Synergistic path planning for ship-deployed multiple UAVs to monitor vessel pollution in ports

Lixin Shen [a], Yunxia Hou [a,*], Qin Yang [a], Meilin Lv [a], Jing-Xin Dong[b], Zaili Yang [a,c] and Dongjun Li[d]

[a] *Maritime Economics and Management College, Dalian Maritime University, Dalian 116026, China*

[b] *Business School, Newcastle University, 5 Barrack Road, Newcastle upon Tyne, NE1 4SE, UK*

[c] *Liverpool Logistics, Offshore and Marine Research Institute, Liverpool John Moores University, Liverpool, UK*

[d] *Newcastle Business School, Northumbria University, Sutherland Building, Ellison Pl, Newcastle Upon Tyne, NE1 8ST, UK*

E-mail: shenlixin@dlmu.edu.cn; houyx@dlmu.edu.cn; yangqin918@dlmu.edu.cn; lvmeilin_jax@qq.com; jingxin.dong@ncl.ac.uk; Z.Yang@ljmu.ac.uk; dongjun.li@northumbria.ac.uk;

Correspondence information: YunXia Hou; *Maritime Economics and Management College, Dalian Maritime University, Dalian 116026, China;* houyx@dlmu.edu.cn; 13591127218.

Abstract

Traditionally, vessel air emissions are monitored onboard vessels or at fixed points at sea. These methods ineffectively meet the needs of monitoring pollution from vessels travelling. Unmanned aerial vehicles (UAVs) equipped with pollution monitoring sensors are becoming a research focus. However, due to battery capacity constraints,

the monitoring scope of UAVs is still not optimal. Thus, using a ship (such as a patrol ship) as a UAV mobile supply base can overcome battery limitations and increase monitoring coverage. This paper investigates the joint routing and scheduling problem of ship-deployed multiple UAVs (SDMU) for the monitoring of pollution from vessels. The artificial bee colony (ABC) algorithm based on simulated annealing is employed to minimize the total monitoring time. The model and solution algorithm are verified by real-time dynamic vessel data from Tianjin Port.

Keywords:

UAVs, Vessel air pollution, Two-level path planning, Bee colony algorithm

1. Introduction

Maritime transport plays an important role in international supply chains and carries approximately 90% of global trade in terms of volume (United Nations, 2019). The 2019 United Nations Shipping Report (UNCTAD, 2019) shows that the total global shipping capacity reached 11 billion tons in 2018, an increase of 300 million tons compared to 2017 levels. With the rapid growth of the shipping industry, the number of vessels in the port is increasing, and consequently, environmental pollution from vessels is becoming increasingly critical (Lee et al., 2020). Currently, air pollution from vessels has become the third-largest source of air pollution after motor vehicle exhaust emissions and industrial production emissions (Liang, 2016). Since 2015, the International Maritime Organization (IMO) has successively established four vessel air pollution emission control areas (ECAs), and more ECAs are being planned (Xia et al., 2019). International communities are making efforts to reduce pollution from vessels.

At the 70th meeting of the IMO in October 2016, the Marine Environmental Protection Committee (MEPC) confirmed that January 2020 was the implementation time of a global standard according to which the sulfur content of vessels must not exceed 0.50% m/m (CHINA MSA, 2019). In 2019, the Marine Bureau of the Ministry of Transport in China published the Implementation Plan of the Global Marine Fuel Limit in 2020 and the Guidelines for the Supervision and Management of Marine Atmospheric Pollutant Emissions and proposed stricter requirements to reduce vessel exhaust emissions. The plans suggest that all types of sea-going vessels should use marine fuel oil with a sulfur content no more than 0.1% m/m in inland river ECAs and no more than 0.5% m/m in coastal ECAs (CHINA MSA, 2019). To ensure that vessels obey these standards of international regulations, it is necessary for all law enforcement agencies to effectively implement the relevant laws and regulations, and monitoring air pollution from vessels is essential. Traditional monitoring methods require that environmental law enforcement officers board vessels or install sensors at fixed points. These methods are not cost-effective and have inadequate capacity. For instance, due to the outbreak of COVID-19, monitoring air pollution on board vessels has been suspended in many countries due to concerns over spreading the virus.

Using UAVs to monitor air pollution from vessels has attracted widespread attention from researchers due to the advantages of their high flexibility, few geographical restrictions, and less manual work involved (Shen et al., 2020; Zhou et al., 2020; Xia et al., 2019). This is evident from their applications in areas such as military surveillance (Galbraith et al., 2020), maritime search and rescue (Wang et al., 2018),

and waterway mapping and logistics distribution businesses (Li et al., 2020). China has carried out pilot projects in which UAVs were used to monitor air pollution from vessels. For example, in November 2019, the Dalian Maritime Safety Administration in China tested the use of UAVs carried by a patrol ship to monitor the air pollution from vessels in high traffic waters, such as the Laotieshan Waterway and Dasanshan Waterway. On April 23, 2020, the Bureau of Yangzhou Maritime Affairs of China used Dajiang M210 UAVs to monitor the emissions from dozens of vessels and carried out real-time visual analysis on air pollution. On May 20, 2020, a DJI M210 UAV equipped with an air pollution detection system called Lingxi V2 measured the vessel exhaust gas in the area surrounding the Hanjiang Maritime Department of the Yangzhou Maritime Safety Administration. The findings of these pilot projects indicate that the use of UAVs to monitor air pollution from vessels can significantly expand the monitoring scope.

The application of UAVs in monitoring air pollution from vessels is, however, in its infancy. Among the main challenges is UAV operations suffering from limited battery capacity and consequently very limited flying distance and time (Shen et al., 2020; Xia et al., 2019). In practice, the maximum coverage by UAVs is not in proportion with ECAs requiring air emission detection. Meanwhile, the number of vessels to be monitored in ECAs is often high (Xia et al., 2019), revealing a complicated traffic scenario in which ships come in and out the areas with high uncertainty and randomness. Furthermore, vessels with emissions exceeding the regulated limits make every effort to avoid the monitoring of environmental law enforcement agencies and the disclosure of their ships' true emission levels (Shen et al., 2020; Xia et al., 2019). In light of these

difficulties, UAVs have advantages in terms of their high frequency, broad monitoring coverage and routine/nonroutine detection. To further improve the UAV's detection ability, the use of ship to carry and deploy multiple UAVs in ECAs has been proposed (Xia et al., 2019). Using ship-deployed multiple UAVs (SDMU) to monitor air pollution at sea is a new idea, and many theoretical issues still need to be resolved. In this paper, we specifically focus on the synergistic path planning problem for UAVs and the ships carrying them simultaneously. This problem involves path planning coordination between the ship carrier and UAVs, coordination between UAVs, and the prediction of the locations of vessels to be monitored. In this paper, a two-level path planning model is developed to address the problem of minimizing the total monitoring task completion time. Our model can also ensure that the routes of a ship carrier and its carried UAVs are synchronized, the workloads of UAVs are balanced, and each UAV is reasonably scheduled.

2. Literature review

Zhou et al. (2020) evaluated the use of 27 UAVs to monitor air pollution from vessels in the Yangtze River Estuary. The main contribution of their research is to compare and analyse levels of SO_x, NO_x and other gases contained in the atmospheric pollutants emitted by moving vessels. Their research suggests that the use of UAVs to monitor vessel air pollution is feasible in practice. However, Zhou et al. (2020) did not consider the path planning issue relating to their UAVs and their carrying ships.

There are only limited studies examining the path planning problem involved in UAVs monitoring air pollution from vessels in maritime transport. Xia et al. (2019)

investigated path planning for UAVs to monitor air pollution from vessels sailing in ECAs and developed a mixed integer linear programming model based on time-extended networks. The authors also proposed a solution method based on Lagrangian relaxation to solve the model. In their study, UAVs needed to fly to and from fixed stations on the shore to be charged. Shen et al. (2020) examined the problem of using multiple UAVs to monitor air pollution for vessels sailing in ports. The authors developed a path planning model for multiple dynamic targets and adopted the particle swarm algorithm to solve the problem. In both Shen et al. (2020) and Xia et al. (2019), UAVs needed to fly to and from fixed stations to gain power supply. To the best of our knowledge, no study has investigated a path planning problem wherein UAVs are charged at their carrying ships.

Inspired by the traditional vehicle routing problem (VRP), researchers worldwide have carried out research on UAV routing. Huang et al. (2020) proposed a dynamic UAV path planning framework for the purpose of the rapid collection and reliable transmission of information when dealing with emergency tasks. Sun et al. (2018) theoretically analysed the target detection problem based on an underwater wireless sensor network (UWSN) and derived the optimal path planning algorithm for UAVs to detect moving targets. Studies have been carried out to address the problem of path planning for UAVs as well as the vehicles that carry UAVs or provide power supply. Liu et al. (2019) investigated the route planning problem of UAVs and their carrying vehicles to complete intelligence, surveillance and reconnaissance missions for armies. Yu et al. (2019) proposed algorithms for path planning issues in two scenarios where

UAVs may be recharged at fixed locations or mobile stations. Hu et al. (2019) considered a single carrying vehicle and multiple UAVs and proposed two-level path planning algorithms for vehicles and UAVs. Wang et al. (2019) studied the joint distribution of drones and trucks. A truck carrying drones from the distribution centre where the drone stops to serve the customer and lands on another truck with flying range and loading capacity limits will complete the mission. Meanwhile, a mixed-integer programming model and branching pricing algorithm are proposed. Extensive experiments have been performed in practical environments to verify the effectiveness and accuracy of the proposed algorithm. Moshref-Javadi et al. (2020) proposed a route planning model for the combined distribution of trucks and multiple UAVs. Their model can determine the optimal route of trucks and the optimal launch and collection locations of UAVs along truck routes. Booth et al. (2020) proposed the joint use of UAVs and ground mobile charging vehicles (modular robotic vehicles (MRVs)) to search for multiple moving targets. The problem is solved using MILP and constraint programming (CP). The authors applied real road network data from Scotland to test the proposed model and explore the effect of UAV speed on path optimization results.

Table 1 A summary of UAV routing studies

Literature	**UAV routing**	**Carrier routing**	**Carrier types**	**Moving target**	**UAV monitors air pollution from vessels**
Xia et al. (2019)	Y	N	N	Y	Y
Zhou et al. (2020)	N	N	Ship	Y	Y
Shen et al. (2020)	Y	N	N	Y	Y

Huang et al. (2020)	Y	N	N	Y	N
Sun et al. (2018)	Y	N	N	Y	N
Liu et al. (2019)	Y	Y	Vehicle	N	N
Hu et al. (2019)	Y	Y	Vehicle	N	N
Wang et al. (2019)	Y	Y	Vehicle	N	N
Moshref-Javadi et al. (2020)	Y	Y	Vehicle	N	N
Booth et al. (2020)	Y	Y	Vehicle	N	N
This study	Y	Y	Ship	Y	Y

Our research aims to fill the knowledge gap in the literature in relation to *joint path planning for both UAVs and their carrying ships.* Table 1 shows that there are currently only three studies (e.g., Xia et al 2019, Zhou et al 2020 and Shen et al 2020) that have addressed the use of UAVs to monitor air pollution from vessels. One of these works (Zhou et al 2020) has contributed to the use of UAVs to monitor the content of SO_x, NO_x and other gases contained in the air pollutants emitted by moving vessels. The value and content value of manual monitoring are compared and analysed. The results show that it is feasible to use UAVs to monitor the air pollution of sailing vessels. The other two papers (Xia et al 2019, Shen et al 2020) investigate the path planning of multiple UAVs to monitor air pollution from vessels, but the UAVs considered in the two studies are limited by battery capacity and have to return to fixed onshore charging stations. Studies on the path planning problem for vehicle carriers and multiple UAVs in road transportation cannot be directly applied to maritime transport. When studying

the problem of path planning coordination between a carrier and multiple UAVs, it is necessary to determine the path of the carrier based on the transportation mode involved. For road transportation, the determination of a carrier's path is subject to the distribution of infrastructure construction and road regulations. In the case of maritime transportation, the determination of a carrier's travel path is subject to the distribution of port channels and port management regulations.

3. SDMU model establishment

In this section, we present a model for planning paths for UAVs and their carrying ship.

3.1. Problem description

We assume that a ship carries multiple UAVs to monitor air pollution from the vessels sailing at sea. UAVs need to be released based on the number and locations of vessels to be monitored. The deployment of UAVs in a port water is simulated in Fig 1.

Scenario 1: A single UAV is shown in states I and III in Fig 1. According to the number of monitored vessels, the ship carriers from D_0 to A_{k_1} and from A_{k_2} to A_{k_3} only need to release one UAV to perform the pollution monitoring task;

Scenario 2: Multiple UAVs are shown in states II and V in Fig 1. According to the number of monitored vessels, the ship carriers from A_{k_1} to A_{k_2} and A_{k_4} to A_{k_5} need to launch multiple UAVs to perform monitoring tasks;

Scenario 3: The UAV is idle, as shown in state IV in Fig 1. At this time, there is no vessel to be monitored in the surrounding area. During this particular journey, no UAV will be released to perform any monitoring task.

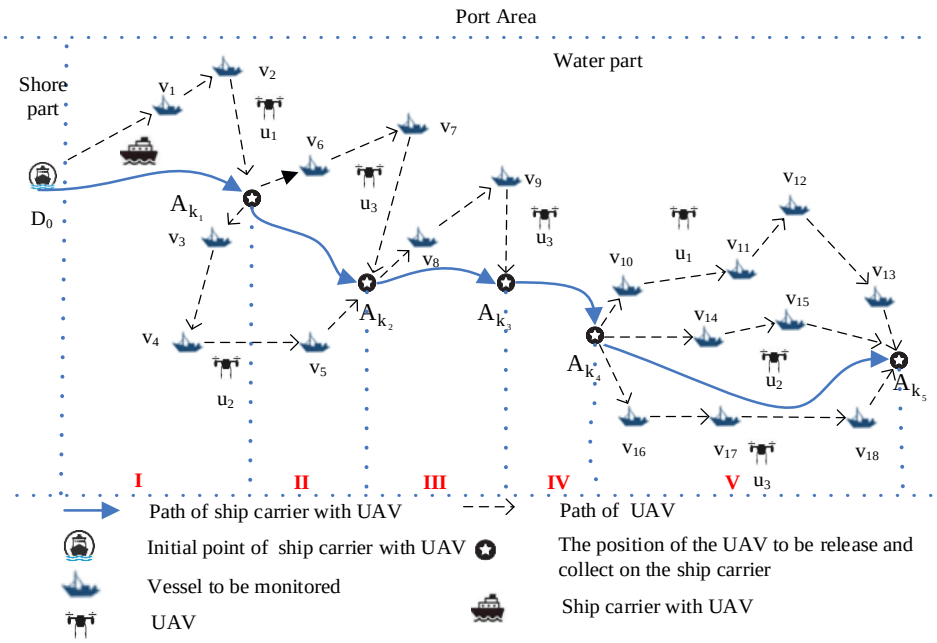

Fig 1 Path planning of ship-deployed UAVs for air pollution monitoring

The primary decision-making issues emerging from the above scenarios are path planning for the ship carrier and the multiple UAVs. More specifically, the following three questions need to be resolved:

(1) How can path planning be coordinated for the ship carrier as well as UAVs? This involves determining the path for the ship carrier, the released and collected locations of UAVs, the number of UAVs to be released each time, and the flying paths of UAVs.

(2) How can path planning be coordinated between UAVs? This problem mainly involves the coordination and path planning of multiple UAVs to minimize the total monitoring time.

(3) How can the positions of the monitored target vessels be predicted in real time? This problem involves predicting the locations of the vessels to be monitored and rescheduling UAV monitoring tasks when the location of the vessel to be monitored

changes.

To model the above UAV-based air pollution monitoring system, the following assumptions are made:

(1) The travel route of a ship carrier is modelled as a connected graph.

(2) The time for UAVs to monitor the pollution from a vessel is fixed.

(3) The travel time for a UAV to travel from one vessel to another is calculated according to the distance between the two vessels and the UAV's speed. The impact of wind on UAVs is not considered to make the problem tractable.

(4) The time needed to replace UAV batteries is negligible.

(5) The ship carrier and the UAVs travel at constant speeds.

(6) One UAV can monitor multiple vessels, and each vessel to be monitored can only be monitored by one drone.

3.2. Notation and terminology

In this subsection, we define the index and set to be used in our formulation.

N: the total number of UAVs available for the monitoring tasks.

N_r: the number of UAVs that the ship carrier needs to release each time, $N_r \in \{1,2,3,\cdots,N\}$.

$V = \{v_1, v_2, \cdots, v_n\}$: a set of vessels to be monitored. v_i is the i^{th} vessel to be monitored, $i \in \{1,2,3,\cdots,n\}.v_i \in V$.

$U = \{u_1, u_2, \cdots, u_N\}$: a set of UAVs. u_i is the i^{th} UAV and $i \in \{1,2,3,\cdots,N\}$, $u_i \in U$.

$A_{kr} = \{A_{kr_1}, A_{kr_2}, \ldots, A_{kr_m}\}$: a set of locations where the ship carrier releases UAVs on the traffic route. A_{kr_i} is the i^{th} location, $i \in \{1,2,3,\cdots,m\}$, $A_{kr_i} \in A_{kr}$.

$A_{kc} = \{A_{kc_1}, A_{kc_2}, \ldots, A_{kc_m}\}$: a set of locations where the ship carrier collects UAVs on the traffic route. A_{kc_i} is the i^{th} location, $i \in \{1,2,3,\cdots,m\}$, $A_{kc_i} \in A_{kc}$.

$G = \{G_1, G_2, G_3, \cdots, G_H\}$: cluster centres of adjacent vessels to be monitored. G_i is the i^{th} cluster centre of adjacent vessels to be monitored, $i \in \{1,2,3,\cdots,H\}$, $G_i \in G$.

$K = \{k_1, k_2, \cdots k_{N_r}\}$: the set of UAV flying paths, where k_i represents the path of a UAV, and $|k_i|$ represents the number of vessels to be monitored in the UAV path set, that is, the number of vessels to be monitored on the i^{th} UAV path. $k_i \cap k_j = \emptyset$, $i \in \{1,2,3,\cdots,N_r\}$, and $j \in \{1,2,3,\cdots,N_r\}$ are used to ensure that for each vessel to be monitored in set K, k_i is monitored only once, and set K includes all vessels to be monitored.

D_0: the departure point of the ship carrier from the port.

T_v: the time required to monitor a vessel.

S_u: the flying speed of a UAV.

T_w: the maximum flying time of a UAV.

$T_{k_i}(start)$: the moment when a UAV starts path k_i.

$T_{k_i}(end)$: the moment when a UAV ends path k_i.

p_{u_i,t_j}: the location of the i^{th} UAV at time t_j, $t \in [\, T_{k_i}(strat), T_{k_i}(end) \,]$, $t_j \in t$.

$T_{\delta_{u_i}, p_{u_i,t_j}}$: the current remaining available flying time of the UAV.

S_{ship}: the running speed of the ship carrier.

A_k: the location where the ship carrier releases or collects UAVs on a traffic route.

$T_{A_k}(collect)$: the time when the ship carrier collects the UAV at A_k.

$T_{A_k}(release)$: the time when the ship carrier releases the UAV at A_k.

$N_{A_k}(arrive)$: the number of idle UAVs when the ship carrier arrives at A_k.

$N_{A_k}(leave)$: the number of idle UAVs when the ship carrier leaves A_k.

d_{G_i}: the sum of the distances between all vessels to be monitored in the i^{th} cluster.

$R_{A_k}(start)$: the set of UAV paths where the ship carrier starts to release at A_k, where $|R_{A_k}(start)|$ refers to the number of elements contained in the set of UAV paths, that is, the number of UAVs released by the ship carrier at A_k.

$R_{A_k}(end)$: the set of UAV paths where the ship carrier ends at A_k, where $|R_{A_k}(end)|$ refers to the number of elements contained in the UAV path set, that is, the number of UAVs collected by the ship carrier at A_k.

Decision variable

$d_{A_{kr_i},A_{kc_i}}$: the length of the travel route of a ship carrier that starts from A_{kr_i} and reaches A_{kc_i}.

$d_{A_{kr_i}k_i,A_{kc_i}}$: the length of the monitoring route of a UAV released from A_{kr_i} that visits each vessel to be monitored in path k_i and that returns to A_{kc_i}.

3.3. SDMU path planning model

The objective function of the SDMU path planning model includes two parts denoted as F_1 and F_2. F_1 represents the ship carrier's travel time, which is equal to the total length of the ship carrier's travel route divided by the ship carrier's travel speed. Taking state I in Fig 1 as an example, F_2 represents the situation where a UAV (u_1) leaves for work, and the ship carrier is also moving forward. The time spent is equal to

the larger value between the time it takes for the UAV to leave the ship carrier to perform the current round of monitoring tasks to return to the ship carrier and the time it takes for the ship carrier to travel from the location where the UAV is released (D_0) to the location where the UAV is collected (A_{k_1}).

Obj:

$$F_1 = \sum_{i=1}^{m-1} (d_{D_0,A_{kr_1}} + d_{A_{kc_i},A_{kr_{i+1}}} + d_{A_{kc_m},D_0}) / S_{ship} \tag{1}$$

$$F_2 = \sum_{i=1}^{m} max\{ d_{A_{kr_i},k_i,A_{kc_i}}/S_u + |k_i| \times T_v, d_{A_{kr_i},A_{kc_i}}/S_{ship}\} \tag{2}$$

$$F_{min} = F_1 + F_2 \tag{3}$$

s.t.

$$T_{k_i}(end) = T_{k_i}(strat) + |k_i| \times T_v + d_{A_{kr_i},k_i,A_{kc_i}}/S_u$$
$$i \in \{1,2,3,\cdots,N_r\},\ k_i \in K \tag{4}$$

$$T_{\delta_{u_i},p_{u_i,t_j}} - d_{p_{u_i,t_j},c_i}/S_u \geq 0$$
$$t \in [\ T_{k_i}(strat), T_{k_i}(end)\ ],\ \ t_j \in t \tag{5}$$

$$T_{\delta_{u_i},p_{u_i,t_j}} = T_w$$
$$t \in [\ T_{k_i}(strat), T_{k_i}(end)\ ],\ \ t_j \in t \tag{6}$$

$$t_j = T_{k_i}(strat)$$
$$0 \leq T_{\delta_{u_i},p_{u_i,t_j}} \leq T_w \tag{7}$$

$$t \in [\, T_{k_i}(strat), T_{k_i}(end)\ \,],\ \ t_j \in t$$

$$d_{A_{kr_i},k_i,A_{kc_i}}/S_u + |k_i| \times T_v \le T_w$$

$$i \in \{1,2,3,\cdots,N_r\}, k_i \in K \tag{8}$$

$$T_{A_{kc_i}}(collect) = T_{A_{kr_i}}(release) + d_{A_{kr_i},A_{kc_i}}/S_{ship}$$

$$i \in \{1,2,3,\cdots,m\},\ \ A_{kr_i} \in A_{kr}, A_{kc_i} \in A_{kc} \tag{9}$$

$$N_{A_{k_i}}(arrive) = N_{A_{k_{i-1}}}(leave)$$

$$i \in \{2,3,\cdots,m\} \tag{10}$$

$$N_{A_{k_i}}(leave) = N_{A_{k_i}}(arrive) + |R_{A_{k_i}}(end)| - |R_{A_{k_i}}(start)|$$

$$i \in \{1,2,3,\cdots,m\} \tag{11}$$

$$T_{A_{k_i}}(collect) = T_{k_j}(end)$$

$$i \in \{1,2,3,\cdots,m\}$$

$$j \in \{1,2,3,\cdots,N_r\} \tag{12}$$

$$T_{A_{k_i}}(release) = T_{k_j}(strat)$$

$$i \in \{1,2,3,\cdots,m\}$$

$$j \in \{1,2,3,\cdots,N_r\} \tag{13}$$

Constraint (4) means that the time when the UAV returns to the ship carrier after this round of monitoring tasks is equal to the time when the UAV leaves on the ship

carrier plus the time spent by the UAV traveling and the time spent by the UAV in the monitoring node period; Constraint (5) ensures that the remaining power of the UAV can return to the ship carrier; Constraint (6) ensures that the power of the UAV upon it leaving the ship carrier is the battery capacity; Constraint (7) denotes that the current remaining available time of the UAV cannot be less than zero or greater than the maximum flying time of the UAV; Constraint (8) is used to ensure that the period of time the UAV goes out to perform a task (the UAV leaves the ship carrier + undertakes monitoring tasks + returns to the ship carrier) does not exceed the maximum endurance period; Constraint (9) denotes the time at which the ship carrier collects the UAV, which is equal to the time at which the ship carrier releases the UAV, plus the travel time of the ship carrier when UAV is performing the monitoring task; Constraint (10) indicates that “the number of idle UAVs when the ship carrier releases (collected) UAVs” is equal to the number of idle UAVs upon leaving the position of the previous ship carrier to collect (release) the UAV; Constraint (11) indicates that the number of idle UAVs when the ship carrier leaves the UAV release location is equal to "the number of idle UAVs upon arriving at this location " plus "the number of UAVs collected at this location" subtracted by "the number of UAVs flying at this location"; Constraint (12) denotes that the position where the ship carrier collects the UAV is the same as the position where the UAV ends its path; Constraint (13) means that the position of the ship carrier to release the UAV is the same as the position of the UAV's starting path.

4. Algorithm implementation

We establish an SDMU two-level path planning model in the previous section.

This section presents an artificial bee colony (ABC) algorithm. In Section 4.1 discusses the ABC algorithm enhanced by the simulated annealing idea and its implementation process in detail. In Section 4.2, we introduce an improved ABC algorithm for the SDMU path planning model.

4.1. Improved ABC algorithm based on the idea of simulated annealing

In this section, we first discuss the standard ABC algorithm and then the application of the algorithm to the path planning problem of UAVs and their carriers.

4.1.1. The basic ABC algorithm

The ABC algorithm is mainly composed of food sources and a leader, scouter, and follower. When solving a problem, each food source represents a viable solution to the SDMU problem. The quality of the solution is represented by the quality of the food source. Each leader is associated with a food source it is developing and records relevant information about the food source, such as distance and quality information. Every time the leader returns to the nest, it will share the information with other bees with a certain probability. The follower selects a leader to follow based on the information shared by the leader and follows the food source corresponding to the leader to select a new food source. The scouter can actively and independently explore new food sources in the environment around the nest. Although the behaviour of the characters varies, the role of the bees is not fixed. When the nectar from a food source is exhausted, the corresponding leader will become a scouter. When a scouter finds a new food source, it becomes a leader. This process of role conversion also corresponds

to the two most basic behaviour patterns of an ABC: giving up a certain food source and recruiting for a certain food source.

The ABC designed in this paper mainly improves the original ABC from the generation mechanism of the neighbourhood solution of the food source and the new solution generated by the scouter. In solving the previously discussed problems, a large-scale damage and repair algorithm is used to ensure that the algorithm does not fall into a local optimal solution.

The first-level path of a ship carrier is clustered by the greedy strategy to cluster the vessels to be monitored. The cluster centres obtained are numbered according to the value of the abscissa of the cluster centres in an increasing order and then connect the cluster centres of adjacent vessels to be monitored into a line. We take the midpoint of the two adjacent cluster centres as key points that the ship carrier must visit and plan the sailing path of the ship carrier as shown in Algorithm 1:

Algorithm 1: first -level

//greedy strategy to cluster the vessels to be monitored

//randomly select H initial cluster centre points

//return clustering results $G = \{G_1, G_2, G_3, \cdots, G_H\}$

List greedyCluster(H) ;

//VRP

results = greedyCluster(H)

//take the midpoint of the two adjacent cluster centres $mp = (G_i + G_{i+1})/2$ as the

key points that the ship carrier must visit

```
mp = []
i = 0
 for i in (1, ceil(results.length()/2)))
        mp.add(results[i] + result[i + 1])/2)
 firstLevelPathResult=VRP(mp)
 return  firstLevelPathResult
```

For the second-level multiple UAVs pursuing path planning, a damage and repair algorithm needs to be used to improve the local search ability of the ABC. Since the greedy algorithm of the ABC algorithm finds a local optimal solution, this paper uses the idea of a simulated annealing algorithm to introduce the concepts of "temperature" and "acceptance probability" to obtain a global optimal solution. This enables the ABC to find new solutions with a larger step size in early stages of the algorithm, which is convenient for leaving the local optimum; In the later stages of the algorithm, it accepts new solutions with a longer step size with low probability, which is convenient for convergence and prevents the algorithm from falling into the local optimal solution.

4.1.2. The improved ABC algorithm

In this paper, a two-dimensional vector is used to represent the solution to the SDMU problem. The corresponding solution representation of the path in Figure 1 is shown in Table 2. The first row of the table represents the first-level path of the ship carrier, and the other rows represent the flying paths of the second-level UAVs

belonging to two adjacent UAVs that leave the ship carrier to then return to the ship carrier. D_0 is the base station at which the ship carrier departs from the port, and u_1, u_2, and u_3 are the corresponding numbers of UAVs.

Table 2 Representation of the solution to the SDMU problem

<table>
<tr><td>First-level path</td><td colspan="3">$D_0 \rightarrow A_{k_1} \rightarrow A_{k_2} \rightarrow A_{k_3} \rightarrow A_{k_4} \rightarrow A_{k_5} \rightarrow D_0$</td></tr>
<tr><td rowspan="8">Second-level path</td><td>$D_0 \rightarrow A_{k_1}$</td><td>u_1</td><td>$D_0 \rightarrow v_1 \rightarrow v_2 \rightarrow A_{k_1}$</td></tr>
<tr><td rowspan="2">$A_{k_1} \rightarrow A_{k_2}$</td><td>$u_2$</td><td>$A_{k_1} \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow A_{k_2}$</td></tr>
<tr><td>u_3</td><td>$A_{k_1} \rightarrow v_6 \rightarrow v_7 \rightarrow A_{k_2}$</td></tr>
<tr><td>$A_{k_2} \rightarrow A_{k_3}$</td><td>$u_3$</td><td>$A_{k_2} \rightarrow v_8 \rightarrow v_9 \rightarrow A_{k_3}$</td></tr>
<tr><td>$A_{k_3} \rightarrow A_{k_4}$</td><td colspan="2">NULL</td></tr>
<tr><td rowspan="3">$A_{k_4} \rightarrow A_{k_5}$</td><td>$u_1$</td><td>$A_{k_4} \rightarrow v_{10} \rightarrow v_{11} \rightarrow v_{12} \rightarrow v_{13} \rightarrow A_{k_5}$</td></tr>
<tr><td>u_2</td><td>$A_{k_4} \rightarrow v_{14} \rightarrow v_{15} \rightarrow A_{k_5}$</td></tr>
<tr><td>u_3</td><td>$A_{k_4} \rightarrow v_{16} \rightarrow v_{17} \rightarrow v_{18} \rightarrow A_{k_5}$</td></tr>
<tr><td>$A_{k_5} \rightarrow D_0$</td><td colspan="2">NULL</td></tr>
</table>

Based on the above mentioned process, the steps of the ABC algorithm, which are enhanced based on the simulated annealing idea, to realize the coordinated monitoring of air pollution and the path planning problem of ship carriers and UAVs are listed as follows:

(1) First, initialize m initial solutions y_i, $i \in \{1,2,\cdots,m\}$ representing m food sources where each food source corresponds to a leader. Each solution can be considered a D-dimensional vector where D can also be defined as a number of parameters.

(2) Initialize $w_1 = w_2 = \cdots = w_m = 0$ and $n = 0$. w_i means that solution y_i has not been optimized in the past successive w_i rounds of iterations, and n means that the algorithm has already performed n rounds of iterations.

(3) In each iteration, the leader searches the fields of their respective food sources and obtains new food source y_i', $y_i' = y_i^j + \varphi(y_i^j - y_k^j)$. In the formula, $i, k \in \{1,2,3,4,\cdots,m\}$, $j \in \{1,2,3,4,\cdots,D\}$, $k \neq i$ and φ are random numbers between [-1, 1]. Calculate fit for new food sources. If the fitness value of the new food source $fit(y_i')$ is higher than the fitness value of the current food source, replace the current food source with the new food source, and set the number of searches w_i to 0; otherwise, w_i increases by 1. The neighbourhood operation accepts a solution as input and returns a neighbourhood solution of the solution.

(4) In each iteration, each follower selects a food source in a roulette manner according to the fitness of each current food source and searches it to obtain new solution y_i'. The probability that food source y_i is selected can be expressed as $p(y_i) = 1/\sum_1^m fit(y_i)$. If the fitness of new solution y_i' found by the follower is less than the fitness of current optimal solution y_i $(fit(y_i') > (1-\theta)fit(y))$, the new solution may perform a local search. If the fitness of the last y_i' is higher than y_i, replace y_i with $y_i'$and set w_i to 0; otherwise, increase w_i by 1.

(5) Add the search for the current optimal solution so that the current optimal solution can be fully searched.

(6) When certain food source y_i is not optimized in N_{limit} iterations, the crossover operation is used for the solution, and its descendant is used as new food

source s_i' for subsequent searches $s_i^{'} = s_{min}^{j} + \varphi(s_{max}^{j} - s_{min}^{j})$; in the formula, $i \in \{1,2,3,4,\cdots,m\}$, $j \in \{1,2,3,4,\cdots,D\}$, and φ are random numbers between [-1, 1].

(7) The number of group iterations n is increased by 1.

(8) The algorithm ends when the maximum number of iterations N_{max} is reached.

To construct the neighbourhood solution, this paper uses a damage and repair algorithm. For a complete solution, first select a damage operation to partially destroy the path of the solution, and then choose a repair operation from the repaired set R to rerepair the broken solution, thereby obtaining a new solution. Assuming that the set of destruction operations is F, input solution f is to perform a damage and repair operation ($f' = r(d(f))$) and return the new solution after repair, as shown in Algorithm 2:

Algorithm 2: second -level

```
NeighborhoodOperator(f, F)
    damage = DamageOperator(F)
    repaire = RepaireOperator(R)
    f' = r(d(f))
    f' = initializer(f')
    count = 0
    while the count is less than the maximum number of iterations
        s'' = NeighborhoodOperator(s')
        if f(s'') < f(s')
            s' = s''
    return s'
```

In the algorithm designed in this paper, the damage and repair operations used are only designed for second-level UAV path planning. The damage operations are mainly divided into the random removal of vessels to be monitored, the removal of adjacent

vessels to be monitored and the random removal of the UAV path. The repair operation involves reinserting the vessels to be monitored into the current solution by putting the damage operations into the pool of vessels to be monitored, mainly including greedy insertion, greedy insertion with disturbance, and forced greedy insertion.

(1) Random removal of the vessels to be monitored

The random removal of the vessels to be monitored involves randomly selecting g ships to be monitored from the second-level path in the current solution, and then these vessels are removed from the path of the current solution and stored as the pool of vessels to be monitored. Whenever a vessel to be monitored is removed, the algorithm developed in this paper does not replan the path of the first-level ship carrier, instead it waits until all selected vessels to be monitored are removed. In the subsequent repair operation, the path of the ship carrier is reoptimized.

(2) Removal of adjacent vessels to be monitored

This operation randomly selects a vessel to be monitored as a seed point and then marks the vessel to be monitored that is the closest to the seed point. The selection of the marked seed point is unlimited. Finally, the seed points and marked points are removed from the UAV path of the current solution to the pool of vessels to be monitored. The removal process is the same as the random removal of the vessels to be monitored.

(3) Random removal of UAV path

This operation randomly removes $b(b > 1)$ paths from all second-level paths of the current solution and places all vessels to be monitored on the removed UAV paths

among the pool of vessels to be monitored.

(4) Greedy insertion

The vessels to be monitored among the pool of vessels to be monitored are removed one by one in random order and inserted into the current solution to minimize current insertion cost ($d_{A_{kr_i},k_i,A_{kc_i}}/S_u + |k_i| \times T_v$). This operation needs to meet the following conditions: (a) A vessel can only be inserted into the position where the ship carrier releases or collects drones. (b) A vessel can be inserted into the existing UAV path where the ship carrier releases the UAV, but after its insertion, it ensures that the UAV can return to the ship carrier after monitoring all the vessels to be monitored on the path, satisfying $d_{A_{kr_i},k_i,A_{kc_i}}/S_u + |k_i| \times T_v \leq T_w,\ \ i \in \{1,2,3,\cdots,N_r\},\ \ k_i \in K$. (c) Regardless of the effect of the first-level ship carrier on the path, after all greedy insertion operations are completed, the first-level ship carrier path is replanned.

(5) Greedy insertion with disturbance

This operation is similar to the above greedy insertion operation, while the only difference is that to increase randomness, the greedy insertion with disturbance needs to multiply the random number in the interval [0.8, 1.2] when calculating the insertion cost.

(6) Forced greedy insertion

The insertion method used in this paper always ensures that the time consumed on the UAV flying path is not greater than the maximum endurance time of the UAV during the calculation process.

Therefore, this easily occurs when all UAV paths released by the ship carrier are

fully loaded or the remaining available UAV flying time is less than the current vessel to be monitored and inserted and the cost of creating a new UAV path is relatively high. According to the calculation method of the greedy insertion operation presented in this paper, the vessel to be monitored cannot be assigned to the position where the ship carrier releases the UAV and must be inserted into the position where the next ship carrier releases the UAV, even if the vessel to be monitored is more suited to the position assigned to the current ship carrier to release the UAVs. To solve this problem, forced greedy insertion operation ignores whether the UAV has enough time to monitor the vessel to be monitored when calculating the cost of insertion.

4.2. An improved ABC algorithm for the SDMU path planning model

SDMU path planning is examined in this section based on three aspects: path planning coordination between the ship carrier and UAVs, coordination between UAVs, and UAV task reassignment in a dynamic environment. A flowchart introducing the algorithmic framework is shown in Fig 2.

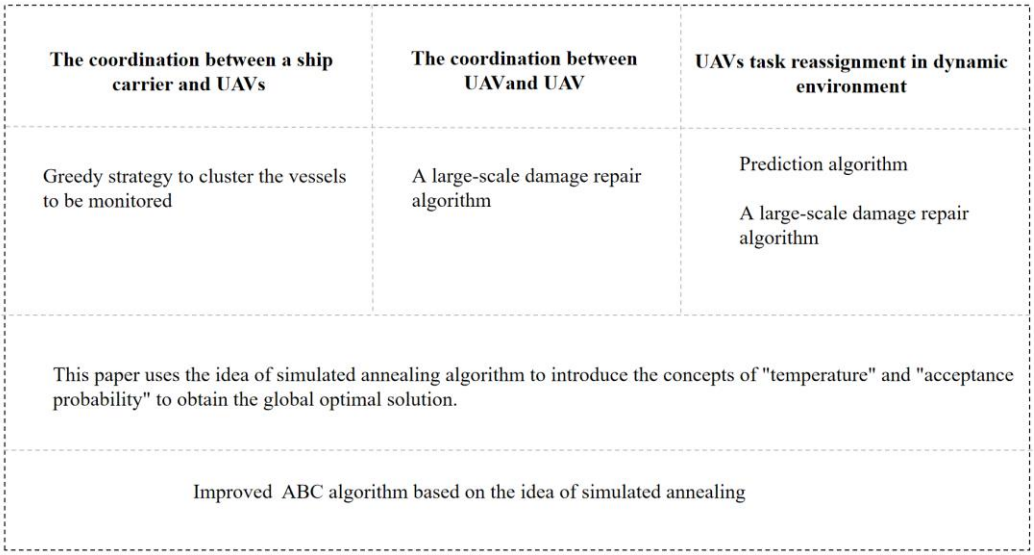

Fig 2 Algorithmic framework

4.2.1. Path planning coordination between a ship carrier and UAVs

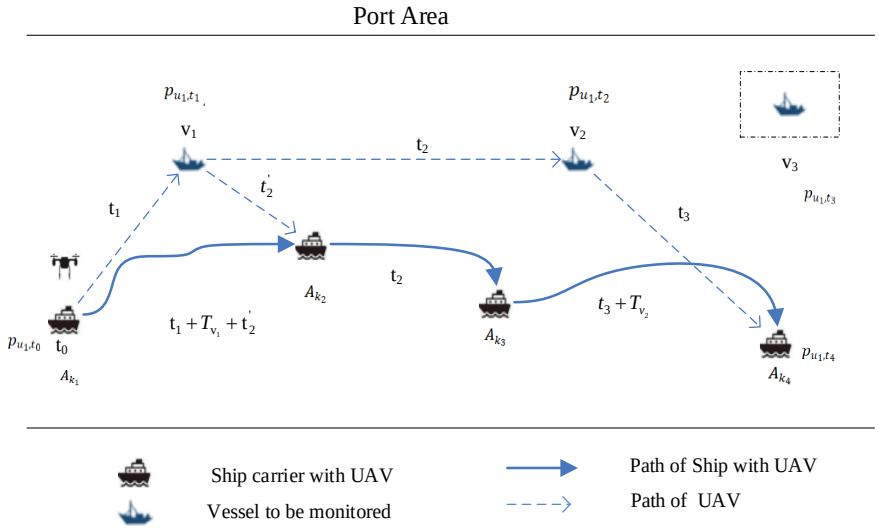

Fig 3 Coordination of a ship carrier and a UAV

Fig 3 presents as an example to explain how the path planning of the ship carrier and UAVs is coordinated. At t_0, a UAV leaves the ship carrier, and the UAV travels from A_{k_1} to p_{u_1,t_1} to monitor the air pollution of the vessel to be monitored, v_1. The time needed to perform this task is $t_1 = d(A_{k_1}, p_{u_1,t_1})/S_u$. At this time, the ship carrier continues to move forward following the predetermined route. After the UAV has finished checking the emissions of vessel v_1, it needs to identify the vessel to be monitored that is closest to its current position. After calculation, it is found that the vessel to be monitored is v_2. The time it takes for the UAV to fly from vessel v_1 to vessel v_2 is $t_2 = d(p_{u_1,t_1}, p_{u_1,t_2})/S_u$. At this time, it is necessary to judge in advance whether the UAV can return to the ship carrier (A_{k_4}) after monitoring vessel v_2, that is, to compare the remaining time for the UAV to monitor vessel v_2, $T_L = T_w - t_1 - t_2 -$

$T_{v_1} - T_{v_2}$, to the time needed for the UAV to travel from vessel v_2 to ship carrier (A_{k_4}), $t_3 = d(p_{u_1,t_2}, A_{k_4})/S_u$. When $T_L > t_3$, the UAV can return to ship carrier (A_{k_4}) after monitoring vessel v_2, and the UAV will continue to perform the task of monitoring vessel v_2. When $T_L < t_3$, the UAV cannot return to ship carrier (A_{k_4}) after monitoring vessel v_2. At this time, the UAV is limited by its remaining electrical energy. After monitoring vessel v_1, the UAV cannot continue to monitor vessel v_2. The UAV needs to immediately return to the ship carrier (A_{k_2}).

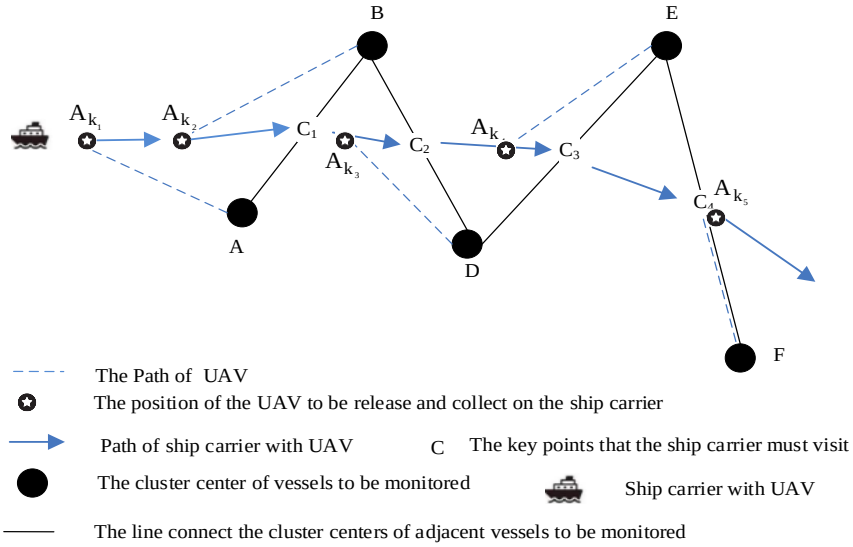

Fig 4 Schematic diagram of solving the collaborative idea of ship carrier and UAV

The ABC algorithm based on the simulated annealing idea is employed to solve the problem of first-level cooperative path planning between the ship carrier and the UAVs. Take Fig 4 as an example to explain the solution to this problem:

(1) Identify the number of vessels N_v to be monitored in the port and use the greedy strategy to cluster the vessels to be monitored. Randomly select H initial cluster centre points, set the number N_H of iterations, continuously update the center points of the clusters, and stop the iteration when the center points do not change or

when the number of iterations is reached to obtain the clustering results $G = \{G_1, G_2, G_3, \cdots, G_H\}$. For example $\mathrm{A, B, D, E, F}$ in Fig 4.

(2) The cluster centers obtained in step (1) are numbered according to the value of the abscissa of the cluster centers from small to large and then connect the cluster centers of adjacent vessels to be monitored G_i and G_{i+1} into a line, where $i \in \{1,2,3,\cdots,q\}$ $(i+1) \in \{1,2,3,\cdots,q\}$. For example $\mathrm{A-B-D-E-F}$ in Fig 4. Taking the midpoint of two adjacent cluster centers $mp = (G_i + G_{i+1})/2$ as the key points that the ship carrier must visit, such as $\mathrm{C_1 - C_2 - C_3 - C_4}$ in Fig 4, after which the sailing path of the ship carrier is planned;

(3) Calculate the total time required to use a UAV to monitor each cluster. This is equal to the sum of the distances of all vessels to be monitored in each cluster divided by the flying speed of the UAV plus the number of vessels to be monitored in the category multiplied by the monitoring time required for each vessel to be monitored. Let N_r denote the number of UAVs that need to be released, and let N_r be determined by the total monitoring time and maximum flying time of a single UAV. When calculating, it is necessary to ensure that the value of Eq (2) is the smallest, and the following constraints must be met. Constraint (9) is used to ensure that the time it takes for the UAV to perform a task (the UAV leaves the ship carrier + performs monitoring tasks + returns to the ship carrier) does not exceed the maximum endurance time. Constraint (14) holds that because a UAV needs to consume electricity when traveling to and from the ship carrier, the number of UAVs released is greater than the number of UAVs required to monitor all vessels to be monitored in the cluster. Otherwise, a UAV

may not return to the ship carrier after completing a monitoring task. Constraint (15) indicates that the number of UAVs released cannot exceed the total number of UAVs carried by the ship carrier.

$$d_{G_i}/S_u >= N_r \tag{14}$$

$$i \in \{1,2,3,\cdots,q\}$$

$$N_r <= N \tag{15}$$

where d_{G_i}/S_u represents the number of UAVs required to monitor all the vessels to be monitored in the cluster without considering the electricity consumption of the UAVs traveling to and from the ship carrier.

(4) When the ship carrier sails a distance of L from the cluster centre of the vessels to be monitored, it will start to release the UAV to undertake the monitoring task. In this paper, L is half the farthest distance from adjacent cluster centers, that is, $\max\{d_{G_1,G_2}, d_{G_2,G_3}, \cdots, d_{G_i,G_{i+1}}, \cdots, d_{G_{H-1},G_H}\}/2$. As shown in Fig 4, the distance between cluster E and cluster F is the largest, so L=$d_{E,F}/2$. That is, when the ship carrier travels from left to right to $A_K = \{A_{k_1}, A_{k_2}, A_{k_3}, A_{k_4}, A_{k_5}\}$, the UAV starts to fly to monitor the vessels to be monitored in each cluster. The positions of C_5 and A_{k_5} are coincident. According to the triangle theorem, the distance range from the cluster center to the ship's route is [0, L]. When the connection line of the two clusters with the farthest distance between the two adjacent cluster centers is perpendicular to the driving route of the ship carrier, the ship carrier will release the drone at the position $L >= max\{d_{G_1,G_2}, d_{G_2,G_3}, \cdots, d_{G_i,G_{i+1}}, \cdots, d_{G_{H-1},G_H}\}/2$, so when the set L is less than

$max\{d_{G_1,G_2}, d_{G_2,G_3}, \cdots, d_{G_i,G_{i+1}}, \cdots, d_{G_{H-1},G_H}\}/2$, It will appear that the vessels to be monitored in the two clusters with the farthest distance between the two adjacent cluster centers will not release UAVs for monitoring. The L we choose in this way can ensure that the ship carrier can fly the UAV on the driving route to monitor vessels to be monitored in all clusters.

(5) Determine the number of UAVs released in each cluster according to step (3) to obtain the monitoring path of N_r UAVs and calculate the connection distance from the positions of all UAVs and vessels to be monitored at a certain time. The distance between all UAVs and vessels to be monitored is L_1, and distance L_2 is the distance connecting all vessels to be monitored. Based on the principle of the shortest total route, disconnect one of the two adjacent lines of each UAV and obtain N_r paths starting from the UAV, and the UAV will follow this path during monitoring.

4.2.2. Multiple UAVs collaborative path optimization

The ABC algorithm based on the simulated annealing idea is applied to solve the problem of second-level cooperative path planning between two UAVs. The solution is as follows:

During the monitoring process, if there any idle UAV is not assigned any monitoring tasks, we need to consider whether it needs to be deployed and cooperate with other UAVs to reduce the overall time required to complete the monitoring task. The procedure employed to make the decision is presented as follows:

(1) Calculate the remaining tasks of other working UAVs on their respective paths (the number of vessels to be monitored multiplied by the time that the UAV monitors

one vessel to be monitored + the length of the route divided by the flying speed of the UAV). According to the extent of the remaining tasks, sort the paths of the UAVs performing the tasks in sequence;

(2) According to the sorted working UAV paths, calculate whether the idle UAV monitoring of each vessel can reduce the total monitoring time needed to determine the first vessel that can reduce the total monitoring time. For an example, see vessel v_3 in Fig 5;

(3) If the vessel is not the last vessel in the path of a working UAV (vessel v_3 in Fig 5), disconnect the edge between the vessel and the adjacent vessel in the opposite direction of the flying path of the working UAV (vessels v_3、v_4 and v_5 in Fig 5). If the vessel is the last on the flying path of a working UAV, the vessels on the route are relatively evenly allocated to two UAVs for monitoring and at the same time cancel out the tasks of other UAVs to monitor these vessels;

(4) Because of the limitations of UAV battery capacity, when the UAV battery is insufficient, the UAV needs to return to the ship carrier for charging. Each UAV will continue to participate in monitoring tasks after charging is complete.

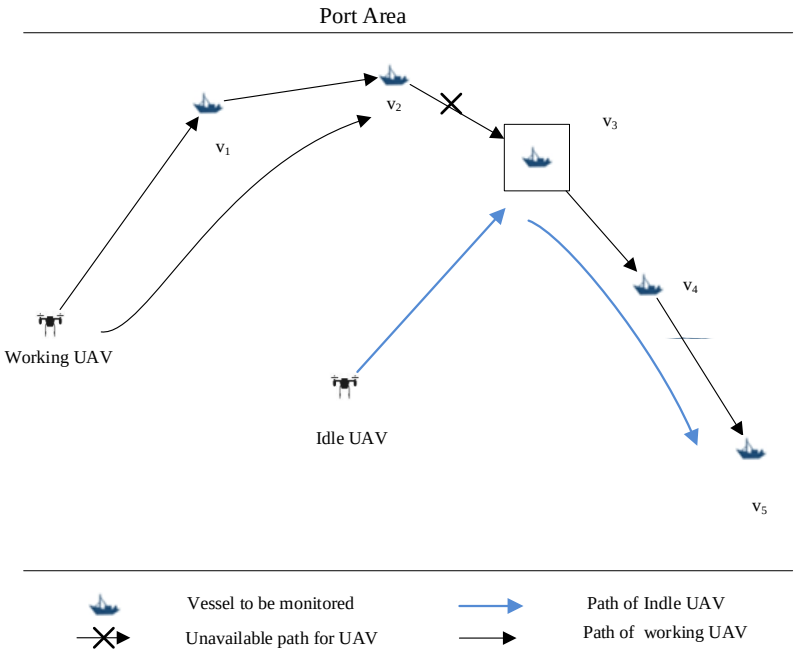

Fig 5 Schematic of multiple UAVs collaboration

4.2.3. UAVs task reassignment in a dynamic environment

Since the vessels to be monitored in the port are sailing at sea and their positions change in real time, it is necessary to redistribute UAV tasks depending on the dynamic changes of these ships. For multiple UAVs monitoring of vessels in the port, we need to consider the following three situations that may lead to the redistribution of UAVs: new vessels entering the port, unmonitored vessels leaving the port, and the location updating of vessels to be monitored in the port. UAVs have to be redistributed when any of these three situations happen individually or collectively. Assuming that the sequence of tasks that a UAV needs to undertake is shown in Fig 6(a), at time $T = t_0$, the UAV deployed to monitor vessel v_1 starts to conduct the vessel's air pollution monitoring task.

The time needed for the UAV to monitor vessel v_1 is T_{v_1}. At time $T = t_0 + T_{v_1}$,

the UAV completes the monitoring of vessel v_1. Fig 6(b) shows that at $T = t_0 + T_{v_1}$, a new vessel to be monitored v_4 enters the port, which triggers the redistribution procedure as well as the path replanning of the UAV. After the UAV monitors vessel v_1, the next monitoring point of the UAV will change from vessel v_2 to vessel v_4. Fig 6(c) shows that at $T = t_0 + T_{v_1}$, the vessel to be monitored v_2 leaves the port, which again triggers the path replanning procedure. After the UAV monitors vessel v_1, the next monitoring point of the UAV will change from vessel v_2 to vessel v_3. Fig 6(d) shows that at $T = t_0 + T_{v_1}$, since the vessel to be monitored in the port is dynamic, after time T_{v_1}, the position of the vessel to be monitored in the port changes, which leads to a replanning of the path of the UAV. After the UAV monitors vessel v_1, the next monitoring point of the UAV will change from vessel v_2 to vessel v_3.

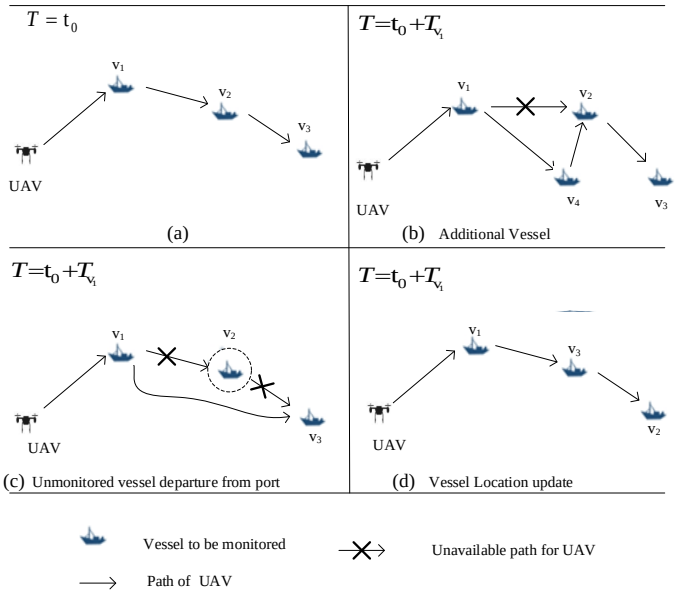

Fig 6 Scenario that triggers dynamic reallocation

A prediction algorithm is required in the above path replanning procedure to predict the real-time, dynamic positioning of the monitored vessels. We design the

prediction algorithm as follows:

(1) When triggering multiple UAVs path replanning, the prediction algorithm is utilized to predict the location of the vessel to be monitored at time $t + T_{v_i}$ based on the location data of the vessel to be monitored in the Ship Automatic Identification System (AIS) at time t.

(2) According to the dynamic environmental changes, the planned path of UAV monitoring is reacquired. Since the vessel is always in motion, after monitoring a vessel, the vessel is taken as a fixed point, and distance L_2 connecting all vessels to be monitored at this moment is calculated. If the difference between the current length of L_2 and the minimum value of L_2 at all moments exceeds the preset threshold, vessels that have not been monitored by UAVs, vessels that have not been monitored, and vessels entering the port are regenerated. A new path is composed (Shen et al., 2020).

5. Experiment and analysis

5.1. Data preparation

In this section, to demonstrate the effectiveness of the improved ABC algorithm based on the idea of simulated annealing, this paper uses MATLAB R2018a to implement the algorithm. This experiment was carried out on a laptop with an Intel(R) Core(TM) i5-7500 CPU @ 3.4 GHz processor and 4 GB of RAM. In Section 5.1, the UAV flying paths were formulated based on the real vessel position data. Section 5.2 analyses the sensitivity of different parameters to the experimental results.

5.2. Data preparation and UAV flying path solution

In the experiment, we investigated the pollution monitoring of the vessels entering

and leaving Tianjin Port at 13:00 on November 29, 2020. We obtained the position data of 90 vessels sailing in Tianjin Port from the AIS, as shown in Fig 7. We use the obtained position data of 90 vessels to perform a simulation verification. Based on the actual geographic location of the port, we scale the monitored range and vessel location to a 1000*800 area to run our algorithm. We first use the greedy strategy to cluster the vessels to be monitored. The cluster centers obtained are numbered according to the value of the abscissa of the cluster center from small to large and then connect the cluster centers of adjacent vessels to be monitored into a line, take the midpoint of the two adjacent cluster centers as the key points that the ship carrier must visit, and plan the sailing path of the ship carrier. Through the coordination of the first level of a ship carrier and UAV, we define the route of the ship carrier as $(100,740) \rightarrow (200,600) \rightarrow (350,450) \rightarrow (540,340) \rightarrow (650,300) \rightarrow (800,200) \rightarrow (1040,60) \rightarrow (100,740)$. The default values of the parameters involved in this example are shown in Table 3.

The data are obtained from ShipXun.com at http://www.shipxy.com/.

Fig 7 Schematic of vessels entering and leaving Tianjin Port

Table 3 Default parameter description

Number of UAVs	3 racks
Average flying speed of UAVs	1000 m/min
Average monitoring time per vessel	2 min/vessel
Number of ships with UAVs	1 ship
Speed of a ship with UAVs	26 m/min
Number of vessels to be monitored	90 vessels
Maximum UAV flying time	35 min

The main parameters listed in Table 3 are defined as follows:

The number of UAVs used: The number of UAVs refers to the number of UAVs considered in Shen (2020).

Number of ship carriers: Based on the measurement of vessel exhaust gas employed by the Maritime Safety Administration, we set the number of ship carriers to 1.

Average monitoring time per vessel: This period refers to the average monitoring time per vessel set by Shen (2020).

The speed of a ship carrier: The law enforcement officers of the local Maritime Safety Administration use UAVs on a patrol ship to carry out the comprehensive air pollution monitoring of vessels in dense waters. With a cumulative cruising mileage of 51 nautical miles and a water area of 133 square kilometres, the speed of the patrol ship is calculated as 26 m/min.

Maximum UAV flying time: This paper uses the maximum flying time of the DJI M210 UAV in the experiment.

This paper lists the initial positions of 90 vessels to be monitored, one ship carrier and three UAVs. At the initial moment, the 3 UAVs and ship carrier are in the same position (v_i represents the i^{th} vessel, and D_0 represents the starting point of the ship carrier leaving the port as well as the initial position of the 3 UAVs; (x, y) represents the position coordinates), as shown in Table 4. Fig 8 shows the initial positions of the vessels to be monitored, ship carrier and UAVs. Fig 9 shows the SDMU path planning result calculated by the improved ABC algorithm for the above mentioned scenario, and Table 5 shows the corresponding specific UAV flying path. Fig. 10 shows the two newly added vessels entering the port based on Fig. 8 and invokes the path replanning algorithm to obtain the collaborative route of the first level of a ship carrier and UAV as $(200,600) \rightarrow (350,450) \rightarrow (540,340) \rightarrow (650,300) \rightarrow (800,200) \rightarrow (1040,60) \rightarrow (0,750)$ and the collaborative route of the second level of the UAVs as shown in Fig 11. Table 6 shows the specific UAV flight path replanning results.

Table 4 Initial positions of the vessels to be monitored, ship carrier and UAVs

	x	y
v_1	14.1612200435730	33.7690631808279
v_2	701.341139812628	685.857317220493
v_3	42.5925925925926	703.067417321720
v_4	26.9063180827887	646.276655127741

v_5	37.6906318082789	641.116193237439
⋮	⋮	⋮
v_{86}	859.259259259259	257.525670986474
v_{87}	867.102396514161	317.756616137001
v_{88}	874.945533769063	305.712226679497
v_{89}	1006.31808278867	61.3778810865753
v_{90}	1030.82788671024	54.4992022727935
D_0	0	750

Table 5 SDMU path planning results

	UAV number	UAV monitoring path	Monitoring time (min)
A_{k_1}	d_1	$A_{k_1} \rightarrow v_3 \rightarrow v_2 \rightarrow v_7 \rightarrow v_6 \rightarrow v_8 \rightarrow A_{k_2}$	10.31420
	d_2	$A_{k_1} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_{11} \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{16} \rightarrow v_{17} \rightarrow A_{k_2}$	14.28266
	d_3	$A_{k_1} \rightarrow v_1 \rightarrow v_4 \rightarrow v_5 \rightarrow v_9 \rightarrow v_{10} \rightarrow v_{20} \rightarrow v_{21} \rightarrow A_{k_2}$	14.41853
A_{k_2}	d_1	$A_{k_2} \rightarrow v_{28} \rightarrow v_{27} \rightarrow v_{29} \rightarrow v_{30} \rightarrow v_{32} \rightarrow A_{k_3}$	10.28357
	d_2	$A_{k_2} \rightarrow v_{18} \rightarrow v_{19} \rightarrow v_{34} \rightarrow v_{37} \rightarrow v_{35} \rightarrow v_{36} \rightarrow v_{38} \rightarrow v_{39} \rightarrow v_{40} \rightarrow A_{k_3}$	18.34384
	d_3	$A_{k_1} \rightarrow v_{23} \rightarrow v_{22} \rightarrow v_{24} \rightarrow v_{25} \rightarrow v_{26} \rightarrow v_{33} \rightarrow A_{k_2}$	12.35794
⋮	⋮	⋮	⋮
A_{k_5}	d_3	$A_{k_5} \rightarrow v_{69} \rightarrow v_{71} \rightarrow v_{74} \rightarrow A_{k_6}$	6.22970
	d_2	$A_{k_5} \rightarrow v_{70} \rightarrow v_{72} \rightarrow v_{73} \rightarrow A_{k_6}$	6.28424

	d_1	$A_{k_5} \rightarrow v_{68} \rightarrow v_{75} \rightarrow v_{76} \rightarrow A_{k_6}$	6.30475
A_{k_6}	d_3	$A_{k_6} \rightarrow v_{78} \rightarrow v_{77} \rightarrow v_{87} \rightarrow v_{88} \rightarrow v_{86} \rightarrow v_{79} \rightarrow A_{k_7}$	12.48854
	d_2	$A_{k_6} \rightarrow v_{83} \rightarrow v_{85} \rightarrow v_{89} \rightarrow v_{90} \rightarrow A_{k_7}$	8.28930
	d_1	$A_{k_6} \rightarrow v_{80} \rightarrow v_{81} \rightarrow v_{82} \rightarrow v_{84} \rightarrow A_{k_7}$	8.30835
Total monitoring time		126.69466 min	

Table 6 SDMU path replanning

	UAV number	UAV monitoring path	Monitoring time (min)
A_{k_1}	d_1	$A_{k_1} \rightarrow v_{15} \rightarrow v_{14} \rightarrow v_{13} \rightarrow v_{16} \rightarrow v_{17} \rightarrow v_{18} \rightarrow A_{k_2}$	12.22149
	d_2	$A_{k_1} \rightarrow v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_{20} \rightarrow v_{21} \rightarrow A_{k_2}$	14.39031
	d_3	$A_{k_1} \rightarrow v_3 \rightarrow v_7 \rightarrow v_8 \rightarrow v_9 \rightarrow v_{11} \rightarrow v_{10} \rightarrow v_{12} \rightarrow A_{k_2}$	14.34933
A_{k_2}	d_1	$A_{k_2} \rightarrow v_{22} \rightarrow v_{24} \rightarrow v_{25} \rightarrow v_{26} \rightarrow v_{33} \rightarrow v_{92} \rightarrow A_{k_3}$	12.38559
	d_3	$A_{k_2} \rightarrow v_{23} \rightarrow v_{28} \rightarrow v_{27} \rightarrow v_{29} \rightarrow v_{30} \rightarrow v_{31} \rightarrow v_{32} \rightarrow A_{k_3}$	14.32686
	d_2	$A_{k_2} \rightarrow v_{19} \rightarrow v_{34} \rightarrow v_{37} \rightarrow v_{35} \rightarrow v_{36} \rightarrow v_{39} \rightarrow v_{40} \rightarrow v_{38} \rightarrow A_{k_3}$	16.50014
⋮	⋮	⋮	⋮
A_{k_5}	d_3	$A_{k_5} \rightarrow v_{64} \rightarrow v_{70} \rightarrow v_{72} \rightarrow v_{73} \rightarrow A_{k_6}$	8.31453
	d_1	$A_{k_5} \rightarrow v_{69} \rightarrow v_{71} \rightarrow v_{74} \rightarrow A_{k_6}$	6.22970
	d_2	$A_{k_5} \rightarrow v_{75} \rightarrow v_{76} \rightarrow v_{77} \rightarrow A_{k_6}$	6.29898
A_{k_6}	d_1	$A_{k_6} \rightarrow v_{78} \rightarrow v_{87} \rightarrow v_{88} \rightarrow v_{86} \rightarrow v_{17} \rightarrow v_{18} \rightarrow A_{k_2}$	10.29106
	d_2	$A_{k_6} \rightarrow v_{81} \rightarrow v_{83} \rightarrow v_{85} \rightarrow v_{86} \rightarrow v_{89} \rightarrow A_{k_7}$	8.29414

	d_3	$A_{k_6} \rightarrow v_{80} \rightarrow v_{82} \rightarrow v_{84} \rightarrow v_{90} \rightarrow A_{k_7}$	8.30923
Total monitoring time		134.72770 min	

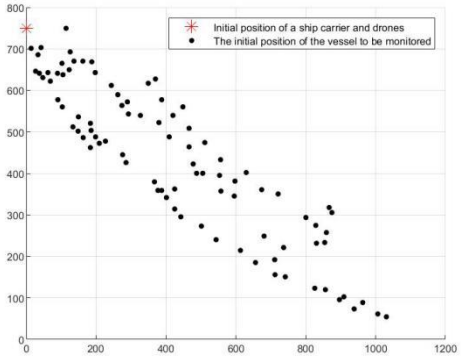

Fig 8 Schematic of the initial positions of the vessels to be monitored, ship carrier, and UAVs

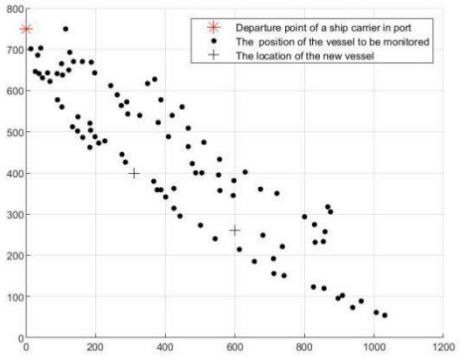

Fig 10 Schematic of newly added vessels in the port

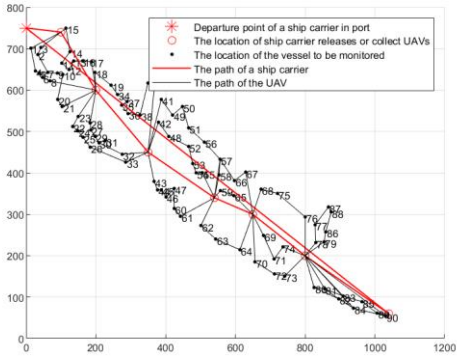

Fig 9 Schematic of SDMU path planning results

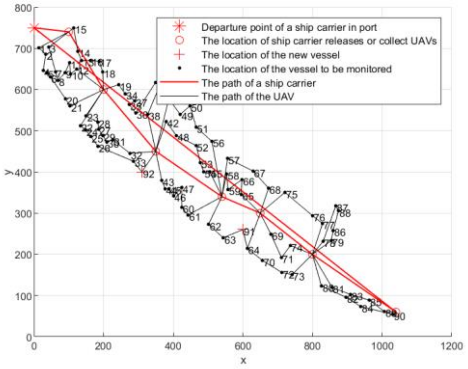

Fig 11 Schematic of SDMU path replanning results

5.3. Sensitivity analysis

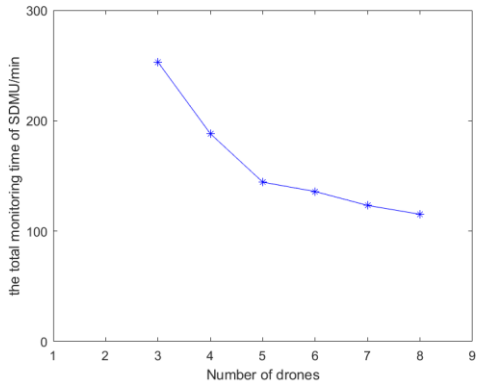

Fig 12 Relationship between the total monitoring time of SDMU and the number of UAVs

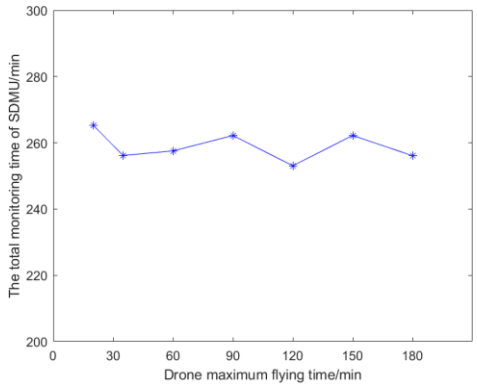

Fig 13 Relationship between the maximum flying time of UAVs and the total monitoring time of SDMU

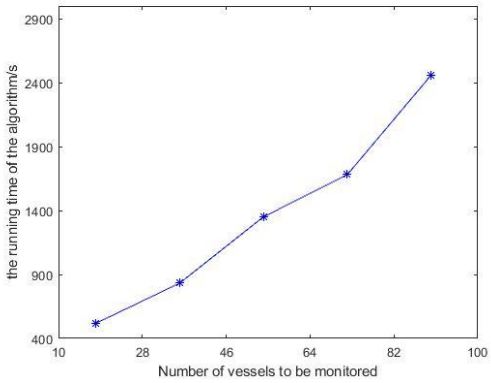

Fig 14 Relationship between the number of vessels to be monitored and the running time of the algorithm

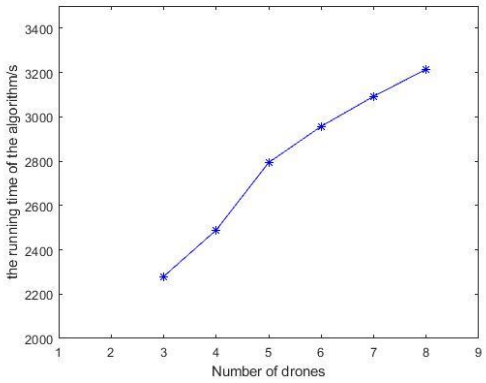

Fig 15 Relationship between the number of UAVs and the running time of the algorithm

Fig 12 shows that as the number of UAVs increases, the total time taken for the SDMU monitoring task will decrease; however, the reduction rate of the total monitoring time of the SDMU decreases. When the number of UAVs exceeds a certain threshold, further growth in the number of UAVs will have a minimal effect on the reduction of the total time spent on SDMU monitoring tasks. In this particular situation, among the main factors that restrict the total monitoring time of the SDMU are the monitoring time spent on a single vessel to be monitored and the time taken to travel from the departure point of the port to the return point of the ship carrier.

Fig 13 shows that when the maximum flying time of the UAV is increased, the

total time of the SDMU monitoring task does not change significantly. Xia et al. (2019) concluded that as the maximum endurance time of the UAV increases, the total time of UAV monitoring missions shows a decreasing trend. Compared to the finding in Xia et al. (2019) show that the introduction of ship carriers as a mobile supply base for UAVs can effectively address UAV electric power limitations.

Fig 14 shows that when the number of UAVs is unchanged, the number of vessels to be monitored increases from 15 to 90, and the calculation time of the algorithm increases by 1941.962 s. When the number of vessels to be monitored in the port is greater than 73, the increase in the running time of the algorithm is relatively significant, and the algorithm can show better computing power when the number of vessels to be monitored in the port is not greater than 73. The reason for this result is that the number of UAVs is limited. When the number of monitoring target nodes increases, the number of calculations that the algorithm requires to analyse damage and repair operations increases; thus, the running time of the algorithm increases.

Fig. 15 shows that when there are 90 vessels to be monitored, the number of available UAVs increases from 3 to 8, and the calculation time increases from 2279.677 s to 3214.958 s. When the number of UAVs is greater than 4, the increase in the running time of the algorithm is relatively significant, and the algorithm can show better computing power when the number of UAVs carried by a ship is not greater than 4. The reason for this result is that when the number of UAVs increases, there are more node pairs from origin to destination during monitoring tasks, which causes an increase in the calculation time when the algorithm is running.

According to the calculation results in Figure 14 and 15, we may find that the computational time required may be too long for dynamic cases where the algorithm needs to be executed repetitively to deal with the dynamic location changes of drones and vessels. Therefore, additional methods need to be employed to apply our algorithm in dynamic case. The possible approaches may include the adoption of parallel computing to reduce the computational time or the development of heuristic rules to update UAVs' routes based on the dynamic locations of monitoring targets.

Conclusion

This paper proposes an improved ABC algorithm designed to solve an SDMU problem. The SDMU supports the cooperation of a ship and multiple UAVs in the air pollution monitoring of vessels entering and leaving a port. The SDMU allows multiple UAVs to be launched and collected at different locations, minimizing the time needed for the coordinated monitoring by a ship and UAVs. The real-time AIS data of ships in Tianjin Port were used as an input for as a simulation experiment to verify our proposed algorithms, and the improved ABC algorithm was developed to solve the problem. The algorithm used in this paper is proven to be able to provide a better solution over a relatively short period when managing large-scale data. It can therefore significantly improve ship emission detection for sustainable shipping and ports.

However, this study still has limitations. For instance, the influence of sea wind speeds and the law of air control on the flying paths of UAVs and the use of larger-scale data need further analysis. In addition, the planning scene considered in this paper involves a two-dimensional environment, and the flying height of UAVs is not

considered. In the future, the SDMU problem may be investigated in a three-dimensional scenario, and more constraints can be added to the path planning model. Furthermore, how to choose the positions for the ship carrier to release UAVs need more discussions. Finally, the research should be tested in the actual port environment. The model and the algorithms of this paper should be continuously optimized according to the real practical application scenarios.

The future research direction may be the development of exact solution to SDMU. Although the meta-heuristic method employed in the study has been proved to be effective in the previous literature and many practical applications, it could not guarantee to provide the exact solution.

Acknowledgements

This work was supported by The National Natural Science Foundation of China (72171032; 71702019) and The Ministry of Education of Humanities and Social Science Project (21YJAZH070; 21YJA630048).

References

United Nations. Review of Maritime Transport 2019: United Nations Conference on Trade and Development[R], UN, 2019.https://unctad.org/en/pages/PublicationWebflyer.aspx?Publicati-on id =2563.

UNCTAD, 2019. Song Jin, Shipping report. Accesed November 1, 2019. http://news.cri.cn/20191101/8535259f-fd26-17e5-6f14-f63cd74ba10f.html.

Lee H, Park D, Choo S, et al. Estimation of the Non-Greenhouse Gas Emissions Inventory from Ships in the Port of Incheon[J]. Sustainability, 2020, 12(19): 8231. https://doi.org/10.3390/su12198231.

Liang G Y. Neglected pollution sources[J]. Environment. 2016, (06):17-19.

Xia J, Wang K, Wang S. Drone scheduling to monitor vessels in emission control areas[J]. Transportation Research Part B: Methodological, 2019, 119: 174-196. https://doi.org/10.1016/j.trb.2018.10.011.

CHINA MSA, 2019, 2020 global marine fuel sulphur limit implementation plant. Accesed October, 2019. https://baijiahao.baidu.com/s?id=1648416313478014400&wfr=spider&for=pc.

CHINA MSA, Ministry of Transport of the People's Republic of China,2019. by Maritime Safety Administration of the People’s Republic of China on 20 Nov. 2019. https://www.msa.gov.cn/page/search.do.

Shen L, Wang Y, Liu K, et al. Synergistic path planning of multiple UAVs for air pollution detection of ships in ports[J]. Transportation Research Part E: Logistics and Transportation Review, 2020, 144: 102128. https://doi.org/10.1016/j.tre.2020.102128.

Zhou F, Hou L, Zhong R, et al. Monitoring the compliance of sailing ships with fuel sulfur content regulations using unmanned aerial vehicle (UAV) measurements of ship emissions in open water[J]. Atmospheric Measurement Techniques, 2020, 13(9): 4899-4909. https://doi.org/10.5194/amt-13-4899-2020.

Galbraith, J. (2020). US drone strike in Iraq kills Iranian military leader Qasem Soleimani. American Journal of International Law, 114(2), 313-323. https://doi.org/10.1017/ajil.2020.15.

Wang S, Han Y, Chen J, et al. A Deep-Learning-Based Sea Search and Rescue Algorithm by UAV Remote Sensing[C]//2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC). IEEE, 2018: 1-5.

Li X, Gong L, Liu X, et al. Solving the last mile problem in logistics: A mobile edge computing and blockchain‐based unmanned aerial vehicle delivery system[J]. Concurrency and Computation: Practice and Experience, 2020: e6068. https://doi.org/10.1002/cpe.6068.

Huang Z, Chen C, Pan M. Multiobjective UAV path planning for emergency information collection and transmission[J]. IEEE Internet of Things Journal, 2020, 7(8): 6993-7009. https://doi.org/10.1109/JIOT.2020.2979521.

Sun P, Boukerche A. Performance modeling and analysis of a UAV path planning and target detection in a UAV-based wireless sensor network[J]. Computer Networks, 2018, 146: 217-231. https://doi.org/10.1016/j.comnet.2018.09.022.

Liu Y, Luo Z, Liu Z, et al. Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions[J]. IEEE Access, 2019, 7: 63504-63518. https://doi.org/10.1109/ACCESS.2019.2914352.

Yu K, Budhiraja A K, Buebel S, et al. Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations[J]. Journal of Field Robotics, 2019, 36(3): 602-616. https://doi.org/10.1002/rob.21856.

Hu M, Liu W, Lu J, et al. On the joint design of routing and scheduling for vehicle-assisted multi-UAV inspection[J]. Future Generation Computer Systems, 2019, 94: 214-223. https://doi.org/10.1016/j.future.2018.11.024.

Wang Z, Sheu J B. Vehicle routing problem with drones[J]. Transportation research part B: methodological, 2019, 122: 350-364.

Moshref-Javadi M, Hemmati A, Winkenbach M. A truck and drones model for last-mile delivery: A mathematical model and heuristic approach[J]. Applied Mathematical Modelling, 2020, 80: 290-318. https://doi.org/10.1016/j.tre.2020.101887.

Booth K E C, Piacentini C, Bernardini S, et al. Target Search on Road Networks with Range-Constrained UAVs and Ground-Based Mobile Recharging Vehicles[J]. IEEE Robotics and Automation Letters, 2020, 5(4): 6702-6709.