Multi-Objective Multi-Drone Collaborative Routing Problem With Heterogeneous Delivery and Pickup Service

Fangyu Hong, Guohua Wu, *Senior Member, IEEE*, Yalin Wang, *Senior Member, IEEE*, Qizhang Luo, Ling Wang, *Senior Member, IEEE*, and Jianmai Shi

***Abstract*— With the development of e-commerce, the types of logistics services have become diverse. In response to the logistics requirements in urban environments, this paper introduces a logistics system that multiple drones and smart parcel lockers (SPLs) collaborate to provide package pickup, delivery and intra-city on-demand delivery services for customers. Different from pickup and delivery services, the intra-city on-demand delivery services need drones to pick up a package from a customer and deliver it to another customer. The multi-drone collaborative routing problem is crucial to find a reasonable tour over customers with flexible time-window. A multi-objective mixed-integer programming model is formulated to describe the proposed problem with simultaneously minimizing transportation costs and maximizing customer satisfaction. The model integrates dynamic energy consumption, soft time-windows, and task precedence constraints arising from the single unit capacity of drones. To tackle this problem, an adaptive-large-neighborhood-search based multi-objective algorithm (ALNSMO) is devised. CPLEX is used to verify the accuracy of the model and the quality of the proposed algorithm. Meanwhile, numerous experiments and analyses are conducted to demonstrate the superiority and practicability of the proposed mode and ALNSMO.**

***Index Terms*— Multi-drone collaborative routing problem, pickup and delivery, intra-city on-demand delivery, multi-objective optimization, multi-objective algorithm.**

I. Introduction

THE expeditious expansion of e-commerce, such as online shopping and on-demand retail [1], [2], has orchestrated a metamorphosis in logistics services and given rise to the diversification of logistics service industry [3]. The inherent attributes of drones, characterized by their adaptability, sustainability, and cost-effectiveness, have made the drone delivery system a focal point of attention within both the industrial and academic spheres in recent years. Workhorse has conceived the Horsefly system [4], a novel configuration integrating an electric-powered van with a drone to facilitate last-mile delivery. Concurrently, collaborative drones and parcel lockers delivery mode has become a promising application because of the superiority of parcel lockers in the last-mile [5], [6]. A notable instance is Meituan's exploration of the viability of collaborative drones and smart parcel lockers [7] for on-demand food delivery in specific scenarios, as illustrated in Fig. 1. Within the academic domain, there is a heightened interest in the intricacies of drone system operation, with a particular emphasis on the exploration of more robust and dependable delivery systems.

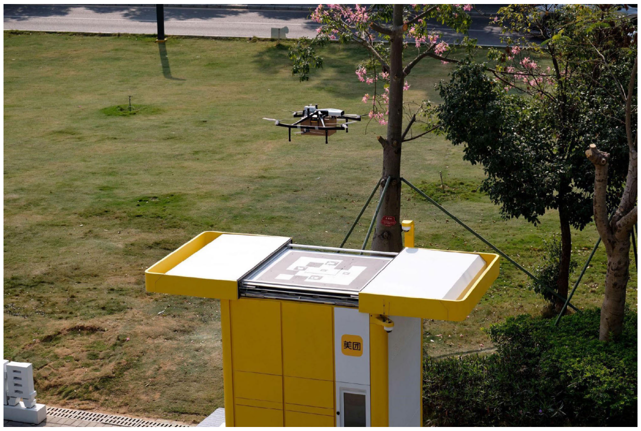


Fig. 1. Collaborative drones and smart parcel lockers of Meituan [8].

Two predominant delivery systems prevail: the truck-drone collaborative system and the multiple-drone collaborative delivery system [9]. The truck-drone collaborative routing problem was first proposed by Murray and Chu [10], with subsequent investigations introducing heightened complexities, such as integrating additional drones [11], considering energy constraints [12] and incorporating time windows [13]. Nevertheless, trucks are often affected by road traffic conditions and compounded by the inherent challenges associated with securing suitable parking spaces within densely urban environments. Therefore, the combined truck-drone system is considered more suitable for rural areas. Concomitantly, scholarly attention has been directed towards the exploration of

Received 28 December 2023; revised 1 July 2024 and 18 December 2024; accepted 2 March 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62373380 and Grant 92267205. The Associate Editor for this article was H. Jula. *(Corresponding author: Guohua Wu.)*

Fangyu Hong is with the School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China (e-mail: fangyuhong@csu.edu.cn).

Guohua Wu, Yalin Wang, and Qizhang Luo are with the School of Automation, Central South University, Changsha 410083, China (e-mail: guohuawu@csu.edu.cn; ylwang@csu.edu.cn; qz_luo@csu.edu.cn).

Ling Wang is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: wangling@mail.tsinghua.edu.cn).

Jianmai Shi is with the College of Systems Engineering, National University of Defense Technology, Changsha 410073, China (e-mail: jianmaishi@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2025.3548426, provided by the authors.

Digital Object Identifier 10.1109/TITS.2025.3548426

the multiple-drone collaborative delivery system, with a focus on the collaborative operation of drones and the development of route planning methods. For instance, Song et al. [14] studied a mode wherein drones collaborate with multiple service stations to effectuate package deliveries. Wen and Wu [15] addressed the heterogeneous multi-drone routing problem for package delivery. Compared with the truck-drone collaborative system, the multiple-drone collaborative delivery system is not susceptible to road conditions and exhibits a heightened level of automation. However, it may exist security risks when drones take off or land at customer locations.

To response to these challenges, this paper considers a collaborative system of multiple drones and smart parcel lockers for the automatic distribution of packages. Smart parcel-lockers (SPLs) are positioned on top of community buildings or in open places for package deposition, and drones transport packages between depots and these SPLs. Furthermore, customers can use SPLs to pick up or deliver by themselves or employ a mobile robot to transport packages between their location and SPLs. To enhance system efficiency, drones are allowed to take off, land and replace batteries at SPLs. Particularly, SPLs can be installed on any idle areas (such as tops of buildings) and bring benefits for carriers because of their small volumes and enhanced convenience [5]. Since the entire delivery/pickup process of drones placed on the SPLs, this system has small security risks and space limitations. For delivery services provided by mobile robots, we can refer to the literature [16]. This paper focuses on the routing problem encountered by drones during transit between depots and SPLs. To avoid confusion, we denote these smart parcel lockers as customer points in the following.

Improving the efficiency of the drone system is crucial for its application in the future. Due to the fact that most famous drones can only carry one package at a time, most literatures related to drone delivery suggest that drones must be returned to depots or trucks after delivery [17], [18]. However, this conventional viewpoint may overlook the latent potential of drones to provide service after delivery. Some studies have considered drone pickup and delivery services and allowed drones to pick up packages after delivery [19], [20]. This presents an opportunity for drones to cater to customers with reverse logistics requests, thereby maximizing the overall efficacy of the drone delivery system. Further, enterprises, such as Amazon, JD and Shunfeng, have realized that using drones to provide the intra-city on-demand distribution service is a promising attempt for efficiency improvement and cost reduction [21]. Diverging from conventional pickup or delivery services, intra-city on-demand delivery requires point-to-point service. This requires couriers to pick up packages from customers and deliver them to other specific customers directly. Very few studies have considered combining the above pickup and delivery services [22]. To improve the efficiency of the proposed drone system, this study integrates three types of pickup and delivery services: package delivery, package pickup and package intra-city on-demand delivery, respectively. Notably, drones can complete pickup or intra-city on-demand distribution services for customers after completing delivery service. Alternatively, they can execute pickup service after completing intra-city on-demand delivery services. This is a multi-visit problem, which has been considered by few previous studies [19], [23].

The proposed mode in this paper derives a route-planning problem for multiple depots and drones, which can be seen as a variant of the vehicle routing problem (VRP). Traditionally, optimization objectives in VRP include minimizing traveling distance [24], minimizing serving time [25], or maximizing the number of completed tasks [26]. Remarkably, for last-mile package delivery, many conflicting objectives are often involved in real-world scenarios. Generally, enterprises are motivated to deliver packages at a low cost, whereas customers often have expectations for timeliness of distribution. The failure to provide services within the expected time window may impact customer satisfaction and affect the long-term interests of enterprises [27]. Nevertheless, these two objectives usually conflict when minimizing the transportation costs and maximizing the customer satisfaction. VRP typically uses weighted sum methods to determine weights for several objectives and models them as a weighted combination objective [19]. However, measuring the weight of different objectives and understanding the relationships between them remains a challenge [18]. For these reasons, multi-objective optimization methods have become popular recently. These methods provide a number of solutions that represent the trade-offs between the objectives, rather than a single solution [27].

In light of these considerations, our work addresses a Multi-Objective Multi-Drone Collaborative Routing Problem with Delivery, Pickup, and Intra-City On-Demand Delivery Service (MDRP-DPOD), which has two objectives: minimizing travel costs and maximizing customer satisfaction. The MDRP-DPOD confronts several challenges: First, to our knowledge, this is the first study to consider the collaborative system of drones, SLPs, and mobile robots for three types of customers, wherein the routing problem is an extension of the typical NP-Hard combinatorial optimization problem [28] and becomes more difficult to obtain Pareto-optimal solutions for the two conflicting objectives. Thereby, it is necessary to design an efficient multi-objective algorithm for the specific problem. Second, the coupling relationships between three types of customers (requiring pickup, delivery, and intra-city on-demand delivery services) are significantly more complex than the previous vehicle routing problems with drones (VRP-D). Finally, the introduction of intricate constraints (e.g., flexible time windows, impact of payload on energy consumption at each point, multi-visit) makes it difficult to obtain high-quality solutions for the MDRP-DPOD.

To address the above issues, the main contributions of this paper are as follows.

- First, we consider the collaborative pickup and delivery system of drones, SLPs and mobile robots for three types of customers and introduce a multi-objective multi-drone collaborative routing problem with delivery, pickup, and intra-city on-demand delivery service (MDRP-DPOD).
- Second, the MDRP-DPOD is modeled as a multi-objective mixed-integer programming model, in which two optimization objectives are involved: minimizing

transportation costs and maximizing customer satisfaction. This model encompasses considerations for the dynamic impact of payload on energy consumption, flexible time windows, and the task precedence constraints arising from the single unit capacity of drones.

- Third, a multi-objective algorithm rooted in Adaptive Large Neighborhood Search (ALNSMO) is designed to solve this problem. In the global optimization phase, an adaptive large-scale neighborhood search strategy with six problem-specific operators is designed to improve each individual within the population. Meanwhile, an extended metropolis strategy is presented to determine whether to accept new individuals. In the local optimization phase, an adaptive Pareto local search (APLS) featuring variable neighborhood descent (VND) is introduced to improve the non-dominated solutions of the population. The collaborative exploration integrates global and local search strategies to identify a set of competitive Pareto solutions.
- Finally, extensive experimental studies are conducted using benchmark instances and an instance derived from real-world data to verify the superiority of the proposed algorithm. Moreover, CPLEX is used to identify the correctness of the established model. The comparative results with five competitors also demonstrate the effectiveness and practicability of ALNSMO in solving the proposed problem.

The remainder of this paper is organized as follows. Section II summarizes related works. Section III formally describes the MDRP-DPOD and the construction of the multi-objective MIP model. Section IV presents the proposed ALNSMO. Section V describes computational experiments. Section VI presents conclusions and future study trends.

II. Related Studies

This section presents a literature review related to the application of drones for package delivery/pickup services, organized into three different categories: (i) Package delivery/pickup services using drones. (ii) Consideration of constraints for drones. (iii) Related algorithms for multi-objective optimization problems.

A. Package Delivery/Pickup Services Using Drones

With the relaxation of regulations governing low-altitude airspace [32] and the maturity of drone-related technologies [33], [34], drones have emerged as promising tools for package distribution [35], [36]. Owing to the flexibility and low cost of drones, drones have demonstrated the potential for logistics distribution [35]. In recent years, many remarkable studies have been proposed. For example, Roberti and Ruthmair [25] studied the variants of the truck-drone collaborative delivery system and introduced some exact methods to solve these problems. Vásquez et al. [37] proposed a mixed-integer programming (MIP) formulation for the traveling salesman problem with Drone (TSP-D) and designed a Benders-type algorithm to solve it. Tamke and Buscher [38] considered a vehicle routing problem with drones and drone speed selection. Additionally, Dukkanci et al. [31] presented a post-disaster delivery problem using drones and trucks, and they developed a two-stage method to solve it.

Some studies have considered package pickup and delivery services for drones simultaneously. Luo et al. [20] introduced a truck-drone collaborative routing problem with delivery and pick-up services. Their study allowed drones to provide pick-up services for customers after finishing a delivery task. Hong et al. [30] proposed a drone delivery mode involving the collaborative pickup and delivery of packages by multiple drones and automatic devices in the last-mile. More recently, Meng et al. [19] studied a truck–drone combined pickup and delivery routing problem and they assumed that drones can carry multiple packages in a flight. However, they explored the pickup and delivery service with drones, and did not consider intra-city on-demand delivery service. Meng et al. [22] studied the one-to-one pair service by the combined truck-drone, assuming that drones can carry multiple packages in a flight, but did not consider a flexible time window.

B. Consideration of Constraints for Drones

As far as we know, many works concerning VRP-D with time window typically assumed that drones must serve each customer within the time window [19], [22]. Meng et al. [22] considered a truck–drone routing problem with time windows, which allowed drones to visit multiple customers per flight. They developed adaptive large neighbourhood search (ALNS) algorithm to find a high-quality solution with the minimum cost. However, some package deliveries may violate time windows in real life due to limited package delivery capacities or some emergency situations. Few studies have considered flexible time windows in VRP-D. For instance, Luo et al. [20] considered that the time of trucks or drones serving customers may affect the service reliability of the delivery system.

With the advancement of research, the capacity constraint of drones has been relaxed. For example, a few studies assumed that drones could carry multiple packages simultaneously [39]. Sun et al. [40] introduced a UAV-rider parallel coordinated delivery scheduling mode with multi-visit, wherein drones were allowed to deliver packages to several customers per flight. Nevertheless, prevailing literature still assumes that drones have a unit capacity, limiting them to carrying only one package at a time. This premise is grounded in the prevalent characteristics of widely-used drones [41].

Additionally, some constraints have been considered to make the model more practical. For instance, few studies have explored the possibility that drones can visit multiple customers in one flight to improve delivery efficiency [42], which has also been known as a multi-visit delivery problem [43]. Luo et al. [20] explored the potential for drones to provide pickup services after delivery. The dynamic impact of payloads on energy consumption of drones has also been considered in some works to make the model closer to real-life [22], [23], [29]. Meng et al. [22] considered extra hovering energy consumption when drones arrive before the customer's expected time window. It is worth mentioning that the previous works have not considered the above practical constraints in

TABLE I
THE ARTICLES RELATED TO DRONE PICKUP/DELIVERY

Reference	Objective	Vehicles	type of customers			constraints				Method
			P	D	P-D	capacity	multi-visit	EC	FTW	
Zhu et al (2022)[29]	minimizing flight distance	drones	×	√	×	M	√	√	×	SOA
Wen and Wu (2022)[15]	minimizing flight distance	drones	×	√	×	M	√	√	×	SOA
Gu et al (2022)[23]	minimizing total costs	truck-drone	×	√	×	M	√	√	×	SOA
Hong et al (2023)[30]	minimizing total costs	drones	√	√	×	1	√	√	×	SOA
Dukkanci et al (2023)[31]	minimizing total unsatisfied demand	truck-drone	×	√	×	1	×	×	×	SOA
Luo et al (2023)[20]	minimizing transportation costs, minimizing waiting time of vehicles, maximizing service reliability	truck-drone	√	√	×	1	√	×	√	MOA
Meng et al (2024)[22]	minimum total costs	truck-drone	√	√	√	M	√	√	×	SOA
this paper	minimizing travel cost, maximizing customer satisfaction	drones	√	√	√	1	√	√	√	MOA

the collaborative system of drones and SPLS for intra-city on-demand delivery service, such as flexible time windows and the dynamic impact of payloads on energy consumption.

C. Related Algorithms for Multi-Objective Optimization Problems

Common optimal objectives revolve around minimizing travel time [41] or costs [24]. However, the attention of decision-makers is typically multifaceted. For example, enterprises concurrently pay attention to customer satisfaction with services while also focusing on minimizing costs, even though these two objectives may conflict with each other. For multi-objective optimization problems, a prevalent strategy involves computing the weighted sum of various objectives when constructing a mathematical model. This approach can transform multi-objective optimization problems into single-objective optimization problems and then use single-objective optimization algorithm to solve it. The above approach has been employed in several studies, such as Salama and Srinivas [44], who considered minimizing the weighted sum of fixed cost of drones and traveling distance of both trucks and drones, employing K-means to solve the problem. Similarly, Dayaria et al. [26] focused on minimizing the weighted sum of traveling distance and time of both trucks and drones, developing a heuristic algorithm to solve the problem. A significant challenge arises in accurately determining the weights assigned to various objectives, as the preferences of decision-makers for these multiple objectives are difficult to ascertain in advance.

In response to the above challenges, numerous studies have adopted population-based multi-objective optimization algorithms to tackle multi-objective optimization problems. Currently, many literature have introduced population-based multi-objective optimization algorithms to address multi-objective vehicle routing problems. For example, Das et al. [45] proposed an improved ant colony optimization algorithm to solve the collaborative truck-drone route-planning problem in package delivery, aiming to minimize traveling cost and maximizing customer service level. Luo et al. [18] proposed a hybrid multi-objective algorithm that integrates multiple mechanisms, such as NSGA-II and Pareto local search, to solve the route-planning problem involving multiple drones and trucks. Duan et al. [46] proposed a multi-objective particle swarm optimization algorithm to solve the multi-objective vehicle route-planning problem with time windows. Liu et al. [47] proposed an improved evolutionary algorithm to solve multimodal multi-objective traveling salesman problems. Nevertheless, it is noteworthy that the algorithms proposed in these studies were developed for specific problems in their studies and may not effectively solve the proposed problem in this paper.

Table I lists the other relief package pickup/delivery problems using drones, as well as our problem, based on the following factors: (i) the objective function, (ii) the mode(s) of pickup/delivery, (iii) types of services: pickup (P), delivery(D), one-to-one pair delivery service (P-D), (iv) constraints: capacity of drones (drones carry one (1) or multiple (M) packages per flight), multi-visit, energy consumption constraints of drones (EC), flexible time window (FTW), (v) solution method: single-objective optimization problem (SOA) or multi-objective optimization algorithm (MOA). Compared to other related works, this paper proposes a new multi-objective multi-drone collaborative routing problem, which aims at maximizing customer satisfaction and minimizing transportation costs. Drones with unit capacity increase the efficiency of heterogeneous pickup/delivery tasks through multi-visit. Our problem also includes complex constraints, such as flexible time window and energy consumption, which distinguish our study from other studies.

III. MODEL OF THE MDRP-DPOD

A. Problem Definition

We consider a scenario where multiple depots are distributed within a city, each equipped with multiple drones. The smart parcel lockers are set in densely populated communities, functioning as repositories for package deposition. These smart parcel lockers serve as pickup/delivery points for drones and are treated as customers. Drones are allowed to take off and land at smart parcel lockers, as detailed in [8]. Drones are used to transport packages between depots and customers.

All depots belong to set $B = \{1, 2, \cdots, b\}$. The set of all customers is denoted by $C = \{1, 2, \cdots, c\}$. Three types of customers are considered in this paper: customers in C^D need package delivery service, customers in C^P need package pickup service, customers in C^{PD} need intra-city on-demand delivery service (drones need to pick up packages from customers and deliver them to the designated customers directly, instead of returning to the depot). Among them, $C^D \cup C^P \cup C^{PD} = C$. All packages transported by drones must not exceed the maximum payload of drones. In addition, drones can automatically replace their batteries on depots or smart parcel lockers. Subsequently, the replaced batteries can be charged on-site, thereby contributing to the improvement of transportation efficiency.

Fig. 2 provides an illustration of the MDRP-DPOD. Drones can start from the depot for package pickup or delivery (the orange line). Meanwhile, drones can visit multiple customers in a flight. Since a drone can carry only one package each time, drones must return to a depot after visiting a pickup point (the green line). Notably, a drone can start from any location to visit an intra-city pickup point and then visit the corresponding intra-city delivery point for package delivery. A drone can visit as many intra-city pickup & delivery points as possible within flight range (the yellow line). The assumptions in this study are as follows: (i) Each drone can carry only one package each time. (ii) The energy consumption of takeoff and landing is not considered. (iii) Drones are homogeneous, and the pick-up and drop-off operations of drones are autonomous. (iv) The service time at each point is constant and the time required for battery replacement is included in it. (v) The number/capacity of the SPLs is sufficient, and all packages can be placed in the SPLs.

All customers and depots are indexed to indicate the visiting order of drones. To distinguish between the intra-city pickup task and the intra-city delivery task for customers who need intra-city on-demand delivery service, each customer $i \in C$ is indexed into a real task $i \in T_0$ and a virtual task $i + c \in T_1$. T_0 is the real task set and T_1 is the virtual task set, wherein $T_0 \cup T_1 = T$. For each customer $i \in C^{PD}$, for instance, the real task i and the virtual task $i + c$ denote the intra-city pickup task and the intra-city delivery task, respectively, both of which need to be visited by a drone. Conversely, for each task $i \in C^P \cup C^D$, the virtual task $i + c$ will not be visited by any drones. Each customer $i \in C$ is associated with an expected service time window (w_i^a, w_i^b) and a tolerable time window (w_i^e, w_i^l). We should calculate the arrival time for the drone at each customer to determine if the customer is served within the expected time window. As each drone may return to the same depot after visiting any point $i \in T$, a virtual depot set $B_v = \{2*c+1, 2*c+2, \cdots, 2*c+b*c\}$ is added to distinguish the time when the drone arrives at the depot each time. In addition, each depot $i \in B$ is numbered as a departing virtual depot $(i + n) \in B_0$ and a returning virtual depot $(i + b + n) \in B_1$ to distinguish the start and end times of drones, wherein $n = 2*c + b*c$. The node set V includes all points contained in routes of all drones, wherein $V = T \cup B_v \cup B_0 \cup B_1$. Meanwhile, each drone has the same battery capacity, and the maximum flight range of the drone depends on its payload. The main notations are described in Table II.

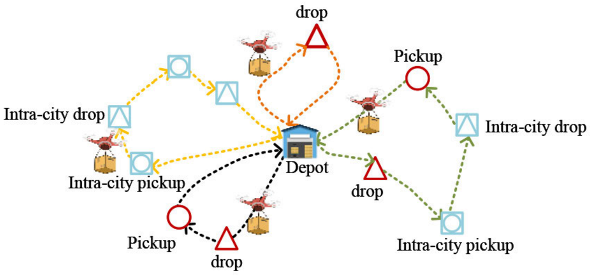


Fig. 2. An illustration of the MDRP-DPOD.

TABLE II
DESCRIPTION OF MAIN NOTATIONS

Notations	Description
B	Set of depots, $B = \{1, 2, \cdots, b\}$
U	Set of drones, $U = \{1, 2, \cdots, u\}$
C	Set of customers, $C = \{1, 2, \cdots, c\}$
T	Set of tasks, $T = \{1, 2, \cdots, 2*c\}$
V	Set of all points, $V = \{1, 2, \cdots, N\}, N = 2*b+n$
i, j, m, k	Index of nodes, depots and drones
C^D	Set of customers requiring delivery service only.
C^P	Set of customers requiring pickup service only.
C^{PD}	Set of customers requiring intra-city on-demand delivery service.
t_0	Serving time of drones at customers (unit: h).
(w_i^a, w_i^b)	Desired time window of customer i (unit: h).
(w_i^e, w_i^l)	Tolerable time window of customer i (unit: h).
E	The battery capacity of empty drones (unit: Wh).
P	Output power of drones' batteries (unit: W).
α	Battery energy consumption per distance per weight of each drone (unit: $Wh/(km \cdot kg)$)
$W_0^{m,k}$	The self-weight of the drone k in depot m (unit: kg).
$W_i^{m,k}$	Payload weight of drone k in depot m leaving point i (unit: kg).
$d_{i,j}$	The distance between point i and point j (unit: km).
$t_i^{m,k}$	Arrival time at point i of drone k in depot m (unit:$second$).
$t_{i,j}^{m,k}$	The flying time of the drone k in depot m from point i to point j.
$e_i^{m,k}$	The remaining energy of drone k in depot m when leaving point i.
t_i^{wait}	The waiting time of the drone k in depot m before serving point i.
$x_{i,j}^{m,k}$	$x_{i,j}^{m,k} = 1$ if the drone k of depot m flies from point i to point j; 0, otherwise.

B. Model

The objective of the MDRP-DPOD aims to find collaborative route-planning schemes for drones that minimizing transportation cost and maximizing customer satisfaction. The transportation cost refers to the cost associated with drones, includes the flying distance and the number of active drones. Considering the number of active drones is crucial because starting a drone each time results in depreciation costs, implying that completing the same tasks with fewer drones over the same flight distance is more cost-effective. The transportation cost of drones is calculated by the following formula:

$$\begin{aligned}\text{Minimize} f_1 = \sigma \sum_{m \in B} \sum_{k \in U} \sum_{i \in B \cup T} \sum_{j \in B \cup T, j \neq i} d_{i,j} \cdot x_{i,j}^{m,k} \\ + \rho (b \cdot u - \sum_{m \in B} \sum_{k \in U} \sum_{i \in B_0} \sum_{j \in B_1} x_{i,j}^{m,k}) \qquad (1)\end{aligned}$$

where σ and ρ are the weight coefficients of the two parts, $\sigma, \rho \in [0, 1], \sigma + \rho = 1$ [48].

In addition, in practical delivery scenarios, unforeseen challenges may arise, rendering the delivery of certain packages within the desired time window challenging. Therefore, we allow drones to serve customers outside the desired time window with a specified tolerance, which can affect customer satisfaction with the service. The customer satisfaction is related to the arrival time of a drone. If a drone serves a customer within the desired time window, customer satisfaction is 1. Conversely, if a drone fails to serve a customer within the desired time window but remains within the earliest or latest time window that the customer tolerates, customer satisfaction undergoes a reduction commensurate with the deviation from the expected time window. If the drone arrives outside of any acceptable time window, customer satisfaction is 0. The customer satisfaction is calculated by the following formula:

$$S_i^{m,k} = \begin{cases} \dfrac{t_i^{m,k} - w_i^e}{w_i^a - w_i^e}, \text{ if } w_i^e \leq t_i^{m,k} < w_i^a \\ 1, \text{ if } w_i^a \leq t_i^{m,k} < w_i^b \\ \dfrac{t_i^{m,k} - w_i^b}{w_i^l - w_i^b}, \text{ if } w_i^b \leq t_i^{m,k} < w_i^l \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

The overall customer satisfaction can be calculated by the following formula:

$$\text{Maximize} f_2 = \sum_{m \in B} \sum_{k \in U} \sum_{i \in C} S_i^{m,k} \tag{3}$$

The constraints of MDRP-DPOD are introduced as follows.

$$\sum_{j \in V} \sum_{m \in B} \sum_{k \in U} x_{i,j}^{m,k} = 1, \forall i \in C \tag{4}$$

$$\sum_{i \in V} \sum_{m \in B} \sum_{k \in U} x_{i,j}^{m,k} = 1, \forall j \in C \tag{5}$$

$$\sum_{j \in V} x_{i,j}^{m,k} - \sum_{j \in V} x_{j,i}^{m,k} = 0, \forall i \in V, k \in U, m \in B \tag{6}$$

$$\sum_{j \in V} x_{n+m,j}^{m,k} = 1, \forall k \in U, m \in B \tag{7}$$

$$\sum_{i \in V} x_{i,n+b+m}^{m,k} = 1, \forall k \in U, m \in B \tag{8}$$

$$\sum_{i \in B_0 \cup B_1} \sum_{j \in V} x_{i,j}^{m,k} - \sum_{j \in V} x_{n+m,j}^{m,k} = 0, \forall k \in U, m \in B \tag{9}$$

$$\sum_{i \in V} \sum_{j \in B_0 \cup B_1} x_{i,j}^{m,k} - \sum_{i \in V} x_{i,n+b+m}^{m,k} = 0, \forall k \in U, m \in B \tag{10}$$

$$\sum_{m \in B} \sum_{k \in U} x_{i,i}^{m,k} = 0, \forall i \in V \tag{11}$$

$$\sum_{j \in V} \sum_{m \in B} \sum_{k \in U} x_{i+c,j}^{m,k} = 0, \forall i \in C^P \cup C^D \tag{12}$$

$$\sum_{i \in V} \sum_{m \in B} \sum_{k \in U} x_{i,j+c}^{m,k} = 0, \forall j \in C^P \cup C^D \tag{13}$$

$$\sum_{j \in B_v \cup B_0 \cup B_1} \sum_{m \in B} \sum_{k \in U} x_{i,j}^{m,k} = 1, \forall i \in C^P \tag{14}$$

$$\sum_{i \in B_v \cup B_0 \cup B_1} \sum_{m \in B} \sum_{k \in U} x_{i,j}^{m,k} = 1, \forall j \in C^D \tag{15}$$

$$\sum_{j \in V} \sum_{m \in B} \sum_{k \in U} x_{i+c,j}^{m,k} = 1, \forall i \in C^{PD} \tag{16}$$

$$\sum_{i \in V} \sum_{m \in B} \sum_{k \in U} x_{i,j+c}^{m,k} = 1, \forall j \in C^{PD} \tag{17}$$

$$\sum_{m \in B} \sum_{k \in U} x_{i,i+c}^{m,k} = 1, \forall i \in C^{PD} \tag{18}$$

$$\sum_{m \in B} \sum_{k \in U} x_{i+c,i}^{m,k} = 0, \forall i \in C^{PD} \tag{19}$$

$$\sum_{i \in V} x_{i,j+c}^{m,k} - \sum_{i \in V} x_{j,i}^{m,k} = 0, \forall j \in C^{PD}, m \in B, k \in U \tag{20}$$

$$\sum_{j \in B_0 \cup B_1} \sum_{m \in B} \sum_{k \in U} x_{i,j}^{m,k} - \sum_{m \in B} \sum_{k \in U} x_{i,i+b}^{m,k} = 0, \forall i \in B_0 \tag{21}$$

$$\sum_{i \in B_0 \cup B_1} \sum_{m \in B} \sum_{k \in U} x_{i,j}^{m,k} - \sum_{m \in B} \sum_{k \in U} x_{j-b,j}^{m,k} = 0, \forall j \in B_1 \tag{22}$$

$$x_{m+n,m+b+n}^{m,k} = \begin{cases} 1, \text{ if } \sum_{i \in C} \sum_{j \in V} x_{i,j}^{m,k} = 0 \\ 0, \text{ else} \end{cases}$$
$$\forall i, j \in V, m \in B, k \in U \tag{23}$$

$$t_i^{m,k} + t_{i,j}^{m,k} + t_0 + t_j^{wait} - M \times (1 - x_{i,j}^{m,k}) \leq t_j^{m,k},$$
$$\forall i, j \in V, i \neq j, m \in B, k \in U \tag{24}$$

$$t_{i,j}^{m,k} = \frac{d_{i,j} \times (W_0^{m,k} + W_i^{m,k}) \times \alpha}{P},$$
$$\forall i, j \in V, m \in B, k \in U \tag{25}$$

$$0 \leq t_{i,j}^{m,k} \leq \frac{E}{P}, i, j \in V, m \in B, k \in U \tag{26}$$

$$e_i^{m,k} - t_{i,j}^{m,k} \times P + M \times (1 - x_{i,j}^{m,k}) \geq e_j^{m,k} \tag{27}$$

$$e_i^{m,k} = \begin{cases} E, \text{if } e_i^{m,k} \leq t_{i,j}^{m,k} \times P \times x_{i,j}^{m,k} \\ e_i^{m,k}, \text{ else} \end{cases}$$
$$\forall i, j \in V, m \in B, k \in U \tag{28}$$

$$\sum_{m \in B} \sum_{k \in U} \sum_{i \in V} x_{i,j}^{m,k} \leq 1, \forall j \in B_v \tag{29}$$

$$\sum_{m \in B} \sum_{k \in U} \sum_{j \in V} x_{i,j}^{m,k} \leq 1, \forall i \in B_v \tag{30}$$

$$\sum_{i \in V} x_{i,j+c*(m+1)}^{m,k} - x_{j,j+c*(m+1)}^{m,k} - x_{j+c,j+c*(m+1)}^{m,k} = 0,$$
$$\forall j \in C, m \in B, k \in U \tag{31}$$

$$\sum_{j \in B_v \cup B_0 \cup B_1} x_{i+c*(m+1),j}^{m,k} = 0, \forall i \in C, m \in B, k \in U \tag{32}$$

$$\sum_{i \in V} \sum_{j \in B_4} x_{i,j}^{m,k} = 0, \forall m \in B, k \in U, B_4 = B_v - B_3,$$
$$B_3 = c * (m+1) + 1, c * (m+1) + 2, \cdots, c * (m+2) \tag{33}$$

$$x_{i,j}^{m,k} \in 0, 1, \forall i, j \in V, m \in B, k \in U \tag{34}$$

$$t_i^{m,k}, t_i^{wait}, e_i^{m,k} \geq 0, \forall i \in V, m \in B, k \in U \tag{35}$$

Constraint (4)-(5) indicate that each task can only be visited exactly once by a drone. Constraint (6) indicates a flow balance constraint. Constraint (7)-(10) indicate that each drone must start from its depot and return to the depot after completing all tasks. Constraint (11) indicates that each route of drones cannot have isolated sub-loops. Constraint (12)-(13) indicate that drones need not to visit virtual task points corresponding to customers who require pickup or delivery services. Constraint (14) indicates that drones must return to the depot after completing pickup tasks. Constraint (15) indicates that drones must start from the depot for completing delivery tasks. Constraint (16)-(19) place the preorder and predecessor task of intra-city pickup/delivery tasks. Constraint (20) indicates that each intra-city pickup task and its intra-city delivery task can only be visited by a drone. Constraint (21)-(22) ensure that the drone k in depot m cannot fly to other depots. Constraint (23) means that a drone stops at the depot when it isn't assigned any tasks. Constraint (24) describes the time consumption of the drone k in depot m when it flies from point i to j.

Constraint (25)-(28) prescribe the energy consumption constraints of drones. Constraint (25) defines the time consumed by the drone from point i to j. Without considering the impact of weather, the energy consumed to fly from point i to j depends on the travel distance $d_{i,j}$ and the payload $W_i^{m,k}$ [19]. Constraint (26) ensures the flying time range of drones, that is, the flying time of drones from point i to j doesn't exceed the maximum flying time. Constraint (27) defines the energy consumed by the drone from point i to j. Constraint (28) ensures that the drone's battery is fully charged when its remaining energy is insufficient to serve the next customer, that is, the drone's battery is replaced. Constraint (29)-(30) prescribe that each virtual depot can be visited at most once. Constraint (31)-(33) place the preorder and predecessor point of virtual depots. Constraint (34)-(35) define the value range of decision variables.

IV. Solution Approach

This section proposes a multi-objective optimization algorithm based on adaptive large neighborhood search (ALNSMO). The algorithm framework, initial solution construction strategy, neighborhood search operators, adaptive pareto local search algorithm are described in detail below.

A. Algorithm Framework

Considering the complexity of the proposed MDRP-DPOD, traditional multi-objective optimization algorithms may find it difficult to solve the problem efficiently. Thereby, the ALNSMO is designed to obtain a set of satisfactory Pareto solutions in a reasonable time, enabling decision-makers to choose an appropriate solution according to their practical requirements. ALNSMO is divided into two phases: population initialization and iterative optimization. In the population initialization phase, we designed an initial solution construction strategy (INP) that includes three task allocation strategies to generate the initial population. For each individual in the initial population, all tasks are assigned to the drones in depots based on principles such as distance first, time first, and randomness. Based on the initial task-allocation scheme, an initial route-planning scheme and a scheduling time window that meet all constraints are generated by rules. In iterative optimization phase, an adaptive large-scale neighborhood search with six problem-specific operators (ALNS) is designed to reallocate tasks and adjust scheduling time-windows. Meanwhile, an adaptive Pareto local search featuring variable neighborhood descent (APLS) is proposed to search for high-quality solutions from the current Pareto Front. In each iteration, the fitness of each individual in the population is evaluated by the two objective functions and the population evolves through fast non dominated solution sorting, crowding degree, and binary tournament mechanism. Notably, the performance of the operator can also be evaluated by the objective functions, and high-performance operators are preferred to be used in each iteration. The ALNSMO is performed iteratively to obtain an approximate Pareto set. The algorithm framework of ALNSMO is presented in Algorithm 1.

First, the weight coefficients W_1, operator scores S_1, and usage count N_1 for the six operators are initialized (Line 1). The initial population M_1 is generated by the INP, detailed in Algorithm S1 of the supplementary material, wherein the fitness of each individual in the population is initialized (Lines 2-3). Subsequently, an iterative optimization process is performed on all solutions in the population M_i (lines 4-24). Fast non-dominated solution sorting and crowding distance determine the rank and crowding of each individual in M_i (lines 5-6). The binary tournament selection mechanism is used to select parent population F_i from M_i (lines 7).

Then, the ALNS with six problem-specific operators is designed to generate offspring population Q_i from the parent population F_i (line 8-18). For each individual f in F_i, an operator o_j is chosen from the six problem-specific operators o by the roulette wheel operator, generating a new individual q (line 10-11). The objective functions $f_1(q)$ and $f_2(q)$ of q are calculated to evaluate the performance of o_j (line 12). Then, the usage count $N_i(o_j)$ and score $S_i(o_j)$ of the operator o_j are updated based on the number of uses and its performance (line 13-14). Additionally, individuals closer to the Pareto Front have a higher probability of exploring high-quality solutions, wherein the operator is randomly chosen from problem-specific operators (line 15-16). Specifically, an improved metropolis strategy is introduced to determine the acceptance of the new solution (line 17). The improved metropolis strategy is as follows:

$$P = \begin{cases} 1, \text{ if } \exists df_i < 0, i = 1, 2 \\ exp(-(\sum_{i=1}^{2} df_i \times iter \times rank)) \text{ , otherwise} \end{cases} \tag{36}$$

where $df_i = f_i(q) - f_i(f)$, $f_i(q)$ is the value of objective function i for the new solution q. $f_i(f)$ is the value of objective function i for the previous solution f. P denotes the probability of accepting a new solution. If $df_i < 0$ exists, the new solution is accepted with a probability of 1. Otherwise, the new solution is accepted with a probability of

$exp(-(\sum_{i=1}^{2} df_i \times iter \times rank))$. Here, $iter$ represents the number of current iterations, and $rank$ is the rank of q in the non-dominated sort. This configuration is motivated by the observation that solutions with higher non-dominant rank exhibit an evident tendency toward the Pareto Front. Furthermore, reducing the probability of accepting low-quality solutions in later iterations contributes to ensuring that more solutions converge towards the Pareto Front.

After the offspring population Q_i is generated by the ALNS, the weights of operators W_i need to be updated (Line 18-19), which is formulated as follows [49]:

$$W_{i+1,j} = (1-\epsilon)W_{i,j} + \epsilon \frac{S_{i+1,j}}{N_{i+1,j}} \tag{37}$$

where $W_{i,j}$ represents the weight of operator j in the $i-th$ iteration. ϵ represents adjustment coefficient for weight. $S_{i+1,j}$ and $N_{i+1,j}$ represent the score and usage count of operator j in the $i+1-th$ iteration, respectively.

Then, APLS is used to search for local-optimal solutions from the current Pareto Front PF_i, detailed in Algorithm S2 of the supplementary material, wherein a simple strategy del_repeat is introduced to delete the duplicated individual from PF_i (lines 20-22). Two trigger strategies are incorporated in APLS to strike a balance between the computational cost and algorithm performance (Lines 20). The first strategy adaptively determines when to trigger APLS at each iteration, while the second strategy determines when to stop APLS. That is, if the Pareto Front shows no continuous improvement within the maximum number of iterations, the APLS wouldn't be used in subsequent iterations. The population Q_i is combined with the Pareto Front PF_i and the Elitism strategy guides the selection of the next-generation population M_{i+1} from Q_i (Lines 23-24). The pseudocode of Elitism is shown in Algorithm S3 of the supplementary material. Finally, the ALNSMO iterates repeatedly until the maximum number of iterations is met and returns the final Pareto Front PF_i.

B. Solution Representation

Since the solutions of the MDRP-DPOD need to be described in both spatial and temporal dimensions, we design two chromosome structures: chromosome 1 represents the route-planning schemes for all drones, while chromosome 2 represents the scheduling time window for all drones.

1) Route-Planning Scheme Representation: A route-planning solution is synthesized through all route-planning schemes of drones in each depot. We assume that there are three drones in a depot to provide services for thirteen customers. In Fig. 3(a), we present a route-planning solution representation for this case. The numbering rules for all points are described detailedly in part A of section III. Different types of customers are represented by different colors. Each row indicates the route of different drones. Notably, the successive cells in each row signify the permutation of nodes visited by a drone. A feasible solution must satisfy all constraints. Fig. 3(b) expresses the route of drone 3 in depot 1. First, the drone launches from the depot, serves the customer 10 who needs delivery service only, followed by serving two customers 11, 12 who need intra-city on-demand delivery service, then visiting the customer 13 who needs pickup service in sequence, finally returns to the depot. Specifically, each intra-city on-demand delivery task includes two customer nodes a and b, necessitating the drone to pick up a package from customer a and deliver it to customer b.

```
Algorithm 1 ALNSMO
Input: population size $p$, depots number $b$, drone
       number $u$, all tasks $T$, distance matrix $D$,
       maximum iteration time $iter$, historical Pareto
       Front size $e$, reward score $\theta_1$,$\theta_2$ of operators;
Output: Final Pareto Front $PF_i$;
1  initialization: weight of operators $W_1 \leftarrow 1$, score of
   operators $S_1 \leftarrow 0$, usage time of operators $N_1 \leftarrow 0$,
   $time \leftarrow 0$;
2  $M_1 \leftarrow InitialPopulation(p, T, b, u, D)$;
3  Calculate two objective function values $obj_1$ for each
   solution in $M_1$;
4  for $i = 1 : iter$ do
5      $PF_i, rank_i \leftarrow non\_dominated\_sorted(M_i, obj_i)$;
6      $Crow_i \leftarrow Crowding\_distance\_soring(M_i, obj_i)$;
7      $F_i \leftarrow tournamentSelect(M_i, rank_i, Crow_i, p)$;
8      $Q_i \leftarrow \varnothing$,$obj_0 \leftarrow \varnothing$;
9      for each solution $f$ in $F_i$ do
10         $o_j \leftarrow roulette(o, W_i)$;
11         $q \leftarrow Neighbor\_search1(o_j, f)$;
12         $obj(q) \leftarrow$ Calculate the objective function values
           $f_1(q)$, $f_2(q)$ of $q$;
13         $N_{i+1}(o_j) \leftarrow N_i(o_j) + 1$;
14         $S_{i+1}(o_j) \leftarrow$
           $update\_scroe(S_i(o_j), obj(q), \theta_1, \theta_2)$;
15         if $rank(f) \leq R\_max$ && $rand() \leq rank \times 0.1$
           then
16             $q, obj(q) \leftarrow Neighbor\_search2(o, q)$;
17         $O, obj(O) \leftarrow metropolis(q, f, obj(q), obj_i(f))$;
18         $Q_i \leftarrow Q_i \cup O$,$obj_0 \leftarrow obj_0 \cup obj(O)$;
19     $W_{i+1} \leftarrow update\_weight(W_i, S_{i+1}, N_{i+1})$;
20     if $rand > 1/i$ and $time < 10$ then
21         $PE \leftarrow del\_repeat(PF_i)$;
22         $PF_i, time \leftarrow APLS(PE, time)$;
23     $Q_{i+1} \leftarrow Q_i \cup PF_i$,$obj_0 \leftarrow obj_0 \cup obj(PF_i)$;
24     $M_{i+1}, PF_{i+1}, obj_{i+1} \leftarrow elitism(Q_{i+1}, p, obj_0)$;
```

2) Time Windows Representation: In expressing the solution for each drone, beyond the permutation of customer nodes served, the scheduling time window is integral. As shown in Fig. 4, (w_j^a, w_j^b) is the desired time window for customer j expected to be served. w_j^e and w_j^l denote the earliest time and the latest time, respectively. The service time of drone k at customer i is denoted as t_0, and $t_{i,j}^{m,k}$ is the travel time for drone k from customer i to j. Additionally, t_j^{wait} is the waiting time for drone k before serving customer j. Following the start of the time window, each drone visits all customers in sequence according to the prescribed route-planning scheme. The arrival time $t_j^{m,k}$ of drone k at customer j is determined

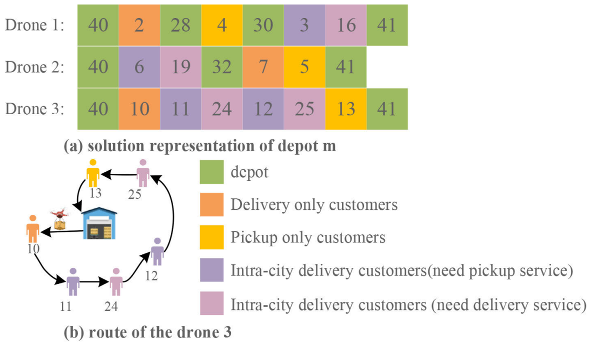

Drone 1: 40 2 28 4 30 3 16 41
Drone 2: 40 6 19 32 7 5 41
Drone 3: 40 10 11 24 12 25 13 41
(a) solution representation of depot m
13
25
10
12
11
24
depot
Delivery only customers
Pickup only customers
Intra-city delivery customers(need pickup service)
Intra-city delivery customers (need delivery service)
(b) route of the drone 3

Fig. 3. An illustration of the route-planning solution representation. (a) a part of chromosome 1 and (b) the route of drone 3.

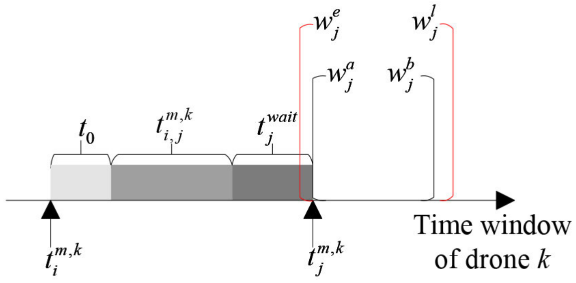


Fig. 4. An illustration of the drone's time window.

as the sum of the arrival time $t_i^{m,k}$ at the preceding customer i, the serving time t_0 for customer i, the travel time $t_{i,j}^{m,k}$ from customer i to j, and the waiting time t_j^{wait} at customer j.

C. Initial Solution Construction Strategy

The solutions posited for MDRP-DPOD in this paper encompass both the sequential order of visiting customers by each drone and the scheduling time windows for drones. The pseudocode of the initial solution construction strategy is presented in Algorithm S1 of the supplementary material. First, the initial population M_1 and the waiting time of M_1 are initialized (line 1). Subsequently, for each individual in population M_1, three heuristic rules are employed to allocate all tasks to depots, followed by the assignment of tasks within each depot to the respective drones (lines 2-8). The three heuristic rules are as follows:

a. Distance-Greedy: first, unassigned tasks in task set T are assigned to the depot closest to the task, determined by Euclidean distance. Second, tasks in each depot are sorted in ascending order according to the customer's desired time window. Finally, tasks are assigned to different drones in a sequential manner.

b. Time-Greedy: The unassigned tasks in task set T are sorted based on the customer's desired time window. Subsequently, adjacent tasks are assigned to different depots. Following the assignment of tasks to depots, each depot allocates adjacent tasks to different drones according to the order dictated by the customer's desired time window.

c. Random-Allocation: Each unassigned task in task set T is randomly assigned to a drone in depots.

It is necessary to generate a route-planning scheme for each drone to complete tasks since the task-allocation schemes are unordered. The route-planning of drones must adhere to the constraints of the model, and all route-planning schemes of drones in all depots form the overall route-planning scheme. In addition, the scheduling time window for each drone is initialized by the greedy principle (line 10-14). The construction of the scheduling time window for drones is guided by the imperative of maximizing current customer satisfaction. As shown in Fig. 4, the scheduling time window for drones is composed of the travel time, the serving time, and the waiting time before serving each customer point. Assuming that the serving time at each point is determined, the travel time between point i and j is determined by the distance between them and the payload of the drone (Eq. 25). Therefore, if the arrival time $t_j^{m,k}$ for the drone k at point j exceeds w_j^a, the waiting time t_j^{wait} before serving customer j is set to zero; therwise, t_j^{wait} equals w_j^a minus $t_j^{m,k}$.

D. Neighborhood Structures

To improve the quality of solutions, we have designed six problem-specific operators for ALNSMO:

a. o_1: This operator is designed to reorder the task execution sequence for drone k. First, we randomly select a drone k in depot m. If the number of tasks allocated to the drone k surpasses 10, a random permutation is executed on 30% of the tasks within the assigned task list for drone k. Conversely, if the number of tasks assigned to the drone k falls below 10, two tasks are randomly selected from the assigned task of drone k, and their positions are interchanged.

b. o_2: This operator is designed with the objective of altering the task order between two different drones i and j in the same depot. First, two drones in a depot are selected randomly. When the number of tasks assigned to the drone i is less than 10, one task is removed from the task list assigned to drone i. Conversely, if the number of tasks assigned to drone i exceeds 10, 30% of the tasks are removed from the task list assigned to drone i. Subsequently, the removed tasks are then added randomly to the task list of drone j.

c. o_3: This operator aims to relocate tasks between two different drones in two different depots. First, two depots are selected randomly. Then, two drones i and j are selected from the above depots respectively. Assuming that drone i is assigned more tasks than drone j. Subsequently, tasks are randomly selected from drone i and relocated to drone j for execution.

d. o_4: This operator aims to decrease the number of active drones. First, two active drones i and j are selected randomly. Then all tasks assigned to drone i are reassigned to drone j for execution, which means that drone i becomes an inactive drone since it has no allocated tasks.

e. o_5: This operator is designed to improve the customer satisfaction for a specific drone. The operation is as follows: a drone k is selected randomly from all drones. Then, its tasks are reordered in ascending order based on the desired time window of the latest customer.

f. o_6: This operator is designed with the objective of exchanging the order of pickup tasks. The operation is as follows: We randomly select a drone k in depot m. Subsequently, a task i is selected from the task list assigned to drone k randomly. If the task i is a pickup task or intra-city on-demand delivery task, it is relocated to the nearest feasible position. Otherwise, another delivery task j is chosen from the task list assigned to drone k, and the positions of tasks i and j are interchanged.

E. Adaptive Pareto Local Search Algorithm

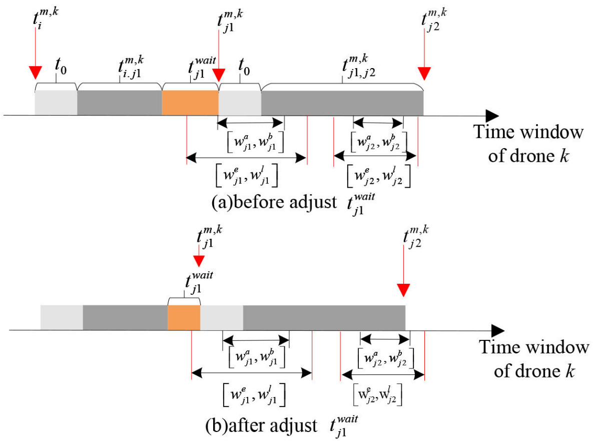
Traditional Pareto local search typically involves an exhaustive exploration of all nondominated solutions, thereby contributing significantly to computational complexity and resulting in insupportable running times. In response to this challenge, this paper introduces an adaptive Pareto local search (APLS). APLS only focuses on the non-repeating Pareto Front rather than the whole population during local search operations, aiming to enhance its search efficiency. The pseudocode of APLS is shown in Algorithm S2 of the supplementary material.

Firstly, the Pareto Front PF is initialized (line 1). For each individual $PE(i)$ in PE, a pareto local search operator based on variable neighborhood descent (PLS-VND) is devised to search for the local optimal individual pe' (lines 3-12). The operators in LS1 are sequentially used for local search on $PE(i)$. If the new solution pe' fails to dominate the preceding solution $PE(i)$, the subsequent operator is used for further local search on $PE(i)$. Otherwise, the new solution pe' is accepted and pe' is refined by LS1 sequentially. The above LS-VND is repeated until no further improvement is attainable by all operators. Additionally, another local search operator LS-wait is introduced to refine the scheduling time window of each individual in PC (lines 13-16). Finally, the new Pareto Front PF is returned. Additionally, the value of $time$ is updated based on the performance of APLS (lines 18-19).

The details of the two local search operators are as follows:

(1) LS-VND: the operator is designed to explore the current Pareto solutions iteratively by changing neighborhood operators one by one. LS1 includes five basic operators o_1, o_2, o_3, o_4, and o_5, along with two local operators, namely o_7 and o_8. The o_7 operator follows a two-step process: first, it randomly selects a drone and calculates the travel time for the drone from the previous customer to the current customer i. Second, it adjusts the position of the customer with the longest travel time. The o_8 operator randomly selects a drone and adjusts the position of the customer exhibiting the greatest discrepancy between the arrival time $t_j^{m,k}$ and the latest time the customer expects to be served, denoted as w_j^a.

(2) LS-wait: The scheduling time window, initially constructed by the greedy principle, may lead to low customer satisfaction. As exemplified in Fig. 5(a), assuming the waiting time before serving customer $j1$ is t_{j1}^{wait}, the arrival time of drone k to customer $j1$ is exactly within the customer's expected time window. However, it encounters a challenge in concurrently meeting the expected time of customer $j2$. Thus, a local search operator, LS-wait, is introduced. As shown in Fig. 5(b), by adjusting the waiting time of the preceding customer $j1$, LS-wait can ensure that the drone serves customer $j2$ within the expected time, so as to maximize the overall customer satisfaction across the entire solution.



(a)before adjust t_{j1}^{wait}

(b)after adjust t_{j1}^{wait}

Fig. 5. The operation of LS_wait.

V. Computational Experiments

This paper conducted numerous experiments to evaluate the performance of ALNSMO on different scale benchmarks. The proposed algorithm and the other compared 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.

A. Experimental Setup

1) Benchmark Instances: Due to the absence of available benchmark instances for the proposed MDRP-DPOD, we modified the well-known instances proposed by Homberger and Gehring [50] (C1_2_1) for our evaluation. Each instance consists of five parts: task ID, task type, package weight, task coordinates, and customer desired time window associated with each task. This paper extends 29 instances derived from those introduced by Homberger and Gehring [50], with task scales ranging from 20 to 80. Considering real-world conditions and drone characteristics, we divided all original coordinates in the instances by 5 and multiplied all time windows by 0.12, then rounded them. Drawing insights from Amazon's survey, which reports that 86% of its packages weigh less than 5 pounds (approximately 2.27 kilograms) [51], the weights of packages in our instances are randomly distributed within the range of $(0, 3]$ kilograms. Besides, considering the statistics from the Ministry of Transport of China [2] that intra-city on-demand delivery service accounts for 11.8% of delivery service, we assumed that 60%, 30%, and 10% of the tasks need delivery, pick-up, and intra-city on-demand delivery services, respectively. As the proposed problem allows for serving customers before or after their desired time windows, we generated a flexible time window (w_i^e, w_i^l) for each task i based on the desired time window (w_i^a, w_i^b) of the original instance [45], wherein $w_i^e = w_i^a - \eta(w_i^b - w_i^a)$, $w_i^l = w_i^b + \eta(w_i^b - w_i^a)$, and the coefficient η is

TABLE III
THE PARAMETER OF DRONES

parameters	value
Self-weight $W_0^{m,k}$ of drone k	6 kg
Battery capacity of empty drones E	504 Wh
Output power of drones' batteries P	1008 W
Serving time t_0	0.05 h
Battery energy consumption coefficient α	3.5 $Wh/(km \cdot kg)$

set to 0.2. Furthermore, all drones are homogeneous, and the traveling distance of the drone is exclusively determined by the Euclidean distance between two points. The drone parameters are set based on literature [19], as detailed in Table III.

2) Evaluation Indicators: To evaluate the performance of ALNSMO for the proposed MDRP-DPOD, two indicators, hypervolume (HV) [52] and coverage metric (C-metric) [53], are selected since the real Pareto Front is unknown. HV values represent the volume of the objective space enclosed by a solution set and a reference point. Larger HV values indicate a broader coverage of the objective space by the solution set, indicating better algorithm performance. For the normalization of solution sets obtained through algorithms, we select (1, 1) as the reference point. The normalization formula is shown as follows.

$$X' = \frac{X - min(X)}{max(X) - min(X)} \tag{38}$$

where $min(X)$ is the minimum objective value among all solutions, and $max(X)$ is the maximum objective value among all solutions. X is the objective value needed to be normalized.

C-metric can be calculated by the following formula:

$$C(E, F) = \frac{|\sum_{y \in F} \exists x \prec y, x\ inE|}{|F|} \tag{39}$$

where the numerator represents the number of solutions in set F dominated by at least one solution in set E. $|F|$ represents the size of set F. $C(E, F) = 1$ indicates that each solution in F is dominated by solutions in E. Conversely, $C(E, F) = 0$ indicates that no solution in F is dominated by solutions in E.

B. Model Validation and Analysis

Since CPLEX can only find an exact solution to a single objective optimization problem, this paper solves two objective functions of minimizing transportation cost and maximizing customer satisfaction in CPLEX, respectively, to verify the correctness of the established model. The results are summarized in Table S1 and Table S2 of the supplemental material, respectively. The results presented in Table S1 and Table S2 demonstrate the accuracy of the established model and the effectiveness of the proposed ALNSMO in comparison to CPLEX. Although CPLEX finds the optimal solution for each instance, it can be found that the solving time of CPLEX increases rapidly with the increase of the number of tasks, depots, and drones, particularly when it solves the function of maximum customer satisfaction. As shown in Table S2, CPLEX requires more than 1676 seconds to find the optimal solution with maximum customer satisfaction for n5m2k2. In contrast, the solving time of ALNSMO is stable and fast in instances of various sizes, wherein it finds the optimal solutions for small-size instances and the near-optimal solution for transportation cost of n20m2k2 with a gap of 0.31%. This verifies that ALNSMO is able to find high-quality Pareto Fronts with low transportation cost or high customer satisfaction, indicating that ALNSMO is suitable to solve the proposed MDRP-DPOD.

C. Comparison With Other Algorithms

Since CPLEX is not capable of solving multi-objective optimization problems [54], it is not a suitable comparative algorithm for evaluating the performance of the ALNSMO in generating high-quality Pareto Front. Meanwhile, since the studied problem is a new problem and no existing algorithms can be directly used for comparisons, we modified five algorithms for the problem. The first algorithm is ALNSMO-l, which doesn't use adaptive Pareto local search (APLS); the second comparison algorithm are ALNSMO-s, which doesn't adopt the extended metropolis strategy and accept a new solution only if it dominates the old one; the third comparison algorithm is ALNSMO-o, which is ALNSMO without problem-specific operators; the other components of the above three comparison algorithms are the same as ALNSMO. The fourth and fifth algorithms are state-of-the-art algorithms: ENSGA-II [55] and MDEA [56], for comparison evaluation. ENSGA-II introduces an extended non-dominated-sorting-genetic-based multi-objective optimization framework and combines a multi-dimensional local search strategy. The MDEA adopts the decomposition-based multi-objective optimization framework, which is integrated with a multi-directional search strategy. The ENSGA-II and MDEA are used to verify the performance of the proposed multi-objective optimization approach (ALNSMO) in solving the original problem. ALNSMO-l, ALNSMO-s and ALNSMO-o are used to determine the effectiveness of the main strategies of the ALNSMO.

The parameter settings of the above six algorithms are shown in Table S3 of supplemental materials. The maximum number of iterations $iter$ and the population size p for the above algorithms are the same. The other parameters of the ALNSMO and its variants were determined from relevant literature [49] or via the trial-and-error method [30]. Due to the high time complexity of the MDEA, its depth of search is limited to 1 to ensure that it can solve the problem in a reasonable time. The remaining parameters of ENSGA-II and MDEA are consistent with the original reference.

Each algorithm was independently run 15 times to address the above 29 instances. HV values are used to evaluate the diversity and convergence of the Pareto Front obtained by the above algorithms, while C-metric values are employed to compare the quality of the Pareto Front obtained by ALNSMO and other algorithms. The average HV values of the 15 experimental results is shown in Table IV. The average C-metric

TABLE IV
THE HV VALUES OF 5 ALGORITHMS ON 29 INSTANCES

Instance	ALNSMO		ALNSMO-l		ALNSMO-s		ALNSMO-o		MDEA		ENSGA-II	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
n20m2d2	**0.875**	0.05	0.787	0.05	0.872	0.06	0.854	0.04	0.793	0.08	0.849	0.09
n20m2d3	0.915	0.02	0.867	0.04	0.901	0.03	0.879	0.02	**0.925**	0.03	0.905	0.04
n20m2d4	**0.955**	0.01	0.929	0.04	0.952	0.02	0.954	0.02	0.952	0.03	0.940	0.03
n20m2d5	0.995	0.01	0.957	0.04	**0.999**	0.00	0.946	0.02	0.983	0.02	0.969	0.02
n30m2d2	0.909	0.06	0.721	0.07	**0.928**	0.05	0.817	0.03	0.741	0.08	0.794	0.07
n30m2d3	**0.918**	0.06	0.726	0.06	0.898	0.04	0.857	0.04	0.758	0.06	0.794	0.02
n30m2d4	**0.954**	0.02	0.797	0.03	0.931	0.03	0.922	0.03	0.881	0.05	0.842	0.04
n30m2d5	**0.973**	0.01	0.823	0.03	0.958	0.01	0.920	0.03	0.935	0.03	0.865	0.03
n40m2d2	**0.816**	0.10	0.626	0.08	0.793	0.10	0.795	0.06	0.659	0.06	0.664	0.05
n40m2d3	0.886	0.06	0.621	0.06	**0.911**	0.04	0.852	0.03	0.749	0.07	0.760	0.09
n40m2d4	0.911	0.05	0.680	0.06	**0.918**	0.03	0.866	0.02	0.838	0.04	0.783	0.04
n40m2d5	**0.944**	0.03	0.775	0.04	0.933	0.02	0.814	0.01	0.824	0.03	0.823	0.04
n50m2d2	**0.891**	0.06	0.663	0.06	0.538	0.04	0.845	0.05	0.649	0.08	0.607	0.07
n50m2d3	**0.878**	0.09	0.654	0.04	0.564	0.08	0.797	0.05	0.691	0.07	0.724	0.07
n50m2d4	**0.946**	0.03	0.626	0.06	0.778	0.05	0.933	0.03	0.772	0.07	0.757	0.04
n50m2d5	**0.950**	0.04	0.643	0.05	0.773	0.04	0.877	0.04	0.783	0.03	0.783	0.04
n60m2d2	**0.898**	0.07	0.677	0.06	0.578	0.08	0.837	0.04	0.704	0.07	0.656	0.10
n60m2d3	**0.905**	0.10	0.680	0.06	0.514	0.06	0.896	0.04	0.730	0.07	0.727	0.04
n60m2d4	**0.817**	0.08	0.588	0.05	0.497	0.06	0.800	0.04	0.684	0.03	0.694	0.03
n60m2d5	**0.934**	0.04	0.608	0.04	0.670	0.06	0.880	0.03	0.778	0.05	0.772	0.05
n80m2d2	**0.890**	0.05	0.632	0.05	0.872	0.08	0.745	0.06	0.539	0.07	0.514	0.05
n80m2d3	0.832	0.08	0.669	0.05	**0.882**	0.07	0.801	0.06	0.648	0.05	0.586	0.03
n80m2d4	0.858	0.05	0.635	0.06	**0.887**	0.07	0.759	0.04	0.689	0.04	0.668	0.02
n80m2d5	0.886	0.06	0.609	0.03	**0.899**	0.06	0.774	0.03	0.728	0.05	0.711	0.03
n40m3d2	**0.903**	0.07	0.692	0.03	0.883	0.04	0.865	0.03	0.627	0.06	0.672	0.06
n40m3d3	**0.944**	0.03	0.672	0.08	0.943	0.03	0.920	0.03	0.752	0.03	0.756	0.04
n40m3d4	**0.950**	0.03	0.713	0.03	0.933	0.03	0.928	0.00	0.856	0.05	0.823	0.03
n40m3d5	**0.983**	0.01	0.780	0.05	0.983	0.02	0.961	0.00	0.899	0.02	0.885	0.03
n40m4d2	0.898	0.04	0.759	0.05	0.891	0.02	**0.926**	0.03	0.664	0.03	0.667	0.07
$+, -, \approx$	-		0/29/0		9/19/1		4/25/0		0/29/0		1/28/0	

values, and computational time of the 15 experimental results are summarized in Table S4, and Table S5 of the supplemental material, respectively. Also, the Wilcoxon rank-sum test with $p = 0.05$ is implemented to verify the significant differences between the results. The best results for each instance are marked in boldface. The symbols $+$, $-$, and $\approx$ indicate that the results of an algorithm are significantly better than, worse than, and similar to the proposed ALNSMO, respectively.

In Table IV, it is evident that ALNSMO outperforms the other five comparative algorithms in terms of mean HV values, wherein ALNSMO obtains significantly better mean HV values than other competitors for most instances. For more details, ALNSMO obtains the best mean HV values on all instances compared with ALNSMO-l and ENSGA-II. Compared with ALNSMO-o and MDEA, ALNSMO obtains the best mean HV values on all instances except for the results obtained by ALNSMO-l on n40m4d2 and the results obtained by ENSGA-II on n20m2d3. Meanwhile, the mean HV values of the proposed ALNSMO are better than ALNSMO-s on 22 out of 29 test instances. According to the Wilcoxon rank-sum test, ALNSMO obtains the best HV values on 29, 19, 25, 29, 28 out of 29 instances compared with ALNSMO-l, ALNSMO-s, ALNSMO-o, MDEA and ENSGA-II. In summary, with respect to the HV, ALNSMO has better performance than ALNSMO-l, ALNSMO-s, ALNSMO-o, MDEA and ENSGA-II.

In terms of C-metric, the proposed ALNSMO also shows promising performance compared with the other five competitors. From Table S4, ALNSMO obtains the best mean C-metric values on all instances, except for the results generated by ALNSMO-s on n40m3d4, n20m2d5, n40m2d4, n40m3d5. The similar superiority of ALNSMO can also be observed in the results of Wilcoxon rank-sum test. It is noteworthy that the majority of *C(A,B)* values are closely approximating or equal to 1, while the corresponding values of *C(B,A)* are approaching or equal to 0 when compared to ALNSMO-l, ALNSMO-o, MDEA and ENSGA-II. This suggests that the Pareto Front obtained by ALNSMO tends to dominate most solutions obtained by the other four algorithms. However, it can be found that although most values of *C(ALNSMO, ALNSMO-s)* significantly outperform *C(ALNSMO-s, ALNSMO)*, some values of *C(ALNSMO, ALNSMO-s)* are close to 0.5. This indicates that a proportion of solutions obtained by ALNSMO and ALNSMO-s are non-dominated by each other.

For a direct comparison of the convergence trends among all algorithms, Fig. S1 of the supplemental material illustrates the trajectories of average HV values of all algorithms on instances n20m2k2, n30m2k2, n40m2k2, n50m2k2, n60m2k2 and n80m2k3. In Fig. S1, the proposed ALNSMO obtains the similar HV values on n20m2k2 and n30m2k2, which are consistent with the results in Table IV. For other instances, although the HV values of all algorithms are similar at the initial stage, ALNSMO exhibits superior performance to the other five algorithms after a certain number of generations. It is intuitive that other competitors show poor performance with increasing number of tasks. Besides, ALNSMO-s shows unstable performance in these instances, since it lacks the component to jump out of the local optima.

These results can be explained from various perspectives. First, despite ENSGA-II and MDEA are classic multi-objective optimization algorithms, their adaptability for efficiently exploring high-quality Pareto Fronts of the presented problem is limited. Especially, ENSGA-II tends to fall into local optima during the early stages of iterations. Hence, further modifications to the algorithm are necessary for obtaining a representative Pareto Front for specific problems. Second, the global search mechanism and the six problem-specific operators obviously improve the efficiency and quality of ALNSMO. Third, the comparison results between ALNSMO and ALNSMO-l indicate the effectiveness of the proposed APLS strategy in searching for local-optimal solutions. Further, the proposed ALNSMO leverages the extended metropolis strategy to select prominent solutions to improve the robustness of ALNSMO in terms of solution quality.

In terms of time consumption, the average running time (in seconds) on 29 instances for the five algorithms is presented in Table S5 and Fig. S2 of the supplemental material. It is noteworthy that ALNSMO-l exhibits the shortest computational time, followed by ALNSMO-o, ALNSMO, ALNSMO-s and ENSGA-II, with MDEA spending the longest computational time. This indicates that the APLS strategy of ALNSMO extends its computational time in comparison to ALNSMO-l to a certain degree. Besides, the running times of ALNSMO-o and ALNSMO-s are similar to ALNSMO, whereas the quality of Pareto solutions generated by ALNSMO surpasses those of ALNSMO-o and ALNSMO-s. Compared to MDEA or ENSGA-II, the computation time of ALNSMO falls within a moderate range. Observing the results for computational time, HV and C-metric, we can further summarize that the proposed ALNSMO can obviously improve the quality of Pareto solutions with a small time consumption.

D. Impact Analysis of Key Parameter

A series of sensitivity analyses is conducted to discern the impact of different parameters on the results. All experiments and comparisons were conducted on the instance n40m2k2, with ALNSMO running independently 15 times to solve it.

1) Impact Analysis of Algorithm Parameters: ALNSMO involves three major parameters: adjustment coefficient ϵ, operator scores θ_1 and θ_2. To evaluate the sensitivity of ALNSMO, we vary the three parameters ϵ, θ_1, and θ_2 from 0 to 1, respectively. Then ALNSMO is independently run for 15 times on n40m2k2 while maintaining other parameters constant. The experimental results are shown in Figs. S3 to S5 of the supplemental material. From Figs. S3 to S5, we can observe that the HV values exhibit fluctuations with the variation of the three parameters. However, there is a lack of linear correlation between the HV values of ALNSMO and the three parameters. Meanwhile, the running time of ALNSMO is significantly affected by the adjustment coefficient ϵ and operator scores θ_1 and θ_2. Specifically, when ϵ is set to 0.5, the running time of ALNSMO is shortest while the quality of the Pareto population is lower. Moreover, ALNSMO obtains the highest HV when θ_1 and θ_2 are set to 0.8 and 0.4, respectively, whereas the running time is higher when θ_1 and θ_2 are set to 0.5 and 0.2. Therefore, it needs to be measured simultaneously on algorithm performance and computational time when setting algorithm parameters.

2) Impact of the Number of Drones: In our study, the majority of instances are set up such that each depot is equipped with two drones. To evaluate the impact of drones on the results, we extend the instance n40m2k2 by adjusting the number of drones from 2 to 5, representing these instances as n40m2k2, n40m2k3, n40m2k4, and n40m2k5, respectively. Fig. S6 in the supplemental material depicts the distribution of the Pareto Front obtained by ALNSMO. Notably, an observable trend emerges as the number of drones increases: the approximate Pareto Front shows a decrease in transportation costs, while customer satisfaction of the Pareto Front also increases accordingly. This phenomenon is attributed to the increased drone deployment in the overall logistics system, which improves customer satisfaction and reduces transportation costs. However, for logistics companies, employing additional drones entails augmented fixed costs. Therefore, decision-makers should consider operating costs and profits simultaneously and select a suitable scheduling scheme according to the specific situation.

3) Impact of the Number of Depots: To evaluate the impact of depots on the results, we extend the instance n40m2k by varying the number of depots from 2 to 4, naming them n40m2k2, n40m3k2, and n40m4k2, respectively. Fig. S7 in the supplemental material illustrates the Pareto Front of the three instances obtained by ALNSMO. From Fig. S7, it can be seen that the customer satisfaction of the Pareto Front significantly increases as the number of depots increases, coupled with a substantial reduction in transportation costs. This observation suggests that a higher number of depots can enhance the

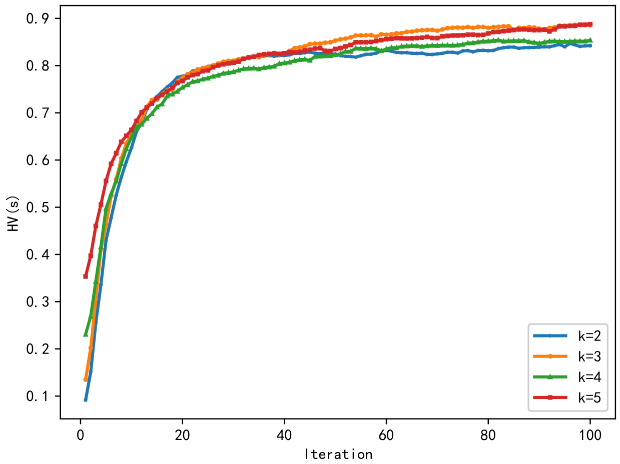
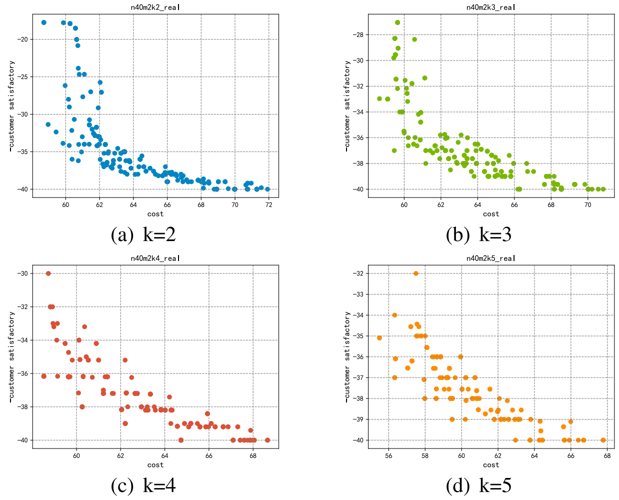


Fig. 6. The trajectories of average HV values of ALNSMO on real-world instances.



(a) k=2

(b) k=3

(c) k=4

(d) k=5

Fig. 7. The Pareto Front of ALNSMO on real-world instances.

efficiency of the entire logistics system. Nevertheless, it is crucial to acknowledge that an increase in depots incurs additional operating costs, necessitating a further evaluation of the balance between costs and profit.

E. Experiment in a Realistic Scenario

To further validate the effectiveness of the proposed model and the ALNSMO, a simulation instance based on data collected in Changsha City, Hunan Province, China was developed for experimentation. As shown in Fig. S8 of the supplemental material, this instance consists of 40 tasks located within Changsha City, Hunan Province, China. The travel distance of drones between points is measured in Euclidean distance. Besides, each task is subject to a desired time-window spanning from 8:00 am to 11:00 am, with the length of each time window randomly distributed between 20 minutes and 40 minutes. The instance includes 2 depots, each equipped with 2 to 4 drones. To avoid confusion, instances with different numbers of drones are denoted as n40m2k2_real, n40m2k3_real, n40m2k4_real, n40m2k5_real. The other parameters of drones and ALNSMO are the same as those used in the previous experiments.

Since a multi-objective algorithm generates a Pareto Front that contains a set of solutions, decision-makers have the opportunity to select an optimal solution that aligns with their preferences and to understand the distribution and trends of different objectives within the Pareto set. Especially, for better observation, the customer satisfaction values are converted to negative values. Fig. 6 illustrates the trajectories of average HV values of the ALNSMO on instances n40m2k2_real, n40m2k3_real, n40m2k4_real, n40m2k5_real. Evidently, ALNSMO exhibits a remarkable convergence performance, manifesting a rapid enhancement in the quality of the population within a mere 20 iterations. Compared with the initial population, ALNSMO increases the HV value of the population by 8.19, 5.57, 2.70, and 1.51 times, respectively. This substantiates the efficacy of ALNSMO in progressively refining the population's quality.

For a comprehensive assessment of the Pareto Front distribution, Fig. 7 visualizes the Pareto Front of the four instances. Here, the customer satisfaction is also converted to negative values for better observation. From Fig. 7, it can be seen that the approximate Pareto Front of the four instances exhibits a similar distribution, and there is a conflict between customer satisfaction and cost. The pursuit of reduced transportation costs is inherently conflicting with the goal of maximizing customer satisfaction, and vice versa. Therefore, the decision-makers are confronted with the imperative of concurrently weighing both customer satisfaction and cost, thereby necessitating a judicious selection of an appropriate scheduling scheme tailored to individual preferences.

Further, we provide visualized routing results in Fig. S9 and Fig. S10 of the supplemental material to help understand the comparison study. As shown in Fig. S9(a), it can be found that when considering the best transportation-cost solution, drones tend to complete more tasks in a flight, such that the overall transportation cost can be reduced. If the solution with the best customer-satisfaction is considered, it is observable that drones take off multiple times to complete a task in a flight, as Fig. S9(b) shows. Similar observation results can be found in Fig. S10. Meanwhile, the customer satisfaction of the best transportation-cost solution in n40m2k2_real is 17.74, and customer satisfaction of the best-transportation cost solution in n40m2k3_real is 32.97. It can be found that an increase in the number of drones can increase the lower limit of customer satisfaction for the whole population, similar observation results can be found in Fig. 7. Furthermore, we consider two optimization objectives that conflict with each other. Although the visualized solutions are based on objective-wise results, the algorithm still optimizes the other objective simultaneously.

VI. Conclusion and Future Work

This paper considers the collaborative pickup and delivery system of drones, SLPs and mobile robot for three types of customers. The multi-objective multi-drone collaborative routing problem with delivery, pickup, and intra-city on-demand delivery service (MDRP-DPOD) is crucial for the system. A mixed-integer programming model is constructed involving flexible time windows, the dynamic impact of payload on energy consumption, and the task precedence constrains derived from the single unit capacity of drones, wherein two conflicting objectives of minimizing transportation cost and maximizing customer satisfaction are included. The multi-objective optimization algorithm based on adaptive large neighborhood search (ALNSMO) is proposed for obtaining the well-distributed Pareto solutions. In global optimization phase, the adaptive large-scale neighborhood search with six problem-specific operators and the improved metropolis strategy are adopted to enhance the quality of the whole population. In local optimization phase, APLS combined with two local search strategies is implemented to exploit the current Pareto solutions. This approach aimes to obtain a number of high-quality scheduling schemes with a trade-off between transportation cost and customer satisfaction through repeated iterations.

Numerous experiments were conducted to evaluate the effectiveness of both the proposed model and algorithm. To verify the correctness of the established model and the quality of ALNSMO, CPLEX was used to solve the objective functions of minimizing transportation cost and maximizing customer satisfaction on small instances, respectively. The comparison experiments with CPLEX showed that the proposed ALNSMO can obtain the near-optimal or optimal solution for small-scale instances with significant computational time savings. The experimental results of ALNSMO and the other five comparative algorithms further verified the superiority of ALNSMO. Concurrently, a series of sensitivity analyses verified the impact of key parameters on ALNSMO. Especially, the results indicated that an increase in the number of drones or depots correlates with a reduction in transportation costs and an elevation in customer satisfaction. Furthermore, ALNSMO exhibited the capacity to generate high-quality Pareto Fronts within a reasonable time when it was applied to solve instances derived from real-world scenario data in Changsha City, Hunan Province, China. This demonstrates the great performance of ALNSMO in practical scenarios.

This paper has some limitations that need to be addressed in future research. For example, the proposed algorithm is not suitable for dynamically adjusting the route-planning scheme of drones in emergency situations. Furthermore, for minimizing installation costs and meeting the common delivery/pickup requirements of customers, it would also be interesting to consider the location and number of SPLs [57].

References

[1] Amazon. (Dec. 2023). *Free Same-Day From Stores Near You*. [Online]. Available: https://www.amazon.com/b?ie=UTF8

[2] M-Transport of China. (2023). *Operation of China's Postal Industry in 2022*. [Online]. Available: https://www.spb.gov.cn/gjyzj/c200039/202304/e4a775da6c074576a40f131e1330d91d.shtml

[3] A. M. Arslan, N. Agatz, and M. A. Klapp, "Operational strategies for on-demand personal shopper services," *Transp. Res. C, Emerg. Technol.*, vol. 130, Sep. 2021, Art. no. 103320.

[4] Workhorse. (Dec. 2021). *Meet Our Fleet*. [Online]. Available: https://workhorse.com/

[5] S. S. Mohri, N. Nassir, R. G. Thompson, and H. Ghaderi, "Last-mile logistics with on-premises parcel lockers: Who are the real beneficiaries?" *Transp. Res. E, Logistics Transp. Rev.*, vol. 183, Mar. 2024, Art. no. 103458.

[6] X. Zang, L. Jiang, C. Liang, and X. Fang, "Coordinated home and locker deliveries: An exact approach for the urban delivery problem with conflicting time windows," *Transp. Res. E, Logistics Transp. Rev.*, vol. 177, Sep. 2023, Art. no. 103228.

[7] Y. Weilong, "A intelligent power exchange station and a method of intelligent power exchange for inspection drone," China Patent 1 13 511 102B, Jul. 3, 2021.

[8] Y. Wu. (2022). *The Meituan Drone Carrying Food is Landing on the Top of a Dedicated Delivery Locker*. [Online]. Available: https://finance.sina.cn/tech/2022-07-13/detail-imizmscv1342574.d.html

[9] 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.

[10] 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.

[11] Z. Wang and J.-B. Sheu, "Vehicle routing problem with drones," *Transp. Res. B, Methodol.*, vol. 122, pp. 350–364, Apr. 2019.

[12] Y. Xia, W. Zeng, C. Zhang, and H. Yang, "A branch-and-price-and-cut algorithm for the vehicle routing problem with load-dependent drones," *Transp. Res. B, Methodol.*, vol. 171, pp. 80–110, May 2023.

[13] 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.

[14] 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.

[15] X. Wen and G. Wu, "Heterogeneous multi-drone routing problem for parcel delivery," *Transp. Res. C, Emerg. Technol.*, vol. 141, Aug. 2022, Art. no. 103763.

[16] G. Kim, H. Chung, and B.-K. Cho, "MOBINN: Stair-climbing mobile robot with novel flexible wheels," *IEEE Trans. Ind. Electron.*, vol. 71, no. 8, pp. 9182–9191, Aug. 2024.

[17] 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.

[18] Q. Luo, G. Wu, B. Ji, L. Wang, and P. N. Suganthan, "Hybrid multi-objective optimization approach with Pareto local search for collaborative truck-drone routing problems considering flexible time windows," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13011–13025, Aug. 2022.

[19] S. Meng, X. Guo, D. Li, and G. Liu, "The multi-visit drone routing problem for pickup and delivery services," *Transp. Res. E, Logistics Transp. Rev.*, vol. 169, Jan. 2023, Art. no. 102990.

[20] Q. Luo, G. Wu, A. Trivedi, F. Hong, L. Wang, and D. Srinivasan, "Multi-objective optimization algorithm with adaptive resource allocation for truck-drone collaborative delivery and pick-up services," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 9642–9657, Sep. 2023.

[21] K. Zheng. (2022). *How Do JD, SF Express, and Meituan Layout Drone Delivery?* [Online]. Available: https://m.thepaper.cn/baijiahao_19114333

[22] S. Meng, Y. Chen, and D. Li, "The multi-visit drone-assisted pickup and delivery problem with time windows," *Eur. J. Oper. Res.*, vol. 314, no. 2, pp. 685–702, Apr. 2024.

[23] R. Gu, M. Poon, Z. Luo, Y. Liu, and Z. Liu, "A hierarchical solution evaluation method and a hybrid algorithm for the vehicle routing problem with drones and multiple visits," *Transp. Res. C, Emerg. Technol.*, vol. 141, Aug. 2022, Art. no. 103733.

[24] P. Stodola and J. Nohel, "Adaptive ant colony optimization with node clustering for the multi-depot vehicle routing problem," *IEEE Trans. Evol. Comput.*, vol. 27, no. 6, pp. 1866–1880, Dec. 2023.

[25] 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.

[26] I. Dayarian, M. Savelsbergh, and J.-P. Clarke, "Same-day delivery with drone resupply," *Transp. Sci.*, vol. 54, no. 1, pp. 229–249, 2020.

[27] H. Zhang, Q. Zhang, L. Ma, Z. Zhang, and Y. Liu, "A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows," *Inf. Sci.*, vol. 490, pp. 166–190, Jul. 2019.

[28] 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.

[29] T. Zhu, S. D. Boyles, and A. Unnikrishnan, "Two-stage robust facility location problem with drones," *Transp. Res. C, Emerg. Technol.*, vol. 137, Apr. 2022, Art. no. 103563.

[30] F. Hong, G. Wu, Q. Luo, H. Liu, X. Fang, and W. Pedrycz, "Logistics in the sky: A two-phase optimization approach for the drone package pickup and delivery system," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 9175–9190, Sep. 2023.

[31] O. Dukkanci, A. Koberstein, and B. Y. Kara, "Drones for relief logistics under uncertainty after an earthquake," *Eur. J. Oper. Res.*, vol. 310, no. 1, pp. 117–132, Oct. 2023.

[32] C. C. A. Network. (Aug. 2022). *China Civil UAS Development International Forum*. [Online]. Available: http://www.caacnews.com.cn/thpd/202208/t20220825.html

[33] Q. Hou and J. Dong, "Cooperative output regulation of MASs with a novel parameter convergence condition and prescribed-time event-triggered observers and its application to robot formation," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 4168–4179, 2025.

[34] Q. Hou and J. Dong, "Fully distributed cooperative fault-tolerant output regulation of linear heterogeneous MASs: A dynamic memory event-triggered distributed observer approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 3, pp. 4548–4560, Jul. 2024.

[35] O. F. W. H. Zhang, Z. Ren, and H. Liu, "Logistics unmanned aerial vehicle flight plan pre-allocation in urban low-altitude airspace," *Syst. Eng. Electron.*, vol. 45, no. 9, pp. 2802–2811, 2023.

[36] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458, 2018.

[37] S. A. Vásquez, G. Angulo, and M. A. Klapp, "An exact solution method for the TSP with drone based on decomposition," *Comput. Oper. Res.*, vol. 127, Mar. 2021, Art. no. 105127.

[38] F. Tamke and U. Buscher, "The vehicle routing problem with drones and drone speed selection," *Comput. Oper. Res.*, vol. 152, Apr. 2023, Art. no. 106112.

[39] N. Yanpirat, D. F. Silva, and A. E. Smith, "Sustainable last mile parcel delivery and return service using drones," *Eng. Appl. Artif. Intell.*, vol. 124, Sep. 2023, Art. no. 106631.

[40] X. Sun, M. Fang, S. Guo, and Y. Hu, "UAV-rider coordinated dispatching for the on-demand delivery service provider," *Transp. Res. E, Logistics Transp. Rev.*, vol. 186, Jun. 2024, Art. no. 103571.

[41] H. Zhou, H. Qin, C. Cheng, and L.-M. Rousseau, "An exact algorithm for the two-echelon vehicle routing problem with drones," *Transp. Res. B, Methodol.*, vol. 168, pp. 124–150, Feb. 2023.

[42] X. Li, J. Tupayachi, A. Sharmin, and M. Martinez Ferguson, "Drone-aided delivery methods, challenge, and the future: A methodological review," *Drones*, vol. 7, no. 3, p. 191, Mar. 2023.

[43] Z. Luo, M. Poon, Z. Zhang, Z. Liu, and A. Lim, "The multi-visit traveling salesman problem with multi-drones," *Transp. Res. C, Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103172.

[44] 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.

[45] 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.

[46] J. Duan, Z. He, and G. G. Yen, "Robust multiobjective optimization for vehicle routing problem with time windows," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8300–8314, Aug. 2022.

[47] Y. Liu, L. Xu, Y. Han, X. Zeng, G. G. Yen, and H. Ishibuchi, "Evolutionary multimodal multiobjective optimization for traveling salesman problems," *IEEE Trans. Evol. Comput.*, vol. 28, no. 2, pp. 516–530, Apr. 2024.

[48] Y.-Q. Han, J.-Q. 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, Mar. 2020, Art. no. 172988142092003.

[49] 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.

[50] J. Homberger and H. Gehring, "Two evolutionary metaheuristics for the vehicle routing problem with time windows," *Inf. Syst. Oper. Res.*, vol. 37, no. 3, pp. 297–318, Aug. 1999.

[51] Allain. (2013). *Physics of the Amazon Octocopter Drone*. [Online]. Available: https://www.wired.com/2013/12/physics-of-the-amazon-prime-air-drone/

[52] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[53] L. Ke, Q. Zhang, and R. Battiti, "Hybridization of decomposition and local search for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1808–1820, Oct. 2014.

[54] X. Guo, T. Wei, J. Wang, S. Liu, S. Qin, and L. Qi, "Multiobjective U-shaped disassembly line balancing problem considering human fatigue index and an efficient solution," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 4, pp. 2061–2073, Aug. 2023.

[55] S. Zhang, S. Liu, W. Xu, and W. Wang, "A novel multi-objective optimization model for the vehicle routing problem with drone delivery and dynamic flight endurance," *Comput. Ind. Eng.*, vol. 173, Nov. 2022, Art. no. 108679.

[56] Y.-L. Lan, F. Liu, W. W. Y. Ng, M. Gui, and C. Lai, "Multi-objective two-echelon city dispatching problem with mobile satellites and crowd-shipping," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15340–15353, Sep. 2022.

[57] Y. Liu, "An elliptical cover problem in drone delivery network design and its solution algorithms," *Eur. J. Oper. Res.*, vol. 304, no. 3, pp. 912–925, Feb. 2023.

Fangyu Hong received the B.S. and M.S. degrees in traffic and transportation engineering from Central South University, Changsha, China, in 2017 and 2023, respectively, where she is currently pursuing the Ph.D. degree in traffic and transportation engineering. Her research interests include computational intelligence and urban logistics distribution.

Guohua Wu (Senior 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 Automation, Central South University, Changsha, China. He has authored more than 80 refereed articles, including IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. His current research interests include planning and scheduling, computational intelligence, and machine learning. He serves as an Editorial Board Member for *International Journal of Bio-Inspired Computation* and a Guest Editor for *Information Sciences and Memetic Computing*. He serves as an Associate Editor for *Swarm and Evolutionary Computation*. He is a Regular Reviewer of more than 20 journals, including IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, and *Information Sciences*.

Yalin Wang (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees from the Department of Control Science and Engineering, Central South University, Changsha, China, in 1995 and 2001, respectively. She is currently a Professor with the School of Automation, Central South University. Her research interests include modeling, optimization, and control for complex industrial processes, intelligent control, and process simulation.

Qizhang Luo received the B.S. and Ph.D. degrees in traffic and transportation engineering from Central South University, Changsha, China, in 2018 and 2023, respectively. He was a Ph.D. Visiting Student with the National University of Singapore from 2021 to 2022. His current research interests include computational intelligence and scheduling, with a focus on their applications in traffic and transportation fields.

Ling Wang (Senior Member, IEEE) received the B.Sc. degree in automation and the Ph.D. degree in control theory and control engineering from Tsinghua University, Beijing, China, in 1995 and 1999, respectively.

Since 1999, he has been with the Department of Automation, Tsinghua University, where he became a Full Professor in 2008. He has authored five academic books and more than 300 refereed articles. His current research interests include intelligent optimization and production scheduling. He was a recipient of the National Natural Science Fund for Distinguished Young Scholars of China, the National Natural Science Award (Second Place) in 2014, the Science and Technology Award of Beijing City in 2008, and the Natural Science Award (First Place in 2003 and Second Place in 2007) nominated by the Ministry of Education of China. He is the Editor-in-Chief of *International Journal of Automation and Control* and an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, *Swarm and Evolutionary Computation*, and *Expert Systems With Applications*.

Jianmai Shi received the Ph.D. degree in management science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2011. From 2008 to 2009, he was a Visiting Ph.D. Student with the University of Windsor, Canada. From 2014 to 2015, he was a Visiting Scholar with McGill University. He is currently an Associate Professor with the School of System Engineering, NUDT. He has authored more than 30 refereed articles, including those published in IEEE TRANSACTIONS ON INFORMATION THEORY, SMC, COR, and *Applied Energy*. His current research interests include vehicle routing, supply chain management, and heuristic design.