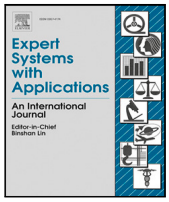
Expert Systems With Applications 249 (2024) 123472

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



# Ensemble multi-objective optimization approach for heterogeneous drone delivery problem

Xupeng Wen [a], Guohua Wu [a,*], Shuanglin Li [a], Ling Wang [b]

[a] School of Traffic & Transportation Engineering, Central South University, Changsha Hunan, 410075, China
[b] Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

*Keywords:*
Multiobjective evolutionary algorithm
Ensemble algorithms
Heterogeneous multi-drone
Routing

ABSTRACT

Recently, driven by advancements in the payload capacity and endurance of drones, the logistics industry has shown significant interest in drone Last-Mile logistics. Efficient routing are crucial scientific challenges in drone delivery problems. In this study, we address the routing problem in heterogeneous drone delivery, which involves a large drone transporting multiple small drones to sub-regions for parcel delivery, aiming to both reduce the drones' distance costs and improve customer satisfaction, termed HDDPBO To tackle the HDDPBO problem effectively, we propose a voting-based ensemble multi-objective genetic approach, named VEMOGA, in which an improved clustering algorithm is developed to divide customers into $K$ clusters, enabling each drone to handle multiple parcel deliveries within a sub-region. In this way, it reduces the complexity of HDDPBO by transforming it into multiple sub-problems. Secondly, a multi-objective genetic approach with heuristic operators is proposed to explore high-quality solutions, in which customized crossover and mutation operators are designed in the genetic approach, and a voting-based ensemble algorithm is designed to robustly select the Pareto frontier with high-quality convergence and diversity. Extensive experiments are conducted on synthetic instances to evaluate the proposed algorithm, and the experimental results demonstrate superior performance compared to three other baselines. Additionally, a real-world instance has been scrutinized to ascertain the applicability of Last-Mile logistics, and sensitivity analyses of pivotal factors have been conducted and several managerial insights pertinent are given to the drone-based Last-Mile logistics.

## 1. Introduction

Since drone delivery has the advantages of energy-saving, flexible, timely, and unaffected by ground traffic congestion or road conditions, it has garnered significant attention from logistics enterprises, including Amazon, SF, and JD. research institute (2020). Drone-based delivery modes have been applied for Last-Mile logistics in recent years, such as the truck & drone delivery mode (Lemardelé et al., 2021; Masmoudi et al., 2022). Particularly, Wen and Wu (2022) proposed the heterogeneous multi-drone delivery mode, which offers unparalleled advantages in terms of efficient and low-consumption straight-flight parcel delivery, independent of ground road networks. The heterogeneous multi-drone delivery system comprises a large drone and multiple small drones. With advancements in drone aviation logistics technology, the payload capacity of small drones can reach up to 20 kg. Research conducted by Amazon revealed that 86% of parcels weigh less than 2.27 kg, Guglielmo (2013), making it feasible for a small drone to handle multiple parcel deliveries in a single flight. On the other end of the spectrum, large drones boast payloads of up to 1.5 tons, facilitating transportation over distances exceeding 2000 km. Noteworthy achievements in this domain include Google's Wing drones, which have successfully delivered over 100,000 parcels (Wing, 2021). Furthermore, some countries have opened pilot areas for drone logistics in urban and rural regions, offering ample airspace for parcel delivery. Furthermore, SF has analyzed that drone delivery can reduce the cost by 30% and improve logistics efficiency by 50%.[3] Consequently, the heterogeneous multi-drone delivery problem is endowed with significant practical utility and holds promise for diverse applications in Last-Mile logistics.

This study presents a novel investigation into the heterogeneous multi-drone delivery problem with bi-objective, termed, HDDPBO, building upon previous research on the heterogeneous multi-drone delivery problem (HDDP) (Wen & Wu, 2022). Existing studies on HDDP have certain limitations. Firstly, they do not consider the time

* Corresponding author.
*E-mail addresses:* wenxupeng@csu.edu.cn (X. Wen), guohuawu@csu.edu.cn (G. Wu), lishuanglin@csu.edu.cn (S. Li), wangling@mail.tsinghua.edu.cn (L. Wang).

[3] http://www.wuasc.cn/page130?article_id=1905, 2022-4-20.

https://doi.org/10.1016/j.eswa.2024.123472
Received 27 July 2023; Received in revised form 30 January 2024; Accepted 10 February 2024
Available online 19 February 2024

window factor, despite its importance in real-world delivery scenarios where parcels are often delivered based on time priorities. Secondly, violating the time window constraint can negatively impact customer satisfaction. To address these shortcomings, we extend HDDP by incorporating minimization distance cost and maximization customer satisfaction as objective functions to construct a bi-objective model for HDDP. Specifically, in HDDPBO, a large drone carries $K$ small drones to serve $K$ sub-regions which are graphically shown in Fig. 2(a), and the small drone delivers multiple parcels in a single flight and finally flies to the automatic airport for recycling, depicted in Fig. 2(b). The schematic diagram of the proposed HDDPBO is graphically shown in Fig. 2. As shown in Fig. 2, the large drone, denoted as *mDrone*, is responsible for transporting small drones, denoted as *eDrone*, to sub-regions rather than directly delivering parcels to customers. Instead, small drones delivering the parcels within each sub-region before flying to the airport for recycling.

Given the conflicting nature of distance cost and customer satisfaction as objective functions in HDDPBO, single-objective optimization algorithms are inadequate for addressing this problem. In contrast, multi-objective evolutionary algorithms (MOEAs) have the potential to effectively balance multiple conflicting objective functions simultaneously, making them suitable for solving multi-objective optimization problems (Yu & Gen, 2010). Consequently, researchers have proposed various MOEAs to optimize problems with multiple conflicting objective functions. Hybrid MOEAs, which combine multiple complementary algorithms, have garnered significant attention due to their competitive performance (Tang & Wang, 2012; Wang, Wang, & Sun, 2022). These evolutionary algorithms aim to leverage the strengths of different approaches to solve complex optimization problems (Zhou et al., 2011).

To address the above issues, we propose a voting-based ensemble multi-objective genetic approach, named VEMOGA. The proposed VEMOGA consists of two main modules. First, since the computation complexity of HDDPBO increases exponentially with the increase of customer scale, we propose an improved clustering algorithm considering the drone's payload to decompose the original problem into $K$ sub-problems that are relatively easy to solve, significantly reducing the complexity of HDDPBO. Second, we design several specific crossover and mutation operators according to the features of HDDPBO in the genetic approach, and develop a voting-based environment selection algorithm. Further details of the proposed approach will be elaborated in Section 4.

**Contributions.** The contributions of this study are summarized as follows:

- We extend the heterogeneous multi-drone delivery problem (HDDP) and propose a bi-objective model for the HDDP. Concretely, a large drone carrying $K$ small drones, flies to $K$ sub-regions, and launches a small drone at each sub-region. While a small drone delivers multiple parcels in a flight in a sub-region. Furthermore, we introduce maximization customer satisfaction as an objective function and combine it with minimization distance cost to construct a bi-objective model for HDDP, namely HDDPBO.
- We propose a voting-based ensemble multi-objective genetic approach, namely VEMOGA, to assist the HDDPBO solving. First, an improved clustering algorithm considering drone payload is proposed to divide the distribution region into $K$ sub-regions to generate high-quality initial solutions. Second, several specialized crossover and mutation operators are designed in the genetic algorithm, and a voting-based ensemble algorithm is proposed to obtain Pareto frontiers with high-quality convergence and diversity.
- We conduct extensive experiments on the extended synthetic instances with small-scale, medium-scale, and large-scale. The experimental results show that the proposed VEMOGA is superior to the other three state-of-the-art comparison algorithms. Additionally, sensitivity analyses of critical factors of the drones are conducted in a real-world instance and several managerial insights are given.

In the next section, we provide a concise review of relevant literature on the routing problem in drone delivery. Section 3 presents the details of the voting-based ensemble multi-objective genetic approach. In Section 4, we present and discuss the experimental results. Finally, Section 5 concludes the paper and outlines potential directions for future research.

## 2. Related works

To date, there have been limited studies on the 'large drone + small drone' mode for addressing heterogeneous multi-drone logistics problems. Jeong et al. (2020) introduced a flight warehouse delivery system and designed a rolling time-domain method. Savuran and Karakaya (2016) proposed large aircraft as a mobile warehouse to launch and recycle drones. Wang, Pesch, Kress, Fridman and Boysen (2022) designed a distribution model of 'flight warehouse + shuttle + small drone'. The flight warehouse is a large airship. The shuttle obtains parcels from the ground warehouse and supplies the flight warehouse. The flight warehouse launches a small drone for parcel delivery. However, these works just regard the large drone as a fixed flight warehouse in the sky, and the small drone delivers one parcel in a flight. In addition, the truck and trailer routing problem (TTRP) is similar to our research problem, which requires a subset of customers to be visited by a truck-trailer pair, while other customers are visited by the truck alone (Derigs et al., 2013; Villegas et al., 2013). Dondo et al. (2011) surveyed the vehicle routing problem with more than two echelons in TTRP. Mirmohammadsadeghi et al. (2020) studied the truck and trailer routing problem with stochastic travel and service time, its travel and service times between customers are considered stochastic, and multi-point simulated annealing is applied to assist the problem-solving. Accorsi and Vigo (2020) proposed the extended single truck and trailer routing problem, which contains a variety of vertex types previously considered only separately: truck customers, vehicle customers with and without parking facilities, and parking-only locations. Rothenbächer et al. (2018) proposed a branch-price-cut algorithm to solve the truck-and-trailer routing problem with time windows.

Additionally, different kinds of hybrid multi-objective algorithms have already been applied to several variants of routing problems and have shown excellent performance in recent years (Zhang et al., 2019). Schermer et al. (2019) constructed mathematical models with minimization of transportation cost and minimization of completion time for the model with drone station, and then they constructed mathematical models with cost minimization and completion time minimization for the hybrid distribution mode with robot stations (Schermer et al., 2020), respectively. Zhang et al. (2019) considered the objective of reducing the vehicle number and minimizing wasting time, and proposed a hybrid MOEA with global search based on fast sampling strategy and local search based on route sequence difference to solve this problem. Ha et al. (2020) built a mathematical model for TSP drones (TSP-D) to minimize the cost and the completion time. Wang et al. (2020) built a bi-objective model for TSP-D with the objective of minimizing operation cost and completion time. Salama and Srinivas (2020) introduced two conflicting objectives (i.e., completion time and delivery cost) to extend the TSP-$m$D model. Han et al. (2020) proposed a tri-objective optimization model to minimize vehicle energy consumption, drone energy consumption, and the number of vehicles. Das et al. (2020) considered the travel cost and customer service level, and proposed an improved ant colony optimization algorithm to solve the proposed problem. Kumar et al. (2014) proposed a genetic algorithm for multi-objective routing problem, in which the objectives are minimizing the travel distance, total vehicle number, and the route balance simultaneously. Moradi (2020) studied a learnable evolution model, and

combined it with robust strength Pareto evolutionary algorithm to optimize the multiobjective routing problem. Zhang et al. (2019) proposed a multi-objective meme algorithm based on an adaptive local search chain, which combines a multi-directional local search strategy with enhanced local search chain technology. Luo et al. (2021) considered flexible time windows in a bi-objective truck-drone collaborative routing problem and proposed a hybrid multi-objective genetic approach incorporated with a Pareto local search algorithm.

Through the above literature review, it can be found that the HDDPBO studied in this paper is significantly different from that of truck drone delivery. The details can be summarized as follows: (1) In the truck-drone delivery problem, the truck is construed as a mobile carrier, and the small drone ultimately returns to the truck for retrieval, forming a closed loop of two-echelon routing problem. However, in the heterogeneous multi-drone model proposed in this study, the large drone do not recycle the small drones; instead, the latter are recycled by ground-based automatic airport. This configuration represents an instance of the open two-echelon routing problem variant. In contrast, the act of a small drone returning to the truck necessitates substantial waiting time for either the truck or the drone to ensure temporal synchronization. This introduces a novel challenge: determining the optimal automatic airport selection to minimize the overall distance cost. (2) In the traditional two-echelon routing problem 2E-VRP, the location of the transfer station remains relatively fixed, with the truck transporting the small drones to and from this predetermined station. Conversely, in the heterogeneous multi-drone model introduced in this paper, the large drone have the capability to transport small drones to various customer points within a given sub-region. This implies that the location of the transfer station becomes a flexible parameter. Consequently, an additional challenge arises: identifying the optimal location for flying small drones to minimize the total distance cost.

Additionally, the preceding optimization algorithm literature review reveals the proposed HDDPBO and solution algorithm VEMOGA are quite different. In HDDPBO, we adopt customer satisfaction as a metric for evaluating the adherence to time windows, and combine it with distance cost to formulate a bi-objective model for HDDPBO. To address this problem effectively, we propose a voting-based ensemble multi-objective genetic approach. In this manner, the proposed ensemble approach VEMOGA can obtain Pareto frontiers with high-quality convergence and diversity while solving complex multi-objective routing problems.

## 3. Mathematical formulation

### 3.1. Problem descriptions and assumptions

The Heterogeneous Drone Delivery Problem for Last-Mile Logistics (HDDPBO) is formally defined as follows: Within the Last-Mile delivery paradigm, all customer nodes are serviced by a heterogeneous drone system characterized by diverse configurations. The large drone departs from the depot, carries $K$ small drones and several parcels, launches a small drone at each sub-region, and finally returns to the depot. Each small drone is capable of executing multiple parcel deliveries during a single flight and subsequently navigates to an automatic airport for recycling purposes. In this delivery process, the large drone does not service the customer node, and all the customer nodes are serviced by the small drones. It is noteworthy that the automatic airport serves as a temporary depot within the Last-Mile logistics paradigm, facilitating the storage of parcels and small drones. The schematic diagram of HDDPBO is shown in Fig. 1.

To describe the HDDPBO accurately and define the scope of this study, several assumptions are presented as follows.

(1) The distribution system consists of a large drone, and $K$ small drones of the same type. The large drone is a fuel type with sufficient payload and range. The small drone is a rechargeable drone with payload constraints;

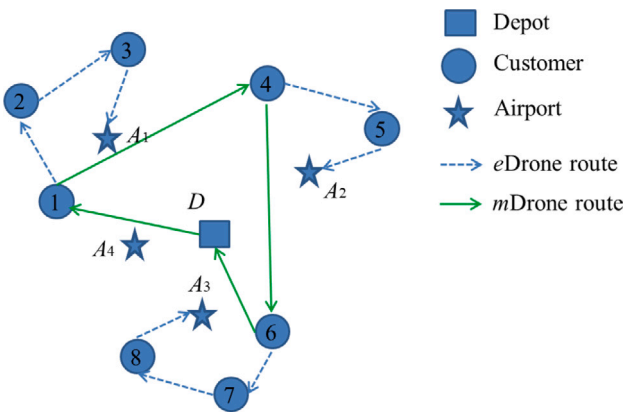



**Fig. 1.** The schematic diagram of the proposed HDDPBO.


(a) A large drone launches small drones[2]

(b) An automatic airport recycles small drones[3]

**Fig. 2.** The heterogeneous multi-drone delivery problem scenario. (a) A large drone launches small drones[2]. (b) An automatic airport recycles small drones[3].

(2) The large drone visits $K$ sub-regions, and launches a small drone in each sub-region, and finally returns to the depot $D$ for recycling;

(3) A small drone delivers multiple parcels in a flight in a sub-region, and finally flies to an automatic airport $A$ for recycling;

(4) Each customer node can only be visited by a small drone once;

(5) There is no sub-loop in the routes of the large drone and small drones;

(6) The automatic airport acts as a temporary depot, storing parcels and end drones, and the operating and maintenance costs and automatic airports are not considered.

### 3.2. Symbols and definitions

According to the literature (Poikonen & Golden, 2020), we define the large drone as the mother drone in this paper, and the small drone as the end drone. To accurately express the model, we define the following symbols, including the set, parameters, and decision variables of the mother drone and end drones. The definition of each symbol is given in detail in Table 1.

### 3.3. Model development

Given a directed graph network $G = (V, E)$, where $V$ is the node-set, $E$ is the set of routes between two nodes. Specifically, $V = D \cup A \cup C$, where $D$ is the depot, $A$ is the automatic airport set, $C$ is the customer node set. To facilitate the model of HDDPBO, several definitions are presented as follows:

[2] https://weibo.com/tv/show/1034:4356606616663880?from=old_pc_videoshow, 2022-6-5.

[3] http://www.foiadrone.com/, 2022-5-10.

**Table 1**
Symbols and Definitions.

| Symbols | Definitions |
|---|---|
| *Sets*: | |
| $V$ | $=D \cup A \cup C$. $D$ is the depot, $A$ is the automatic airport set, $C$ is the customer node set. |
| $R$ | $=\{R_m, R_e\}$. The set of routes of mother drone and end drone; |
| $R_m$ | The routes of the mother drone; |
| $R_e$ | The routes of the end drones; |
| $C$ | $=\{c_1, c_2, \ldots, c_n\}$, the set of customer nodes; |
| $C_k$ | The set of customer nodes in the $k$th cluster, where $C_k \subseteq C$; |
| $C_k^l$ | The node of the mother drone launches an end drone in the $k$th cluster, where $C_k^l \in C_k$; |
| $C^l$ | $=\{C_1^l, C_2^l, \ldots, C_k^l\}$, the node set of the mother drone launches end drones, where $C^l \subseteq C$; |
| $D$ | $=\{0, K+1\}$, the depot; |
| $A$ | The set of automatic airports; |
| *Paras*: | |
| $d_{i,j}$ | The distance between the node $i$ and the node $j$, $i, j \in V$; |
| $w_e$ | The weight of the drone; |
| $v_m$ | The flying speed of the mother drone; |
| $v_e$ | The flying speed of the end drones; |
| $q_{i,k}$ | The parcel's weight of the $i$th customer in $k$th cluster; |
| $n$ | The number of customers; |
| $n_k$ | The number of customers in the $k$th cluster; |
| $M$ | A finite mother enough number; |
| $Q_s$ | The max payload capacity of an end drone |
| $a_i$ | The lower limit of the best service time window for customers $i$; |
| $b_i$ | The upper limit of the best service time window for customers $i$; |
| $c_i$ | The lower limit of the customer $i$ acceptable service time window; |
| $h_i$ | The upper limit of the customer $i$ acceptable service time window; |
| $S_k(et_i)$ | Customer satisfaction function of customer node $i$; |
| $\rho_1$, $\rho_2$ | The transportation costs per unit of distance of mother drone and end drones, respectively; |
| $mt_i$ | The arrival time for the mother drone to the customer node $i$; |
| $et_{i,k}$ | The arrival time for the $k$th end drone to the customer node $i$; |
| *Vars*: | |
| $x_{i,j}$ | $x_{i,j}$ =1, if the mother drone goes from node $i$ to node $j$. otherwise, $x_{i,j}$ =0, where $i, j \in C_l \cup D$; |
| $y_{i,j,k}$ | $y_{i,j,k}$ =1, if the $k$th end drone goes from node $i$ to node $j$ in the $k$th cluster; otherwise, $y_{i,j,k}$ =0, where $i \in C_k$, $j \in C_k \cup A$; |

### 3.3.1. The definitions of mother and end drone routes

We first define the mother drone's route and the end drone's route.

Mother drone route: Each route $r \in R_m$ starts from the depot $D = \{0\}$, visits several customer nodes in $C$ to launch $K$ end drones, and then returns to the depot $D = \{0\}$. Given two nodes $i$ and $j$ ($i, j \in D \cup C^l$), if the mother drone route visits the customer nodes $i$ and $j$, then $x_{i,j} = 1$. Otherwise, $x_{i,j} = 0$. The mother drone route consists of the edges and can be defined as $R_m = \{(i, j) | i, j \in D \cup C^l, i \neq j\}$. Taking Fig. 1(b) as an example, the route of the mother drone is ($D$, 1, 4, 6, $D$).

End drone route: Each route $r \in R_e$ departs from the launch node, visits several customer nodes, and returns to the automatic airport. Given two nodes $i$ and $j$ ($i, j \in C_k$) in the $k$th cluster, if the end drone visits the customer nodes $i$ and $j$, then the $y_{i,j,k}$ is set to 1. Otherwise, $y_{i,j,k}$ is set to 0. The end drone route consists of the edges and can be defined as $R_e = \{(i, j) | i \in C_k, j \in C_k \cup A, i \neq j\}$. Taking Fig. 1 as an example, the end drone routes are (1, 2, 3, $A_1$), (4, 5, $A_2$), and (6, 7, 8, $A_3$).

### 3.3.2. Optimization objectives of HDDPBO

We introduce customer satisfaction as an optimization objective. Assuming that the customer satisfaction $S_k(et_i)$ equals 1 if a drone arrives at customer node $i \in C$ during the time window $[a_i, b_i]$, while it is approximate to 0 as the drone serves the customer too late or too early. If the time of drone serving the customer node $i$ is within the acceptable time range of the soft time window, i.e., the time of arriving at the customer node $i$ is $[c_i, a_i]$ or $[b_i, h_i]$, the value range of customer satisfaction $S_k(et_i)$ is (0, 1). The customer satisfaction curve is presented in Fig. 3.

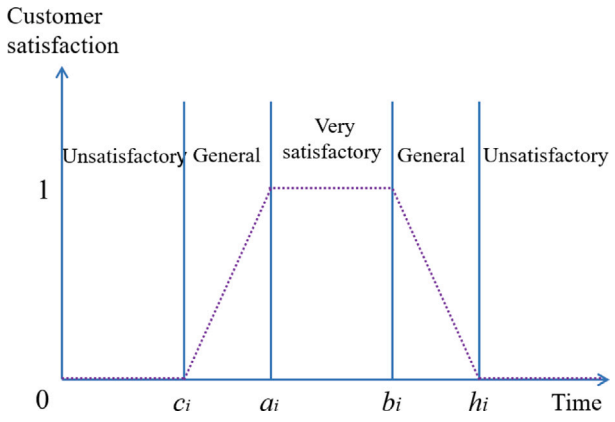



**Fig. 3.** Customer satisfaction curve.

Optimization Objective 1: The overall customer satisfaction can be expressed in (2), where the $i$th node's customer satisfaction can be calculated in (1).

$$S_k(et_i) = \begin{cases} 0 & et_i \le c_i, \\ \frac{a_i - et_i}{a_i - c_i} & c_i \le et_i \le a_i, \\ 1 & a_i \le et_i \le b_i, \\ \frac{et_i - b_i}{h_i - b_i} & b_i \le et_i \le h_i, \\ 0 & et_i \ge h_i, \end{cases} \tag{1}$$

$$f_1 = \max \sum_{k=1}^{K} \sum_{i=1}^{n_k} S_k(et_i), \tag{2}$$

Besides, as the cost is important for a logistics company, we consider the total distance cost as the second optimization objective.

Optimization Objective 2: The total distance cost can be expressed in (3).

$$f_2 = \min(\rho_1 \sum_{i \in C^l \cup D} \sum_{j \in C^l \cup D} d_{i,j} x_{i,j} + \rho_2 \sum_{k=1}^{K} (\sum_{i \in C_k} \sum_{j \in C_k \cup A} d_{i,j} y_{i,j,k})), \tag{3}$$

Objective function (2) maximizes customer satisfaction. Objective function (3) minimizes the total distance cost.

### 3.3.3. Constraints of HDDPBO

Additionally, based on the above-mentioned assumptions, the models should meet the following constraints, including the mother drone's constraints and small drone's constraints.

$$\sum_{i \in C^l \cup D} x_{i,j} = 1, \quad \forall j \in C^l \cup D, \tag{4}$$

$$\sum_{j \in C^l \cup D} x_{i,j} = 1, \quad \forall i \in C^l \cup D, \tag{5}$$

$$\sum_{j \in C^l \cup D} x_{0,j} = 1, \tag{6}$$

$$\sum_{i \in C^l} x_{i,K+1} = 1, \tag{7}$$

$$\sum_{i \in C^l \cup D} \sum_{j \in C^l \cup D} x_{i,j} \le K - 1, \tag{8}$$

$$\sum_{i\in C^l} x_{i,h} - \sum_{j\in C^l\cup D} x_{h,j} = 0 \quad \forall h \in C^l, \quad i \neq j, \tag{9}$$

$$\sum_{i\in C_k} q_{i,k} \leq Q, \qquad \forall k \in \{1,2,\ldots,K\}, \tag{10}$$

$$\sum_{i\in C_k} y_{i,j,k}=1, \quad \forall j \in A \cup C_k \backslash C_k^l, k \in \{1,2,\ldots,K\}, \tag{11}$$

$$\sum_{j\in C_k\cup A} y_{i,j,k}=1, \quad \forall i \in C_k, k \in \{1,2,\ldots,K\}, \tag{12}$$

$$\sum_{i\in C_k} y_{i,h,k} - \sum_{j\in C_k\cup A} y_{h,j,k} = 0, \quad \forall h \in C_k, \quad i \neq j, \tag{13}$$

$$\sum_{i\in C_k}\sum_{j\in C_k\cup A} y_{i,j,k} \leq n_k - 1, \quad \forall k \in \{1,2,\ldots,K\}, \tag{14}$$

$$mt_j \geq mt_i + \frac{d_{i,j}}{v_m} - M(1-x_{i,j}), \quad \forall i,\ j \in C^l \cup D, j \neq i, \tag{15}$$

$$\begin{aligned} &et_{j,k} \geq et_{i,k} + \tfrac{d_{i,j}}{v_e} - M(1-y_{i,j,k}),\\ &\forall i \in C_k, j \in C_k \cup A, j \neq i, \end{aligned} \tag{16}$$

$$ma_i=(\sum_{i\in C^l\cup D}\sum_{j\in C^l\cup D} d_{i,j}x_{i,j})/v_m, \tag{17}$$

$$\begin{aligned} &ea_{i,k}=ma_k + (\textstyle\sum_{i\in C_k}\sum_{j\in C_k\cup A} d_{i,j}x_{i,j})/v_e,\\ &\forall k \in \{1,2,\ldots,K\}, \end{aligned} \tag{18}$$

$$y_{i,j,k} \in \{0,\ 1\}, \quad \forall i \in C_k, j \in C_k \cup A,\ k \in \{1,2,\ldots,K\}, \tag{19}$$

$$x_{i,j} \in \{0,\ 1\}, \quad \forall i,j \in C^l \cup D, \tag{20}$$

$$q_{i,k} \geq 0, \quad \forall i \in C_k, k \in \{1,2,\ldots,K\}, \tag{21}$$

Constraint (4)–(5) restrict the mother drone to visit each sub-region only once. Constraint (6) ensures the mother drone start from the depot. Constraint (7) ensures the mother drone finally return to the depot. Constraint (8) ensures that there are no sub-loops in the mother drone route. Constraint (9) ensures that the mother drone leaves $i$th sub-region after visiting $i$th sub-region. Constraint (10) indicates that the total weight of parcels carried by the end drone cannot exceed its payload. Constraint (11)–(12) restricts that each customer node is visited by an end drone only once. Constraint (13) restricts the end drones from leaving customer node $h$ after delivering parcels to customer node $h$. Constraint (14) imposes that there is no sub-loop in an end drone route. Constraint (15) give the relationship between the arrival time and flight time in different clusters by the mother drone. Constraint (16) give the relationship between the arrival time and flight time of different customer nodes delivered by an end drone. Constraint (17) presents the calculation of arrival time in the $i$th sub-region for the mother drone. Constraint (18) presents the calculation of arrival time to the $i$th customer node delivered by an end drone in the $k$th sub-region. Constraints (19)–(20) provide the decision variable ranges of mother drone and small drones. Constraint (21) stipulates that the weight of each parcel is greater than 0.

## 4. The proposed approach

The traditional exact algorithms (such as branch and bound algorithm and column generation algorithm) have excellent performance in solving small-scale instances of two-echelon VRP. However, the scale of the instance may be much larger, which leads to the complexity increasing exponentially, the traditional exact algorithm may not obtain high-quality feasible solutions in an acceptable time (Sluijk et al., 2022). Therefore, we design an approximate approach that prescribes promising solutions within an acceptable time for HDDPBO, i.e., a voting-based ensemble multi-objective genetic approach (VEMOGA).

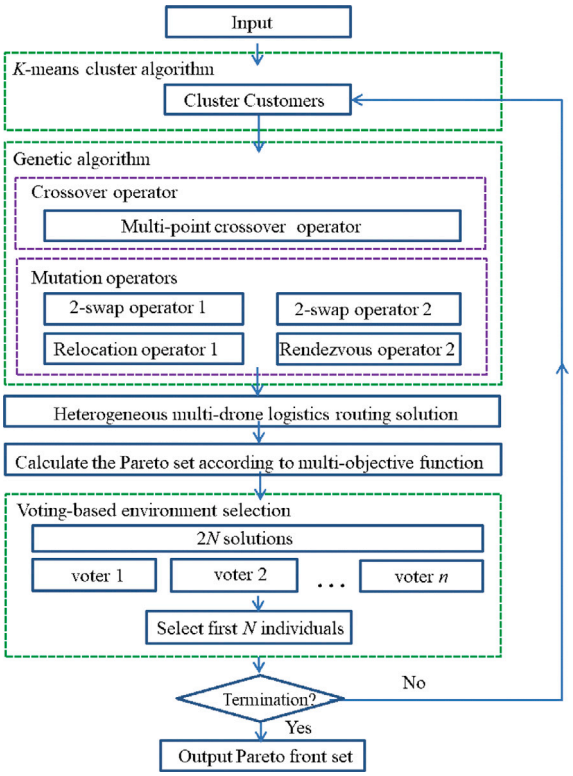



**Fig. 4.** The framework of the proposed approach.

### 4.1. The framework of the proposed VEMOGA

To consider the diversity and convergence of the search process simultaneously, we propose the VEMOGA. First, an improved clustering algorithm considering drone's payload is designed to divide the problem into $K$ subproblems. Then, a multi-objective genetic algorithm with heuristic operators is proposed to search high-quality solutions, in which a variety of crossover operators, mutation operators and chromosome coding rules for HDDPBO are designed in the genetic algorithm to promote the search efficiency, and a voting-based environment selection algorithm that integrates multiple sorting methods is proposed to select promising Pareto frontier with high-quality convergence and diversity. To clearly show the proposed approach, we draw a flow chart of the VEMOGA, which is shown in Fig. 4.

### 4.2. The proposed VEMOGA

To comprehensively consider the algorithm's convergence and diversity, the process of optimization in VEMOGA can be roughly divided into three steps. First, a chromosome encoding rule is designed based on the unique feature of the HDDPBO, and the initial population is generated according to the improved *K-means*($C$) function and the chromosome encoding rules. Second, the *GenerateOffspring* algorithm and several specialized operators are designed to generate offspring solutions. Third, a voting-based *EnvironmentSelection* algorithm is proposed to select the most suitable $N$ solutions to form the Pareto frontier. The pseudocode of the main procedure of the proposed VEMOGA is presented in Algorithm 1.

In Algorithm 1, the inputs are the customer node-set, airport set $A$, and depot $D$. The main framework of the proposed VEMOGA can be roughly divided into the initial population generation step and the population iterative optimization step. In the initial population generation step, first, the *CustomerCluster*($C$) algorithm is called to cluster customers into $K$ sub-regions. Then, chromosome encoding rules are

**Algorithm 1:** Main procedure of VEMOGA

**Input:** The customer node-set $C$; the airport set $A$, the depot $D$;
**Output:** The final population $P'$;
1 # Initial population generation;
2 Generate a new population according to search algorithm $A$;
3 **for** $i = 1 \rightarrow N$ **do**
4 $C_K \leftarrow$ Use the *CustomerClusterAdjust*$(C)$ function to divide the customers into $K$ clusters;
5 $P_i \leftarrow$ Generate an offspring solution using chromosome encoding rules on the $K$ clusters $C_K$;
6 # Population iterative optimization;
7 **while** $FEs < MaxFEs$ **do**
8 $Q \leftarrow$ Use the *GenerateOffspring*$(P)$ to generate offspring, and *UpdateOperatorParameters*() ;
9 $P' \leftarrow$ Update the parent population $P$ using the voting-based *EnvironmentSelection*$(P, Q)$ algorithm, and *UpdateVoterWeight*() ;
10 $FEs \leftarrow FEs + N$;

performed for each sub-region to record the nodes in one chromosome, including the node visited by the end drones and the nodes where the end drone is launched by the mother drone. Then, $N$ solutions are generated according to the clustering algorithm to form the initial population $P$ (lines 4–5). In the process of optimization, first, the *GenerateOffspring*() algorithm is called to generate the offspring population, and the operator parameters are updated (line 9). Second, the *EnvironmentSelection*$(P, Q)$ algorithm is called to select top $N$ solutions with high-quality convergence and diversity among $2N$ individuals from the parent population $P$ (including $N$ individuals) and newly generated offspring population $Q$(including $N$ individuals), and regard the top $N$ individuals as the Pareto frontier and the parent population in the next generation, and the voter weights are updated accordingly (line 10). The following describes the details of each proposed component.

#### 4.2.1. Customer clustering algorithm

Since the $K$-means is a simple and efficient clustering algorithm, it cannot solve the HDDPBO directly. To speed up the solution time and improve the solution efficiency, based on the ideas of 'truck first, drone second' (Murray & Chu, 2015) and 'clustered first, routed second' (Salama & Srinivas, 2020), we propose an improved clustering algorithm considering the end drone payload to divide this problem into $K$ sub-problems, and then optimize them separately. Specifically, first, the $K$-means algorithm is adopted to generate the initial customer clustering scheme, and its pseudocode is presented in Algorithm 2. Then the customer clustering scheme is adjusted according to the end drone payload. Once the clustering scheme is generated, the traversal nodes of the end drones are obtained. The pseudocode of the *CustomerCluster* $(C)$ algorithm is presented in Algorithm 3.

**Algorithm 2:** The *CustomerClusterInitial* $(C)$ algorithm

**Input:** The set of customer nodes $C$; the number of clusters $K$;
**Output:** The updated $K$ clusters $O$=$\{O_1, O_2, \ldots, O_K\}$;
1 Select $K$ customer nodes randomly as initial centroids;
2 **while** *termination criterion is not satisfied* **do**
3 $\{O_1, O_2, \ldots, O_K\} \leftarrow$ Assign each customer node to the nearest centroid to form $K$ clusters ;
4 $\boldsymbol{v} \leftarrow$ Update new centroid of the clusters according to (23);
5 $J \leftarrow$ Calculate the loss function of clusters $O$ according to (22);

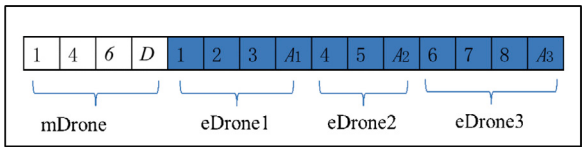



**Fig. 5.** Chromosome encoding diagram.

$$J = \sum_{i=1}^{K} \sum_{u \in O_i} \|u_i - v_j\|^2, \tag{22}$$

$$v_j = \frac{1}{|O_i|} \sum_{u \in O_i} u, \tag{23}$$

From Algorithm 2, the initial customer clustering is generated. Specifically, the cluster centroids can be calculated in (23) (line 4), and the loss function $J$ of customer clustering is (22) (line 5), where $O_i$ is the $i$th cluster, $v_j$ is the position of cluster centroid, and $u_i$ is the position of the $i$th customer.

**Algorithm 3:** The *CustomerClusterAdjust* $(C)$ algorithm

**Input:** The set of customer nodes $C$; the number of clusters $K$;
**Output:** The updated $K$ clusters $O$=$\{O_1, O_2, \ldots, O_K\}$;
1 $\{O_1, O_2, \ldots, O_K\} \leftarrow$ Call the *CustomerClusterInitial* $(C, K)$ algorithm to form $K$ clusters ;
2 # If the clusters $\{O_1, O_2, \ldots, O_K\}$ not satisfied the end drone's payload $Q_S$;
3 $K' \leftarrow K$;
4 **for** $k = 1 \rightarrow K$ **do**
5 **while** $Q_k \geq 2 * Q_S$ **do**
6 $K' \leftarrow K'$+1; Add a new cluster;
7 $Q_k \leftarrow Q_k - Q_S$;
8 $\{O_1, O_2, \ldots, O'_K\} \leftarrow$ Call the *Clustering* $(C, K')$ algorithm to update the $K'$ clusters;
9 **for** $k = 1 \rightarrow K'$ **do**
10 **if** $Q_S < Q_k < 2 * Q_S$ **then**
11 $\{O_1, O_2, \ldots, O'_K\} \leftarrow$ Use the cluster adjustment operators to adjust the $K'$ clusters;

In Algorithm 3, the numbers of customers in different clusters are adjusted appropriately according to the designed heuristic operators to satisfy the drone payload constraints (lines 6–11). Specifically, if the total weight of parcels need to be delivered in a sub-region $Q_k$ is more than 2 times the payload of the end drone $Q_S$, i.e., $Q_k = \sum_{i \in C_k} q_{i,k} \geq 2 * Q_S$, a cluster is added until it is less than 2 times the payload of the end drone (lines 6–9). On this basis, if the total weight of parcels in a sub-region $Q_k$ is still greater than the end drone payload $Q_S$, i.e., $Q_k = \sum_{i \in C_k} q_{i,k} < 2 * Q_S$, the clusters are adjusted according to the designed heuristic operators (lines 10–11).

#### 4.2.2. Chromosome encoding strategy

The chromosome encoding strategy should be properly designed because its quality will significantly affect the genetic algorithm search performance. Thus, we propose a chromosome encoding strategy for the HDDPBO based on the label encoding rule. To be specific, the order of customer nodes visited by the mother drone and end drones (i.e., launching node, end drone delivery node, and automatic airport recycle node) is considered during encoding, and the chromosome encoding diagram of an instance in Fig. 1(b) is presented in Fig. 5. From Fig. 5, this chromosome includes a mother drone route and three end drone routes.

In the process of decoding, taking Fig. 5 as an example, the chromosome contains a combined route of a mother drone and three end

drones, where the first repeated node of the combined route (customer node 1) is the takeoff node of the first end drone, and the landing node of the first end drone is the airport node $A_1$. Similarly, the second repeated node (customer node 4) is the takeoff node of the second end drone, and the landing node of the first end drone is the airport node $A_2$. The third repeated node (customer node 6) is the takeoff point of the third end drone and the landing node of the first end drone is the airport node $A_3$. All takeoff nodes, customer nodes, and landing nodes on each end drone route can be identified by the above rules. At this time, the parcel weight of the corresponding sub-region of the end drone at customer node 6 exceeds the large drone payload; The second $D$ node represents the large drone final return to the warehouse.

According to the above clustering algorithm and chromosome encoding rules, $N$ chromosomes are generated to form the initial population, in which each chromosome is composed of the routes of the mother drone and $K$ end drones.

#### 4.2.3. Offspring generation algorithm

The designed generate offspring algorithm mainly consists of two steps. First, $N$ offspring individuals are generated according to the multi-point crossover operators (De Jong & Spears, 1992) based on the clustering results from Algorithm Second, offspring individuals are mutated according to the designed mutation operators. The pseudocode of the *GenerateOffspring*$(P)$ algorithm is presented in Algorithm 4.

**Algorithm 4:** The *GenerateOffspring* $(P)$ algorithm

**Input:** The set of crossover operator $S_c$, the set of mutation operator $S_M$, mutation probability $pr_m$, parent population $P$; Population size $N$;
**Output:** The offspring population $Q$;

```
1  # Generate offspring population using crossover operator;
2  for i = 1 → |N| do
3      p1, p2 ← Select two parents individuals from P according to the tournament;
4      Q ← Use the individual p1, p2 to generate offspring individuals based on the multi-point crossover operator, and add to the offspring population ;
5  #Local search for offspring population using mutation operator;
6  for i = 1 → |N| do
7      Q_i' ← Select a mutation operator from S_M, and generate offsprings based on Q_i;
8      # If the new offspring Q_i' is infeasible, use the parent individual instead;
9      if Q_i' is not satisfied with the constraints then
10         Q_i ← P_i;
```

From Algorithm 4, the generate offspring algorithm consists of two stages. The first stage is to generate an offspring population using the crossover operator, and the second stage is to local search for the offspring population using the mutation operators. In the first stage, in each generation, two-parent individuals $p_1$, $p_2$ are selected from parent population $P$ according to the tournament rules (line 3). Then, offspring individuals are generated according to $p_1$, $p_2$, and add to the offspring population$Q$. Thus, $N$ offspring individuals are generated during the first stage (lines 2–4). In the second stage, using the mutation operators from $S_M$ and the mutation probability $pr_m$ to generate a new offspring (line 7). If the new generated solution $Q_i'$ does not satisfy the constraints, then the $i$th parent solution $P_i$ in parent population $P$ is adopted directly.

In Algorithm 4, we design the *multi-point crossover* approach to crossover the sub-regions, in which each sub-region corresponds to a crossover point. Thus, it is actually the $K$-point crossover approach. Take the coding in Fig. 5 as an example, the schematic diagram of the

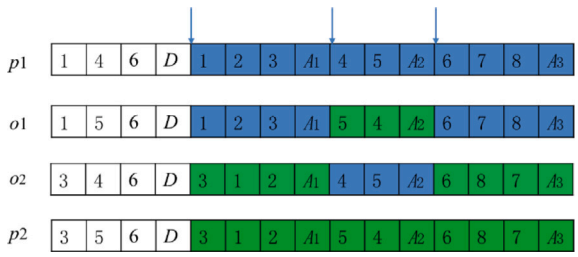
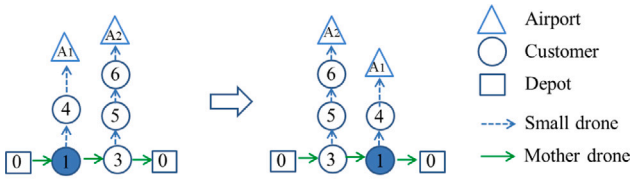
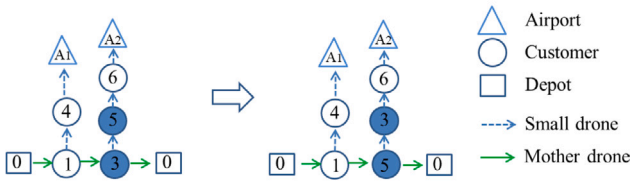



**Fig. 6.** Schematic diagram of multi-point crossover for sub-regions.




(a) Rendezvous operator 1




(b) Rendezvous operator 2

**Fig. 7.** Rendezvous operator for takeoff nodes.

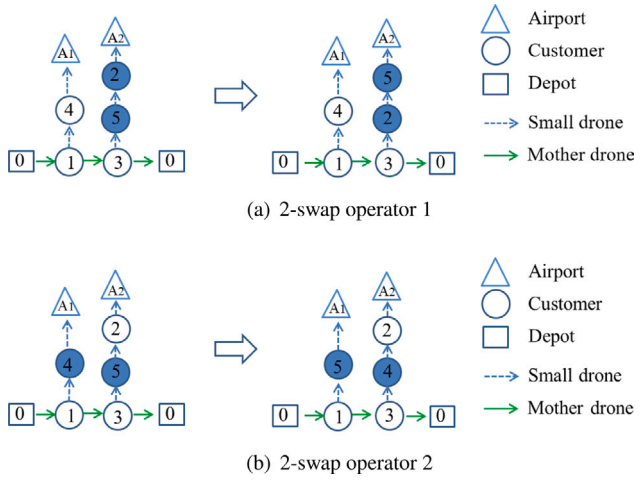
$K$-point crossover for sub-regions is presented in Fig. 6. In Fig. 6, the two-parent solutions have three crossover-point, i.e., $p_1$(1, 4, 6) and $p_2$(3, 5, 8), and two offspring solutions $o_1$ and $o_2$ are generated. If the generated solution is infeasible, its parent solution is adopted directly. In this manner, $N$ offspring solutions are generated. Furthermore, the number of crossover-point depends on the number of sub-regions.

#### 4.2.4. Specialized local search operators

Based on the multiple neighborhood operators Sadati and Çatay (2021), we designed specialized mutation operators for local search. Specifically, specialized perturbation operators are designed to mutate chromosomes, i.e., rendezvous operators, and 2-swap operators. These operators are used to transform the order of takeoff node and non-takeoff node of the mother drone and end drones, and optimize the track point order of the mother drone (because the route of mother drone consists of the nodes where the mother drone launches the end drone) and the order of customer nodes visited by end drones.

*(a)* **Rendezvous operator** The rendezvous operator is to adjust the takeoff node of end drones to obtain a neighborhood solution, so as to change the nodes order visited by the mother drone. As Fig. 7(a) shows, the visiting order of mother drones is adjusted from 0→1→3→0 to 0→3→1→0. As Fig. 7(b) shows, the visiting order of mother drones is adjusted from 0→1→3→0 to 0→1→5→0. It should be noted that the operator only adjusts the takeoff node of the end drone to its adjacent node.

*(b)* **2-Swap operator** The 2-Swap operator is to swap the visiting order of two customer nodes. We design two situations for exchanging the location of two customer nodes: (a) Swap two end drones visiting nodes in a cluster in Fig. 8(a); (b) Swap an end drone visiting node with another end drone visiting node between two different clusters in Fig. 8(b).




(a) 2-swap operator 1

(b) 2-swap operator 2

**Fig. 8.** 2-swap operator.

#### 4.2.5. Voting-based environment selection algorithm

Since the model of HDDPBO is a multi-objective optimization problem, the traditional exact algorithm may not be able to obtain resultful Pareto frontier (*PF*), and traditional dominance-based solution-sorting algorithms may lose selection pressure in addressing MOPs and are not able to get a good approximation of the true *PF* of a given problem. Besides, different multi-objective algorithms use different sorting algorithms, and in different optimization problems, even in different problem-solving stages, different sorting algorithms are quite different. It is crucial to design an ensemble method to dynamically determine the most appropriate sorting algorithm.

Previously, Wu et al. (2021) proposed a framework based on a voting mechanism to integrate multiple algorithms. These composed algorithms are regarded as experts (voters) based on collective wisdom to select individuals with excellent performance, and the weight of composing algorithms is dynamically adjusted according to their historical voting performance. For the multi-objective optimization problems, Qiu et al. (2020) proposed a voting mechanism framework in which multiple sorting methods of solutions are integrated into an ensemble, and the crowding distance calculation algorithm in Deb et al. (2002) is adopted in environment selection to keep the balance between convergence and diversity.

Based on the above motivations, to obtain high-quality Pareto frontier with high-quality convergence and diversity in the process of optimizing HDDPBO, we propose an ensemble framework based on a voting mechanism to integrate multiple sorting methods, which are used to vote on $2N$ solutions ($N$ offspring solutions and $N$ parent solutions, and the top $N$ most compromise solutions are selected according to the proposed *EnvironmentSelection*$(P, Q, S_{sort})$ algorithm as the next generation of the parent population, and then continue iterative evolution. The pseudocode of the proposed voting-based *EnvironmentSelection*$(P, Q, S_{sort})$ algorithm is presented in Algorithm 5.

**Algorithm 5:** *EnvironmentSelection*$(P, Q, S_{sort})$ algorithm

**Input:** Parent population $P$; offspring population $Q$; the set of sorting algorithm set $S_{sort}$;
**Output:** The updated population $P'$;

```
1  # Voting based on multiple sorting methods;
2  Merge the population P ∪ Q;
3  for i = 1 → |S_sort| do
4      for j = 1 → |P ∪ Q| do
5          # Use the ith sorting method S_sort(i) to vote solutions in
             P ∪ Q;
6          if S_sort (i) votes for the solution P ∪ Q(j) then
7              V(i, j) ← 1;
8          else
9              V(i, j) ← 0;
10 # Calculate weighted votes;
11 V_T(i) ← Σ_{i=1}^{|S_sort|} w_i · V(i, j) Calculate the weighted votes of the ith
     solution in all voters;
12 F_p ← Calculate the crowding distance;
13 P' ← Select the most suitable N solutions according to the
     weighted votes V_T(i) and crowding distance F_p;
14 # Update voter weights;
15 for i = 1 → |S_sort| do
16     S_Vi ← Calculate the number of solutions that voted by the
         ith sorting method and final survive to the population P';
17 for i = 1 → |S_sort| do
18     Con_i ← S_Vi / Σ_{i=1}^{|S_sort|} S_Vi  Calculate contributions of the ith voter
         based on S_Vi;
19 # Adjust the voter weights based on their contributions;
20 for i = 1 → |S_sort| do
21     w_i ← Con_i;
```

In Algorithm 5, the voting-based environment selection algorithm is mainly divided into four steps: First, voting on the merged population $P \cup Q$ according to multiple sorting methods (lines 2–9); Second, calculating weighted votes (line 11); Third, selecting most suitable $N$ solutions with high-quality convergence and diversity according to the weighted votes $V_T(i)$ and crowding distance $F_p$(lines 12–13); Fourth, reallocating all voter weights according to their contribution (lines 15–21).

Concretely, each element in the matrix $V(i, j)$ is 1 or 0, where 1 means that the solution is selected by the $i$th sorting method; Otherwise, 0 means that the solution is not selected by the $i$th sorting method. Taking the voting-result matrix with 4 rows and 200 cols as an example, it denotes that the voting-based framework contains 4 sorting methods (voters), and the number of parent and offspring population is 200. If $V(1, 35) = 1$, it means that the 1st sorting method (voter) considers that the 35-th solution in the merged population $P \cup Q$ is a promising solution. Otherwise, $V(1, 35) = 0$. Note that each sorting method is extracted from representative multi-objective algorithms. For example, if the RVEA (Cheng et al., 2016) is integrated into the proposed framework, in which each objective vector is converted into a normalized vector, and a set of reference vectors is required.

In the second step, each sorting method is initialized with the same weight. The weighted votes of the $i$th solution $V_T(i)$ are calculated by the voting results and their weights (line 11). In the third step, the solutions are divided into several levels according to the obtained weighted votes, then the most suitable $N$ solutions are selected from population $P$ according to the divided level and its crowding distance (lines 12–13). In the fourth step, the contribution $Con_i$ of each voter is calculated based on the voting results $V$ and finally survive individuals $S_{V_i}$ to the population $P'$ (lines 17–18), then the voters weights are updated according to their contribution during the optimization process (lines 20–21).

## 5. Experimental study

To investigate the performance of the proposed VEMOGA, we compare it with three other state-of-the-art algorithms on extensive synthetic instances and a real-world instance. All the algorithms are coded in Python, and run on a 64-bit Windows 10 with Intel Core(TM) i7-9800X CPU, 3.80 GHz.

**Table 2**
Drone parameter settings.

| | |
|---|---|
| $Q_s = 10$ kg | The payload capacity of end drone. |
| $w_{max} = 2$ kg | The max weight of a parcel. |
| $r_n = 6$ | The number of rotors. |
| $\mu = 1.204$ kg/m$^3$ | The fluid density. |
| $\zeta = 0.2$ m$^2$ | The rotor disc area. |

### 5.1. Experiment setting

#### 5.1.1. Algorithm settings

The crossover probability $p_c$ is set to 0.6, and the mutation probability $p_m$ is set to 0.4. Besides, the solution-sorting methods (voters) in the voting-based framework are selected from multi-objective evolutionary algorithms (MOEAs) according to the setting of Qiu et al. (2020). The initial weights of these four solution-sorting methods (voters) are set to 0.25.

In addition, we select two representative state-of-the-art multi-objective optimization algorithms, i.e., MOEAD (Zhang & Li, 2007) and NSGA-III (Deb & Jain, 2013) as competitors. MOEAD is a well-known decomposition-based algorithm while NSGA-III is an improved multi-objective combinatorial optimization algorithm based on NSGA-II. MOEAD and NSGA-III decompose multi-objective problems based on Chebyshev approach. In addition, we simplify the proposed VEMOGA by removing local search, named VEMOGA-no-LS, as the third comparison algorithm. It is worth noting that the proposed VEMOGA is developed based on the NSGA-II, thus, the VEMOGA-no-LS can be regarded as the improved version of NSGA-II.

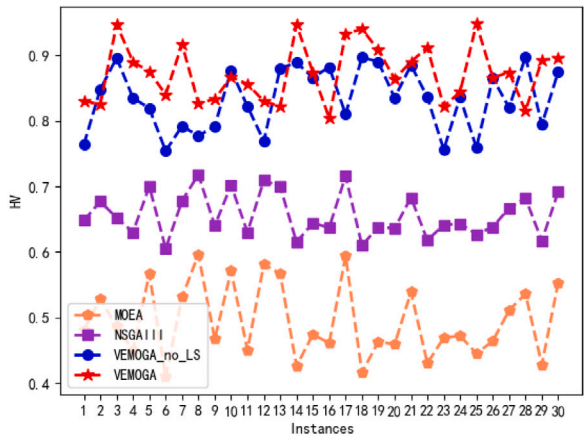
We adopt the Hypervolume (HV) as indicator in this experiment. HV refers to the spatial volume determined by a set of solutions in the reference point and objective space. It can reflect both the convergence and diversity of the population acquired by MOEA. The approach to specifying reference points refers to the work of Ishibuchi et al. (2017).

#### 5.1.2. Benchmark settings

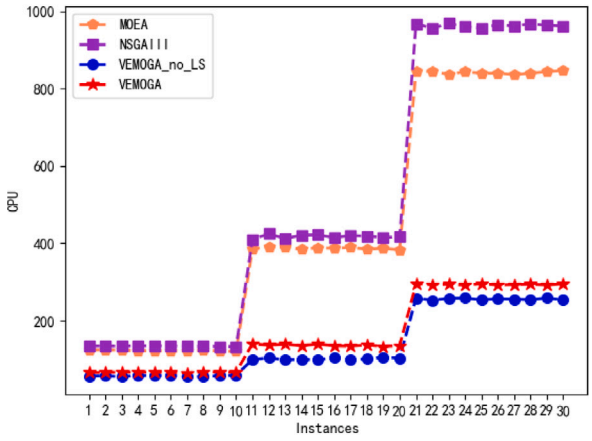
Since the proposed HDDPBO is a new problem, there is no available benchmark for HDDPBO can be directly used to conduct experiments. Therefore, we extend the well-known public benchmark instances proposed by Augerat et al. (1995) and Dumas et al. (1995). To comprehensively investigate the performance of the proposed algorithm, we consider three customer scales, i.e., the small-scale with 40 customers, the medium-scale with 60 customers, and the large-scale with 80 customers. Each scale consists of 10 instances. These instances are named based on the number of customer nodes and the length of time windows, such as *M-n40-t60-i1* denotes the number of customers is 40, the length of the time window is 60, and the instance number is 1.

#### 5.1.3. Drone settings

Since the end drone's payload can be 10–15 kg in industrial applications (Han et al., 2017), the end drone's payload is set to 10 kg and its flight speed range is 30–72 km/h in this experiment. As 86% of parcels are lighter than 2.27 kg (Guglielmo, 2013), an end drone can theoretically deliver up to 5–8 small light parcels. While the mother drone, its payload can be up to 1.5 tons and has a sailing distance of more than 2000 km, which has sufficient range and payload. In addition, the maximum charging capacity of the end drone is set as 12,400 mAh, and the unit energy consumption cost is set as $0.635 \times 10^{-4}$ /mAh (Liu et al., 2020). We only considered the distance costs of drones. In summary, the drone parameters are displayed in Table 2. Within the practical drone logistics scenarios, the selection of various drone types and their corresponding parameter configurations can be tailored to align with the specific requirements of logistics.




(a) HV values




(b) CPU time

**Fig. 9.** HV values and CPU time of instance on three scales.

### 5.2. Experiments and analysis on synthetic instances

Each instance runs 10 times in our experiments. The mean HV values, Std HV values, and the CPU runtime are recorded in Table 3. We introduce the significance test to compare the four algorithms, in which the symbols +, −, ≈ represents the HV value of the comparison algorithm is better than, worse than, or similar to that of the proposed VEMOGA, respectively. Note that the larger the HV value is, the better the performance of the solution is. To more intuitively display the comparison of HV values and running time of different algorithms, we present the comparison diagram of HV value of different algorithms (as shown in Fig. 9(a)) and the comparison diagram of CPU time of different algorithms (as shown in Fig. 9(b)), respectively.

In Table 3 and Fig. 9(a), the Pareto frontier generated by the proposed VEMOGA is significantly superior to that of the other three baseline algorithms on synthetic instances of the small-scale, medium-scale, and large-scale. Specifically, based on the significance test results, the HV values of MOEAD and NSGAIII are worse than the proposed VEMOGA on all instances. Meanwhile, the HV value of VEMOGA-no-LS is most similar to that of VEMOGA. However, the HV values of VEMOGA are worse than the VEMOGA- no-LS on the *M-n40-t60-i2*, *M-n40-t60-i10*, *M-n60-t60-i6*, and *M-n80-t60-i8* instances. The reasons can be inferred as multifold. First, specialized local search operators are designed in the proposed VEMOGA. Although both MOEAD

**Table 3**
The mean and Std. of HV, and CPU runtime of the four comparison algorithms.

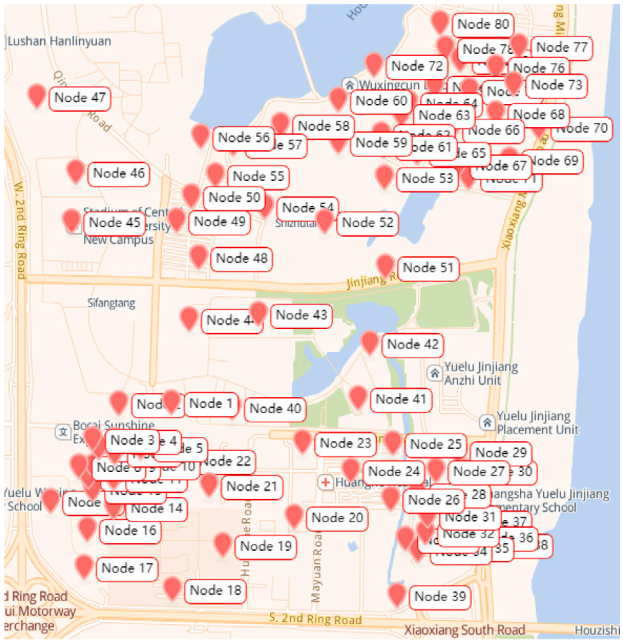
| | MOEAD | | | NSGA-III | | | VEMOGA-no-LS | | | VEMOGA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | Mean | Std. | CPU time | Mean | Std. | CPU time | Mean | Std. | CPU time | Mean | Std. | CPU time |
| *n40-i1* | 0.481 | 0.036 | 123.878 | 0.648 | 0.025 | 134.620 | 0.764 | 0.028 | 56.734 | **0.830** | 0.010 | 66.183 |
| *n40-i2* | 0.528 | 0.060 | 124.999 | 0.677 | 0.022 | 134.726 | **0.848** | 0.032 | 57.866 | 0.825 | 0.010 | 66.729 |
| *n40-i3* | 0.486 | 0.039 | 123.453 | 0.652 | 0.020 | 134.332 | 0.896 | 0.029 | 55.185 | **0.947** | 0.008 | 67.609 |
| *n40-i4* | 0.450 | 0.054 | 122.341 | 0.630 | 0.021 | 135.172 | 0.835 | 0.028 | 58.842 | **0.890** | 0.012 | 67.519 |
| *n40-i5* | 0.567 | 0.047 | 122.742 | 0.703 | 0.020 | 134.950 | 0.819 | 0.030 | 57.845 | **0.875** | 0.008 | 67.368 |
| *n40-i6* | 0.409 | 0.054 | 121.111 | 0.605 | 0.023 | 135.032 | 0.754 | 0.033 | 58.595 | **0.840** | 0.009 | 66.334 |
| *n40-i7* | 0.531 | 0.045 | 121.860 | 0.678 | 0.015 | 133.897 | 0.792 | 0.031 | 55.221 | **0.917** | 0.012 | 65.541 |
| *n40-i8* | 0.596 | 0.040 | 123.428 | 0.718 | 0.017 | 133.836 | 0.777 | 0.033 | 55.693 | **0.827** | 0.012 | 68.338 |
| *n40-i9* | 0.467 | 0.053 | 122.071 | 0.640 | 0.016 | 133.658 | 0.792 | 0.031 | 57.806 | **0.833** | 0.012 | 67.652 |
| *n40i10* | 0.571 | 0.040 | 120.430 | 0.702 | 0.019 | 133.036 | **0.877** | 0.030 | 59.436 | 0.867 | 0.012 | 67.395 |
| *n60-i1* | 0.450 | 0.057 | 385.330 | 0.630 | 0.018 | 410.250 | 0.822 | 0.030 | 99.696 | **0.856** | 0.011 | 140.376 |
| *n60-i2* | 0.581 | 0.045 | 390.863 | 0.709 | 0.016 | 424.597 | 0.769 | 0.030 | 103.939 | **0.830** | 0.010 | 136.954 |
| *n60-i3* | 0.567 | 0.045 | 389.091 | 0.705 | 0.019 | 412.278 | 0.880 | 0.029 | 99.005 | **0.822** | 0.008 | 139.587 |
| *n60-i4* | 0.426 | 0.056 | 385.064 | 0.615 | 0.016 | 420.904 | 0.889 | 0.031 | 99.959 | **0.947** | 0.011 | 135.377 |
| *n60-i5* | 0.474 | 0.046 | 388.333 | 0.644 | 0.017 | 422.095 | 0.866 | 0.029 | 98.969 | **0.874** | 0.013 | 140.095 |
| *n60-i6* | 0.461 | 0.057 | 387.275 | 0.637 | 0.023 | 414.604 | **0.882** | 0.028 | 104.751 | 0.804 | 0.012 | 135.211 |
| *n60-i7* | 0.594 | 0.058 | 389.217 | 0.716 | 0.025 | 420.345 | 0.810 | 0.029 | 98.436 | **0.933** | 0.009 | 134.439 |
| *n60-i8* | 0.416 | 0.056 | 385.049 | 0.610 | 0.023 | 418.724 | 0.897 | 0.033 | 101.595 | **0.940** | 0.012 | 137.309 |
| *n60-i9* | 0.463 | 0.036 | 387.193 | 0.638 | 0.021 | 415.145 | 0.890 | 0.031 | 105.785 | **0.908** | 0.011 | 133.561 |
| *n60i10* | 0.459 | 0.037 | 383.352 | 0.636 | 0.025 | 416.292 | 0.835 | 0.030 | 103.480 | **0.864** | 0.008 | 136.026 |
| *n80-i1* | 0.539 | 0.056 | 844.516 | 0.683 | 0.018 | 967.569 | 0.885 | 0.031 | 258.195 | **0.890** | 0.010 | 295.326 |
| *n80-i2* | 0.430 | 0.053 | 843.719 | 0.618 | 0.016 | 955.406 | 0.836 | 0.031 | 252.957 | **0.912** | 0.011 | 293.484 |
| *n80-i3* | 0.469 | 0.049 | 836.746 | 0.641 | 0.018 | 967.852 | 0.756 | 0.029 | 257.715 | **0.822** | 0.013 | 294.073 |
| *n80-i4* | 0.472 | 0.040 | 843.349 | 0.643 | 0.022 | 959.602 | 0.836 | 0.029 | 258.647 | **0.844** | 0.012 | 291.885 |
| *n80-i5* | 0.445 | 0.037 | 839.254 | 0.627 | 0.023 | 955.719 | 0.759 | 0.029 | 253.857 | **0.948** | 0.011 | 295.625 |
| *n80-i6* | 0.464 | 0.043 | 839.505 | 0.638 | 0.025 | 964.823 | 0.865 | 0.030 | 256.652 | **0.866** | 0.011 | 292.237 |
| *n80-i7* | 0.511 | 0.054 | 835.320 | 0.667 | 0.017 | 961.279 | 0.820 | 0.032 | 255.620 | **0.874** | 0.010 | 293.336 |
| *n80-i8* | 0.536 | 0.040 | 839.219 | 0.682 | 0.019 | 967.029 | **0.898** | 0.030 | 254.238 | 0.815 | 0.012 | 294.219 |
| *n80-i9* | 0.428 | 0.046 | 843.617 | 0.617 | 0.022 | 963.230 | 0.794 | 0.030 | 258.648 | **0.893** | 0.009 | 292.276 |
| *n80i10* | 0.553 | 0.038 | 846.221 | 0.692 | 0.025 | 961.574 | 0.875 | 0.032 | 253.866 | **0.895** | 0.009 | 295.197 |
| +/−/≈ | | | 0/30/0 | | | 0/30/0 | | | 4/26/0 | | | |

and NSGAIII are excellent algorithm frameworks, they cannot solve our studied HDDPBO well, thus we design mutation operators and crossover operators to improve the algorithm's performance. Second, specialized global crossover operators are designed in this study. Compared with MOEAD and NSGAIII, both VEMOGA and VEMOGA-no-LS with designed crossover operators can improve the diversity of the population.

From Table 3 and Fig. 9(b), it can be found that the CPU computation time of the MOEAD and NSGAIII is significantly longer than that of the proposed VEMOGA for solving the HDDPBO, even three times that of the proposed VEMOGA on the large-scale instances. The underlying reasons can be speculated multifold. First, we design a clustering algorithm within the VEMOGA addressing the heterogeneous multi-drone routing problem that plays a pivotal role. This algorithm effectively partitions the Last-Mile customer distribution region into $K$ sub-regions, wherein the customer nodes for each end drone are determined through cluster-based heuristic information, without randomly searching the customer nodes in the entire customer distribution region for Last-Mile logistics. In this manner, it can remarkably reduce the CPU computation time. Furthermore, the increased CPU computation time of VEMOGA relative to VEMOGA-no-LS for the last-mile logistics can be rationalized by the incorporation of multiple local search mutation operators in the former. This inclusion, while enhancing the algorithm's exploration capabilities, concurrently results in heightened CPU consumption. This trend is evident in both Table 3 and Fig. 9(b). Notably, the computational time of NSGAIII surpasses that of the other two algorithms, VEMOGA-no-LS and MOEAD, which do not employ local searches. The extended CPU computation time for NSGAIII is attributed to its adoption of local search operators, necessitating additional search time for the HDDPBO.

### 5.3. Experiments and analysis on the real-world instance

To investigate the proposed VEMOGA performance on the real-world instance for the HDDPBO, 80 customer nodes in Changsha, China, are selected as a real-world instance according to the setting of Hong et al. (2023). The locations and regions of these 80 customer nodes are shown in Fig. 10. The longitude and latitude data of 80 customer nodes are obtained using the tool provided by the Baidu Map.




(a)

**Fig. 10.** The real-world instance includes 80 customer nodes in Changsha, China.

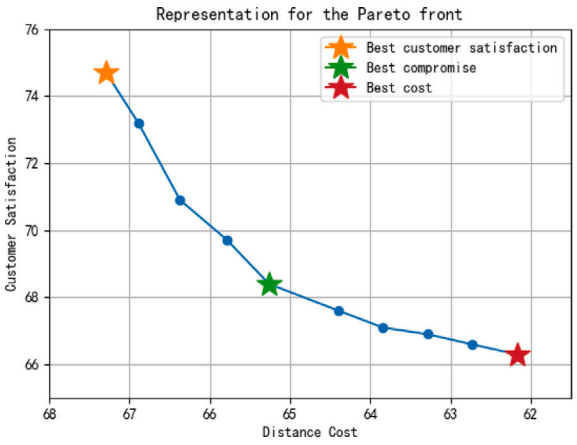



**Fig. 11.** Pareto frontier of the real-world instance.

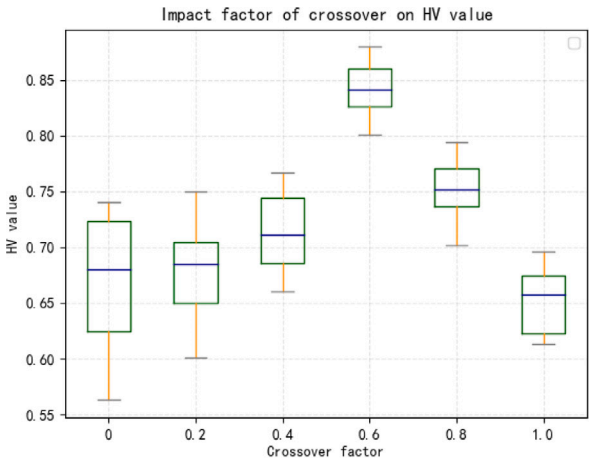
**Table 4**
Best-cost, best-satisfaction, and best-compromise of comparison algorithms.

| | MOEAD | | NSGAIII | | VEMOGA-no-LS | | VEMOGA | |
|---|---|---|---|---|---|---|---|---|
| | *Cost* | *Satis* | *Cost* | *Satis* | *Cost* | *Satis* | *Cost* | *Satis* |
| Best-cost | 79.28 | 66.47 | 74.51 | 67.78 | 66.07 | 68.19 | 61.81 | 70.89 |
| Best-satis | 85.93 | 71.36 | 81.27 | 72.93 | 72.31 | 72.45 | 65.73 | 74.23 |
| Best-compromise | 82.16 | 68.75 | 77.80 | 69.12 | 68.79 | 70.83 | 62.46 | 72.56 |

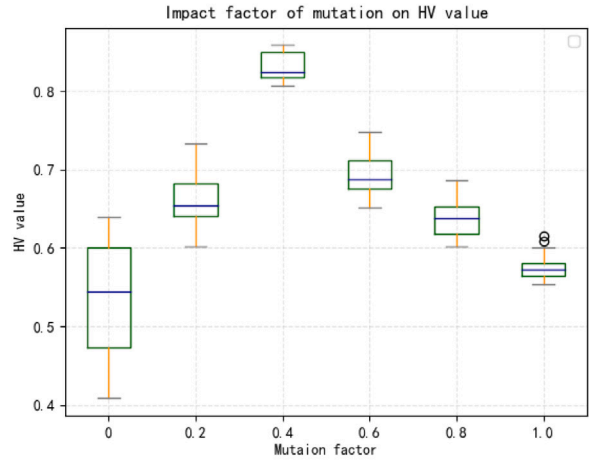
To intuitively show the multi-objective solutions in the real-world instance, we present the Pareto frontier in Fig. 11. From Fig. 11, the best-cost solution, best-satisfaction solution, and best-compromise solution are marked by red, orange, and green colors, respectively, where the best-compromise solution can be obtained based on its distance to the ideal point (Blasco et al., 2008). As we can see from Fig. 11, the best-compromise solution's customer satisfaction and distance cost on the real-world instance are 68.43 and 65.26, respectively. The best-cost solution's customer satisfaction and distance cost on the real-world instance are 66.31 and 62.17, respectively. The best-satisfaction solution's customer satisfaction and distance cost on the real-world instance are 74.69 and 67.29, respectively.

To further analyze the tradeoff between transportation cost and customer satisfaction, we give a comparison among the best-cost, best-compromise, and best-satisfaction solutions of different algorithms, which are recorded in Table 4. The *Satis* represents the value of customer satisfaction.

From Table 4, in terms of the best-compromise solutions for the heterogeneous multi-drone routing problem in Last-Mile logistics, it can be found that the distance cost of VEMOGA decreases 31.54%, 24.56%, and 6.76% compared with MOEAD, NSGAIII, and VEMOGA-no-LS, respectively. While the customer satisfaction of VEMOGA increases 5.54%, 4.98%, and 2.44% from the other three algorithms. The underlying factors contributing to these outcomes can be discerned through several facets. Firstly, a customer clustering algorithm is incorporated into the proposed VEMOGA, affording a high-quality initial solution for heterogeneous multi-drone routing. In this manner, the heterogeneous multi-drone routing problem is methodically deconstructed into several constituent sub-problems that exhibit relative computational tractability. This decomposition approach significantly mitigates the overall complexity associated with solving the HDDPBO. Secondly, an ensemble multi-objective genetic algorithm is formulated, featuring bespoke crossover and mutation operators. Additionally, a vote-based environment selection algorithm is introduced to enhance selection pressure, thereby attaining Pareto frontiers characterized by heightened convergence and diversity, indicative of high-quality solutions for the HDDPBO.




(a) Impact factor of crossover on HV value




(b) Impact factor of mutation on HV value

**Fig. 12.** Sensitivity analysis of the proposed algorithm.

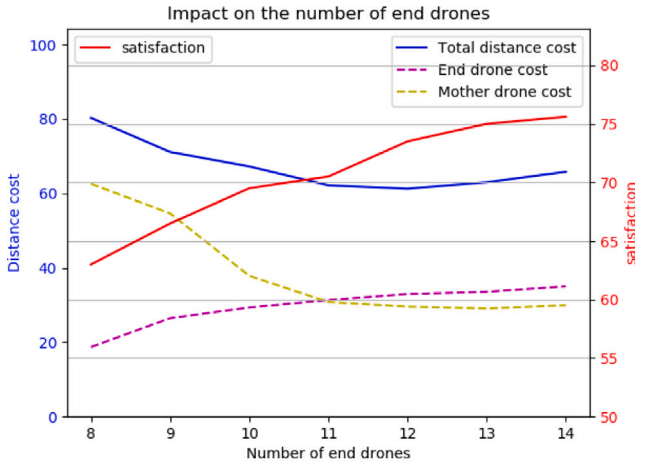
### 5.4. Sensitivity analysis

To analyze the influence of crucial factors on the distance cost of heterogeneous multi drones, first, two crucial factors of the proposed VEMOGA (i.e., crossover rate and mutation rate) on HV values are analyzed, and two crucial factors of the end drones (i.e., the number of end drones and the payload of end drones) on HV values are analyzed, namely, the number of end drones and the payload of end drones. All sensitivity analyses are based on the real-world instance, i.e., the medium scale of 80 customer nodes in Changsha, China.

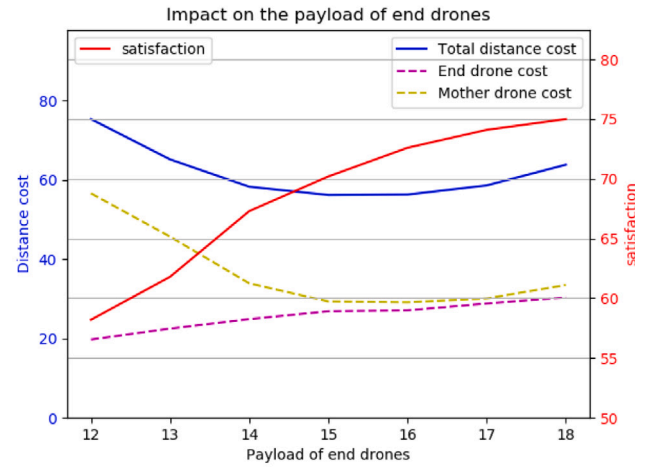
#### 5.4.1. Sensitivity analysis of the proposed VEMOGA

The sensitivity of the crossover rate and mutation rate in the proposed VEMOGA on HV values are analyzed. The factors values of crossover rate and mutation rate vary from 0 to 1 with step 0.2. All the experiments are run 15 times on the real-world instance. The results are displayed in Fig. 12.

In Fig. 12(a), it can be found that when the crossover factor value is 0.6, the mean and median HV values are the largest, meanwhile, the difference between the upper quartile and the lower quartile is the largest. It demonstrates that 0.6 is the most suitable crossover setting for solving the HDDPBO. In Fig. 12(b), it can be found that when the mutation factor value is 0.4, the mean and median HV values are the




(a) Impacts on the number of end drones




(b) Impacts on the payload of end drone

**Fig. 13.** Sensitivity analysis of drone.

largest. The underlying reason can be inferred that if the times of global search (represented as the probability of crossover operator) and local search (represented as the probability of mutation operator) are too large or too small, the tradeoff between the diversity and convergence of the algorithm will deteriorate. In summary, appropriate probability settings of the crossover and mutation operators will affect the tradeoff between convergence and diversity.

### 5.4.2. Sensitivity analysis of drones

The sensitivity of the proposed model on HV values is analyzed, which includes two crucial factors of drone, i.e., the number of end drones and the payload of end drones. The curves of the mother drone cost and the end drone cost and customer satisfaction curves are presented in Figs. 13(a) and 13(b), respectively.

(*a*) Sensitivity analysis of the number of end drones

With a certain number of customer nodes, the impact of the number of end drones on distance cost and customer satisfaction is analyzed. The number of end drones is increased from 12 to 18. The experimental results of distance cost and customer satisfaction are shown in Fig. 13(a).

In Fig. 13(a), when the number of end drones increases from 12 to 15, the total distance cost is significantly reduced and customer satisfaction is significantly increased. Meanwhile, due to the number of end drones increasing, the distance costs increase. It can be inferred that, limited by the payload and endurance of end drones, a single end drone can deliver more customer nodes within a unit distance as the number of drones will increase from 12 to 15 in this instance, and customer satisfaction will continue to rise. In addition, it can be seen from Fig. 13(a) that as the number of end drones is more than 15, although customer satisfaction increases, the total distance cost also increases. The total distance cost cannot be reduced by continuing to increase the number of end drones. In sum, when the number of end drones is around 15, the total distance cost is relatively low, and the customer satisfaction is relatively high.

(*b*) Sensitivity analysis of the payload of end drones

With a certain number of customer nodes, the impact of the maximum payload of end drones on distance cost and customer satisfaction is analyzed. The flight speeds of mother drone and end drones are experimentally fixed, the payload of end drones is increased from 8 kg to 14 kg, and the impacts on the distance cost and customer satisfaction are analyzed. The results of the payload of the end drone are recorded in Fig. 13(b).

In Fig. 13(b), with the payload of end drones increasing from 8 kg to 10 kg, the total distance cost of heterogeneous multi-drone decreased significantly, while customer satisfaction continued to rise. The results show that increasing the distribution burden of end drones can correspondingly reduce the waste of resources caused by the low distribution efficiency of the mother drone. In particular, when the payload is around 11 kg, the total distance cost is the smallest, and customer satisfaction is relatively high. When the payload of the end drone is larger than 11 kg, the distance cost of the end drone increases even more. In addition, it can be seen from Fig. 13(b) that when the payload of the end drone is 8–11 kg, increasing the end drone's payload can improve the end drone's effective utilization rate. In sum, when the payload of the end drone is 11 kg, the total distance cost is relatively low and the customer satisfaction is relatively high.

(*c*) Managerial insights

Building upon the empirical findings from a real-world instance, this study imparts managerial insights aimed at providing decision-makers with pertinent references.

**Managerial Insight on the number of end drones**. The results, as depicted in Fig. 13(a), reveal an inverse relationship between the number of end drones and the distance cost. Within our VEMOGA method, it is notable that aligning the quantity of end drones with the number of clusters is a viable approach. This strategic coupling not only significantly diminishes computation time but also enhances the utilization rate of end drones, thereby diminishing distance costs and elevating customer satisfaction. Consequently, we recommend that decision-makers consider configuring the number of end drones to be equivalent to the number of clusters.

**Managerial Insight on the payload of end drones**. The outcomes delineated in Fig. 13(b) indicate a U-shaped correlation between the distance cost and the payload of end drones. Consequently, in optimizing practical applications, the selection of end drones should be contingent upon their payload constraints. To this end, it is advisable to utilize end drones with payloads that align appropriately with the constraints of the real-world instance.

## 6. Conclusions

As drone delivery offers unparalleled advantages in terms of energy-saving, flexibility, and timeliness, we address the Heterogeneous Drone Delivery Problem with bi-objective (HDDPBO), namely customer satisfaction and distance cost objectives. In HDDPBO, a mother drone carries $K$ end drones and launches them to $K$ sub-regions, and returns to the depot. Each end drone is responsible for delivering multiple parcels in a sub-region, and flies to the automatic airport for recycling. To effectively solve this model, we propose a novel algorithm named

VEMOGA, in which mainly consists of two modules. The first module involves an improved clustering strategy that divides the distribution region into several sub-regions, with each end drone assigned to deliver parcels within a specific sub-region. The second module proposed an ensemble multi-objective optimization algorithm, in which a voting-based ensemble algorithm is proposed to identify promising Pareto frontier with high-quality convergence and diversity.

To investigate the performance of VEMOGA, extensive experiments are conducted on the three scale synthetic instances, the results show that the proposed VEMOGA can generate high-quality Pareto frontier, outperforming three other comparison algorithms in terms of both customer satisfaction and distance cost. To evaluate the impact of Last-Mile logistics, a real-world experiment was conducted in Changsha. The empirical findings affirm the viability of the novel heterogeneous multi-drone delivery model, which integrates multiple optimization objectives within the context of Last-Mile logistics. The model demonstrates effectiveness in furnishing solutions to Last-Mile logistics challenges, concurrently achieving reduced distance costs and heightened customer satisfaction. Consequently, this study offers a decision-making basis for the Last-Mile logistics in modern logistics.

In the follow-up study, considering the reverse logistics problem for recycling small drones by ground vehicles, and optimizing the problem of forward logistics and reverse logistics simultaneously will be an interesting scientific research problem with challenges and practical significance.

## CRediT authorship contribution statement

**Xupeng Wen:** Writing – original draft, Software. **Guohua Wu:** Writing – review & editing, Funding acquisition, Supervision. **Shuanglin Li:** Writing – review & editing, Investigation, Validation. **Ling Wang:** Methodology, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China under grant number 62373380.

## References

Accorsi, L., & Vigo, D. (2020). A hybrid metaheuristic for single truck and trailer routing problems. *Transportation Science*, *54*, 1351–1371.

Augerat, P., Naddef, D., Belenguer, J., Benavent, E., Corberan, A., & Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem.

Blasco, X., Herrero, J. M., Sanchis, J., & Martínez, M. (2008). A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences*, *178*, 3908–3924.

Cheng, R., Jin, Y., Olhofer, M., & Sendhoff, B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *20*, 773–791.

Das, D. N., Sewani, R., Wang, J., & Tiwari, M. K. (2020). Synchronized truck and drone routing in package delivery logistics. *IEEE Transactions on Intelligent Transportation Systems*, *22*, 5772–5782.

De Jong, K. A., & Spears, W. M. (1992). A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, *5*, 1–26.

Deb, K., & Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, *18*, 577–601.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*, 182–197.

Derigs, U., Pullmann, M., & Vogel, U. (2013). Truck and trailer routing—problems, heuristics and computational experience. *Computers & Operations Research*, *40*, 536–546.

Dondo, R., Méndez, C. A., & Cerdá, J. (2011). The multi-echelon vehicle routing problem with cross docking in supply chain management. *Computers & Chemical Engineering*, *35*, 3002–3024.

Dumas, Y., Desrosiers, J., Gelinas, E., & Solomon, M. M. (1995). An optimal algorithm for the traveling salesman problem with time windows. *Operations Research*, *43*, 367–371.

Guglielmo, C. (2013). Turns out Amazon, touting drone delivery, does sell lots of products that weigh less than 5 pounds. https://www.forbes.com/sites/connieguglielmo/2013/12/02/turns-outamazon-touting-drone-delivery-does-sell-lots-of-products-that-weighless-than-5-pounds.

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2020). A hybrid genetic algorithm for the traveling salesman problem with drone. *Journal of Heuristics*, *26*, 219–247.

Han, Y.-q., Li, J.-q., Liu, Z., Liu, C., & Tian, J. (2020). Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones. *International Journal of Advanced Robotic Systems*, *17*, Article 1729881420920031.

Han, R., Li, W., & Shen, D. (2017). Research on UAV logistics distribution selection based on the different external environments and carrying weight. *Air Transport Bus*, *12*, 51–55.

Hong, F., Wu, G., Luo, Q., Liu, H., Fang, X., & Pedrycz, W. (2023). Logistics in the sky: A two-phase optimization approach for the drone package pickup and delivery system. *IEEE Transactions on Intelligent Transportation Systems*.

research institute, K. (2020). Unmanned delivery feld research report. http://www.199it.com/archives/1072522.html.

Ishibuchi, H., Imada, R., Setoguchi, Y., & Nojima, Y. (2017). Reference point specification in hypervolume calculation for fair comparison and efficient search. In *Proceedings of the genetic and evolutionary computation conference* (pp. 585–592).

Jeong, H. Y., Song, B. D., & Lee, S. (2020). The flying warehouse delivery system: A quantitative approach for the optimal operation policy of airborne fulfillment center. *IEEE Transactions on Intelligent Transportation Systems*, *22*, 7521–7530.

Kumar, V. S., Thansekhar, M., Saravanan, R., & Amali, S. M. J. (2014). Solving multi-objective vehicle routing problem with time windows by FAGA. *Procedia Engineering*, *97*, 2176–2185.

Lemardelé, C., Estrada, M., Pagès, L., & Bachofner, M. (2021). Potentialities of drones and ground autonomous delivery devices for last-mile logistics. *Transportation Research Part E (Logistics and Transportation Review)*, *149*, Article 102325.

Liu, Y., Liu, Z., Shi, J., Wu, G., & Pedrycz, W. (2020). Two-echelon routing problem for parcel delivery by cooperated truck and drone. *IEEE Transactions on Systems, Man & Cybernetics, Part A (Systems & Humans)*, *51*, 7450–7465.

Luo, Q., Wu, G., Ji, B., Wang, L., & Suganthan, P. N. (2021). Hybrid multi-objective optimization approach with pareto local search for collaborative truck-drone routing problems considering flexible time Windows. *IEEE Transactions on Intelligent Transportation Systems*.

Masmoudi, M. A., Mancini, S., Baldacci, R., & Kuo, Y. -H. (2022). Vehicle routing problems with drones equipped with multi-package payload compartments. *Transportation Research Part E (Logistics and Transportation Review)*, *164*, Article 102757.

Mirmohammadsadeghi, S., Masoumik, S. M., & Alavi, S. (2020). Multi-point simulated annealing algorithm for solving truck and trailer routing problem with stochastic travel and service time. *Journal of Soft Computing and Decision Support Systems*, *7*, 14–18.

Moradi, B. (2020). The new optimization algorithm for the vehicle routing problem with time windows using multi-objective discrete learnable evolution model. *Soft Computing*, *24*, 6741–6769.

Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C (Emerging Technologies)*, *54*, 86–109.

Poikonen, S., & Golden, B. (2020). The mothership and drone routing problem. *INFORMS Journal on Computing*, *32*, 249–262.

Qiu, W., Zhu, J., Wu, G., Chen, H., Pedrycz, W., & Suganthan, P. N. (2020). Ensemble many-objective optimization algorithm based on voting mechanism. *IEEE Transactions on Systems, Man & Cybernetics, Part A (Systems & Humans)*.

Rothenbächer, A. -K., Drexl, M., & Irnich, S. (2018). Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows. *Transportation Science*, *52*, 1174–1190.

Sadati, M. E. H., & Çatay, B. (2021). A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem. *Transportation Research Part E (Logistics and Transportation Review)*, *149*, Article 102293.

Salama, M., & Srinivas, S. (2020). Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C (Emerging Technologies)*, *114*, 620–642.

Savuran, H., & Karakaya, M. (2016). Efficient route planning for an unmanned air vehicle deployed on a moving carrier. *Soft Computing*, *20*, 2905–2920.

Schermer, D., Moeini, M., & Wendt, O. (2019). The traveling salesman drone station location problem. In *World congress on global optimization* (pp. 1129–1138). Springer.

Schermer, D., Moeini, M., & Wendt, O. (2020). The drone-assisted traveling salesman problem with robot stations.. In *HICSS* (pp. 1–10).

Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., & Van Woensel, T. (2022). Two-echelon vehicle routing problems: A literature review. *European Journal of Operational Research*.

Tang, L., & Wang, X. (2012). A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, *17*, 20–45.

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., & Velasco, N. (2013). A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, *230*, 231–244.

Wang, K., Pesch, E., Kress, D., Fridman, I., & Boysen, N. (2022). The Piggyback Transportation Problem: Transporting drones launched from a flying warehouse. *European Journal of Operational Research*, *296*, 504–519.

Wang, F., Wang, X., & Sun, S. (2022). A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. *Information Sciences*, *602*, 298–312.

Wang, K., Yuan, B., Zhao, M., & Lu, Y. (2020). Cooperative route planning for the drone and truck in delivery services: A bi-objective optimisation approach. *Journal of the Operational Research Society*, *71*, 1657–1674.

Wen, X., & Wu, G. (2022). Heterogeneous multi-drone routing problem for parcel delivery. *Transportation Research Part C (Emerging Technologies)*, *141*, Article 103763.

Wing (2021). Alphabet's drone delivery service wing hits 100,000 deliveries milestone. https://www.asdnews.com/news/aerospace/2021/08/27/alphabets-drone-delivery-service-wing-hits-100000-deliveries-milestone.

Wu, G., Wen, X., Wang, L., Pedrycz, W., & Suganthan, P. N. (2021). A voting-mechanism-based ensemble framework for constraint handling techniques. *IEEE Transactions on Evolutionary Computation*, *26*, 646–660.

Yu, X., & Gen, M. (2010). *Introduction to evolutionary algorithms*. Springer Science & Business Media.

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*, 712–731.

Zhang, H., Zhang, Q., Ma, L., Zhang, Z., & Liu, Y. (2019). A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows. *Information Sciences*, *490*, 166–190.

Zhou, A., Qu, B. -Y., Li, H., Zhao, S. -Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, *1*, 32–49.