous UAVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25546–25556, Dec. 2022.
- [29] M. Deng et al., "A two-phase coordinated planning approach for heterogeneous Earth-observation resources to monitor area targets," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 10, pp. 6388–6403, Oct. 2021.
- [30] H. Liu et al., "An iterative two-phase optimization method based on divide and conquer framework for integrated scheduling of multiple UAVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5926–5938, Sep. 2021.
- [31] S. Ma, W. Guo, R. Song, and Y. Liu, "Unsupervised learning based coordinated multi-task allocation for unmanned surface vehicles," *Neurocomputing*, vol. 420, pp. 227–245, Jan. 2021.
- [32] D. H. Lee, S. A. Zaheer, and J. H. Kim, "A resource-oriented, decentralized auction algorithm for multirobot task allocation," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1469–1481, Oct. 2015.
- [33] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "On the min-cost traveling salesman problem with drone," *Transp. Res. C, Emerg. Technol.*, vol. 86, pp. 597–621, Jan. 2018.
- [34] B. J. Moore and K. M. Passino, "Distributed task assignment for mobile agents," *IEEE Trans. Autom. Control*, vol. 52, no. 4, pp. 749–753, Apr. 2007.
- [35] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transp. Sci.*, vol. 52, no. 4, pp. 965–981, Aug. 2018.
- [36] C. C. Murray and R. Raj, "The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones," *Transp. Res. C, Emerg. Technol.*, vol. 110, pp. 368–398, Jan. 2020.
- [37] Y. Liu, Z. Liu, J. Shi, G. Wu, and W. Pedrycz, "Two-echelon routing problem for parcel delivery by cooperated truck and drone," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 12, pp. 7450–7465, Dec. 2021.
- [38] R. Roberti and M. Ruthmair, "Exact methods for the traveling salesman problem with drone," *Transp. Sci.*, vol. 55, no. 2, pp. 315–335, Mar. 2021.
- [39] Y. Han, J. Li, Z. Liu, C. Liu, and J. Tian, "Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 2, pp. 1–14, Mar. 2020.
- [40] I. Hong, M. Kuby, and A. T. Murray, "A range-restricted recharging station coverage model for drone delivery service planning," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 198–212, May 2018.
- [41] K. Peng et al., "A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery," *IEEE Access*, vol. 7, pp. 49191–49200, 2019.
- [42] L. Gao, M. Chen, Q. Chen, G. Luo, N. Zhu, and Z. Liu, "Learn to design the heuristics for vehicle routing problem," 2020, *arXiv:2002.08539*.
- [43] P. Kitjacharoenchai, B.-C. Min, and S. Lee, "Two echelon vehicle routing problem with drones in last mile delivery," *Int. J. Prod. Econ.*, vol. 225, Jul. 2020, Art. no. 107598.
- [44] H. Li, H. Wang, J. Chen, and M. Bai, "Two-echelon vehicle routing problem with time windows and mobile satellites," *Transp. Res. B, Methodol.*, vol. 138, pp. 179–201, Aug. 2020.
- [45] S. M. Fernandez, T. Harbison, T. Weber, R. Sturges, and R. Rich, "Optimization of a truck-drone in tandem delivery network using K-means and genetic algorithm," *J. Ind. Eng. Manage.*, vol. 9, no. 2, pp. 374–388, 2016.

- [46] H. Liu, X. Li, M. Fan, G. Wu, W. Pedrycz, and P. N. Suganthan, "An autonomous path planning method for unmanned aerial vehicle based on a tangent intersection and target guidance strategy," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3061–3073, Apr. 2022.
- [47] Mikroopter. (Jan. 2017). *Technical Specifications of MK8-3500 Standard*. [Online]. Available: <http://www.mikroopter.de/en/products/nmk8stden/nmk8techdaten/>
- [48] H. Y. Jeong, B. D. Song, and S. Lee, "Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones," *Int. J. Prod. Econ.*, vol. 214, pp. 220–233, Aug. 2019.
- [49] J. J. Yoon, "The traveling salesman problem with multiple drones: An optimization model for last-mile delivery," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, Jun. 2018.
- [50] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, Apr. 1987.
- [51] L. Zhen, C. Ma, K. Wang, L. Xiao, and W. Zhang, "Multi-depot multi-trip vehicle routing problem with time windows and release dates," *Transp. Res. E, Logistics Transp. Rev.*, vol. 135, Mar. 2020, Art. no. 101866.
- [52] Y. Lei, S. Feng, W. Hui, and H. Fei, *Analysis of 30 Cases of MATLAB Intelligent Algorithm*, 2nd ed. Beijing, China: Beihang Univ. Press, Aug. 2015, ch. 19, pp. 178–187.
- [53] W. Gong, Y. Wang, Z. Cai, and L. Wang, "Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 4, pp. 1499–1513, Apr. 2020.
- [54] N. A. Ab. Aziz, Z. Ibrahim, M. Mubin, S. W. Nawawi, and M. S. Mohamad, "Improving particle swarm optimization via adaptive switching asynchronous—Synchronous update," *Appl. Soft Comput.*, vol. 72, pp. 298–311, Nov. 2018.
- [55] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transp. Sci.*, vol. 40, no. 4, pp. 455–472, Nov. 2006.
- [56] R. Masson, F. Lehuédé, and O. Péton, "An adaptive large neighborhood search for the pickup and delivery problem with transfers," *Transp. Sci.*, vol. 47, no. 3, pp. 344–355, Aug. 2013.



Fangyu Hong received the B.S. degree in traffic and transportation engineering from Central South University, Changsha, China, in 2017, where she is currently pursuing the M.S. degree in traffic and transportation engineering. Her research interests include computational intelligence and urban logistics distribution.



Guohua Wu (Member, IEEE) received the B.S. degree in information systems and the Ph.D. degree in operations research from the National University of Defense Technology, China, in 2008 and 2014, respectively.

From 2012 to 2014, he was a Visiting Ph.D. Student with the University of Alberta, Edmonton, Canada. He is currently a Professor with the School of Traffic and Transportation Engineering, Central South University, Changsha, China. He has authored more than 100 refereed articles including those published in *IEEE TRANSACTIONS ON CYBERNETICS*, *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, and *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A*. His current research interests include scheduling, computational intelligence, and machine learning. He serves as an Associate Editor for *Information Sciences* and *Swarm and Evolutionary Computation*, an Editorial Board Member for *International Journal of Bio-Inspired Computation*, and the Guest Editor for *Information Sciences and Memetic Computing*.



Qizhang Luo received the B.S. and M.S. degrees in traffic and transportation engineering from Central South University, Changsha, China, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree in traffic and transportation engineering. He is a Ph.D. Visiting Student with the National University of Singapore. His current research interests include computational intelligence and scheduling, with a focus on their applications in traffic and transportation fields.



Huan Liu received the B.S. and M.S. degrees in traffic and transportation engineering from Central South University, Changsha, China, in 2018 and 2021, respectively, where she is currently pursuing the Ph.D. degree with the School of Traffic and Transportation Engineering. Her research interests include UAV path planning and task assignment.



Xiaoping Fang received the M.S. degree from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from Central South University, Changsha, China. She is currently a Professor with the School of Traffic and Transportation Engineering, Central South University. Her current research interests include enterprise strategic management and logistics economics.



Witold Pedrycz (Life Fellow, IEEE) received the M.Sc., Ph.D., and D.Sc. degrees from the Silesian University of Technology, Gliwice, Poland. He is currently a Professor and a Canada Research Chair (CRC-Computational Intelligence) with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with the Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland. His main research interests include computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering. He has published numerous papers in these areas. He is also a member of a number of editorial boards of other international journals. He is intensively involved in editorial activities. He is the Editor-in-Chief of *Information Sciences*. He serves as an Associate Editor for the *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS* and *IEEE TRANSACTIONS ON FUZZY SYSTEMS*.