

## Article

# Research on Path Optimization for Collaborative UAVs and Mothership Monitoring of Air Pollution from Port Vessels

Lixin Shen <sup>\*</sup>, Jie Sun and Dong Yang 

School of Shipping Economics and Management, Dalian Maritime University, Dalian 116026, China; 1120210498@dlnu.edu.cn (J.S.); yd\_@dlnu.edu.cn (D.Y.)

\* Correspondence: shenlixin@dlnu.edu.cn

**Abstract:** The seriousness of vessel air pollution has forced the International Maritime Organization (IMO) to introduce a series of relevant laws and regulations. This paper proposes a monitoring scheme based on the synergistic operation of motherships and UAVs. This scheme innovatively adopts a harbor sea patrol vessel or the other official vessel (mothership) as the mobile power supply base for UAVs and realizes efficient and accurate monitoring of vessel air pollution in the pre-monitored area at sea by carrying multiple UAVs. The focus of this paper is on the path optimization problem for multi-UAV collaboration with mothership (MUCWM) monitoring, where the objective is to minimize the total monitoring time for MUCWM. The following three main aspects are studied in this paper: (1) multi-UAV monitoring path optimization; (2) the collaboration mechanism between the mothership and multiple UAVs; and (3) mothership traveling path optimization. In order to effectively solve the above problems, this thesis constructs a path optimization model for multi-UAV collaborative mothership monitoring of air pollution from vessels in port waters; solves the model using the improved adaptive differential evolution (IADE) algorithm; and verifies the effectiveness of the model and the algorithm by using the position data in the Automatic Identification System (AIS) of vessels in Ningbo Zhoushan Port. Through the performance comparison and sensitivity analysis of the algorithm, it is confirmed that the algorithm can effectively solve the path planning problem of the collaborative operation between the mothership and multiple UAVs. The research results in this paper not only help to reduce the air pollution level of harbor vessels and improve the efficiency of sea cruising but also play an important supporting role in the enforcement of relevant emission regulations.



check for updates

**Citation:** Shen, L.; Sun, J.; Yang, D. Research on Path Optimization for Collaborative UAVs and Mothership Monitoring of Air Pollution from Port Vessels. *Sustainability* **2024**, *16*, 4948. <https://doi.org/10.3390/su16124948>

Academic Editor: Giada La Scalia

Received: 26 April 2024

Revised: 21 May 2024

Accepted: 4 June 2024

Published: 9 June 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** vessel atmospheric pollution; UAVs; path optimization; collaborative mechanism; differential evolution algorithm

## 1. Introduction

In recent years, international shipping has rapidly expanded as a low-cost way to transport bulk commodities [1], accounting for over 90% of global trade [2]. According to the United Nations' 2023 Review of Maritime Transport report, maritime trade is expected to grow by 2.4% in the post-pandemic era [3]. However, as the shipping industry flourishes, the environmental issues associated with marine fuels are becoming more apparent [1]. Vessel air pollution is now recognized as the third primary source of pollution, after vehicle exhaust and industrial emissions. [4]. Vessels emit significant pollutants such as carbon dioxide (CO<sub>2</sub>), nitrogen oxides (NO<sub>x</sub>), sulfur oxides (SO<sub>x</sub>), and particulate matter (PM) while navigating. Global data indicate that SO<sub>x</sub> emissions from vessels make up 10–15% of total human activity emissions [5]. These pollutants not only lead to ocean acidification [1,6] but also cause various respiratory diseases in humans. Studies show that vessel pollution has become a major source of air pollution in port cities, particularly from container vessels powered by high-sulfur fuels exceeding 3.50% sulfur content, which severely pollute nearby waters and air [7].

Serious environmental and health concerns have compelled the International Maritime Organization (IMO) to take decisive actions. Since 2015, the IMO has established four Emission Control Areas (ECAs) in the Baltic Sea, the North Sea, North America, and the U.S. Caribbean. The organization intends to establish additional Emission Control Areas (ECAs) in various regions to minimize pollution from vessels [8]. The International Maritime Organization (IMO) now mandates that vessels entering these ECAs utilize fuel containing less than 0.1% m/m sulfur content [9]. In 2020, the IMO's Marine Environment Protection Committee (MEPC) set a global limit on sulfur content in marine fuels at 0.50% m/m [10]. In the same year, the Maritime Safety Administration under China's Ministry of Transportation declared the "2020 Global Marine Fuel Sulfur Restriction Implementation Plan" along with the "Guidelines for the Supervision and Management of Air Pollutant Emissions from Ships". According to these guidelines, the sulfur content of fuel used by vessels in inland waterway emission control areas must be no more than 0.1% m/m. Additionally, the sulfur content in the fuel used by ships operating within coastal air pollution control areas and on international journeys in non-emission control areas should not surpass 0.5% m/m [11]. Despite these regulations, the extra costs of purchasing gas purification devices or using low-sulfur fuels lead some shipping companies to violate these rules to cut costs. Methods include using cheaper, high-sulfur fuels, shutting down pollution control equipment in ECAs, and falsifying vessel logs and fuel consumption records [1]. Historical data indicate that non-compliance rates can reach as high as 12.3% [11].

There are two main traditional methods for monitoring air pollution from vessels. The first is manual boarding, which includes taking oil samples, checking oil change records in the logbook, and using a rapid sulfur detector for marine fuel oil. The second method involves real-time tracking of air quality using land-based environmental monitoring stations. However, both methods have drawbacks. They are inefficient, susceptible to environmental influences, and offer limited monitoring coverage. This is particularly problematic during emergencies such as epidemics, when manual boarding and testing become almost impossible.

UAVs are characterized by high flexibility, no geographical restrictions, and high operability [12]. As UAV technology continues to develop, its use in the shipping industry is becoming increasingly widespread. UAVs are now used for port infrastructure security inspections, maritime oil spill monitoring, ship piloting, maritime smuggling detection, and hazardous chemical inspections at ports [13–17]. Additionally, UAVs have started to be employed in vessel air pollution monitoring. For instance, on 20 May 2020, a DJI M210 UAV, named "Rhinoceros V2" and equipped with a system for detecting air pollution, carried out exhaust gas measurements close to the Ganjiang Marine Office of the Yangzhou Maritime Bureau in China [18]. On 5 January 2021, Zhangjiagang Maritime Bureau used a UAV-mounted gas detector to detect a vessel with excessive sulfur emissions [19]; on 26 June 2022, Changshu Maritime Bureau used a UAV to monitor air pollution from vessels and inspected and found a vessel with an exhaust sulfur value of 3.0% m/m or more [20]; on 20 October 2022, the Port of Rotterdam Authority attempted to apply the new Aera3 UAV for monitoring air pollution emissions from vessels [21]. These instances show that UAVs are a feasible and promising tool for monitoring air pollution from vessels [7].

However, the above-mentioned studies are still nascent, with several critical issues needing resolution:

- (1) The energy constraints of UAVs [22].
- (2) The limitation of the size of the sea area that can be monitored [8].
- (3) The limitation on the number of vessels that can be monitored [8].

To further optimize the research on monitoring air pollution from vessels using UAVs, it is essential to thoroughly explore and address these issues. This paper proposes a monitoring scheme involving a mothership and multiple UAVs working in concert. In this case, the mothership could be a port sea patrol ship or the other official ship. We have developed a path optimization model for MUCWM monitoring of air pollution from vessels in port waters, aiming to minimize the total monitoring time.

The core focus of this paper is the path planning problem for UAVs collaborating with a mothership to monitor air pollution from vessels. This involves three main aspects: (1) optimizing the monitoring paths for multiple UAVs, (2) developing a synergy mechanism between the mothership and multiple UAVs, and (3) optimizing the travel path of the mothership. We have created a path planning model for MUCWM monitoring designed to minimize the total time needed to complete the monitoring task.

The purpose of this study is to explore how the mothership and UAVs can monitor air pollution from port vessels in a collaborative manner. Based on the self-constructed path optimization model for the joint operation of the mothership and UAVs to monitor the air pollution from port vessels and solved by using the improved adaptive differential evolution (IADE) algorithm, the optimization of the monitoring paths of the UAVs, the collaborative work between the mothership and the UAVs, and the optimization of the travel paths of the mothership are implemented. In the case study, the effectiveness of the proposed model and algorithm is verified by using the vessel position data from the Automatic Identification System (AIS) of Ningbo Zhoushan Port. The experimental results show that our algorithm can effectively plan the traveling paths of the mothership and UAVs in collaborative operations, which is of great theoretical significance and application value for the study of UAVs monitoring air pollution from vessels in port.

The contributions of this study are significant: (1) We propose a novel path planning model for UAV collaborative mothership monitoring of air pollution from vessels. This model optimizes paths for both UAVs and the mothership working together. (2) We employ the IADE algorithm to solve the model. We have verified the superiority of this algorithm through performance comparisons and sensitivity analysis. (3) We conducted numerical experiments using vessels in the sea area around Zhoushan Port in Ningbo, China. We gathered position data for 100 ships from the Automatic Identification System (AIS) to validate the feasibility of our algorithm.

The innovations of this paper are as follows: (1) Model innovation: This study proposes a path optimization model for multi-UAV collaborative with mothership (MUCWM) surveillance. The model also optimizes the collaborative monitoring routes of the mothership and multiple UAVs, and this method can effectively reduce the overall operation time and improve monitoring efficiency. Therefore, the solution strategy of path planning in this paper can provide a useful reference for other scholars engaged in related research. (2) Algorithm innovation: In this paper, we propose an improved adaptive differential evolution (IADE) algorithm based on the collaborative operation of the mothership and multiple UAVs, which successfully solves the path planning problem of UAVs collaborating with the mothership to monitor the air pollution of port vessels. After comparative analysis, the algorithm demonstrates strong search capability and convergence speed, which significantly improves the efficiency and performance of the problem, and thus better meets the requirements of practical applications. The algorithm is especially suitable for harbor environments, where fast problem-solving is required.

The remaining sections of this paper are structured as follows: Section 2 outlines the related literature. The methodology for establishing the MUCWM model and implementing the algorithm is detailed in Sections 3 and 4, respectively. Section 5 presents the experiments and the results, with the conclusion following in Section 6.

## 2. Literature Review

### 2.1. Vehicle-Mounted Drone Scheduling

The vehicle-mounted drone scheduling problem is typically modeled as a variant of the Traveler Problem (TSP) or the Vehicle Path Problem (VRP), with the choice depending on the number of vehicles involved. The Traveling Salesman Problem with Drones (TSP-D) addresses scenarios where one vehicle and one or more drones are collaboratively optimized for operation. In contrast, the Vehicle Routing Problem with Drones (VRP-D) applies to cases involving multiple vehicles and multiple drones, all optimized together.

The primary optimization objectives include minimizing the total time, reducing the joint routing costs, and decreasing the customer's waiting time.

In the problem presented in this paper, the mothership can be viewed as a truck for joint distribution with multiple UAVs, and the decision objective is also to minimize the total time for mothership and multi-UAV monitoring, which can be viewed as a variant form of the TSP-D scientific problem.

#### 2.1.1. Vehicle-Mounted Drone Scheduling (TSP-D)

The TSP-D draws inspiration from Murray and Chu [23], who introduced two related problems: the Flight Side Kick Traveling Salesman Problem (FSTSP) and the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP). These aim to minimize the total delivery time for a single vehicle and drone. They proposed two Mixed Integer Programming (MIP) formulations and two simple heuristics, tested on scenarios with up to 10 customers. Agatz et al. [24] explored a similar problem called TSP-D, which differs from FSTSP by allowing the return to previously visited nodes and waiting to receive drones at the launch site. It also considers the drones' maximum flight distance and uses an Integer Linear Program (ILP) to choose the best sequence of operations to minimize completion time.

Bouman et al. [25] developed a dynamic programming-based exact approach to TSP-D, capable of solving much larger instances. Ha et al. [26] introduced a Minimum-Cost TSP-D, aimed at reducing the total transportation cost through two algorithms: a local search-based TSP-LS and a Greedy Random Adaptive Search Procedure (GRASP). Yurek and Ozmutlu [27] devised an iterative optimization algorithm for TSP-D that breaks down the problem into finding truck routes and then optimizing these routes using a mixed-integer linear program. Tu et al. [28] extended the TSP-D to include multiple UAVs traveling with the truck, creating a new variant called "TSP-mD".

#### 2.1.2. Vehicle-Mounted Drone Scheduling (VRP-D)

The VRP-D extends the classic Vehicle Routing Problem (VRP) by allowing both trucks and drones to independently serve customers, creating a many-to-many relationship between these vehicles. Wang et al. [29] first proposed the VRP-D, which focuses on delivering goods using multiple vehicles and drones. The objective is to minimize the total task duration, assigning multiple customers to each drone per dispatch. The model tests several worst-case scenarios to explore the potential time savings from integrating trucks with drones compared to using trucks alone. Poikonen et al. [30] extended this work by considering the battery life of drones and expanding the range of the worst-case scenarios examined. Kitjacharoenchai et al. [31] developed a model that coordinates trucks and multiple drones to synchronize package delivery, aiming to minimize the time it takes for both vehicle types to reach the warehouse. Dayarian et al. [32] introduced a "drone resupply" model, termed the "Vehicle Route Problem for Drone Resupply" (VRPDR). This model involves a collaboration between a group of drones and vehicles to deliver online orders from a fulfillment center to a home. Karak and Abdelghany [33] introduced the "Hybrid Vehicle Drone Routing Problem" (HVDRP), in which several drones are deployed from a central mother ship to execute numerous pickups and deliveries at the same time. Poikonen and Golden [34] recently developed the "k-Multi-visit Drone Routing Problem" (k-MVDRP), which considers tandem connections between trucks and k drones, allowing drones to deliver one or more packages to customers.

Another significant research direction in the VRP-D is the Vehicle Routing Problem with Time Window and Drones (VRPTWD). Guerriero et al. [35] proposed a VRPTWD model that considers a soft time window and focuses on customer satisfaction. Phan and Suzuki [36] investigated a variant of this problem that includes simultaneous receiving and sending constraints with a multi-objective optimization approach. Schermer et al. [37], Wang et al. [38], and Sacramento et al. [39] have also proposed variant forms of the VRP-D.

## 2.2. Research Status of UAV Electric Energy Mobile Supply

Typical approaches to tackle the UAV charging problem involve the use of stationary charging stations and mobile charging vehicles (MCVs) [40]. Recently, significant research has focused on mobile replenishment strategies for UAVs [41]. Yu et al. [42] utilized unmanned ground vehicles (UGVs) to replenish UAVs. Their goal was to allow energy-restricted UAVs to visit a series of stations in the least amount of time, introducing an autonomous UAV charging model grounded in the generalized traveling salesman problem (GTSP). They categorized the problem into three types: Multiple Stationary Charging Stations (MSCS), Single Mobile Charging Stations (SMCS), and Multiple Mobile Charging Stations (MMCS). Inspired by aerial refueling, Zhu et al. [43] introduced a novel concept for aerial charging of UAVs on missions through wireless power transmission. They categorized UAVs into charging UAVs (CUAVs) and mission-performing UAVs (MUAVs) and developed a Deep Reinforcement Learning (DRL)-based scheduling algorithm to optimize the flight paths of both UAV types. Zhou et al. [44] showcased a collaborative network involving UAVs and ground vehicles. In this arrangement, a sub-network of UAVs in the air assists a sub-network of ground vehicles via air-to-air and ground interactions. The ground vehicles function as energy supervisors and chargers for the UAVs. This network structure has been effective in areas like pollution monitoring, disaster response, and information distribution. Luo et al. [45] investigated a two-echelon cooperative routing issue involving ground vehicles (GVs) and their onboard unmanned aerial vehicles (UAVs). In this configuration, the GV functions as mobile support units for the UAVs. They formulated a novel 0–1 integer planning model that accounts for the spatio-temporal cooperative constraints affecting the routes of both GV and UAVs. To address the model, they introduced two heuristic methods: the first creates a comprehensive route covering all targets, while the second devises a travel path for the GV and synchronizes the flight paths of the UAVs.

## 2.3. A Study of Path Planning for Monitoring Air Pollution from Vessels with UAVs

The use of unmanned aerial vehicles (UAVs) to monitor air pollution from vessels has attracted considerable academic interest. Scholars have conducted research in this area: Xia et al. [8] solved the path planning problem for UAV monitoring of air pollution from vessels in emission control areas (ECAs). They proposed a mixed-integer linear programming model based on time-expanded networks and developed a solution method using Lagrangian relaxation techniques. Shen et al. [46] addressed the path planning problem of multiple UAVs for collaborative monitoring of marine air pollution in harbors. They constructed a dynamic multi-objective path planning model and proposed a UAV path planning algorithm for dynamic environments. Yuan et al. [47] developed an enhanced tracking algorithm for UAV gas sensor systems to monitor marine vessel emissions. Sun et al. [1] improved the ACO algorithm to solve the UAV path planning problem for monitoring vessel air pollution. They introduced a hierarchical pheromone update strategy and a partition-based pheromone management mechanism to enhance the traditional ACO algorithm. Luo et al. [48] also developed an improved ACO algorithm for scenarios involving a large number of vessels and UAV base stations. Table 1 summarizes the studies on UAV path planning for monitoring air pollution from vessels. Sun et al. [7] introduced a dynamic scheduling strategy based on reinforcement learning (RL) that takes into account the fluctuations in the speed and direction of ships at sea and thus significantly reduces the flight speed of the UAV.

Currently, most of the research on UAV monitoring of air pollution from vessels focuses on the shore-based replenishment method. This method limits the monitoring range of the UAV, and in this mode, the UAV needs to return to shore frequently to recharge, consuming a large amount of electrical energy for the round trip. This reduces the efficiency of energy utilization and increases the time required for the UAV to complete the monitoring task. To address this current problem, Shen et al.'s study [12] explored the use of a ship as a mobile supply station for UAVs. However, the problems not addressed in this study include the following: (1) the possibility of releasing the UAVs prematurely; (2) the charging time of

the UAV is not considered; (3) the travel route of the loaded ship is determined at the initial stage, and the paths of the loaded ship and the UAVs are not planned at the same time; and (4) the simulated annealing artificial swarm algorithm, which is designed to be influenced by the initial solution, is highly affected by the initial solution.

**Table 1.** Summary of research on UAV monitoring of air pollution from vessels.

Literature	Collaboration Object	Electricity Supply Method	UAV Routing	Carrier Routing	Collaborative Approach
Xia et al. [8]	N	Shore-based	Y	N	N
Shen et al. [22]	N	Shore-based	Y	N	N
Yuan et al. [23]	N	N	N	N	N
Sun et al. [1]	N	Shore-based	Y	N	N
Luo et al. [24]	N	Shore-based	Y	N	N
Sun et al. [7]	N	N	Y	N	N
Shen et al. [12]	Ship	Mobile supply	Y	Y	Step by step
This study	Mothership	Mobile supply	Y	Y	Synchronize

Y: Yes; N: No

#### 2.4. Literature Summary

Currently, research on using UAVs to monitor vessel air pollution is still in the early stages. Although there have been several related studies, challenges such as UAV electrical energy constraints remain. This paper utilizes the mothership as the mobile electric energy supply base for UAVs to address the issue of monitoring vessel air pollution. This approach has relevance to research on vehicle-mounted drone scheduling and UAV mobile resupply. Section 2.1 indicates that the vehicle-mounted UAV scheduling problem typically represents a variant of the Traveling Salesman Problem with Drones (TSP-D) or the Vehicle Routing Problem with Drones (VRP-D). Section 2.2 indicates that most existing studies on mobile supply bases for UAV electrical energy are based on shore, with only a few attempts to utilize a loading ship as a mobile supply base for UAVs. Optimization was carried out to address the possible problems in the only article proposing the study of mobile resupply bases, Shen et al. [12]. In the model of this paper, (1) the release and recovery operations of the UAVs are not synchronized, but are carried out independently according to the actual monitoring needs; (2) the UAV charging time is taken into account for the modeling; (3) the collaborative monitoring routes of the mothership and multiple UAVs are optimized simultaneously, which can minimize the overall operation time; (4) the improved adaptive differential evolution (IADE) algorithm is applied to this problem scenario, the search performance and convergence ability of the algorithm are enhanced by the improved strategy, and finally, the effectiveness of the IADE algorithm is verified by an algorithm performance comparison and sensitivity analysis. The monitoring scheme in this paper provides a more ideal solution for multi-UAV collaborative monitoring of air pollution from vessels in a harbor and improves the monitoring efficiency of air pollution from vessels in harbor waters. Table 1 provides a summary of research on UAV monitoring of air pollution from vessels.

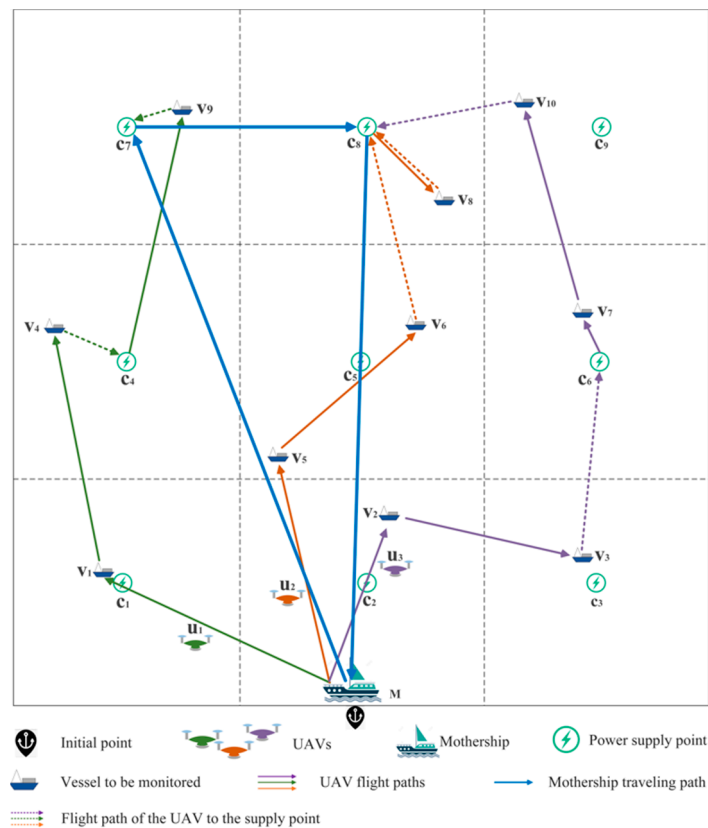
### 3. Construction of MUCWM Model

In this section, we established a path optimization model for multiple UAVs to collaborate with a mothership to monitor vessel air pollution in the port.

#### 3.1. Problem Description

The workflow of the monitoring of air pollution from vessels in the harbor implemented by the mothership in collaboration with multiple shipborne UAVs is shown in Figure 1, where M denotes the mothership,  $v_1-v_{10}$  denote the monitored ships,  $u_1-u_3$  denote the UAVs,  $c_1-c_9$  denote the candidate supply bases, the blue solid line in the figure denotes the traveling path of the mothership, the solid lines of other colors denote the sailing path

of the UAVs, and the dashed lines denote the paths of the UAVs flying to the supply points. First, the mothership carries the UAVs from the harbor shore to cruise along the predetermined monitoring route. The shipborne UAVs take off from the mothership, fly to the location of the vessel to be monitored in the harbor waters, monitor the concentration of its emissions in the vessel's plume, and fly to the location of the next vessel immediately after the monitoring until the energy is not enough to maintain the subsequent monitoring tasks; then they return to the mothership to replenish the electric power and continue to perform the monitoring tasks when all of the monitoring tasks are completed. After completing the monitoring tasks of all the vessels, the UAVs will be recovered and carried away from the harbor area by the mothership and return to the departure point of the harbor.



**Figure 1.** Vessel air pollution monitoring route map.

The main decision problem for the above monitoring effort is how to plan the cruise paths of the mothership and the multiple shipborne UAVs in such a way as to minimize the time of the total monitoring mission. Issues to be considered in this problem include the following:

- (1) How to allocate and optimize the UAV monitoring sequence, that is, the collection of vessels monitored by UAVs and the monitoring sequence.
- (2) How to optimize the driving path of the mothership. This problem involves the selection of the UAV energy supply location.
- (3) How to realize the cooperation between the mothership and UAV. The goal to be achieved in this problem is to achieve collaboration between multiple UAVs and the mothership to minimize the total monitoring time.

### 3.2. Model Assumptions

In the real port environment, it is very complicated to use UAVs to monitor the air pollution of vessels. This paper mainly considers non-environmental factors and the path

optimization problem of vessel air pollution monitoring under relatively ideal conditions. Therefore, the following hypotheses are put forward:

- (1) Both the mothership and the UAV can move freely in  $R^2$  (Euclidean plane) (that is, within the port sea area).
- (2) Both the mothership and the UAV sail in a straight line.
- (3) The UAV travels at maximum speed.
- (4) The charging time of each UAV and the time for monitoring the vessel to be monitored are fixed.
- (5) Each UAV can monitor multiple vessels, and each vessel can only be monitored once.
- (6) The position of the vessel to be monitored is considered unchanged.
- (7) The speed of the UAV > the speed of the mothership.
- (8) Ignore the flight distance of the UAV during the take-off and landing phases.
- (9) The mothership's fuel and the power to replenish the UAV are unlimited.

### 3.3. Symbol Description

The following lists the definitions and expressions of parameters and variables used in constructing a multi-UAV collaborative mothership monitoring port vessel air pollution path optimization model:

The auxiliary variables are shown in Table 2.

**Table 2.** Model symbol description.

Symbol	Meaning
$N_C$	Number of supply points (equal area grids).
$N_U$	Number of UAVs.
$N_V$	Number of vessels to be monitored.
$C$	$C = \{c_1, c_2, \dots, c_C\}$ is a set of $N_C$ supply points, $c_i$ is the $i^{\text{th}}$ supply point, $i \in \{1, 2, 3, \dots, N_C\}$ , $c_i \in C$ .
$U$	$U = \{u_1, u_2, \dots, u_{N_U}\}$ is a set of $N_U$ UAVs, $u_i$ is the $i^{\text{th}}$ UAV, $i \in \{1, 2, 3, \dots, N_U\}$ , $u_i \in U$ .
$V$	$V = \{v_1, v_2, \dots, v_{N_V}\}$ is the set of $N_V$ vessels to be monitored, $v_i$ is the $i^{\text{th}}$ vessel to be monitored, $i \in \{1, 2, 3, \dots, N_V\}$ , $v_i \in V$ .
$Sta$	Initial point.
$d_{(m_i, n_j)}$	The distance between $m_i$ node and $n_j$ node, where $m_i$ and $n_j$ can be the location of the vessel to be monitored or the location of the supply point, $m_i, n_j \in \{V \cup C\}$ .
$(v_i, v_j) \in E_V$	$(v_i, v_j)$ represents the arc connecting the vessel nodes $v_i, v_j$ to be monitored, $v_i, v_j \in V$ ; $E_V$ is the set of arcs between the vessels to be monitored.
$(c_i, c_j) \in E_C$	$(c_i, c_j)$ represents the arc connecting supply points $c_i, c_j$ , $c_i, c_j \in C$ ; $E_C$ is the set of arcs between supply points.
$T_U$	Charging time of UAV.
$T_{max}$	Maximum endurance of UAV.
$T_V$	Time for the UAV monitoring of the vessel.
$t_{(u_i, c_j)}$	The time when UAV $u_i$ arrives at supply point $c_j$ , $u_i \in U$ , $c_j \in C$ .
$t_{(u_i, c_j)}^M$	The time when the mothership arrives at the supply point $c_j$ to serve the UAV $u_i$ , $u_i \in U$ , $c_j \in C$ .
$l_M$	The last supply node passes by mothership, $l_M \in C$ .
$l_{u_i}$	The last supply node passes by UAV $u_i$ , $l_{u_i} \in C$ , $u_i \in U$ .
$S_M$	The speed of the mothership.
$S_U$	The speed of the UAV.
$p_{(u_i, v_j)}$	The previous supply point when UAV $u_i$ travels to vessel node $v_j$ , $u_i \in U$ , $v_j \in V$ .
$L_{(u_i, q_j)}$	The distance traveled by the UAV $u_i$ when it is fully charged and travels to $q_j$ , $u_i \in U$ , where $q_j$ can be the location of the supply point or the location of the vessel $q_j \in \{V \cup C\}$ .
$N_{(u_i, q_j)}^V$	The number of vessels monitored when UAV $u_i$ is fully charged and travels to $q_j$ , $u_i \in U$ , where $q_j$ can be the location of the supply point or the location of the vessel $q_j \in \{V \cup C\}$ .
$x_{(v_j, v_k)}^{u_i} = \begin{cases} 1 \\ 0 \end{cases}$	A value of 1 means that the UAV $u_i$ passes through the arc $(v_j, v_k)$ between vessel nodes, 0 means no.
$y_{(v_j, v_k)}^{u_i} = \begin{cases} 1 \\ 0 \end{cases}$	A value of 1 means that the UAV $u_i$ needs to insert a supply point between vessel nodes $v_j, v_k$ , 0 means no.



Table 2. Cont.

Symbol	Meaning
$z_{(c_i, c_j)} = \begin{cases} 1 \\ 0 \end{cases}$	A value of 1 means that the mothership passes through the arc $(c_i, c_j)$ between supply point nodes, 0 means no.
$w_{(u_i, v_j)} = \begin{cases} 1 \\ 0 \end{cases}$	A value of 1 means that the vessel node $v_j$ is monitored by the UAV $u_i$ , 0 means no.

### 3.4. MUCWM Path Optimization Model

This paper chooses to construct the objective function based on the travel time of the mothership. The total time to complete the monitoring task consists of three parts:

(1) The time spent by the mothership while traveling. (2) The time spent by the mothership waiting for the UAV. (3) The time spent by the mothership charging the UAV. Based on this, a path optimization model for UAVs to collaborate with the mothership to monitor air pollution from vessels in ports is constructed:

Obj:

$$\min T_{total} = \frac{\sum_{(c_i, c_j) \in E_C} (d_{(c_i, c_j)} \times z_{(c_i, c_j)}) + d_{(l_M, Sta)}}{S_M} + \sum_{u_i \in U} \sum_{c_j \in C} (t_{(u_i, c_j)} - t_{(u_i, c_j)}^M) + \sum_{u_i \in U} \sum_{(v_j, v_k) \in E_V} y_{(v_j, v_k)}^{u_i} \times T_U \quad (1)$$

The multi-UAV path optimization model constraints are as follows:

$$\sum_{u_i \in U} \sum_{(v_j, v_k) \in E_V} x_{(v_j, v_k)}^{u_i} = N_U, \quad i = 1, 2, \dots, N_U, \quad j, k = 1, 2, \dots, N_V \quad (2)$$

$$\sum_{u_i \in U} \sum_{v_j \in V} x_{(Sta, v_j)}^{u_i} = 1, \quad i = 1, 2, \dots, N_U, \quad j = 1, 2, \dots, N_V \quad (3)$$

$$\sum_{u_i \in U} \sum_{v_j \in V} x_{(v_j, l_{u_i})}^{u_i} = 1, \quad i = 1, 2, \dots, N_U, \quad j = 1, 2, \dots, N_V \quad (4)$$

$$\sum_{u_i \in U} x_{(v_j, v_k)}^{u_i} = \sum_{u_i \in U} x_{(v_k, v_j)}^{u_i}, \quad i = 1, 2, \dots, N_U, \quad j, k = 1, 2, \dots, N_V, \quad j \neq k, \quad \forall (v_j, v_k) \in E_V \quad (5)$$

$$\sum_{u_i \in U} w_{(u_i, v_j)} = 1, \quad j = 1, 2, \dots, N_V \quad (6)$$

The multi-UAV collaborative with mothership optimization model constraints are as follows:

$$0 < d_{(v_j, c_n)} \times y_{(v_j, v_k)}^{u_i} \times z_{(c_m, c_n)} < T_{max} - L_{(u_i, v_j)} / S_U - N_{(u_i, v_j)}^V \times T_V \text{ and } c_m = p_{(u_i, v_j)}, \quad \forall u_i \in U, \quad i = 1, 2, \dots, N_U, \quad \forall (v_j, v_k) \in E_V, \quad j, k = 1, 2, \dots, N_C, \quad \forall (c_m, c_n) \in E_C, \quad m, n = 1, 2, \dots, N_V \quad (7)$$

$$z_{(c_j, c_k)} \times [t_{(u_i, c_j)} - t_{(u_i, c_j)}^M] > 0, \quad \forall u_i \in U, \quad i = 1, 2, \dots, N_U, \quad \forall (c_j, c_k) \in E_C, \quad j, k = 1, 2, \dots, N_C \quad (8)$$

$$z_{(c_j, c_k)} \times [(t_{(M, c_k)} - t_{(u_i, c_j)}) \times S_U] - d_{(c_j, c_k)} > 0, \quad \forall u_i \in U, \quad i = 1, 2, \dots, N_U, \quad \forall (c_j, c_k) \in E_C, \quad j, k = 1, 2, \dots, N_C \quad (9)$$

$$x_{(v_j, v_k)}^{u_i} \in \{0, 1\}, \quad \forall u_i \in U, \quad i = 1, 2, \dots, N_U, \quad \forall (v_j, v_k) \in E_V, \quad j, k = 1, 2, \dots, N_C \quad (10)$$

$$y_{(v_j, v_k)}^{u_i} = \begin{cases} 1 & 0 < (T_{max} - L_{(u_i, v_j)} / S_U - N_{(u_i, v_j)}^V \times T_V - d_{(v_j, v_k)}) \times x_{(v_j, v_k)}^{u_i} < d_{(v_k, p_{(u_i, v_j)})} \\ 0 & (T_{max} - L_{(u_i, v_j)} / S_U - N_{(u_i, v_j)}^V \times T_V - d_{(v_j, v_k)}) \times x_{(v_j, v_k)}^{u_i} \geq d_{(v_k, p_{(u_i, v_j)})} \end{cases}, \quad \forall (v_j, v_k) \in E_V, \quad j, k = 1, 2, \dots, N_C, \quad \forall u_i \in U, \quad i = 1, 2, \dots, N_U \quad (11)$$

$$z_{(c_i, c_j)} \in \{0, 1\}, \quad \forall (c_i, c_j) \in E_C, \quad i, j = 1, 2, \dots, N_C \quad (12)$$

$$w_{(u_i, v_j)} \in \{0, 1\}, \forall u_i \in U, i = 1, 2, \dots, N_U, \forall v_j \in V, j = 1, 2, \dots, N_V \quad (13)$$

Constraint (2) means that there are  $N_V$  UAVs in total that start from the port's initial point at the same time to monitor the air pollution of vessels in the port sea area. Constraint (3) is the starting route constraint of each UAV, which means that each UAV can only start from the starting point of the port once. Constraint (4) is the constraint that each UAV passes through the last supply point, which means that each UAV can only reach the last supply node once. Constraints (3) and (4) simultaneously constrain each UAV to have only one flight path. Constraint (5) is used to ensure the continuity of the flight path of each UAV; that is, each monitored vessel node has the same degree of entry and exit for its monitoring UAV. Constraint (6) is used to ensure that each monitored vessel can be monitored by a UAV and can only be monitored once. Constraint (7) is the optional supply point constraint and the service sequence constraint provided by the mothership for UAV charging. Constraint (8) is the time window constraint for the mothership to arrive at the supply point  $m$ , which is used to ensure that the mothership must arrive at the target supply point to replenish power for the UAV before the UAV's power is consumed. Constraint (9) is used to determine whether the mothership can reach the target supply node at its maximum speed within the remaining endurance of the UAV. Constraints (10)–(13) are  $\{0, 1\}$  variable constraints, indicating the value range of the decision variable.

#### 4. Model Solution Based on Improved Adaptive Differential Evolution Algorithm

In this paper, an improved DE algorithm is utilized to solve the model. In order to make the DE algorithm more applicable to the problem scenarios in this paper, adaptive and arithmetic improvements are made on the basis of the inherent advantages of the original DE algorithm to increase its performance and ability to solve the problems in this paper, and the improved algorithm is referred to as the improved adaptive differential evolution (IADE) algorithm.

##### 4.1. Adaptive Parameter Settings

(1) Adopt the lifespan mechanism and extinction mechanism for population NP.

This mechanism regulates population size and maintains diversity. The lifespan mechanism operates as follows: Initially, the age of all individuals is set to zero, aligning with the number of iterations. In each generation, when a new individual is created, its lifespan is assigned based on the value of its fitness function. An individual is removed from the population once its age exceeds its predetermined lifespan. The formula for calculating lifespan is as follows:

$$\begin{cases} LF_{min} + \eta \times \frac{f(x_i) - f_{min}}{f_{avg} - f_{min}} & f_{avg} \geq f(x_i) \\ \frac{1}{2} \times (LF_{min} + LF_{max}) + \eta \times \frac{f(x_i) - f_{avg}}{f_{max} - f_{avg}} & f_{avg} < f(x_i) \end{cases} \quad (14)$$

Among them,  $f_{avg}$  represents the average value of the fitness function value of the current population and  $f_{min}$  and  $f_{max}$  represent the minimum value and maximum value of the fitness function in the current population, respectively. The fitness function value of the algorithm in this article is the minimum value of the total time consumption of the monitoring task. That is,  $f = T_{total}$ .  $LF_{max}$  represents the maximum value that the life span can reach, that is, the maximum number of iterations, and  $LF_{min}$  represents the minimum value that the life span can reach. At the same time,  $\eta = \frac{1}{2} \times (LF_{max} - LF_{min})$ . The extinction mechanism strategy is as follows: When the population fitness function value does not change for  $p \times NP$  times in a row, it means that the population in the current iteration may fall into a local optimum. At this time, the extinction mechanism strategy can be used to increase the diversity of the population. The parameter  $p$  is the extinction coefficient. Generally, the larger the extinction coefficient, the higher the probability of an individual being eliminated or reset; the smaller the extinction coefficient, the higher the probability of an individual being retained. This article sets the extinction coefficient

$p = 0.3$  to ensure the timely generation of new individuals. After an individual becomes extinct, a new population of individuals needs to be generated. Here, elite individual replication is used to generate a new population of individuals. The specific formula is as follows:

$$num = \zeta \cdot (NP_{max} - NP_G) \quad (15)$$

$\zeta$  is the replication rate. This article sets the replication rate  $\zeta = 0.9$  to ensure the majority of elite individuals.  $NP_{max}$  is the maximum value of the population size, and  $NP_G$  is the current population size.

When the lifespan mechanism comes into play, the diversity of the population may decline sharply, which increases the risk of the population falling into a local extreme. Once this situation occurs, the extinction mechanism will be activated to enhance the diversity of the population by eliminating some individuals and using the replication strategy of elite individuals. This mechanism helps prevent premature convergence of the population, thereby improving the search efficiency and convergence of the algorithm. Therefore, the lifespan mechanism and the extinction mechanism collaborate with each other to realize the adaptive adjustment of  $NP$  at different stages.

(2) Introducing an adaptive parameter adjustment strategy of Beta distribution for scaling factor  $F$  and crossover probability  $CR$ .

Beta distribution is a common continuous probability distribution in the field of probability statistics, and its value range is limited to the  $[0, 1]$  interval. The Beta distribution is often used to describe the probability distribution characteristics of random variables, such as describing the prior distribution of probability, proportion, or parameters. In many fields, such as machine learning and mathematical statistics, Beta distribution plays an important role. The specific form of the Beta distribution is given by its probability density function, which describes the probability distribution characteristics of the Beta distribution in detail. The probability density function of the Beta distribution is as follows:

$$f(x|a, b) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(a, b)} \quad (16)$$

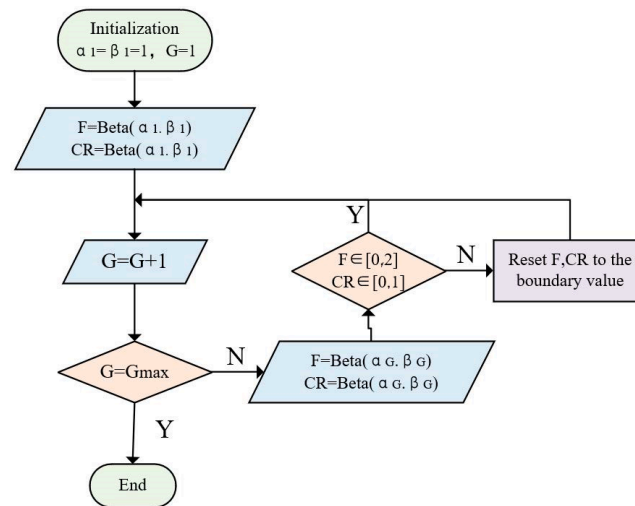
where  $x$  is the value of the random variable,  $\alpha$  and  $\beta$  are the two shape parameters of the Beta distribution, and  $B(a, b)$  is the Beta function:  $B(a, b) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1} dx$ . When  $\alpha = \beta = 1$ , the Beta distribution degenerates into a uniform distribution. When  $\alpha$  and  $\beta$  are large, the Beta distribution presents a bell-shaped shape concentrated in the middle. When  $\alpha$  and  $\beta$  are small, the Beta distribution presents a long-tail shape.

This article introduces a method of dynamically adjusting  $F$  and  $CR$ ; that is, the dynamic adaptive adjustment method of Beta distribution is introduced to improve the performance of the algorithm. Specifically, the Beta distribution is used to generate random numbers, and then the generated random numbers are mapped to the value ranges of  $F$  and  $CR$ . Through this dynamic adjustment method, the DE algorithm is able to select appropriate values of  $F$  and  $CR$  at different stages. In the initial state, we set  $\alpha_1 = \beta_1 = 1$  so that the probability distribution function presents a  $(0, 1)$  uniform distribution. During the population evolution process, the parameters  $F$  and  $CR$  change based on the following Beta probability distribution formula:

$$f(x|\alpha_G, \beta_G) = \frac{x^{\alpha_G-1}(1-x)^{\beta_G-1}}{B(\alpha_G, \beta_G)} \quad (17)$$

where  $\alpha_G$  is the number of individuals with better performance in the current  $F$  (or  $CR$ ) (such as successful evolution or better fitness function value), and  $\beta_G$  is the number of individuals with poorer performance in the current  $F$  (or  $CR$ ) (such as failure to evolve successfully or making the fitness function value better, which does not make the fitness function value more optimal). We substitute the number of individuals with better performance  $\alpha_G$  and the number of individuals with poor performance  $\beta_G$  in the  $G$  generation

population into Equation (16), respectively, to calculate the values of  $CR$  and  $F$  in generation  $G + 1$ , and make constraint boundaries for  $F$  and  $CR$  ( $F \in [0, 2]$ ,  $CR \in [0, 1]$ ), so that  $F$  and  $CR$  can change adaptively according to the iteration results, effectively improving the convergence speed of the algorithm and the diversity of the population. The calculation process of adaptive  $F$  and  $CR$  is shown in Figure 2.



**Figure 2.** Calculation process of adaptive  $F$  and  $CR$ .

#### 4.2. Hybrid Mutation Improvement Strategy

The traditional differential evolution algorithm has limitations in mutation improvement strategies, which limits its versatility. To address this problem, this paper proposes a mutation strategy based on “DE-BB” (differential evolution bare bones) to optimize mutation operations. This strategy is an innovative combination of the backbone particle swarm algorithm and the differential evolution algorithm. Experimental results show that this strategy effectively utilizes the advantages of the backbone particle swarm optimization algorithm in the local depth search, thereby improving the convergence accuracy of the differential evolution algorithm. In the strategy, we incorporate elements of the classic differential mutation and combine them with the adaptive improvement strategy of the  $F$ -value to generate individual trial vectors. The specific mutation improvement strategy formula is as follows:

$$u_{i,j,G} = \begin{cases} p_{i,j,G} + rand \times (x_{r_1,j,G} - x_{r_2,j,G}) & \text{if } rand(0, 1) \leq CR \\ x_{i,best,G} + F \times (x_{r_1,j,G} - x_{r_2,j,G}) + F \times (x_{r_3,j,G} - x_{r_4,j,G}) & \text{otherwise} \end{cases} \quad (18)$$

In the formula,  $r_1 \sim r_4$  are random integers that are different from each other and randomly selected from the population set  $\{1, 2, \dots, NP\}$ , and  $F$  is the scaling factor, where  $p_{i,j,G} = r_j \times pbest_{i,j,G} + (1 - r_j) \times NPbest_{j,G}$ , where  $r_j$  is also a random decimal in the interval  $[0, 1]$ .  $pbest_{i,j,G}$  is the historical local optimal value of the individual, and  $NPbest_{j,G}$  is the optimal individual value of the current population.

#### 4.3. Hybrid Cross Improvement Strategy

In traditional differential evolution algorithms, the binomial crossover method (binomial) is often used to cross the contribution vector  $v$  generated by the mutation operation with the current target vector  $x$  to generate a test vector  $u$ . Subsequently, the algorithm competes between the trial vector  $u$  and the target direction  $x$  through a greedy selection mechanism, and only the solution vector with better fitness is retained to enter the next round of iteration, while the worse solution vector is eliminated. However, when the contribution vector  $v$  is already very close to the global optimal position, the crossover operation may destroy its complete solution structure. If the test vector  $u$  can fully absorb

the information of vector  $v$ , it will help increase the diversity of potential solutions and reduce the risk of the algorithm falling into a local optimum. To this end, this article quotes the crossover strategy  $uu_{i,t}$  of the multi-selection mechanism and combines it with the adaptive improvement strategy of CR to obtain the crossover improvement strategy formula as follows:

$$uu_{i,j,G+1} = \begin{cases} u_{i,j,G} & \text{if } rand \leq \theta \text{ and } rand \leq CR \\ v_{i,j,G} & \text{otherwise} \end{cases} \quad (19)$$

where  $\theta$  represents the mixed crossover probability threshold;  $u_{i,j,G}$  represents the value of the  $j$  dimension of individual  $i$ , calculated by the traditional binomial crossover method.  $\theta$  represents the choice probability of performing a binomial crossover or a fully absorbing crossover. According to the experiment of the proposer of this strategy, when the value of parameter  $\theta$  is set in the range of [0.4, 0.6], the average convergence speed and convergence accuracy of the algorithm are better than those in other intervals. Therefore, this article sets the parameter  $\theta$  to 0.5.

#### 4.4. IADE Algorithm Solution Steps

##### 4.4.1. Codec Strategy

The encoding method used in this article is natural number encoding. Each individual is composed of two chromosomes. The natural number  $x_k$  in the first chromosome represents the serial number of the UAV, that is,  $(x_1, x_2, \dots, x_k, \dots, x_{N_V})$ , where  $N_V$  represents the number of monitored vessels; the natural number  $y_k$  in the second chromosome represents the serial number of the monitored vessel; that is,  $(y_1, y_2, \dots, y_k, \dots, y_{N_V})$  also represents the number of monitored vessels. Among them, the upper and lower correspondence between chromosomes represents the vessel to be monitored by the UAV. The chromosome encoding strategy is shown in Figure 3. The UAV with serial number 1 corresponds to the serial number of the monitored vessel {3, 8, 1, 10}, which means that the order of the vessels monitored by UAV No. 1 is {3→8→1→10}.

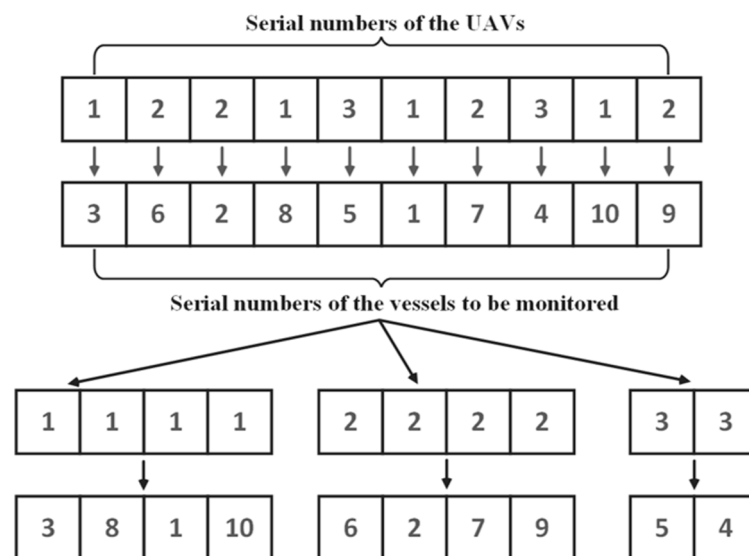


Figure 3. Chromosome coding strategy.

Decoding requires designing an appropriate decoding solution based on the characteristics of the problem and converting the encoding after the algorithm is executed into the solution to the problem. The specific steps are as follows:

Step 1: According to the order in which each UAV monitors the vessel, consider the UAV endurance time  $T_{endu}$ . If, before the UAV goes to the location of the next monitored vessel,  $T_{endu} < d_{(v_{now}, v_{next})} + d_{(v_{next}, c_{pre})} + T_V$ , that is, the UAV cannot return to the supply

point after monitoring the next vessel, it is necessary to insert a new supply point gene after the gene of the current vessel position, as shown in Figure 4.  $T_{endu}$  represents the remaining range of the UAV,  $d_{(x,y)}$  represents the distance between the two positions  $x$  and  $y$ ,  $v_{now}$  represents the current location of the vessel being monitored by the UAV, and  $seq_{next}$  represents the position of the next vessel that needs to be monitored, and  $c_{pre}$  represents the location of the previous charging point of the UAV;  $T_V$  represents the monitoring time.

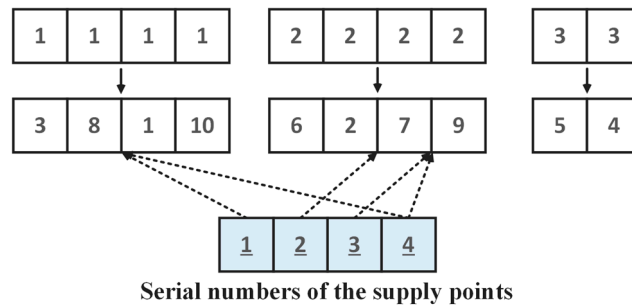


Figure 4. Replenishment point gene insertion strategy.

Step 2: Select a charging point that meets the conditions for the UAV based on the remaining power of the UAV. If  $T_{endu} > d_{(v_{now}, c)}$  is satisfied, add the current charging point to the gene sequence and add the feasible charging points to the UAV monitoring sequence that is added to the UAV feasible monitoring sequence set  $r_u$ , as shown in Figure 4. Finally, a recovery and supply point need to be added at the end of the genetic sequence of each UAV. After the UAV completes its monitoring mission, the mothership is used to recover it.

Step 3: According to the time window constraint in  $u\_route$ , that is,  $t_{c_{now}} > t + d_{(c_{now}, c_{pre})}$ , select the charging point driving path that meets the mothership speed constraint, as shown in ①→②→⑤ in Figure 5 and the ①→④→⑤ sequence.  $t_c$  represents the maximum time point when the UAV reaches supply point  $c$  and hovers above it,  $t$  represents the current time point, and  $S_M$  represents the speed of the mothership.

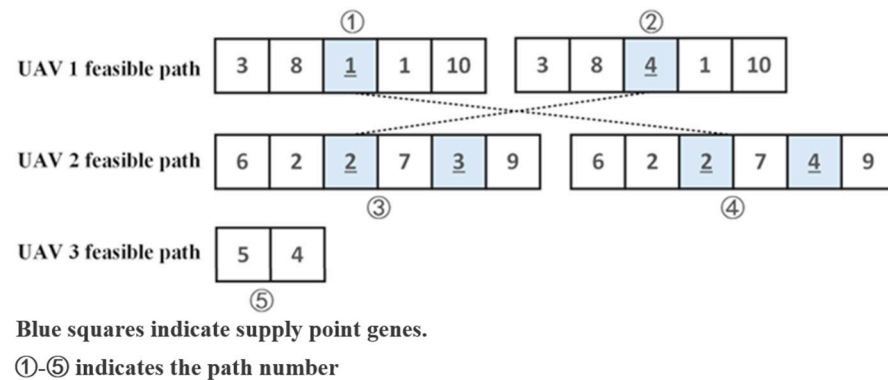


Figure 5. Viable paths for UAV monitoring.

Step 4: Calculate the total duration of the monitoring task based on the charging point driving path of the mothership, return the fitness function value, and end.

Table 3 gives the solution form of MUCWM path optimization.

Table 3. Solution form of MUCWM path planning.

UAV monitoring sequence	$u_1$ $u_2$ $u_3$	$v_3 \rightarrow v_8 \rightarrow c_1 \rightarrow v_1 \rightarrow v_{10}$ $v_6 \rightarrow v_2 \rightarrow c_2 \rightarrow v_7 \rightarrow c_4 \rightarrow v_9$ $v_5 \rightarrow v_4$
Mothership sailing sequence	$M$	$c_2 \rightarrow c_1 \rightarrow c_4$
Minimum time consumption of monitoring tasks		$min T_{total}$

#### 4.4.2. Algorithm Process

The process of implementing the IADE algorithm is as follows:

Step 1: Initialize parameters, including population size  $NP$ , extinction coefficient  $p$ , elite individual replication rate  $\zeta$ , individual component value range  $bounds$ , mutation scaling factor  $F$ , shape parameters  $\alpha_1, \beta_1$ , crossover probability  $CR$ , mixed crossover probability threshold  $\theta$ , the maximum number of iterations  $G_{max}$ .

Step 2: Generate an initial population, with each individual containing 10D parameters, and the value range of the parameters is  $bounds$ .

Step 3: Decode the initial population, multiply the first dimension of the individual by  $N_U$ , and perform a rounding operation to determine the set of vessels monitored by the UAV.  $N_U$  represents the number of UAVs and corresponds to the second-dimension parameters of the individual. Attach a set of sequences  $[1, 2, \dots, N_V]$  to the position, sort the parameters from small to large, retain the attached serial number in the sorting, and use this sorting as the order of the UAV monitoring vessel collection;  $N_V$  means the number of vessels to be monitored.

Step 4: Let  $G = 1$  represent the first iteration.

Step 5: Let  $i = 1$  represent the first individual.

Step 6: Mutation operation: randomly select 4 individuals  $x_{r_1}, x_{r_2}, x_{r_3}$ , and  $x_{r_4}$  in the population, and generate mutated individuals according to the mixed mutation improvement strategy of Formula (19)  $v_{i,G}$ .

Step 7: Crossover operation: according to the hybrid crossover improvement strategy of Formula (19), generate new test individuals  $u_{i,G}$ .

Step 8: Greedy selection: calculate the fitness function value (minimum value of monitoring task completion time); if the fitness function value  $f(u_{i,G}) < f(x_{i,G-1})$ , update the individual of the previous generation  $x_{i,G-1} = u_{i,G}$ , otherwise the individual of the previous generation is retained.

Step 9: If  $i < NP$ , let  $i = i + 1$  and return to step 5, otherwise go to step 10.

Step 10: Adaptive parameter adjustment. According to the fitness function value, use Formulas (15) and (16) to calculate the adjusted population; use Formula (18) to calculate the adjusted mutation scaling factor  $F$  and crossover probability  $CR$ .

Step 11: If  $G < G_{max}$ ,  $G = G + 1$ , go to step 4, otherwise go to step 11.

Step 12: Output the historical optimal solution  $f$  and the algorithm ends.

The pseudo-code of the improved adaptive differential evolution algorithm is shown in Algorithm 1.

---

#### Algorithm 1: Improved adaptive differential evolution algorithm

---

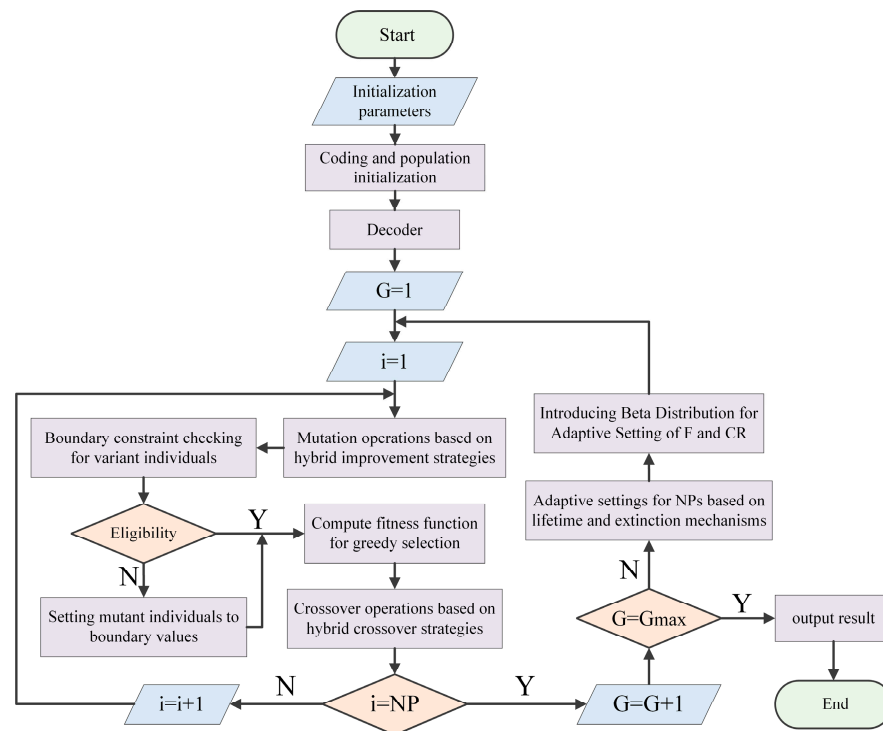
```

1 Begin
2 Initialize related parameters:  $NP, p, \zeta, bounds, F, \alpha_1, \beta_1, CR, \theta, G_{max}$ 
3 Generate initial population
4 Decoding
5 for  $G$  in  $G_{max}$ 
6   for  $i$  in  $NP$ 
7      $v_{i,G}$  = The optimal individual and four random individuals generate mutant individuals
       through Formula (17)
8     Boundary constraint check after mutation
9      $u_{i,G}$  = Generate new test individuals through crossover using Formula (18)
10    if  $f(u_{i,G}) < f(x_{i,G-1})$  then
11       $x_{i,G-1} = u_{i,G}$ 
12    end if
13  end for
14   $NP$  = Calculate the adjusted population by Formulas (14) and (15)
15   $F, CR$  = scaling factor and crossover probability adaptively adjusted through Formula (16)
16 end for
17 Output the optimal solution  $f$ , the optimal individual  $NP_{best}$ , the algorithm execution time
   and the algorithm convergence chart
18 Finish

```

---

The flow chart of the algorithm is shown in Figure 6.



**Figure 6.** Flowchart of improved adaptive differential evolution algorithm.

#### 4.5. Algorithm Design for MUCWM Path Optimization Problem

This section mainly describes the basic ideas for solving the MUCWM path optimization problem in three aspects: optimization of UAV monitoring paths, collaboration between motherships and UAVs, and optimization of mothership monitoring paths.

##### 4.5.1. UAV Monitoring Path Optimization

First, the mothership carries multiple UAVs and sets off from the port starting point  $Sta$  at the same time. The UAVs will take off from the mothership at the beginning to perform monitoring tasks. At this time, monitoring needs to be planned for multiple UAVs. To achieve this goal, two problems need to be solved: (1) How to determine the collection of UAVs to monitor vessels; (2) how to determine the order of UAVs to monitor vessels.

Ignoring the power constraints of the UAV, that is, treating the UAV power as infinite, we randomly assign vessel collections and monitoring sequences to the UAV. The specific strategy is as follows: First, add an extra dimension  $[1, 2, \dots, N_U]$  sequence to the test individual  $u_{i,G}$ . Secondly, after multiplying the values of the first dimensional elements of the test individual  $u_{i,G}$  by  $N_U - 1$ , respectively, since the value range of the individual elements belongs to  $[0, 1]$ , and after rounding off the elements and adding 1, the value range of the first dimension element value of  $u_{i,G}$  will become  $[1, N_U]$ ,  $q$ , where  $N_U$  is the number of UAVs, and then we sort the second-dimensional elements of  $u_{i,G}$  from large to small. During the sorting process, the corresponding positions of the additional sequences remain unchanged. Through the above two strategies, the random allocation of the UAV monitoring vessel collection and monitoring sequence is achieved. The random allocation algorithm for UAV monitoring is shown in Algorithm 2.



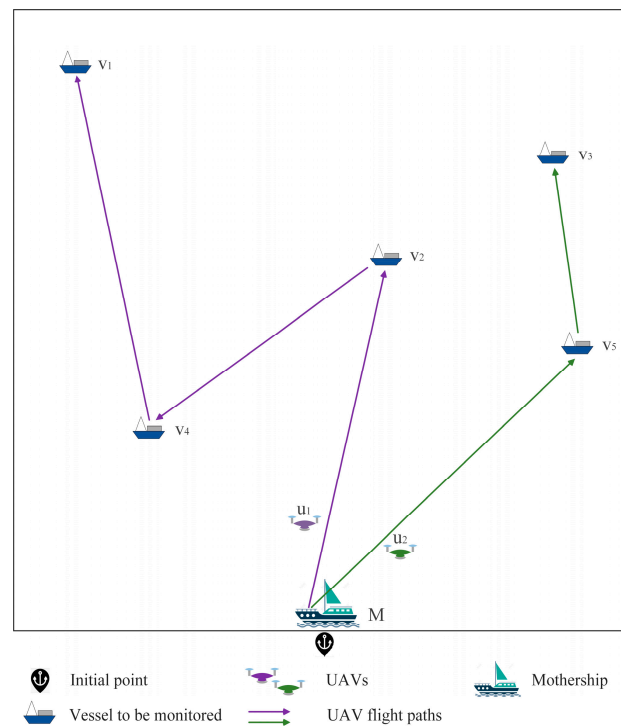
**Algorithm 2:** Random allocation algorithm for UAV monitoring

```

1  $u_{i,G(1)} = \text{np.round}(N_U - 1)$ 
2  $seq = \text{np.arange}(N_V).reshape(1, N_V)$ 
3  $u_{i,G} =$  The trial individual formed by appending  $seq$  to the third dimension of  $u_{i,G}$ 
4
5  $unique = \text{np.unique}(u_{i,G(1)})$ 
6 for  $k$  in  $\text{enumerate}(unique)$ 
7    $u_{i,G} = \text{Split } unique\_values$  according to the value of  $unique\_values$ 
8 end for
9 Sequential allocation strategy for UAV monitoring of vessels:
10 for  $u_{i,G(2)(m)}$  in  $u_{i,G(2)}$ 
11    $u_{i,G} = \text{Sort}$  according to the value of  $u_{i,G(2)(m)}$ 
12 end for

```

Figure 7 shows a random allocation strategy in a scenario with 2 UAVs and 5 monitored vessels. The sequence of UAV monitoring vessels is divided into the following:  $u_1 : v_2 \rightarrow v_4 \rightarrow v_1$ ,  $u_2 : v_5 \rightarrow v_3$ .

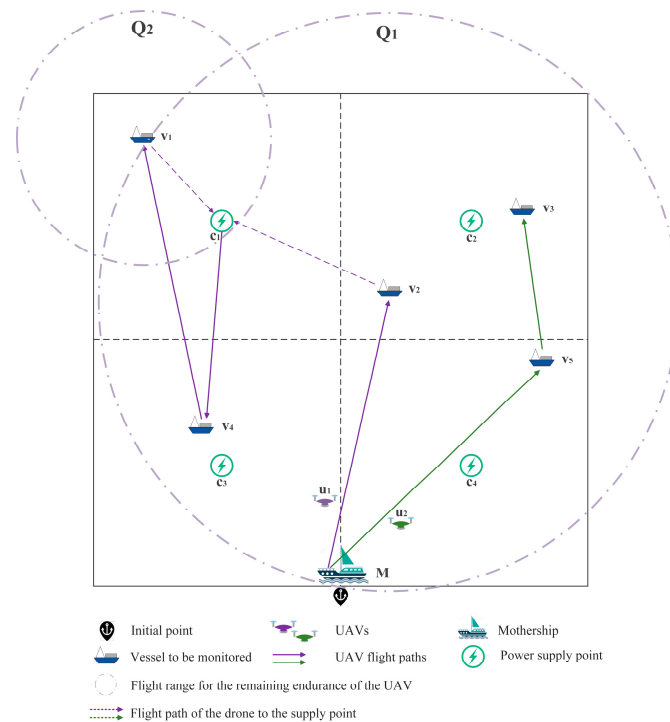


**Figure 7.** A random assignment strategy for monitoring vessels by UAVs.

#### 4.5.2. Collaboration between the Mothership and UAVs

Since, in the MUCWM path optimization problem, the mothership plays the role of mobile resupplying for UAVs, this means that the mothership needs to meet the UAVs at the supply point and replenish power for the UAVs in time. Therefore, the mothership and the UAVs need to face the problem of human–machine collaboration, which will be realized based on the selection of UAV power supply points.

First, divide the monitored sea area into equal-area grids, and use the midpoints of all grids as the set of candidate power supply points for UAVs:  $C = [1, 2, \dots, N_C]$ , where  $N_C$  is the divided grid quantity. Based on the value of the individual  $u_{i,G}$  after the random allocation strategy is executed, and based on the maximum endurance time of the UAV  $T_{max}$ , it is judged whether it is necessary to insert a power supply point for the UAV between the two vessels to be monitored during the monitoring process. Take the UAV  $u_1$  in Figure 8 as an example to describe the supply point selection strategy, taking into account the UAV power constraints during the UAV monitoring of vessels.



**Figure 8.** Synergistic strategy for UAVs and the mothership.

In the initial case, set  $flag = 1$  as the flag that shows that the UAV is fully charged and starts to perform monitoring tasks, and loop through each UAV  $u_i, u_i \in U, i \in \{1, 2, 3, \dots, N_U\}$  in sequence; then select the supply points it needs to pass during its voyage. Let  $T_{cost} = 0$ ;  $T_{cost}$  represents the time consumed by the current UAV to perform the task, loop through the vessel nodes  $v_j, v_j \in V, j \in \{1, 2, 3, \dots, N_V\}$  that the UAV needs to monitor, and judge whether it requires the insertion of feasible supply points between two consecutive monitored vessels. The specific strategies are as follows:

(1) Determine whether the vessel node is the first node monitored by the UAV when it departs from the port, that is, whether it meets  $flag = 1$  and  $j == 1$  at the same time. If the conditions are met, it is judged that the UAV flies to the first node. The vessel node to be monitored will return to the port  $Sta$  after monitoring it, that is,  $T_{max}$  will be greater than or equal to  $2d_{(Sta, v_j)} / S_U + T_V$ , where  $S_U$  represents the UAV flight speed; if the conditions are met, the UAV will fly to the location of  $v_j$  and monitor it, as shown in Figure 8, the process of UAV  $u_1$  flying to  $v_2$ . In the algorithm, it is necessary to record the remaining endurance time of the UAV  $T_{endu}$ , the UAV monitoring sequence  $r_{u_i}$  and  $T_{cost}$  and set the flag to 0, indicating that the next vessel node is not the first vessel in the UAV monitoring sequence. It is worth mentioning that since the UAV may have multiple candidate power supply points that meet the conditions in subsequent monitoring tasks, the latter node will be calculated from the previous supply point. Therefore, the solution space is an unmanned power supply point. The first monitored vessel node in the UAV monitoring path is a tree structure array with the root as the root. Therefore, the current vessel node  $v_j$  is assigned to  $r_{u_1}$  as the root node of the UAV monitoring path  $r_{u_1}$ .

(2) Determine whether the next vessel node is the last vessel node in the UAV monitoring sequence. If not, we need to determine whether the UAV can return to the UAV after flying to the next node and monitoring the vessel. Then we determine the last power supply point of the machine, that is, determine whether  $T_{endu}$  is greater than or equal to  $[d_{(v_j, v_{j+1})} + d_{(v_{j+1}, c_{pre})}] / S_U + T_V$ ; if it can, the UAV will fly to the next supply point. In Figure 8, the UAV  $u_1$  flies from the location of  $v_4$  to the location of  $v_1$  for monitoring. The reason for using the above strategy to determine whether the UAV needs to replenish electrical energy is that it must ensure that the UAV has a candidate supply point where

it can land. Even if the UAV reaches the next vessel node and completes the monitoring task, the UAV's remaining power is not satisfied. The recharge point with electric energy constraint can also be returned to the previous recharge point for charging. This strategy ensures that the UAV can land and charge at any point in time and will not run out of power and have nowhere to land during the monitoring mission. If it is not satisfied, it means that the remaining range of the UAV is not enough to support its subsequent monitoring tasks. At this time, all electric energy supply points  $c_k, c_k \in C, k \in \{1, 2, 3, \dots, N_C\}$  are traversed, and the unmanned machine selects feasible power supply points. If  $T_{endu}$  is greater than or equal to  $d_{(v_{j+1}, c_k)}$ , then  $c_k$  is added to the solution of the tree structure, and branches may appear, as shown in Figure 8. The area in  $Q_1$ , the supply point in the area, represents the position of the supply point that can be reached from the location of the vessel  $v_2$  to be monitored under the constraints of UAV endurance. Then we update  $T_{endu}$  and  $T_{cost}$ . At this time,  $T_{cost} = T_{cost} + T_u$ , where  $T_u$  represents the charging time of the UAV, and we let  $flag = 1$ . Finally, the time consumption  $T_{endu}$  from each time the UAV passes through the power supply point is added to the time window set at  $t_{window}$  in order to solve the path planning of the mothership. In Figure 8, the UAV  $u_1$  is traveling from the location of the vessel  $v_2$  to the vessel  $v_4$ . Between the locations, the  $c_1$  supply point needs to be inserted. At this time, the next node is the first vessel node monitored by the UAV after it is fully charged and takes off from the mothership; that is, it satisfies  $flag = 1$  and  $j > 1$ . Then it is judged whether the UAV can return to the location of the current supply point after flying to the first vessel node to be monitored and monitoring it, that is,  $T_{max} \geq 2d_{(c_{now}, v_j)} / S_U + T_V$ ; if satisfied, then we update  $T_{endu}$  and  $T_{cost}$ , and set the flag to 0. As shown in Figure 8, the UAV  $u_1$  flies from  $c_1$  to the vessel node  $v_4$  to be monitored.

(3) If the next node is the last node in the UAV monitoring sequence, select an electric energy supply point for the UAV that satisfies the remaining power constraint (recovery point: the UAV will be recovered by the mothership at this node, and the mothership returns to the port together and ends the monitoring task), then  $c_k$  is added to the solution of the tree structure, branches may appear, and then  $T_{endu}$ ,  $T_{cost}$ , and  $t_{window}$  are updated. At this time,  $T_{cost} = T_{cost} + d_{(v_j, c_k)} / S_U$ .

The algorithm for selecting UAV power supply points is shown in Algorithm 3 (the first node time monitored when the UAV takes off from the port).

**Algorithm 3:** The algorithm for selecting UAV power supply points

---

```

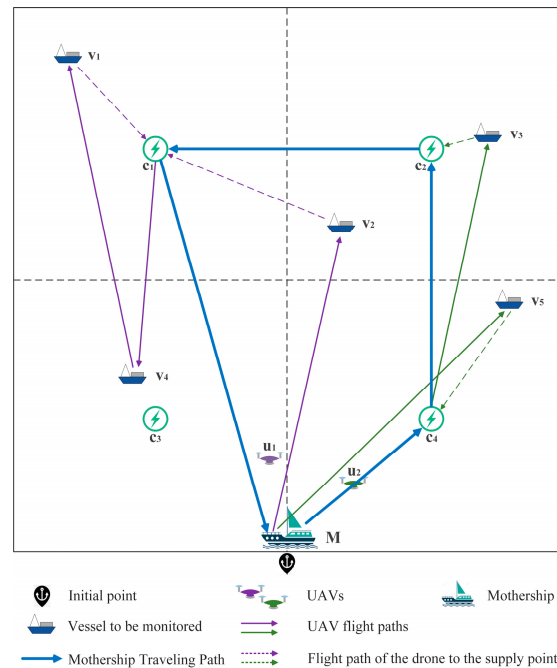
1 Begin
2 flag = 1
3 for i in  $N_U$ 
4  $T_{cost} = 0$ 
5   for j in  $N_V$ 
6      $T_{max} = 2d_{(v_j, v_1)} / S_U + T_V$  The first node monitored when the UAV takes off from the port
7     if flag = 1 and j = 1
8       if  $T_{max} \geq 2d_{(v_j, v_1)} / S_U + T_V$ 
9          $T_{endu} = E_U - 2d_{(v_j, v_1)} / s_u - t_D$ 
10         $T_{cost} = T_{cost} + T_{endu}$ 
11        flag = 0
12         $r_{u_1} \leftarrow v_j$  Add to solution structure
13      end if
14    end if
15  end for
16 end for
17 Output  $r_u, T_{endu}, t_{window}$ 
18 Finish

```

---

#### 4.5.3. Mothership Traveling Path Optimization

The results in Section 4.5.2 are used as parameters and input into the optimization algorithm of the mothership monitoring path, which is illustrated in Figure 9. The optimization strategy of the mothership monitoring path is as follows:



**Figure 9.** Monitoring path optimization strategy for the mothership.

(1) First, select one of the feasible monitoring paths of each UAV for calculation. Sort  $r_u$  in time order according to  $t_{window}$  to get  $seq_M$ , which is the sequence in which the mothership passes through each supply point. As shown in Figure 9, the navigation path of the mothership  $M$  is  $\{Sta \rightarrow c_4 \rightarrow c_2 \rightarrow c_1 \rightarrow Sta\}$ . Then assign the supply point corresponding to the sorted  $t_{window}$  to the set  $W$ . Traverse the elements in  $W$ , and use the speed constraint of the mothership to determine whether the mothership can reach the corresponding supply point to charge the UAV before the UAV's power is exhausted, that is, whether  $d_{(c_k, c_{k+1})}/S_M$  is less than  $t_{window}(c_k) + T_{endu}$ ; if satisfied, add  $c_k$  to the mothership's travel path  $r_M$ , and update  $T_{cost}$ , as shown in Figure 9, each time the mothership moves from the current supply point to the next supply point, such as  $c_4 \rightarrow c_2$ .

(2) Since the mothership and the UAV can wait for each other at the supply point, it is necessary to determine the order in which the mothership and the UAV arrive at the supply point and then update the time. Specifically, it needs to be judged whether  $t_{c_k}^M$  is greater than  $t_{window}(c_k)$ , where  $t_{c_k}^M$  represents the time when the mothership arrives at the supply point. If it is satisfied, it means that the mothership arrived after the UAV arrived at the supply point  $c_k$ . At this time,  $T_{cost}$  needs to be updated,  $T_{cost} = T_{cost} + t_{c_k}^M - t_{window}(c_k)$ , and the UAV waits for the mothership's time to be added to the time consumption.

(3) After executing a set of data, assign  $T_{cost} + d_{(l_m, Sta)}/S_M$  to  $f$ .  $f$  represents the completion time of the monitoring task.  $l_m$  represents the mothership passing the last supply point. This formula represents the mothership passing through the last supply point. The time when the vessel-borne UAV returns to the port is added to  $T_{cost}$  and then assigned to  $f$ . After all the data have been traversed, the smallest  $f$  is selected as the optimal solution  $T_{total}$  of the algorithm. Finally, output  $r_M$  and  $T_{total}$ .

The pseudocode of the mothership traveling path optimization algorithm is shown in Algorithm 4.

**Algorithm 4:** The Mothership Traveling Path Optimization Algorithm

---

```

1 Begin
2 Input  $r_u, T_{endu}, t_{window}$ 
3 for  $i$  in  $r_{u1}$ 
4   for  $j$  in  $r_{u2}$ 
5   ...
6   for  $n$  in  $r_{uN_u}$ 
7      $seq_M = \text{Sort } r_u$  in time order according to  $t_{window}$ 
8      $W =$  corresponding supply point sequence after sorting
9      $T_{cost} = 0$ 
10    for  $k$  in  $W$ 
11      if  $T_{cost} + d_{(c_k, c_{k+1})} / S_M < t_{window(c_k)} + T_{endu}$ 
12         $r_M \leftarrow c_k$ 
13         $T_{cost} = T_{cost} + T_{cost} / S_M + T_V$ 
14      end if
15    end for
16     $T_{total} \leftarrow T_{cost} + d_{(l_m, Sta)} / S_M$ 
17  end for
18  ...
19 end for
20 end for
21  $\min T_{total} = \min(f)$ 
22 Output  $r_M, \min T_{total}$ 
23 Finish

```

---

**5. Experiment and Analysis**

In this section, in order to prove the effectiveness of the improved adaptive differential evolution algorithm, this article uses Python 3.12.0 to implement the solution of the algorithm. This experiment was run on a computer with an Intel (R) Core (TM) i5-7200U CPU at 2.50 GHz to 2.70 GHz and a memory of 8 GB.

*5.1. Data Preparation and Parameter Settings*

This paper conducted a numerical experiment on the vessels entering and leaving the port in a certain port area of Ningbo Zhoushan Port. At 17:40 on 3 April 2023, 100 vessels in the above port were obtained from the automatic identification system (AIS). The position data of vessels sailing in the sea area are partly shown in Table 4. The table gives the vessel serial number, vessel name, vessel type, vessel location (including longitude and latitude), and the update time of the current position. In order to more intuitively display the position of the vessel in the port sea area, a real-time satellite image of the selected sea area is given, as shown in Figure 10. The area within the dotted line is the monitored port sea area.

Before parameter setting, we consider some more realistic factors, such as whether the sea state and weather environment have any influence on the path optimization problem of air pollution monitoring of port vessels by shipboard UAVs.

First of all, the performance and stability of the current DJI UAV are very high, and as an example, the wind resistance of the DJI Warp M300 RTK model UAV can reach 12 m/s, i.e., level 6. According to the survey, in some open sea areas, such as the central part of the ocean or some particularly windy sea areas (such as the roaring westerly wind belt in the South Atlantic Ocean), the average wind speed may be between 7 m/s (about 25 km/h) and 10 m/s (about 36 km/h). Wind speeds are somewhat lower in harbor-based waters, so the monitoring mission is largely unaffected by winds at sea. Secondly, wave heights average a few centimeters to 20 m and can only reach 30 m in very extreme natural conditions [49]. However, when monitoring vessels, UAVs need to fly into the vessel's plume to do so, and the average vessel's air height (i.e., the height of a vessel in the ocean between sea level and the very top of the vessel) can vary greatly depending on the type and use of the vessel, and the following are the approximate ranges of air heights for some common types of vessels: container vessels: larger container vessels, usually in the range of 40 to 60 m; bulk carriers: medium-sized bulk carriers may be between 20 and 30 m; tankers: roughly between 30 and 50 m; cruise vessels: usually over 50 m; and medium-sized passenger vessels: these are generally between 20 and 30 m. The vessel's plume is at a position even higher than the

highest point of the vessel, so under normal circumstances, the plume height is higher than the waves, and therefore, the UAV monitoring process will not be affected by the waves as long as it maintains the flight height at the plume position.

Table 4. Data from selected test vessels.

Serial Number	Type	Name	Longitude	Latitude	Update Time
1	cruise vessel	MINFUDINGYU02378	122.210367	29.87655	17:38:30
2	freighter	ZHESHENGYU07203	122.367693	29.876857	17:39:23
3	other types	SHUNAN118	122.167765	29.877635	17:39:24
4	passenger vessel	ZHELINGYU96006	122.168633	29.878233	17:38:30
5	cruise vessel	MINFUDINGYU04373	122.248718	29.879403	17:22:29
6	other types	ZHELINGYUYUN20079	122.313133	29.882153	17:39:19
7	cruise vessel	ZHEXIANGYUYUN01008	122.313125	29.882275	17:34:58
8	passenger vessel	ZHEPUYU81003	122.312587	29.882623	17:14:21
9	other types	MINFUDINGYU09892	122.30088	29.882937	17:24:20
10	cruise vessel	ZHESANYUYUN80399	122.300633	29.883198	17:39:20
11	freighter	XIN HAI 125	122.300817	29.8832	17:33:44
12	other types	02387-21 51%	122.373223	29.883888	17:11:32
13	other types	FUDINGYU07855-6 90%	122.305915	29.88447	16:51:14
14	cruise vessel	ZHEPUYUYUN98788	122.287335	29.886522	17:37:51
15	other types	07853-25 80%	122.210898	29.887745	16:57:28
16	port supplies	RUN ZHOU 7	122.186388	29.888345	17:33:42
17	passenger vessel	MA ZHI DU 2	122.372592	29.888377	17:39:06
18	freighter	SUQIYU02669	122.372337	29.888437	17:34:28
19	port supplies	PU MAO GONG 4	122.37232	29.888593	17:24:03
20	other types	SUQIYU02669-33%	122.260595	29.891088	17:30:34

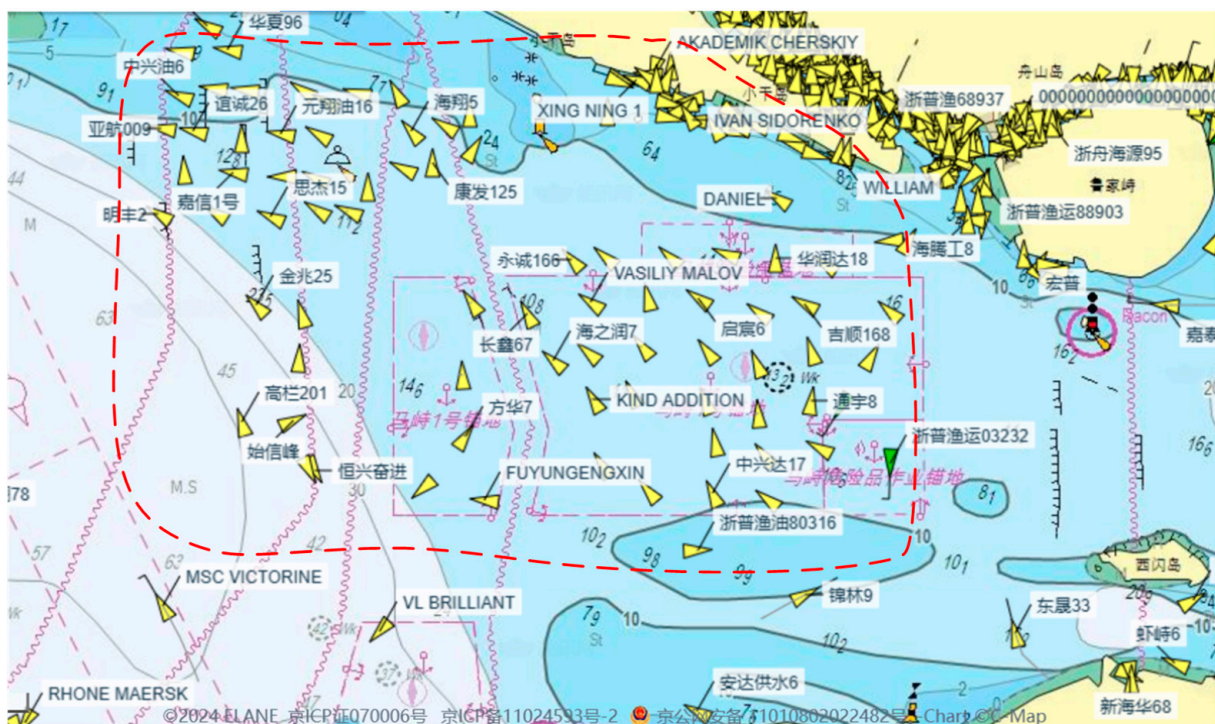


Figure 10. Schematic diagram of the location of the test vessels in Ningbo Zhoushan Port.

The default values of the parameters involved in the experiment are shown in Table 5.

**Table 5.** Parameter settings involved in the test.

Parameter	Value
Number of UAVs	3
Number of equal-area divisions within the sea area (number of candidate supply points)	9
Harbor initial point coordinates	(153.8975, 172.3765)
UAV maximum flight speed	18 m/s
Maximum endurance of UAV	45 min
Average monitoring time per vessel	3 min
UAV charging time	2 min
Mothership maximum sailing speed	621 m/min
Number of vessels to be monitored	100 vessels

The description of the main parameters in Table 5 is as follows:

**Number of UAVs:** in the scenario of the vessel exhaust gas measurement trial carried out by the Hanjiang Maritime Safety Department of the Yangzhou Maritime Safety Administration, the number of UAVs was established at three.

**Number of equal area divisions in the sea area (number of candidate supply points):** In the sea area, it is not advisable to select too many candidate supply points, because too many supply points will greatly increase the running time of the algorithm; at the same time, it is not advisable to select too few candidate supply points, because there may be no solution if there are too few supply points. Therefore, this article divides the sea area into nine equal areas; that is, there are nine candidate supply points.

**Port starting point coordinates:** set according to the coordinates of a central pier in the port.

**Maximum UAV flight speed and maximum UAV endurance time:** Based on the UAV produced by DJI that can be used in water conservancy: M300 RTK flight parameters: 55 min maximum flight time, 23 m/s maximum flight speed, and 12 m/s wind resistance. Taking into account the impact of the environment and special circumstances, the maximum flight speed of the UAV is set to 18 m/s, and the maximum endurance time of the UAV is set to 45 min.

**Average monitoring time of each vessel:** based on previous monitoring experience, the time for UAV monitoring of each vessel is set to 3 min [50].

**UAV charging time:** In order to save the time required for the total monitoring task, the battery is directly replaced on the mothership to replenish the UAV's power. Therefore, the battery replacement time is set to 2 min.

**The maximum sailing speed of the mothership:** Calculated based on the previous average speed of China Maritime Safety Administration port patrol vessels of 20.13 knots. After converting the unit into m/min, it is 621 m/min.

Table 6 shows the parameter settings in the improved adaptive differential evolution algorithm.

**Table 6.** Parameterization of improved adaptive differential evolutionary algorithms.

Parameter	Definition	Value
NP	Number of populations	100
G_max	The maximum number of iterations	300
F	Variation scaling factor	Adaptive (initially 1)
CR	Crossover probability	Adaptive (initially 1)
bounds	Element upper and lower bounds	[0, 1]

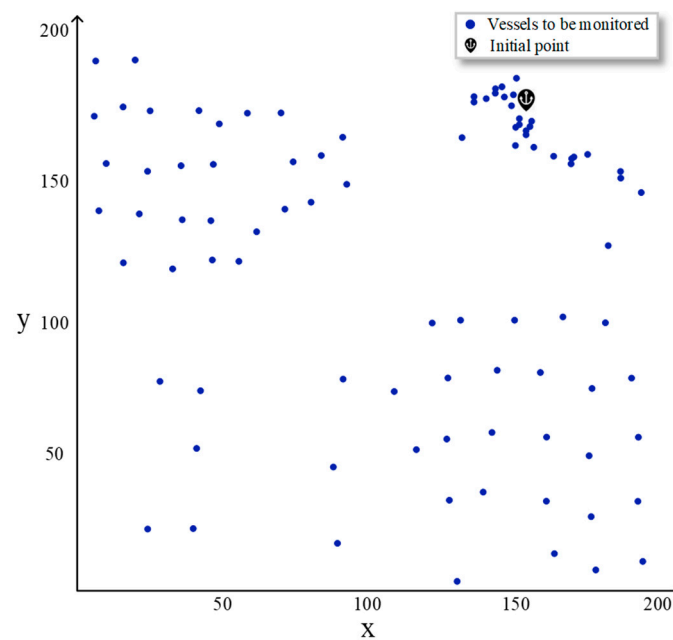
## 5.2. Solution to MUCWM Path Optimization Algorithm Based on IADE Algorithm

According to the actual geographical location of the port, this article scales the monitored sea area to the *xoy* plane rectangular coordinate system of  $200 \times 200$  to run the algorithm of this article. Following scaling, the initial positions of the 100 vessels to be

monitored, one mothership, and three UAVs were determined, represented by (x, y) coordinates. These data are partially displayed in Table 7. Initially, the three UAVs and the mothership share the same location. Figure 11 visually illustrates the initial locations of the vessels to be monitored, the mothership, and the UAVs.

**Table 7.** Partially scaled vessels' position data.

Vessel	X	Y
$v_1$	46.99850459302554	0.0728363324858847
$v_2$	133.20431783052203	3.016281062558292
$v_3$	181.7962476594918	7.069408740377959
$v_4$	167.35237568646986	13.041988003445184
$v_5$	91.48372017039544	16.482433590408455
$\vdots$	$\vdots$	$\vdots$
$v_{90}$	146.9947346595266	178.70608397601762
$v_{91}$	148.79173630574027	179.34875749784328
$v_{92}$	154.04953692653433	182.60497000858064
$v_{93}$	6.501878683514789	188.69751499570967
$v_{94}$	20.46319916557688	188.9203084832821
Harbor initial point	153.8975	172.3765



**Figure 11.** Schematic of the vessels' positions after zooming.

The data are substituted into the improved adaptive differential evolution algorithm to solve the problem. Figure 12 shows the MUCWM path optimization results obtained in the above scenario. Different colors represent different UAV flight paths, and the dotted line is when the UAV goes to the supply point for charging. Since the figure involves many line segments, the path of the mothership will be displayed in the table. Table 8 shows the specific results of path optimization for the mothership and three UAVs.



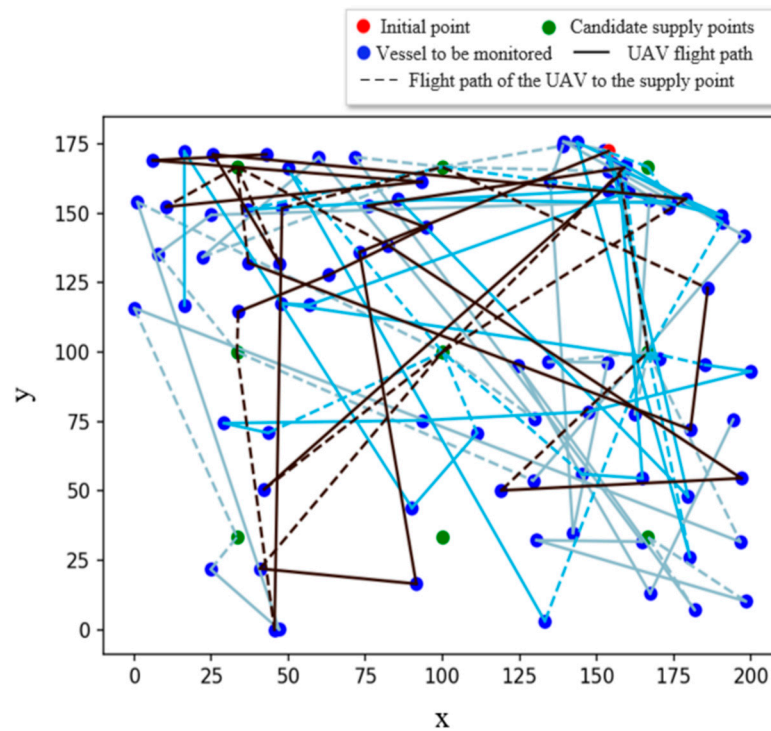


Figure 12. Schematic diagram of MUCWM path optimization results.

Table 8. Specific results of MUCWM path optimization.

UAV monitoring sequence	$u_1$	Sta → v <sub>80</sub> → v <sub>68</sub> → v <sub>60</sub> → v <sub>75</sub> → c <sub>4</sub> → v <sub>22</sub> → v <sub>66</sub> → v <sub>36</sub> → v <sub>84</sub> → v <sub>17</sub> → c <sub>6</sub> → v <sub>46</sub> → v <sub>74</sub> → v <sub>39</sub> → v <sub>28</sub> → v <sub>61</sub> → c <sub>5</sub> → v <sub>24</sub> → v <sub>42</sub> → v <sub>12</sub> → v <sub>81</sub> → c <sub>9</sub> → v <sub>63</sub> → v <sub>85</sub> → v <sub>96</sub> → v <sub>41</sub> → v <sub>14</sub> → c <sub>3</sub> → v <sub>21</sub> → v <sub>55</sub> → v <sub>73</sub> → c <sub>5</sub> → v <sub>99</sub> → v <sub>34</sub> → v <sub>100</sub> → v <sub>65</sub> → v <sub>3</sub> → c <sub>3</sub> → v <sub>13</sub> → v <sub>58</sub> → v <sub>87</sub> → c <sub>4</sub>
	$u_2$	Sta → v <sub>93</sub> → v <sub>83</sub> → v <sub>57</sub> → v <sub>26</sub> → v <sub>23</sub> → v <sub>27</sub> → c <sub>5</sub> → v <sub>88</sub> → v <sub>77</sub> → v <sub>50</sub> → v <sub>76</sub> → v <sub>72</sub> → c <sub>5</sub> → v <sub>47</sub> → v <sub>32</sub> → v <sub>67</sub> → v <sub>40</sub> → c <sub>6</sub> → v <sub>35</sub> → v <sub>7</sub> → v <sub>78</sub> → v <sub>59</sub> → v <sub>48</sub> → c <sub>5</sub> → v <sub>97</sub> → v <sub>62</sub> → v <sub>91</sub> → v <sub>82</sub> → v <sub>95</sub> → v <sub>1</sub> → c <sub>3</sub> → v <sub>10</sub> → v <sub>30</sub> → v <sub>70</sub> → v <sub>86</sub> → c <sub>8</sub> → v <sub>44</sub> → v <sub>56</sub> → v <sub>54</sub> → c <sub>8</sub> → v <sub>89</sub> → v <sub>5</sub> → c <sub>7</sub>
	$u_3$	Sta → v <sub>49</sub> → v <sub>51</sub> → v <sub>11</sub> → v <sub>20</sub> → c <sub>5</sub> → v <sub>45</sub> → v <sub>2</sub> → v <sub>53</sub> → c <sub>2</sub> → v <sub>4</sub> → v <sub>6</sub> → v <sub>71</sub> → v <sub>90</sub> → c <sub>7</sub> → v <sub>64</sub> → v <sub>69</sub> → v <sub>92</sub> → v <sub>29</sub> → c <sub>3</sub> → v <sub>18</sub> → v <sub>43</sub> → v <sub>25</sub> → v <sub>52</sub> → c <sub>4</sub> → v <sub>9</sub> → v <sub>94</sub> → v <sub>15</sub> → v <sub>38</sub> → c <sub>4</sub> → v <sub>33</sub> → v <sub>19</sub> → v <sub>31</sub> → v <sub>37</sub> → c <sub>8</sub> → v <sub>79</sub> → v <sub>98</sub> → v <sub>16</sub> → v <sub>8</sub> → c <sub>7</sub>
Mothership sailing sequence	M	Sta → c <sub>4</sub> → c <sub>5</sub> → c <sub>6</sub> → c <sub>2</sub> → c <sub>7</sub> → c <sub>6</sub> → c <sub>9</sub> → c <sub>3</sub> → c <sub>5</sub> → c <sub>3</sub> → c <sub>4</sub> → c <sub>3</sub> → c <sub>5</sub> → c <sub>8</sub> → c <sub>7</sub> → c <sub>4</sub> → Sta
Minimum time consumption of monitoring tasks		112.25619 min

### 5.3. Analysis of Results

#### 5.3.1. Algorithm Performance Comparison

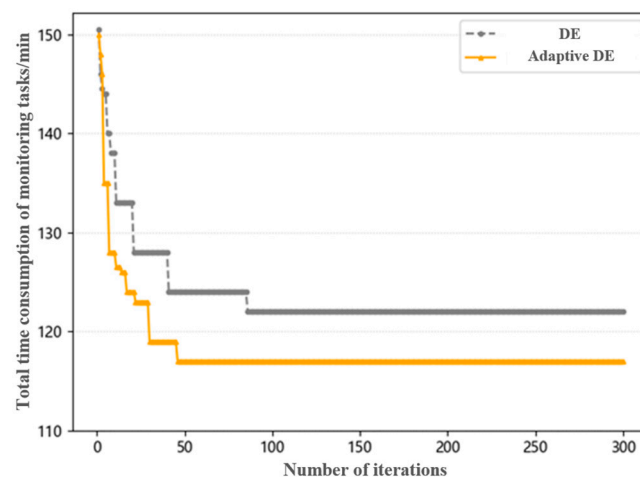
In order to verify the accuracy of the algorithm in small-scale calculation examples, the results of IADE solving this problem in small-scale cases are compared with the Gurobi running results. We set the problem scales,  $N_U = \{1, 2, 3\}$ ,  $N_V = \{5, 10, 20, 50\}$ , to conduct 12 independent experiments for each scale, and we recorded the GAP between the algorithm and Gurobi in each experiment, where  $GAP = (f(IADE) - f(Gurobi)) / f(Gurobi) \times 100\%$ . It is stipulated that if Gurobi runs for more than 30 min and still does not obtain a result, it is considered that Gurobi cannot obtain the solution to the calculation example. The IADE algorithm records the results and running time of each generation and selects the running time of the generation with the smallest difference from the Gurobi result. The comparison results are shown in Table 9.

**Table 9.** Experimental comparison of the results of IADE and Gurobi.

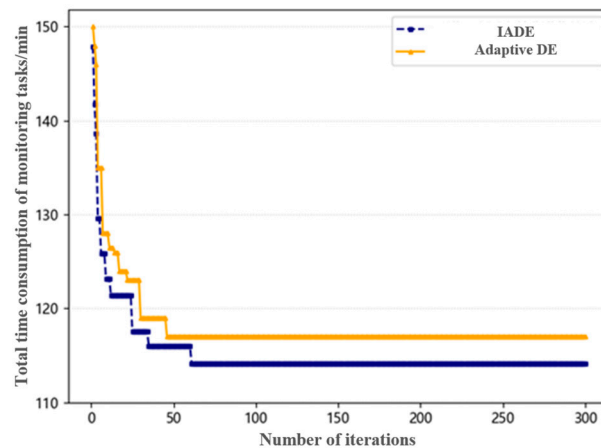
Problem Size		Optimal Solution (min)			CPU Running Time (s)	
		This Article	Gurobi	GAP	This Article	Gurobi
$N_U = 1$	$N_V = 5$	83.26657	83.26657	0	1.52	4.98
	$N_V = 10$	93.83875	93.83875	0	8.93	21.92
	$N_V = 15$	100.64764	100.64764	0	36.54	118.34
	$N_V = 20$	115.90341	115.90341	0	114.06	523.03
$N_U = 2$	$N_V = 5$	69.33498	69.33498	0	3.58	27.05
	$N_V = 10$	80.93034	80.93034	0	12.95	99.07
	$N_V = 15$	99.06371	97.43652	1.67%	40.65	256.83
	$N_V = 20$	105.99024	103.95277	1.96%	136.98	976.64
$N_U = 3$	$N_V = 5$	66.98347	66.98347	0	6.98	87.54
	$N_V = 10$	75.54691	74.96964	0.77%	18.97	386.83
	$N_V = 15$	84.96447	84.86348	1.19%	49.54	1605.75
	$N_V = 20$	93.71631	-	-	145.86	-

It can be seen from the table that, in the independent experiment with  $N_U = 3$ ,  $N_U = 20$ , the GAP value between IADE and the accurate result is less than 5%, and the solution speed has been greatly improved. Although Gurobi can obtain accurate results, it takes a long time to solve, so the comparison with the Gurobi results proves the effectiveness of this algorithm in solving the collaborative routing optimization problem of UAVs and motherships, and it has practical application significance in real-life scenarios.

When solving the model, in order to verify the superiority of the improved adaptive differential evolution algorithm compared to the traditional differential evolution algorithm, this paper compares the parameters of the adaptive improved differential evolution algorithm with the basic differential evolution algorithm. The group size and the number of iterations are set to  $NP = 100$  and  $G_{max} = 300$ , respectively. The resulting operation iteration diagram is shown in Figure 13.

**Figure 13.** Adaptive strategy iteration graph.

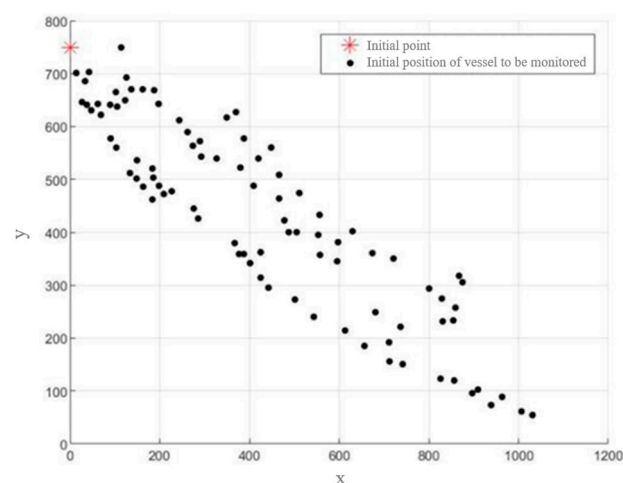
From Figure 14, it can be seen that the DE algorithm, after the adaptive improvement strategy, shows a trend of being better than the traditional DE algorithm in terms of convergence speed and optimal solution performance.



**Figure 14.** Iterative diagram of operator improvement strategy.

On the basis of the adaptive parameter differential evolution algorithm, after adding the hybrid mutation improvement strategy and the hybrid crossover improvement strategy, the test example is also run when  $NP = 100$  and  $G_{max} = 300$ , and the resulting convergence diagram is shown in Figure 14. It can be seen from the figure that although the convergence speed has dropped slightly, the improved algorithm has strong search capabilities and performs well in manifesting the optimal solution. Therefore, the overall solution performance of the improved adaptive differential evolution algorithm is higher than that of the adaptive differential evolution algorithm.

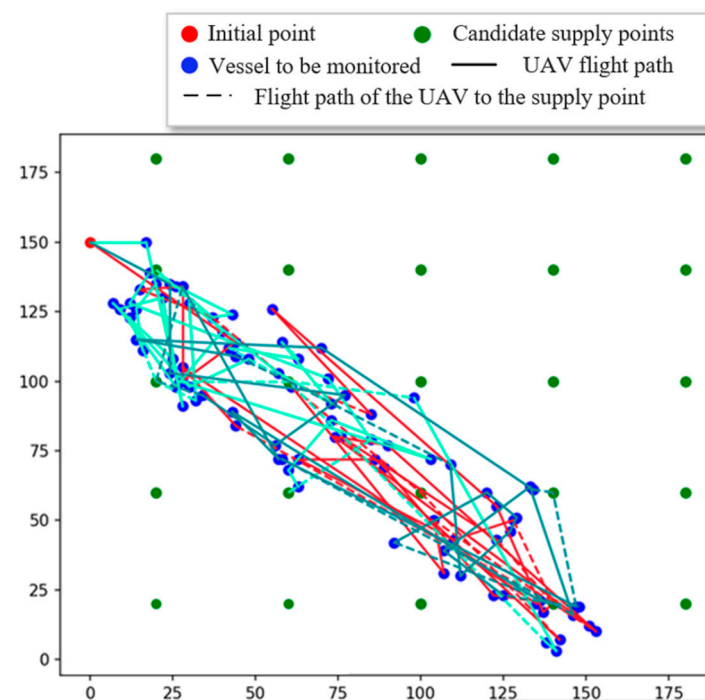
We took the test case data from Shen et al. [12], as shown in Figure 15, and adjusted the parameters to the values in the article, as shown in Table 10. Since the vessel distribution is relatively concentrated, the number of candidate supply points can be increased to improve the efficiency of the algorithm. Since Shen et al. [12] did not consider the UAV charging time, the UAV charging time was set to 0. Bringing the adjusted data into the algorithm of this article, the result is 116.76982 min, which is smaller than the optimal solution of 126.69466 min in the article. The optimization performance of the improved adaptive differential evolution algorithm is further demonstrated. Figure 16 shows the MUCWM path optimization results obtained by using the algorithm of this article to execute the test case data in Shen et al. [12].



**Figure 15.** Test data from Shen et al. [12].

**Table 10.** Test data parameter settings.

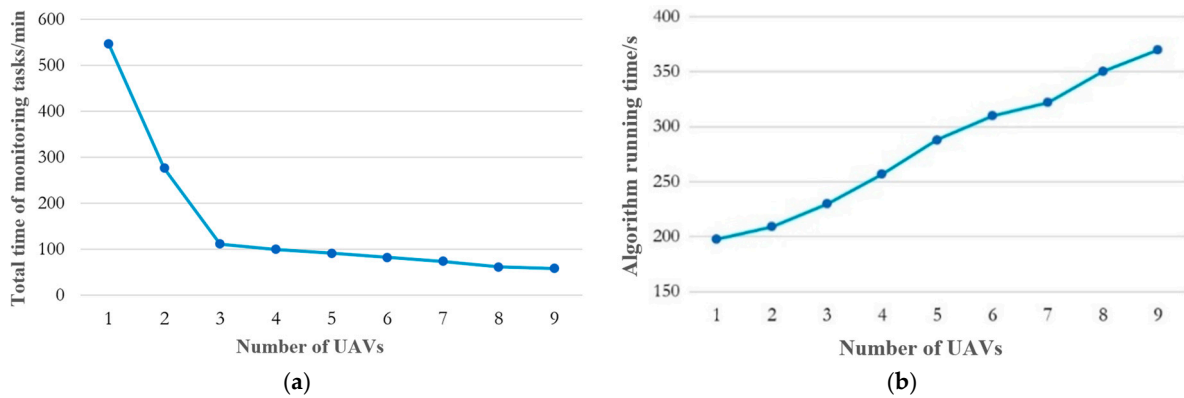
Parameter	Value
Number of UAVs	3
Number of equal-area divisions within the sea area (number of candidate supply points)	25
Harbor initial point coordinates	(0, 125)
UAV maximum flight speed	1000 m/min
Maximum endurance of UAV	35 min
Average monitoring time per vessel	2 min
UAV charging time	26 m/min
Mothership maximum sailing speed	90 vessels

**Figure 16.** MUCWM path optimization results for running test data.

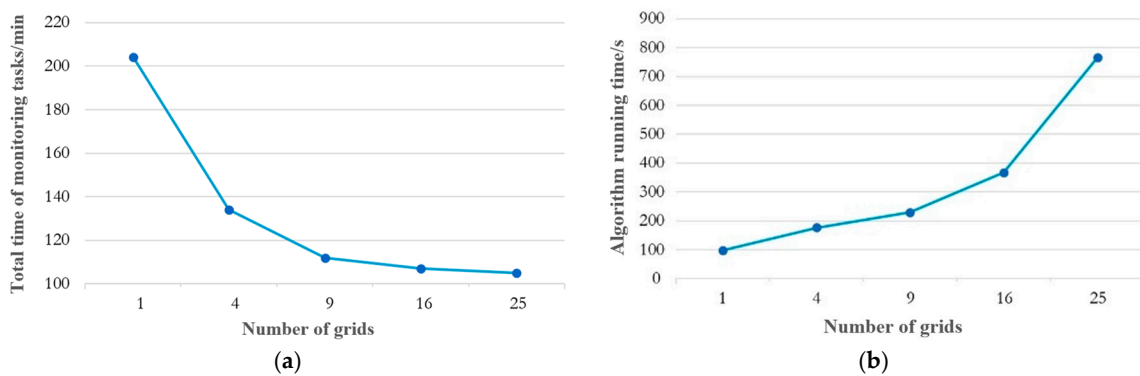
### 5.3.2. Sensitivity Analysis

Figure 17a illustrates that with an increasing number of UAVs, the overall time for MUCWM monitoring decreases, although the rate of reduction in total monitoring time gradually diminishes. Thus, beyond a certain number of UAVs, further increases will not significantly decrease the total duration of the MUCWM monitoring task. Figure 17b reveals that as the number of UAVs grows, the total execution time of the algorithm consistently increases, suggesting a roughly positive correlation between the number of UAVs and the algorithm's running time.

Figure 18a demonstrates that as the number of grid divisions rises, there is a decrease in the total MUCWM monitoring time, particularly noticeable when the grid count increases from 1 to 9. However, with further increases in grid divisions, the rate of reduction in the total monitoring task time starts to taper off. Hence, beyond a certain number of grid divisions, the advantages of further increases in reducing the overall time of the MUCWM monitoring task become marginal. Figure 18b shows that as the number of grid divisions grows, the total execution time of the algorithm also increases, with the rate of increase accelerating, suggesting that the number of grid divisions significantly affects the algorithm's running time.

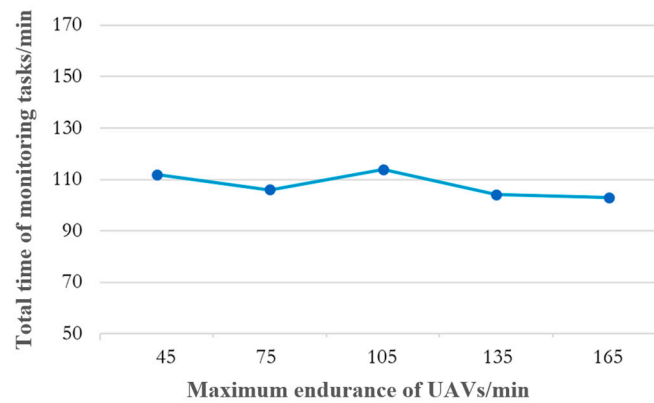


**Figure 17.** The relationship between the number of UAVs and the optimal solution and the running time of the algorithm: (a) the relationship between the total time of monitoring tasks and the number of UAVs; (b) the relationship between the algorithm running time and the number of UAVs.

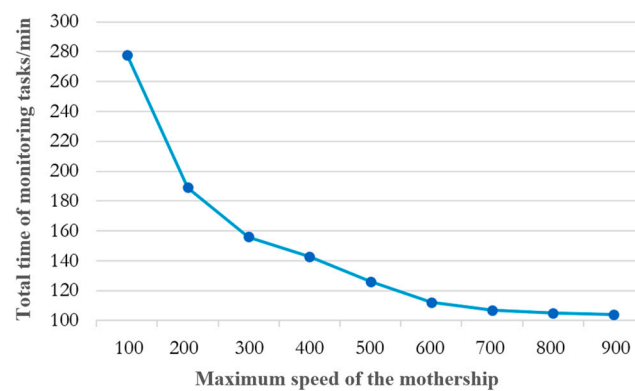


**Figure 18.** The relationship between the number of grids and the optimal solution and the running time of the algorithm: (a) the relationship between the total time of monitoring tasks and the number of grids; (b) the relationship between the algorithm running time and the number of grids.

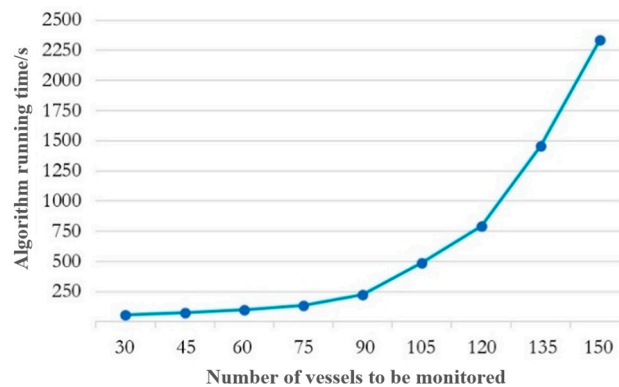
Figure 19 indicates that extending the maximum endurance time of the UAV has a minimal effect on the total duration of the MUCWM monitoring task. This minimal impact is largely due to the introduction of the mothership serving as a mobile power supply for the UAVs, which significantly mitigates the power limitation issues of the UAV. Figure 20 additionally reveals that, while increasing the sailing speed of the mothership reduces the overall monitoring time of MUCWM, this reduction tends to plateau. Thus, after the mothership's speed reaches a certain level, further increases in speed yield diminishing returns in reducing the total monitoring time. Figure 21 illustrates that, with varying numbers of UAVs, when the number of vessels to be monitored rises from 30 to 150, the algorithm's running time increases by 2487.962 s. Notably, when the count of vessels monitored in the port surpasses 120, there is a significant spike in the algorithm's running time. Consequently, the algorithm exhibits improved computational efficiency when the number of vessels monitored remains below 120.



**Figure 19.** The relationship between the maximum endurance of UAVs and the total time of monitoring tasks.

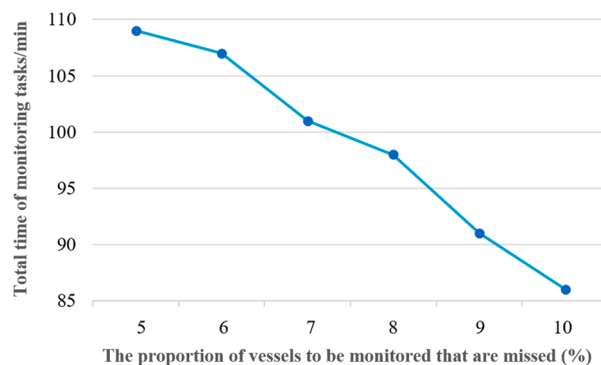


**Figure 20.** The relationship between the maximum speed of the mothership and the total time of monitoring tasks.



**Figure 21.** The relationship between the number of vessels to be monitored and the algorithm running time.

In particular, we discussed the impact on the total monitoring time in the case of leakage of 5–10% of the monitored vessels, and 10 sets of experiments were conducted for each value of leakage, and the results obtained after averaging are shown in Figure 22. From the figure, it can be observed that the total monitoring time tends to decrease as the number of missed vessels increases, and we find that the impact on the total monitoring task is more significant when it is reduced to 10%. This provides a compromise for the port; the port can tradeoff between the time of the total monitoring task and the coverage of the monitored vessels and can choose to give up a small portion of the vessels to obtain a shorter monitoring time, and this analysis provides a certain reference value for the monitoring program of the vessels in the port's sea area.



**Figure 22.** Relationship between the proportion of vessels to be monitored that are missed and the total time of the monitoring tasks.

## 6. Conclusions

This paper presents an innovative monitoring solution for vessel air pollution, utilizing a mothership and multiple UAVs to conduct efficient and comprehensive monitoring in maritime pre-monitoring areas. This significantly enhances monitoring efficiency and extends the monitoring scope. The core focus of this research is the path optimization problem for multi-UAV collaborative mothership (MUCWM) monitoring, which encompasses three critical issues: (1) optimizing the monitoring paths of multiple UAVs to ensure efficient and accurate coverage of the target area; (2) developing a collaboration mechanism between the mothership and multiple UAVs to synchronize their monitoring efforts; and (3) optimizing the mothership's traveling path to maximize overall monitoring efficiency.

To address these challenges, the paper develops a path optimization model for multi-UAV collaborative mothership monitoring of vessel air pollution in port waters. The model is solved using an improved adaptive differential evolution (IADE) algorithm. Utilizing vessel position data from the Automatic Identification System (AIS) of Ningbo Zhoushan Port, we demonstrate the model and algorithm's effectiveness through an example. Performance comparisons and sensitivity analyses of the algorithm highlight its significant advantages in solving the path planning problems of collaborative operation between the mothership and multiple UAVs.

However, the research also acknowledges certain limitations: the mission optimization model lacks practical constraints such as UAV flight altitude, sea wind speed, and air control. Furthermore, the model designates static supply points as the midpoints of a divided grid network. Future research will explore dynamically adjusting the supply points based on navigation conditions to achieve a better optimal solution.

**Author Contributions:** Conceptualization, J.S. and L.S.; methodology, J.S.; software, D.Y.; validation, L.S., J.S. and D.Y.; formal analysis, J.S.; data curation, J.S.; writing—original draft preparation, L.S.; writing—review and editing, L.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Liaoning Province Social Science Planning Fund, grant number 2022-ZSK078, the National Natural Science Foundation of China, grant number 72171032, the National Natural Science Foundation of China, grant number 71702019, and the Humanities and Social Science Foundation of the Ministry of Education, grant number 21YJAZH070.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All data are available in the paper.

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Sun, Z.H.; Luo, X.; Wu, E.Q.; Zuo, T.Y.; Tang, Z.R.; Zhuang, Z. Monitoring Scheduling of Drones for Emission Control Areas: An Ant Colony-Based Approach. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 11699–11709. [CrossRef]
2. Sirimanne, S.N.; Hoffman, J.; Juan, W.; Asariotis, R. *Review of Maritime Transport 2019*; United Nations: New York, NY, USA, 2019; Volume 9.
3. United Nations. *Review of Maritime Transport 2023: United Nations Conference on Trade and Development*. 2023. Available online: <https://unctad.org/publication/review-maritime-transport-2023> (accessed on 27 September 2023).
4. Liang, G.Y. Neglected pollution sources. *Environment* **2016**, *6*, 17–19.
5. Transport & Environment, Air Pollution from Ships. 2018. Available online: <https://www.transportenvironment.org/articles/air-pollution-ships> (accessed on 20 January 2018).
6. Li, L.; Gao, S.; Yang, W. The enforcement of ECA regulations: Inspection strategy for on-board fuel sampling. *J. Comb. Optim.* **2022**, *44*, 2551–2576. [CrossRef]
7. Sun, P.Z.; Luo, X.; Zuo, T.; Bao, Y.; Sun, Y.N.; Law, R.; Wu, E.Q. Emission Monitoring Dispatching of Drones Under Vessel Speed Fluctuation. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 21833–21847. [CrossRef]
8. Xia, J.; Wang, K.; Wang, S. Drone scheduling to monitor vessels in emission control areas. *Transp. Res. Part B Methodol.* **2019**, *119*, 174–196. [CrossRef]
9. IMO Low Sulfur Fuel Emission Control Partial State Requirements. 2018. Available online: [https://www.sohu.com/a/284957167\\_120015135](https://www.sohu.com/a/284957167_120015135) (accessed on 27 December 2018).
10. China MSA. The Latest Global Emission Control Areas and Low-Sulfur Oil Areas Data Collation and Sink. 2022. Available online: [http://www.360doc.com/content/22/0823/07/77972649\\_1044930083.shtml](http://www.360doc.com/content/22/0823/07/77972649_1044930083.shtml) (accessed on 23 August 2022).
11. OECD. Reducing Sulphur Emissions from Ships. 2018. Available online: [https://www.oecd-ilibrary.org/transport/reducing-sulphur-emissions-from-ships\\_5jlwvz8mq9s-en](https://www.oecd-ilibrary.org/transport/reducing-sulphur-emissions-from-ships_5jlwvz8mq9s-en) (accessed on 20 January 2018).
12. Shen, L.; Hou, Y.; Yang, Q.; Lv, M.; Dong, J.X.; Yang, Z.; Li, D. Synergistic path planning for ship-deployed multiple UAVs to monitor vessel pollution in ports. *Transp. Res. Part D Transp. Environ.* **2022**, *110*, 103415. [CrossRef]
13. Stein, M. Conducting safety inspections of container gantry cranes using Unmanned Aerial Vehicles. In *Dynamics in Logistic, Proceedings of the 6th International Conference LDIC 2018, Bremen, Germany, 20–22 February 2018*; Springer International Publishing: Cham, Switzerland, 2018; pp. 154–161.
14. Urbahs, A.; Zavtkevics, V. Oil pollution monitoring of sea aquatorium features with using unmanned aerial vehicles. In Proceedings of the 17th International Conference Transport Means, Kaunas, Lithuania, 23–24 October 2014; pp. 75–78.
15. Zhang, S.; Xin, M.; Wang, X.; Zhang, M. Anchor-free network with guided attention for ship detection in aerial imagery. *J. Appl. Remote Sens.* **2021**, *15*, 024511. [CrossRef]
16. Primeau, N.; Abielmona, R.; Falcon, R.; Petriu, E. Maritime smuggling detection and mitigation using risk-aware hybrid robotic sensor networks. In Proceedings of the 2017 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA), Savannah, GA, USA, 27–31 March 2017; pp. 1–7.
17. Kim, J.S.; Lee, M.J.; Nam, H.; Do, S.; Lee, J.H.; Park, M.K.; Park, B.H. Indoor and Outdoor Tests for a Chemi-capacitance Carbon Nanotube Sensor Installed on a Quadrotor Unmanned Aerial Vehicle for Dimethyl Methylphosphonate Detection and Mapping. *ACS Omega* **2021**, *6*, 16159–16164. [CrossRef] [PubMed]
18. Soarability. How does UAV Estimate the Sulfur Content of Ship Fuel by Monitoring Ship Exhaust [EB/OL]. Available online: [https://www.sohu.com/a/399172599\\_100226347](https://www.sohu.com/a/399172599_100226347) (accessed on 2 June 2020).
19. Flyable technology Soarability. Maritime Administration Explores Implementation of Drone Ship Exhaust Monitoring. 2022. Available online: <https://baijiahao.baidu.com/s?id=1738314327938212810&wfr=spider&for=pc> (accessed on 14 July 2022).
20. Jiangsu Maritime Bureau of the People's Republic of China, News Events. 2022. Available online: [https://www.js.msa.gov.cn/art/2022/7/11/art\\_47\\_1356157.html](https://www.js.msa.gov.cn/art/2022/7/11/art_47_1356157.html) (accessed on 11 July 2022).
21. China seShipping. Port of Rotterdam Is Trialling New Drones to Monitor Ship Emissions. 2022. Available online: [https://www.chineseshipping.com.cn/cninfo/News/202210/t20221021\\_1370484.shtml](https://www.chineseshipping.com.cn/cninfo/News/202210/t20221021_1370484.shtml) (accessed on 21 October 2022).
22. Explanation of the Endurance of Multi-Rotor Drones and Fixed-Wing Drones. Military/Avionics. 2023. Available online: <https://m.elecfans.com/article/2238023.html> (accessed on 7 September 2023).
23. Murray, C.C.; Chu, A.G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 86–109. [CrossRef]
24. Agatz, N.; Bouman, P.; Schmidt, M. Optimization approaches for the traveling salesman problem with drone. *Transp. Sci.* **2018**, *52*, 965–981. [CrossRef]
25. Bouman, P.; Agatz, N.; Schmidt, M. Dynamic programming approaches for the traveling salesman problem with drone. *Networks* **2018**, *72*, 528–542. [CrossRef]
26. Ha, Q.M.; Deville, Y.; Pham, Q.D.; Hà, M.H. On the min-cost traveling salesman problem with drone. *Transp. Res. Part C Emerg. Technol.* **2018**, *86*, 597–621. [CrossRef]
27. Yurek, E.E.; Ozmutlu, H.C. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transp. Res. Part C Emerg. Technol.* **2018**, *91*, 249–262. [CrossRef]
28. Tu, P.A.; Dat, N.T.; Dung, P.Q. Traveling salesman problem with multiple drones. In Proceedings of the 9th International Symposium on Information and Communication Technology, Da Nang, Vietnam, 6–7 December 2018; pp. 46–53.



29. Wang, X.; Poikonen, S.; Golden, B. The vehicle routing problem with drones: Several worst-case results. *Optim. Lett.* **2017**, *11*, 679–697. [[CrossRef](#)]
30. Poikonen, S.; Wang, X.; Golden, B. The vehicle routing problem with drones: Extended models and connections. *Networks* **2017**, *70*, 34–43. [[CrossRef](#)]
31. Kitjacharoenchai, P.; Lee, S. Vehicle routing problem with drones for last mile delivery. *Procedia Manuf.* **2019**, *39*, 314–324. [[CrossRef](#)]
32. Dayarian, I.; Savelsbergh, M.; Clarke, J.P. Same-day delivery with drone resupply. *Transp. Sci.* **2020**, *54*, 229–249. [[CrossRef](#)]
33. Karak, A.; Abdelghany, K. The hybrid vehicle-drone routing problem for pick-up and delivery services. *Transp. Res. Part C Emerg. Technol.* **2019**, *102*, 427–449. [[CrossRef](#)]
34. Poikonen, S.; Golden, B. Multi-visit drone routing problem. *Comput. Oper. Res.* **2020**, *113*, 104802. [[CrossRef](#)]
35. Guerriero, F.; Surace, R.; Loscri, V.; Natalizio, E. A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows constraints. *Appl. Math. Model.* **2014**, *38*, 839–852. [[CrossRef](#)]
36. Phan, D.H.; Suzuki, J. Evolutionary multiobjective optimization for the pickup and delivery problem with time windows and demands. *Mob. Netw. Appl.* **2016**, *21*, 175–190. [[CrossRef](#)]
37. Schermer, D.; Moeini, M.; Wendt, O. A matheuristic for the vehicle routing problem with drones and its variants. *Transp. Res. Part C Emerg. Technol.* **2019**, *106*, 166–204. [[CrossRef](#)]
38. Wang, Z.; Sheu, J.B. Vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **2019**, *122*, 350–364. [[CrossRef](#)]
39. Sacramento, D.; Pisinger, D.; Ropke, S. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transp. Res. Part C Emerg. Technol.* **2019**, *102*, 289–315. [[CrossRef](#)]
40. Zajac, S.; Huber, S. Objectives and methods in multi-objective routing problems: A survey and classification scheme. *Eur. J. Oper. Res.* **2021**, *290*, 1–25. [[CrossRef](#)]
41. Qin, W.; Shi, Z.; Li, W.; Li, K.; Zhang, T.; Wang, R. Multiobjective routing optimization of mobile charging vehicles for UAV power supply guarantees. *Comput. Ind. Eng.* **2021**, *162*, 107714. [[CrossRef](#)]
42. Yu, K.; Budhiraja, A.K.; Tokekar, P. Algorithms for routing of unmanned aerial vehicles with mobile recharging stations and for package delivery. *arXiv* **2017**, arXiv:1704.00079.
43. Zhu, K.; Yang, J.; Zhang, Y.; Nie, J.; Lim, W.Y.B.; Zhang, H.; Xiong, Z. Aerial refueling: Scheduling wireless energy charging for UAV enabled data collection. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 1494–1510. [[CrossRef](#)]
44. Zhou, Y.; Cheng, N.; Lu, N.; Shen, X.S. Multi-UAV-aided networks: Aerial-ground cooperative vehicular networking architecture. *IEEE Veh. Technol. Mag.* **2015**, *10*, 36–44. [[CrossRef](#)]
45. Luo, Z.; Liu, Z.; Shi, J. A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle. *Sensors* **2017**, *17*, 1144. [[CrossRef](#)]
46. Shen, L.; Wang, Y.; Liu, K.; Yang, Z.; Shi, X.; Yang, X.; Jing, K. Synergistic path planning of multi-UAVs for air pollution detection of ships in ports. *Transp. Res. Part E Logist. Transp. Rev.* **2020**, *144*, 102128. [[CrossRef](#)]
47. Yuan, H.; Xiao, C.; Wang, Y.; Peng, X.; Wen, Y.; Li, Q. Maritime vessel emission monitoring by an UAV gas sensor system. *Ocean. Eng.* **2020**, *218*, 108206. [[CrossRef](#)]
48. Luo, X.; Sun, Z.H.; Qiu, S. Ant colony system based drone scheduling for ship emission monitoring. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; pp. 241–247.
49. Drones for Ship Exhaust Monitoring, DroneNet. Available online: <https://www.163.com/dy/article/IEOJKCVR05149OCK.html> (accessed on 16 September 2023).
50. Sea Wave. Key Laboratory of Coastal Disaster and Protection (Hohai University), Ministry of Education. 2019. Available online: <https://coast.hhu.edu.cn/2020/0512/c2585a203485/page.htm> (accessed on 11 November 2019).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.