Monitoring Scheduling of Drones for Emission Control Areas: An Ant Colony-Based Approach

Zhao-Hui Sun, *Member, IEEE*, Xiaosong Luo, *Graduate Student Member, IEEE*, Edmond Q. Wu, *Member, IEEE*, Tian-Yu Zuo, Zhi-Ri Tang, *Student Member, IEEE*, and Zilong Zhuang

***Abstract*— The drone has become a promising tool to improve the efficiency of vessel emission monitoring in emission control areas of the part due to its high mobility. However, how to optimize the flight path of drones to improve the weighted sum of monitored vessels, i.e., drone scheduling problem (DSP), is a not yet fully researched problem. In this paper, different from the classic optimization solution method used by the literature, an efficient ant colony-based algorithm is developed to solve DSP. Given the characteristics of DSP, a hierarchical-based pheromone update strategy and partition-based pheromone management mechanism are proposed to optimize the typical ant colony algorithm. Numerical experiments not only illustrate the feasibility of using the ant colony algorithm to solve DSP, but also show that the algorithm we proposed outperforms other compared methods in terms of the solution quality and the solving speed under different problem scales.**

***Index Terms*— Drone scheduling, emission control area, ant colony optimization, Heuristic algorithm.**

I. Introduction

As A low-cost means of bulk commodity transportation, the international shipping industry is developing rapidly in recent years. Statistics show that there are more than 80% of globally traded goods being carried by maritime transportation [1]–[3]. Behind the prosperity of the international shipping industry, the problem of environmental pollution caused by bunker fuels has become increasingly apparent. 95% of the world's shipping fleet is driven by bunker fuels, particularly residual oil [4]. It leads to a significant number of environmental pollutants emitted by container vessels during the sailing, such as carbon dioxide (CO_2), nitrogen oxides (NO_X), sulphur oxides (SO_X), and inhalable particulate matter (PM) [5]–[8]. Several emission inventory surveys show that vessel pollution has become an important source of air pollution in port cities [9], [10]. Since the pollutants emitted by vessels will have a large negative impact on the air quality of the port and the human health of residents, the government of many countries sets up emission control areas (ECAs) within 12-15 nautical miles near the port to reduce the air pollution caused by vessels [11]–[15].

In ECA, the management department of the port will monitor vessels and detect whether various pollutants emission of these vessels exceeds the standard [16]. Many countries and institutions have formulated emission control standards [17], [18]. For example, the four ECAs established by the International Maritime Organization (IMO) in 2015 require that vessels in ECAs must use fuels containing less than 0.1% of sulphur. However, the standards and decrees established for the ECA are not effective in the actual implementation process. Several historical statistics show that the proportion of vessels with illegal emission reaches 12.3% [19]. This is because to meet the strict emission requirements of ECA, vessel transportation contractors must use clean and high-quality fuels to reduce the content of pollutant emissions. It will greatly increase operating costs and reduce profits. Therefore, it is necessary to increase the monitoring strength of pollutant emission of vessels, and severely crackdown on vessels that do not comply with emission standards.

Traditional vessel pollutant monitoring adopts the method of arranging management personnel on board to verify the technical file of the diesel engine and pollutant after-treatment device. This method requires a lot of manpower and is inefficient. It is not suitable for use in scenes where there are a large number of vessels entering and leaving daily, such as large ports. The use of fixed or mobile sniffing telemetry technology to detect pollutant emissions from vessels is also a widely used monitoring method. For example, the port of Gothenburg in Sweden and the Great Belt Bridge in Denmark combine the fixed sniffers and monitoring analysis platform to monitor the emissions of vessels. However, above monitoring methods generally have a deficiency of low efficiency or easily affected by the environment [20]. Thanks to the development

Manuscript received May 9, 2021; revised July 25, 2021; accepted August 13, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62171274 and Grant U1933125 and in part by the 2019 Shanghai Service Industry Guidance Project supported by the Shanghai Municipal Development and Reform Commission under Grant 2019-01. The Associate Editor for this article was B. F. Ciuffo. *(Corresponding authors: Zhao-Hui Sun; Edmond Q. Wu.)*

Zhao-Hui Sun, Xiaosong Luo, and Zilong Zhuang are with the Department of Industrial Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zh.sun@sjtu.edu.cn; luo_x_s@sjtu.edu.cn; zhuangzilong910126@sjtu.edu.cn).

Edmond Q. Wu is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China, also with the Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: edmondqwu@163.com).

Tian-Yu Zuo is with the School of Automation, Nanjing University of Information Science & Technology, Nanjing 210044, China (e-mail: tian_yu_zuo@163.com).

Zhi-Ri Tang is with the School of Physics and Technology, Wuhan University, Wuhan 430000, China (e-mail: gerintang@163.com).

Digital Object Identifier 10.1109/TITS.2021.3106305

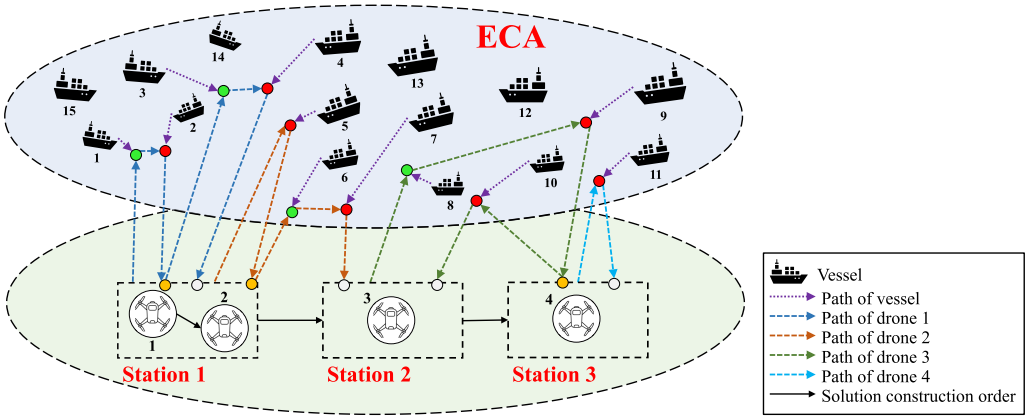

Fig. 1. An example for drone monitoring plan: The colored dot represents the battery status of drones. Green dots indicate that the drone can still conduct the next monitoring task after completing one monitoring task. Red dots indicate that the remaining capacity is not enough to continue the execution of the monitoring task, and it must return to the nearest drone station to its current location to replace the battery. Yellow dots indicate that the drone is replacing the battery. White dots indicate that the drone has completed all the monitoring tasks, then it should return to the nearest station.

of the fields of unmanned aerial vehicles (UAVs) [21], [22] and self-driving vehicles (SDVs) [23], [24], port managers realized the advantages of using drones equipped with sniffer for vessel monitoring [25]–[27]. As a potential new monitoring method, the high mobility of the drone improves the efficiency of monitoring work and the coverage of the monitoring area [27]. There are fewer constraints for drone monitoring due to they can use the airspace of ECA to perform monitoring tasks. It ensures the wide applicability of the drone in pollutant emission monitoring. However, due to the high price of the drone, the reasonable scheduling and planning of in-service drones become particularly important.

Fig. 1 shows an emission monitoring case of 15 vessels by 4 drones within the ECA. The coloured dashed arrow lines in the figure represent the monitoring sequence of different drones. In this case, 4 drones only completed the monitoring of 11 vessels, and the utilization efficiency of drones is not high. Therefore, attention should be paid to how to optimize the flight path of drones to improve the monitoring capability of the drone group, that is, the drone scheduling problem (DSP), is an urgent problem to be solved.

DSP could be described as the variant of the team orienteering problem (TOP) [28], [29] or the traveling salesman problem [30]–[32]. In TOP, a fleet of vehicles is scheduled to visit a set of nodes. Each node is associated with a certain reward [33], [34]. The objective of TOP is to maximize the total received rewards while the travel time of each route must be not more than a time limit [35]–[37]. Similarly, in DSP, a fleet of drones is scheduled to monitor a set of vessels with the monitoring reward. The monitoring reward often comes from the historical violation record. Those vessels with violation records intend to be set the bigger weight as their reward. And, the more the number of violation events, the greater the weight. In this paper, the monitoring weight obtained by the drone group is used as an indicator to evaluate the monitoring performance of the drone group. In other words, the objective of DSP is to search a set of flight paths for drones that make the total monitoring weight maximum in a certain time window.

To describe DSP with the actual constraints in ECA, a mixed-integer linear programming-based time-expanded network model was used in [38]. Then, a Lagrangian relaxation-based method was proposed to solve this model which could speed up the model solving efficiency. However, considering DSP is a kind of NP-hard problem, the increase in the number of drones/ drone stations/ vessels will make it difficult to be solved. Therefore, it is an important task to explore an efficient algorithm with a faster-solving speed and higher solving quality.

There are many previous works that have successfully solved NP-hard problems using computational intelligence methods [39]–[43]. As a typical heuristic method for the NP-hard problem, ant colony optimization (ACO) is widely used to solve various scheduling problems [44], [45]. For example, Gao *et al.* [46] applied ACO to tackle multipoint dynamic aggregation problems in a multi-robot task allocation system. Sun *et al.* [47] applied the ACO-based method to solve the multi-task oriented complex machine layout problem. Liu *et al.* [48] designed the ACO-based algorithm to the allocation problem of cloud virtual machines. Liang *et al.* [49] proposed an improved ant colony system for the public bicycle scheduling problem. These research works show that ACO has strong potential in dealing with complex and dynamic scheduling problems.

In this paper, an ACO-based method is designed for solving DSP, termed ACO-DSP. In detail, we transform the time-expanded network model into a model that could be solved by ACO. Then, the hierarchical-based pheromone update strategy and partition-based pheromone management mechanism are proposed to optimize the performance of ACO. Experiments illustrate that ACO-DSP could obtain better feasible solutions in a shorter time than other compared methods in most cases. Especially in large-scale cases, the solution obtained by ACO-DSP has obvious advantages in solving speed and solving quality.

The main contributions in this paper are as follows.

1) Based on the framework of ACO, the ACO-based method is developed by transforming the time-expanded network

TABLE I
THE LIST OF NOTATIONS

Notation	Description
V	The set of vessels
n	The number of vessels
v_i	The i-th vessel
K	The set of drone stations
k	The number of drone stations
$node$	The vessel node or drone station node
S_i	The coordinate of i-th drone station
$Drone$	The set of all drones
m	The number of all the drones
D_i	The set of drones in i-th drone station
d_j^i	The j-th drone in i-th drone station
r_i	The number of drones in i-th drone station
Γ	The time window of drone scheduling (continuous)
T_{max}	The maximum time window
$\widehat{\Gamma}$	The time window of drone scheduling (discrete)
$\widehat{\Gamma_v}$	The allowed monitoring time points of the v-th vessel
e_v	The start point of $\widehat{\Gamma_v}$
l_v	The end point of $\widehat{\Gamma_v}$
w_i	The monitoring weight of i-th vessel
$w_{i,j}$	The monitoring weight between node i and j
s_{v_i}	The sailing speed of i-th vessel
s_d	The flight speed of drone
Q	The maximum battery capacity
δ_0	The replacement time of battery
δ	The time of pollutant detection
(u_i, t_i)	The element and corresponding time point of node i
N	The set of discretized time points of vessels
N_0	The set of discretized time points of drone stations
$e(i,j)$	The arc between node i and j
Ω	The set of all arcs existing in the network
G	The set of sub-arcs in the entire arc network
$y_{i,t}$	The number of drones in i-th drone station at t
$q_{i,t}$	The remaining capability of the drone reaching node i at t
$x_{i,j}$	A binary value that indicates whether $e(i,j)$ exists in G

model into a model that could be solved by ACO. Moreover, the performance of the proposed methods outperforms the classical optimization solving methods in solving speed and solving quality.

2) To our best knowledge, our work is the first task to solving the monitoring scheduling of drones for ECA via the approach of computational intelligence.

3) Compared to methods from classic optimization, the obvious advantage of ACO-DSP is the heuristic and evolutionary mechanism could bypass the complex mathematical solution process. This progress makes it possible to model and solve more complex and practical problems similar to DSP. (It is discussed in Section IV in detail.)

The rest of this paper is organized as follows. The description and corresponding mathematical model of DSP are introduced in Section II. The detail of the proposed ACO-DSP is described in Section III. Experiments in Section IV illustrate the feasibility and effectiveness of ACO-DSP. Section V summarizes the paper and points out future research directions.

II. PROBLEM STATEMENT AND MATHEMATICAL MODEL

Before going into the establishment of the mathematical model of DSP, we first formally present the details of DSP. Relevant notations for the problem are list in Table I.

A. Preparations

Assume the start point of time is 0, and the time window of drone scheduling is $\Gamma = [0, T_{max}]$. There are n vessels are sailing in ECA within the time window. $V = \{v_1, v_2, \ldots, v_n\}$ represents the set of vessels. Vessels $i \in V$ is assumed to sail at the speed of s_{v_i} and its destination is a certain port. Besides, according to the historical emission information, each vessel is given a certain monitoring weight. w_i represents the monitoring weight of vessel i.

There are k drone stations in ECA. K represents the set of drone stations. The coordinate of drone station i is expressed as $S_i = (\alpha_i, \beta_i)$. Assume there are m drones with the detection device of pollutant. Each drone station is equipped with a certain number of drones at the beginning. $Drone = \{D_1, D_2, \ldots, D_i, \ldots, D_k\}$ represents the set of all drones, where D_i is the set of drones in drone station i. $D_i = \{d_1^i, d_2^i, \ldots, d_{r_i}^i\}$, where d_j^i represents the j-th drone in the drone station i. r_i represent the number of drones in the drone station i. The maximum drone storage capacity of all drone stations is the same. It should be noticed that the number of drones in different drone stations can be different at the beginning and during the scheduling process. For all drones, they have the same physical properties, including the flight speed s_d, the maximum battery capacity Q, operation time to replace fully charged battery δ_0, and the pollutant detect time δ. It should be pointed out that the pollutant detect time given here includes the time for the drone to adjust its sniffer according to actual weather conditions (such as wind speed, wind direction, etc.) and the time for pollutant sniffing on the vessel. We assume the battery consumption is proportional to the operation time of the drone. *i.e.*, one unit of battery consumption corresponds to one unit of time. Therefore, the battery capacity of the drone can be understood as the maximum operating time of the drone.

Similar to TOP, drones in DSP could be considered *competitors*. Vessels and drone stations could be considered as *nodes*. The flight path of drones could be considered as the travel path of *competitors* to visit multiple *nodes* from the starting point and return to the endpoint within the given time window. However, the difference between TOP and DSP is that the coordinates of nodes are fixed in TOP but flexible in DSP. This is because vessels are moving in ECA, so the coordinates of *nodes* (The *nodes* here refer to vessels, not drone stations) change over time. The characteristic of time-dependent makes DSP more difficult than TOP. However, the sailing speed and corresponding target port of the vessel can be obtained in advance, the continuous-time window could be discretized to specifically represent the real-time coordinates of each vessel at each discretized time point. Assume the time window is $\widehat{\Gamma} = \{T_1, T_2, \ldots, T_{max}\}$ and $T_1 = 0$. The set of discretized time points of vessels is $N = \{(v,t) | v \in V,\ t \in \widehat{\Gamma_v}\}$. $\widehat{\Gamma_v} = \{e_v, e_v + 1, \ldots, l_v\}$ represents the allowed monitoring time points of the v-th vessel. *i.e.*, drones can only detect the v-th vessel during the period from e_v to l_v. The set of discretized time points of drone stations is $N_0 = \{(k,t) | k \in K, t \in \widehat{\Gamma}\}$. Different from vessels, drone stations are not time-dependent nodes. It should be noted that although the drone station is considered a kind of *node*, the monitoring reward (weight) will not be obtained when the drone visits it. When the remaining capability of the drone cannot support it

to conduct the monitoring task, it must return to the nearest drone station for replacing the battery.

Based on the above assumptions, the directed arc can be constructed to connect drone stations and vessels and connect between vessels. $\Omega = (N \cup N_0, G)$ represents the set of all arcs existing in the network. G represents the set of sub-arcs in the entire arc network, which corresponds to a set of feasible flight paths designed by decision-makers or output by algorithms.

B. Objective and Constraints

For the convenience of description, u_i represents the element of node i (*i. e.,* vessel or drone station) in G, while t_i represents the corresponding time point of the node i. $e(i, j) \in \Omega$ represents the arc between node i and j. $x_{i,j}$ represents whether the arc between node i and j exists in G. *i.e.*, $x_{i,j} = 1$, if $e(i, j) \in G$. Otherwise, $x_{i,j} = 0$. $y_{i,t}$ represents the number of drones in drone station i at the time t. Especially, $y_{i,0} = r_i$. $q_{i,t}$ represents the remaining capability of the drone when it reaches node i at time t. Based on the above three groups of decision variables $\{x_{i,j} | e(i, j) \in G\}$, $\{y_{i,t} | i \in K, t \in \widehat{\Gamma}\}$, $\{q_{i,t} | i \in N \cup N_0, t \in \widehat{\Gamma}\}$, DSP can be formulated as follows.

$$F: \sum_{e(i,j)\in\Omega} w_{i,j} x_{i,j} \tag{1}$$

$$s.t.$$

$$\sum_{j\in N} x_{i,j} - \sum_{j\in N} x_{j,i} = y_{u_i,t_i-1} - y_{u_i,t_i}, \quad \forall i \in N_0 \tag{2}$$

$$\sum_{j\in N\cup N_0} x_{i,j} - \sum_{j\in N\cup N_0} x_{i,j} = 0, \quad \forall i \in N \tag{3}$$

$$q_{i,t_i} - (t_j - t_i) x_{i,j} \geqslant q_{j,t_j}, \quad \forall i \in N, \ \forall j \in N \cup N_0 \tag{4}$$

$$Q - (t_j - t_i) x_{i,j} \geqslant q_{j,t_j}, \quad \forall i \in N_0, \ \forall j \in N \tag{5}$$

$$t_j - t_i \geqslant \delta_0, \quad \forall e(i, j) \in G, \ \forall i \in N_0, \ t_i > r_i, \ \forall j \in N \tag{6}$$

$$t_j - t_i \geqslant \delta, \quad \forall e(i, j) \in G, \ \forall i, \ j \in N \cup N_0, \ \{u_i, u_j\} \not\subset K \tag{7}$$

$$\sum_{i\in N\cup N_0} \sum_{j\in N, u_j = v} x_{i,j} \leqslant 1, \quad \forall v \in V \tag{8}$$

$$\sum_{j\in N} x_{i,j} + \sum_{j\in N} x_{j,i} \leqslant 1, \quad \forall i \in N_0 \tag{9}$$

$$x_{i,j} \in \{0, 1\}, \quad \forall e(i, j) \in G \tag{10}$$

$$y_{i,t} \in \{0, 1, \ldots, m\}, \quad \forall i \in K, \ \forall t \in \widehat{\Gamma} \tag{11}$$

$$q_{i,t} \geqslant 0, \quad \forall i \in N \cup N_0, \ \forall t \in \widehat{\Gamma} \tag{12}$$

The objective function (1) maximizes the weights of all arcs in G, where $w_{i,j} = (w_i + w_j)/2$ represents the monitoring weight between node i and j. Constraint (2) records the changes in the number of drones in each drone station at each time. Constraint (3) maintains a balance between the in-degree and out-degree of each vessel node. Constraints (4) and (5) are the constraints related to the drone battery. Constraint (4) gives the constraint that the capacity of the drone from the vessel node to the vessel node or the drone station node. Constraints (5) gives the constraint that the battery capability change of the drone from the drone station node to the vessel node. Constraints (6) and (7) restrict that the battery replacement time and the emission detection time, respectively. Constraint (8) enforces that each vessel can be only visited once. Constraint (9) is the safety constraint that requires only one drone is allowed to leave or return to the drone station at the same time to avoid collision of drones. In this paper, the safe time interval is set to 1. The value range of each group decision variable is defined in (10)-(12), respectively. Based on the above, DSP is transformed into a mixed-integer programming model.

III. Proposed Method

In this section, the mechanism and algorithm flow of the proposed ACO-DSP are described in detail as follows.

A. Search Space Definition

In the application of the basic ACO, nodes represent the basic elements of the solution. The space composed of all possible nodes is called the search space. A group of ants is required to explore the search space under the guidance of pheromone and heuristic information to obtain feasible solutions. As described in Section II, the difficulty of DSP comes from the time-varying characteristics of nodes. Although the corresponding mathematical model could be established by discretizing the time, the introduction of the time dimension will lead to the expansion of the search space. Such a huge amount of calculation has been highlighted in experiments of [38]. In consideration of the scale effect of calculation, in ACO-DSP, nodes are defined as vessels and drone stations. The construction of the pheromone matrix does not need to consider the time dimension. Therefore, the number of nodes is $(n + k)$ and the search space is $R^{(n+k)(n+k)}$.

Fig. 2 shows an example of one ant searching a flight path for a drone. In this case, n vessels are navigating within the ECA. There are two drone stations, and three drones in each drone station respectively. The ant needs to explore the optimal drone monitoring path in a search space of size $R^{(n+2)(n+2)}$. The orange node represents the vessel node. The blue and yellow nodes represent drone station 1 (DS_1) and drone station 2 (DS_2), respectively. The solid line with gray color represents the arc that exists in the search space, and the solid arrow line with black color represents the flight path of the drone constructed by the ant. The dashed arrow line with black color indicates the flight path that must be executed due to time window limitation or battery capability limitation. t_i beside the dashed arrow represents the time corresponding to when the drone reaches the target node. When the ant searches the path in the solution space, the corresponding time when the drone arrives at each node is recorded, instead of assigning time attributes to each node, which could greatly reduce the scale of the search space. The path (solution) construction will be introduced in detail in Section III. C.

B. Initialization

1) Initialization of Hierarchical Pheromone: The pheromone value in ACO-DSP adopts a hierarchical system. The initial value of pheromone is τ_0, the upper limit of pheromone is τ_{max}, the lower limit of pheromone is τ_{min}. The incentive

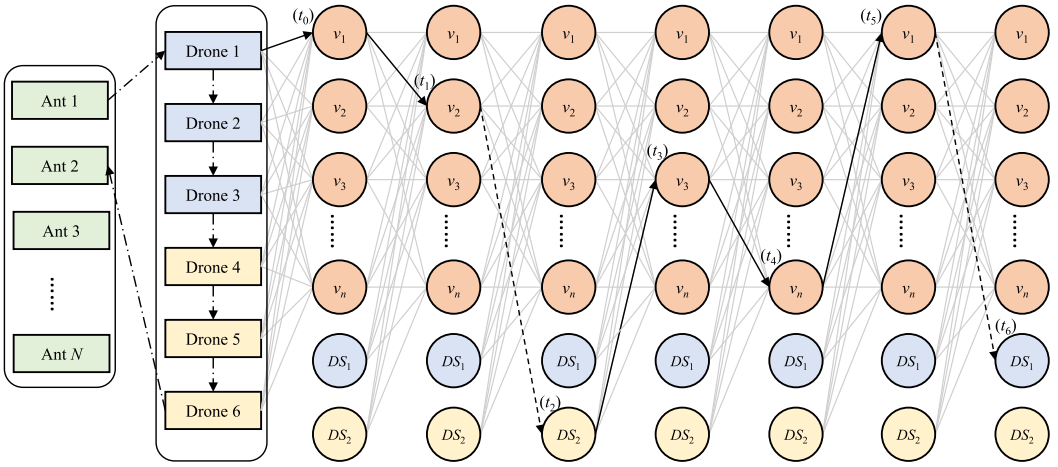

Fig. 2. An example of solution construction in the search space.

of pheromone is set to τ_{up}, and the inhibition of pheromone is to as τ_{down}. Different from using continuous value to increase or decrease the pheromone in typical ACO applications, the increase or decrease values of the pheromone in ACO-DSP adopts discrete values (i.e., τ_{up} and τ_{down}). In the iterative process of the algorithm, if pheromone needs to be increased and it does not reach τ_{max}, increase τ_{up} based on the original pheromone. On the contrary, if pheromone needs to be decreased and it does not reach τ_{min}, increase τ_{down} based on the original pheromone. τ_{up} and τ_{down} are two parameters that control the rise and fall of the pheromone level.

2) Initialization of Static Heuristic Information: Static heuristic information is defined as the weight between two nodes. The elements in the static heuristic information matrix η_s are obtained as follows.

$$\eta_s(i,j) = \begin{cases} w_{i,j}, & i,j \in N \\ w_i/2, & i \in N, j \in N_0 \\ w_j/2, & i \in N_0, j \in N \end{cases} \tag{13}$$

3) Initialization of Dynamic Heuristic Information: Dynamic heuristic information is defined as the reciprocal of Euclidean distance between two time-nodes. The elements in the dynamic heuristic information matrix η_d are obtained as follows.

$$\eta_d(i,j) = \frac{1}{\sqrt{(\varphi_{i,t_i} - \varphi_{j,t_j})^2}}, \quad i,j \in N \cup N_0,\ t_i \in \widehat{\Gamma_i},\ t_j \in \widehat{\Gamma_j} \tag{14}$$

where φ_{i,t_i} and φ_{j,t_j} represent the coordinate of node i at time t_i and the coordinate of node j at time t_j, respectively.

C. Solution Construction

In DSP, A feasible solution means a set of flight paths complied with the constraints of the time window and battery capability. Therefore, each ant needs to complete a set of path planning. In ACO-DSP, each ant acts as each drone in each drone station in sequence, and continuously maintains a vessel-accessible taboo list (termed $Tabu$) during this process. Once an ant visits a certain vessel node in the process of executing the path planning, the node is deleted from $Tabu$ to prevent repeated visits again. Inspired by the rules for constructing feasible solutions in [50], the way of sequential construction is introduced ACO-DSP. In detail, an ant iteratively chooses a drone and selects the next node to be visited from the $Candidate\_list$, until the termination condition is met.

1) Construction of Candidate_list: Since the capacity limit of the drone, before it chooses the next node to visit, it is necessary to determine which of the vessel nodes that have not been visited are feasible nodes or infeasible nodes. We use an example to explain the feasible node and the infeasible node. Let the current time be $t \in \widehat{\Gamma}$, $t \leqslant T_{max}$. The drone is at the node $i \notin Tabu$, and $t_i = t$, $u_i \in V$ or K. The target vessel node is $j \in Tabu$ and $u_j \in V$. To judge whether node j is a feasible node, it is necessary to calculate the earliest arrival time (EAT) between the drone and the node. EAT means the earliest start time when the drone could perform emission detection on the vessel. The calculation process is shown in Alg. 1. In Alg. 1, the sub-function $Get\_nearest\_station(t, node)$ can return the coordinate $S_{nearest}$ of the drone station closest to $\varphi_{j,time+\delta}$. The conditions in rows 2 and 8 of Alg. 1 require that only when the drone can start emission detection on node j within $\widehat{\Gamma_j}$, and its remaining capability can support its return to the nearest drone station after the drones follow the vessel to complete the monitoring task, then node j is considered to be a feasible node. This feasible node and corresponding EAT will be saved into the $Candidate\_list$.

2) Probability Selection of the Next Node: After the determination of the $Candidate\_list$, the next node of the ant will be selected through transition probability and wheel selection. The transition probability from node i to node j is calculated as follows.

$$p(i,j) = \begin{cases} \dfrac{g(i,j)^{\alpha}\eta(i,j)^{\beta}}{\sum_{h \in Candidate\_list} \tau(i,h)^{\alpha}\eta(i,h)^{\beta}}, & j \in Candidate\_list \\ 0, & otherwise. \end{cases} \tag{15}$$

Algorithm 1 Calculate EAT Between Node i and j

Input: t: current time; $q_{i,t}$: current remaining capacity of drone; $\widehat{\Gamma_j}$: sailing time window of vessel j; $\varphi_{i,t}$: coordinate of node i at time t; $\varphi_{j,time}$: coordinate of vessel j, $time \in \widehat{\Gamma_j}$;

Output: Vessel j a feasible node or infeasible node ($True$ or $False$); EAT between node i and j.

for $time = (t+1)$; $time \leqslant max(\widehat{\Gamma_j})$; $time++$ **do**
  if $(distance(\varphi_{i,t}, \varphi_{j,time})/s_d > (time - t))$ **then**
    continue;
  else
    $tmp = q_{i,t} - distance(\varphi_{i,t}, \varphi_{j,time})/s_d - \delta$;
    $S_{nearest} = Get\_nearest\_station(time, j)$;
  end if
  if $(tmp - distance(\varphi_{j,time+\delta}, S_{nearest})/s_d < 0)$ **then**
    continue;
  else
    return $True$, $time$;
  end if
end for
return $False$, 0;

where α and β are the relative weight parameters of pheromone and heuristic information, respectively. $\tau(i, j)$ represents the pheromone level between node i and node j. $\eta(i, j)$ is the heuristic information between node i and node j. It can be calculated as follows.

$$\eta(i, j) = \eta_s(i, j)\eta_d(i, j) \tag{16}$$

And, $g(i, j)$ is the influence factor. It can be calculated as follows.

$$g(i, j) = \frac{\tau(i, j) - 1}{\tau_{ma}} \tag{17}$$

where τ_{ma} is the linear conversion factor of pheromone.

3) Restriction on Returning to the Drone Station: Whenever the ant completes a node selection, it needs to judge whether the remaining capacity of the current drone is enough to support the continued execution of the monitoring task. If not, the drone must return to the drone station closest to its current location and replace the battery.

4) Termination Conditions of the Path Construction: Each ant is limited by the maximum time window when planning the path for each drone. If the maximum time is exceeded, the ant will stop this path construction.

Fig. 1 illustrates the complete solution construction process. In this case, the number of vessels sailing in ECA is 15, and there are 4 drones with the same physical properties are placed in 3 drone stations. The black solid arrow lines represent the sequence of the drone's flight path construction. The ant builds a feasible path for $Drone_1$ in $Station_1$ firstly (the sequence of nodes visited by $Drone_1$ is v_1–v_2–$Station_1$–v_3–v_4–$Station_1$). Then, the ant builds a feasible path for $Drone_2$ in $Station_1$ (the sequence of nodes visited by $Drone_2$ is v_5–$Station_1$–v_6–v_7–$Station_2$). This path implies that at the end of all the monitoring tasks, the drone must return to the nearest drone station to its current location, rather than the drone station from which it starts from. Next, the ant builds a feasible path for $Drone_3$ in $Station_2$ (the sequence of nodes visited by $Drone_3$ is v_8–v_9–$Station_3$–v_{10}–$Station_2$). This path implies that the drone can go to a drone station different from its departure station to replace the battery. Finally, the ant builds a feasible path for $Drone_4$ in $Station_3$ (the sequence of nodes visited by $Drone_4$ is v_{11}–$Station_3$). The rest of the vessels (v_{11} to v_{15}) cannot monitor by drones due to the time window limit or battery capacity restriction.

Algorithm 2 Solution Construction

Input: η_s: static heuristic information matrix; η_d: dynamic heuristic information matrix;

Output: $Path$: Solution of path construction.

$Tabu = \{1, \ldots, n\}$;
$Path = \varnothing$;
for $(Station = 1;\ Station \leqslant k;\ Station++)$ **do**
  $DroneIndex = 0$;
  for $(Drone = 1;\ Drone \leqslant r_{Station};\ Drone++)$ **do**
    $Path = Path \cup Station$;
    $Time\_accumulator = DroneIndex$;
    $q_{i,Time\_accumulator} = Q$;
    Update $Candidate\_list$;
  end for
  while $(Candidate\_list \neq \varnothing)$ **do**
    Select next vessel node i and output the corresponding EAT_i;
    $Path = Path \cup i$;
    Update $q_{i,EAT_i+\delta} = q_{i,Time\_accumulator} - (EAT - Time\_accumulator + \delta)$;
    Update $Time\_accumulator = EAT_i + \delta$;
    Remove vessel i from $Tabu$;
    Update $Candidate\_list$;
    if $(Candidate\_list == \varnothing)$ **then**
      Get the nearest station j and the flight time required to reach this drone station FT;
      $Path = Path \cup j$;
      Update $Time\_accumulator += (FT + \delta_0)$;
      Reload $q_{j,Time\_accumulator} = Q$;
      Update $Candidate\_list$;
    end if
    if $(Time\_accumulator \geqslant T_{max})$ **then**
      break;
    end if
    $DroneIndex += 1$;
  end while
end for

D. Evaluation

After the ant completes the solution construction, the quality of the solution can be evaluated by the weighted sum accumulated by the ant. The greater the weight obtained, the higher the quality of the solution, and vice versa.

E. Pheromone Management

1) Partition Management: Once an ant completes the path construction, a set containing m paths of drones is output. If the pheromone matrix is updated directly in the order of the nodes, it means that the pheromone matrix will record mixed paths of all drones. It is foreseeable that when m is large, it is very difficult to maintain the effective guidance information contained in the pheromone matrix, and multiple different drone flight paths will confuse the effective guidance information contained therein. Based on the above considerations, it is necessary to set up an independent pheromone matrix for each drone. The pheromone matrix of each drone does not interfere with each other, whether in the path construction process or the pheromone update process.

2) Global Pheromone Update: After all the ants complete the path construction and the quality of the solution is evaluated, each pheromone matrix of the drone will be upgraded and downgraded according to the path of the global optimal solution. Taking the pheromone matrix of i-th drone as an example, the pheromone matrix update rule is as follows.

$$\tau_i(h,j) = \begin{cases} min(\tau_{max}, \tau_i(h,j) + \tau_{up}), & e(h,j) \in G_{gb}, u_j \notin K \\ max(\tau_{min}, \tau_i(h,j) - \tau_{down}), & otherwise. \end{cases} \tag{18}$$

where G_{gb} represents the global optimal solution, $e(h, j)$ represents one part path of i-th drone in G_{gb}, and $\tau_i(h, j)$ represents the pheromone between node h and node j of i-th drone. $u_j \notin K$ indicates that the pheromone on the arc from the vessel to the drone station will not be updated. It is because only when $Candidate\_list$ is empty, the drone will return to the drone station.

IV. Numerical Experiments

Based on the proposed method in Section III, DSP, as a variant problem of TOP, could be solved by the heuristic evolutionary algorithms. To illustrate the feasibility of ACO for solving DSP, numerical experiments are designed to test. Also, the comparative experiments are conducted to verify the performance of ACO-DSP. All algorithms are coded in Python 3.8, and experiments are performed on an Intel i9-10900K(3.7GHz) Desktop PC with 32 GB RAM.

A. Experiment Settings

The experiment area (*i.e.*, ECA) is a rectangular area with $[-85, 85]$ nautical miles wide and [0, 20] nautical miles long. The number set of vessels sailing in ECA is $v \in \{20, 30, \ldots, 180\}$. The initial coordinate of each vessel is randomly generated within the area. The fixed coordinate of three ports is (0, 0), (20, 0) and (40, 0). The target port of each vessel is randomly selected in three ports. The allowed monitoring time window of each vessel is defined as the period between the beginning time and the time of arriving at the corresponding target port. The monitoring weight and sailing speed of each vessel are randomly generated in [5,15] and [5,10] knots respectively.

TABLE II
The Parameters Setting of ACO

Parameter	Value	Description
α	1	The intensity of pheromone
β	4	The intensity of heuristic information
ρ	0.01	The decay factor of global pheromone updating
I	2500	The maximum number of iterations
M	50	The number of ants in each iteration

Besides, to better simulate the distribution of vessels, we also designed vessel data based on the vessel channel for experiments. The experimental result of channel-based data is consistent with that of randomly generated data. To avoid the paper being too long, we put this part of the experiment on an open-source platform for reference. More information is available in the GitHub link: https://github.com/polysun/Monitoring-scheduling. It should be noted that the following analysis is based on the experimental results of randomly generated data.

There are two drone stations located at $S_1 = (0, 0)$ and $S_2 = (30, 0)$ respectively. The number of drones in each drone station is 5 (*i.e.*, $|D_i| = 5$, $i \in [1, k]$). The physical properties of each drone including flight speed s_d, maximum battery capacity Q, battery replacement time δ_0, and pollutant detect time δ are set to 30 (knots), 120, 5, and 0 respectively. The upper bound of the time window $\widehat{\Gamma}$ (*i.e.*, T_{max}) is set to 300 minutes. Based on the above settings, we generated 32 datasets with different problem scales for the implementation of experiments.

B. Experiment I: The Feasibility of ACO

In Experiment I, the Lagrangian relaxation-based method (termed Lagrangian-CPLEX) proposed by [38], which is solved by CPLEX solver [51], is used as the compared method to verify the feasibility of ACO for solving DSP. Considering the Lagrangian relaxation-based method has been able to obtain better results than the direct solution from the CPLEX solver [29], this paper will not take the direct solution from the CPLEX solver as the compared method. We set that all algorithms stop running when the maximum number of iterations reached or the predefined upper limit of running time (7200 seconds) exceeded. Notice that the Lagrangian relaxation-based method is iteratively running in rounds and the running time of each round is checked at the end of each round, so the running time may exceed the upper bound (7200 seconds). Therefore the final check time and the corresponding result are recorded for comparison.

For the ACO-based method, the parameters of ACO are set as shown in Table II.

In each experiment, ACO is repeated running 20 times, and the average result of 20 results is recorded. It should be notice that the Lagrangian-CPLEX runs only once since the optimization process of the Lagrangian-CPLEX is a deterministic process. The experiment results are shown in Table III.

It could be found that when the problem scale is small, *i.e.,* from v20k1 to v60k2, in general, the performance of Lagrangian-CPLEX is better than that of ACO. The performance of ACO on v20k2, v40k1, and v40k2 is the

TABLE III
THE RESULTS OF EXPERIMENT I

Dataset	Lagrangian-CPLEX		ACO	
	Mean	Time	Mean	Time
v20k1	**111.00**	6082.66	100.80	66.37
v20k2	**150.00**	4458.35	**150.00**	106.52
v30k1	**216.00**	7226.34	204.30	131.45
v30k2	**291.00**	7402.41	285.70	215.62
v40k1	**217.00**	5655.62	**217.00**	169.02
v40k2	**292.00**	7666.14	**292.00**	278.29
v50k1	**288.00**	7480.55	262.95	195.31
v50k2	**427.00**	7946.40	426.60	382.20
v60k1	**384.00**	7644.29	378.30	349.83
v60k2	388.00	7534.23	**500.10**	596.14
v70k1	364.00	8019.44	**373.85**	401.98
v70k2	426.00	7946.14	**503.00**	627.14
v80k1	302.00	7430.06	**482.20**	533.41
v80k2	497.00	8098.08	**652.85**	954.43
v90k1	350.00	8235.26	**555.65**	725.72
v90k2	507.00	8274.08	**746.15**	1119.12
v100k1	275.00	9152.90	**606.85**	738.45
v100k2	518.00	8769.26	**844.10**	1134.55
v110k1	327.00	10670.23	**681.90**	1294.52
v110k2	634.00	14013.91	**832.15**	1844.71
v120k1	325.00	7552.74	**668.05**	970.12
v120k2	513.00	8151.71	**982.45**	1775.67
v130k1	291.00	10830.89	**795.95**	1288.29
v130k2	659.00	10477.31	**1064.35**	2166.54
v140k1	319.00	8076.47	**753.35**	1409.60
v140k2	593.00	15197.89	**1106.15**	2454.98
v150k1	287.00	8339.47	**797.80**	1499.29
v150k2	738.00	16805.88	**1140.85**	2720.27
v160k1	394.00	10433.91	**950.35**	1722.35
v160k2	802.00	19259.67	**1289.50**	3006.39
v170k1	474.00	14895.45	**1078.50**	2767.05
v170k2	867.00	26047.03	**1433.75**	4157.85
v180k1	386.00	15878.95	**1015.70**	2618.78
v180k2	761.00	28128.80	**1396.55**	4004.45

Note: The naming of each dataset implies the problem scale. For example, v20k1 is mean that 20 vessels are sailing in ECA and only one drone station (Station 1) is in use.

same as that of the comparison methods, and its performance is better than Lagrangian-CPLEX on v60k2. Although the performance of ACO is slightly worse than that of the comparison methods in the remaining instances of the small-scale problem, the solving speed of ACO outperforms Lagrangian-CPLEX. It illustrates that ACO could obtain an acceptable near-optimal solution in a short time when solving a small-scale DSP.

When the problem scale becomes large, *i.e.,* from v70k1 to v180k2, the results of ACO are superior to that of the comparison methods in terms of solution quality and solving speed. Especially in those instances where the number of vessels is greater than or equal to 120, the sum of the monitoring weights obtained by ACO is 109% higher than that of Lagrangian-CPLEX on average. Besides, the solving speed of ACO is 6.18 times faster than that of Lagrangian-CPLEX on average. It shows that ACO has significant potential advantages when solving the large-scale DSP.

C. Experiment II: The Superiority of ACO-DSP

In Experiment II, a hierarchical-based pheromone update strategy and partition-based pheromone management mechanism are used to optimize the typical ACO. The ACO with the hierarchical-based pheromone update strategy is termed ACO-Level. The ACO with the hierarchical-based pheromone update strategy and partition-based pheromone management mechanism is termed ACO-DSP.

TABLE IV
THE PARAMETERS SETTING OF THE HIERARCHICAL-BASED PHEROMONE UPDATE STRATEGY

Parameter	Value	Description
τ_0	1	The initial level of pheromone
τ_{max}	100	The upper limit level of pheromone
τ_{min}	1	The lower limit level of pheromone
τ_{up}	3	The increase level of pheromone
τ_{down}	1	The decrease level of pheromone
τ_{ma}	5	The linear conversion factor of pheromone

Since the hierarchical-based pheromone update strategy is added into ACO_Level and ACO_DSP, several additional hyper-parameters need to be set, as shown in Table IV. The settings of other parameters are consistent with Experiment I. All algorithms run 20 times repeatedly, and the termination conditions for algorithms are the maximum number of iterations reached or the predefined upper limit of running time (7200 seconds) exceeded. The results are shown in Table V.

It can be found that ACO-Level outperforms ACO in most experiments. But the average time consumption is only 4% higher than ACO. It shows that the hierarchical-based pheromone update strategy is effective in terms of improving solution quality. When partition-based pheromone management mechanism is introduced based on ACO-Level, the solution quality could be further improved. Expect the instances of v20k1 and v90k1, the ACO-DSP outperforms both ACO-Level and ACO. The average time consumption of ACO-DSP is 24% higher than ACO and 19% higher than ACO-Level due to the implementation of the partition-based pheromone management mechanism.

Although there is a certain increase in time, from the point of view of the actual scenario application of the algorithm, the scheduling time of ACO-DSP could be acceptable. This is because the path planning tasks of drones are often carried out one day in advance, and there is a high tolerance for the time consumption of the algorithm. Therefore, in this scenario, with the introduction of the partition-based pheromone management mechanism, a better solution can be obtained within an acceptable time consumption.

D. Practical Implications

The flexibility of drones makes drone monitoring become a new supervision technology of ECA, which overcomes the shortcomings of traditional vessel emission monitoring methods that require a lot of manpower. By drone monitoring technology, supervisors could easily conduct real-time pollutant monitoring in the control room. However, in practical applications, especially for port management scenarios with large cargo throughput, the number of drones that could be called is much smaller than the number of vessels that need to be monitored. If drone monitoring scheduling is arranged manually, it will be difficult to ensure that the drones can be used efficiently. Therefore, when the drone is used as a core monitoring tool for emission monitoring, DSP is an important issue that needs to be solved.

TABLE V
THE RESULTS OF EXPERIMENT II

Dataset	ACO				ACO_Level				ACO_DSP			
	Mean	Max	Min	Time	Mean	Max	Min	Time	Mean	Max	Min	Time
v20k1	**100.80**	102.00	98.00	66.37	**100.80**	104.00	98.00	71.13	100.60	104.00	98.00	78.13
v20k2	**150.00**	150.00	150.00	106.52	**150.00**	150.00	150.00	104.67	**150.00**	150.00	150.00	128.62
v30k1	204.30	208.00	197.00	131.45	206.10	208.00	197.00	140.58	**206.70**	208.00	197.00	146.99
v30k2	285.70	291.00	280.00	215.62	282.60	291.00	280.00	216.19	**286.15**	291.00	280.00	244.62
v40k1	**217.00**	217.00	217.00	169.02	**217.00**	217.00	217.00	181.49	**217.00**	217.00	217.00	207.51
v40k2	**292.00**	292.00	292.00	278.29	**292.00**	292.00	292.00	287.90	**292.00**	292.00	292.00	334.54
v50k1	262.95	269.00	254.00	195.31	264.05	265.00	264.00	211.26	**264.25**	265.00	264.00	246.84
v50k2	426.60	427.00	419.00	382.20	426.60	427.00	419.00	398.96	**427.00**	427.00	427.00	469.37
v60k1	378.30	384.00	369.00	349.83	**380.40**	384.00	378.00	386.83	**380.40**	384.00	378.00	442.50
v60k2	500.10	505.00	495.00	596.14	501.75	505.00	499.00	625.41	**504.50**	510.00	499.00	698.07
v70k1	373.85	387.00	358.00	401.98	386.10	400.00	374.00	408.44	**388.10**	400.00	379.00	503.58
v70k2	503.00	516.00	494.00	627.14	513.60	520.00	508.00	682.03	**515.40**	520.00	510.00	806.13
v80k1	482.20	493.00	478.00	533.41	493.30	511.00	479.00	533.01	**498.00**	516.00	478.00	671.71
v80k2	652.85	655.00	647.00	954.43	**655.00**	655.00	655.00	938.14	**655.00**	655.00	655.00	1155.48
v90k1	555.65	566.00	552.00	725.72	**558.60**	566.00	552.00	774.36	556.50	566.00	552.00	878.30
v90k2	746.15	753.00	742.00	1119.12	751.90	753.00	742.00	1202.00	**753.00**	753.00	753.00	1393.40
v100k1	606.85	625.00	593.00	738.45	619.80	639.00	606.00	779.57	**621.30**	635.00	615.00	910.02
v100k2	844.10	866.00	833.00	1134.55	856.65	866.00	846.00	1208.96	**861.40**	879.00	846.00	1473.63
v110k1	681.90	689.00	675.00	1294.52	686.25	695.00	675.00	1339.17	**690.50**	695.00	682.00	1536.85
v110k2	832.15	838.00	825.00	1844.71	836.25	838.00	831.00	1924.31	**836.60**	838.00	831.00	2302.95
v120k1	668.05	680.00	660.00	970.12	671.40	680.00	669.00	1016.35	**675.10**	680.00	669.00	1227.67
v120k2	982.45	988.00	977.00	1775.67	983.35	988.00	982.00	1894.22	**984.00**	988.00	983.00	2289.02
v130k1	795.95	810.00	783.00	1288.29	812.60	831.00	794.00	1337.36	**817.00**	831.00	795.00	1612.50
v130k2	1064.35	1077.00	1047.00	2166.54	1068.80	1083.00	1058.00	2279.73	**1076.70**	1089.00	1058.00	2692.69
v140k1	753.35	777.00	728.00	1409.60	768.20	777.00	752.00	1482.30	**768.30**	777.00	745.00	1846.82
v140k2	1106.15	1111.00	1097.00	2454.98	1110.20	1116.00	1104.00	2457.26	**1112.70**	1122.00	1109.00	3075.90
v150k1	797.80	813.00	784.00	1499.29	809.75	813.00	802.00	1612.74	**811.45**	813.00	797.00	1949.02
v150k2	1140.85	1146.00	1130.00	2720.27	1144.45	1146.00	1135.00	2871.42	**1145.35**	1149.00	1135.00	3488.51
v160k1	950.35	963.00	929.00	1722.35	961.15	972.00	938.00	1850.69	**965.55**	985.00	946.00	2247.43
v160k2	1289.50	1300.00	1276.00	3006.39	1292.55	1299.00	1283.00	3106.08	**1302.40**	1312.00	1293.00	3892.56
v170k1	1078.50	1095.00	1058.00	2767.05	1092.45	1102.00	1077.00	2796.58	**1099.75**	1110.00	1084.00	3451.85
v170k2	1433.75	1446.00	1423.00	4157.85	1436.50	1446.00	1423.00	4357.70	**1439.50**	1452.00	1430.00	5158.29
v180k1	1015.70	1038.00	983.00	2618.78	1034.55	1056.00	1012.00	2459.08	**1040.35**	1061.00	1016.00	3190.51
v180k2	1396.55	1404.00	1393.00	4004.45	1397.00	1404.00	1393.00	4161.09	**1400.90**	1404.00	1393.00	5142.99

Note: The naming of each dataset implies the problem scale. For example, v20k1 is mean that 20 vessels are sailing in ECA and only one drone station (Station 1) is in use.

This fact emphasizes the importance of optimizing the drone monitoring scheduling to cooperate with drone groups to increase the coverage of the monitoring of vessels for the application of drone monitoring technology. For this purpose, the ACO-based method is proposed in our work, which is a computational intelligence method for solving DSP. Compared with the classic optimization method, the ACO-based method outputs a higher quality monitoring scheduling plan within an acceptable time consumption. Experiments illustrate, in cases where the scale of the problem is larger, the superiority of our method is prominent. It shows the potential application advantage of our method in practical scenarios.

E. Theoretical Implications

For classical optimization, non-linear programming is a kind of hard-solving problem [52]. Unfortunately, in several actual scenarios, DSP should contain more complex and non-linear constraints, which may transform the DSP from simple mixed-integer programming to a complex non-linear programming problem. It extremely limits the ability of classical optimization tools such as CPLEX [51], Gurobi [53] to solve non-linear programming problems. The heuristic mechanism in the ACO-based method avoids the complicated and time-consuming mathematical solution process encountered by classical optimization methods in solving DSP [54], [55]. Besides, as a kind of computational intelligence method, the ACO-based method is extensible. It has good support for the expansion and change of the problem model. Therefore, when the problem model becomes complicated or has a non-linear component, the ACO-based method shows its potential advantages in solving complex actual scenarios.

V. CONCLUSION

In this paper, the vessel pollutant monitoring task in ECA is defined as the DSP. Then, to effectively solve the DSP, we redefine the node definition in the time-expanded network and adjust the DSP model to a form that could be solved by ACO. Next, to improve the solving ability of ACO, the ACO-DSP with the hierarchical-based pheromone update strategy and partition-based pheromone management mechanism is proposed. Numerical experiments are designed to verify the performance of the proposed method, including the feasibility of the ACO-based method and the effectiveness of the two proposed mechanisms. Furthermore, through the proposed ACO-DSP, the monitoring scheduling of drones for emission control areas studied in this paper is systematically analyzed and solved. Compared with the classical optimization algorithm, the proposed problem model and ACO-based method could solve the DSP with complex constraints. The proposed method has obvious advantages in solution quality

and solution speed, especially in large-scale cases. In the future, it is necessary to further study the actual constraints of port monitoring requirements and tap the application potential of ACO-DSP in complex actual scenarios.

REFERENCES

[1] K. Cullinane and R. Bergqvist, "Emission control areas and their impact on maritime transport," *Transp. Res. D, Transp. Environ.*, vol. 28, pp. 1–5, May 2014.

[2] P. Balcombe *et al.*, "How to decarbonise international shipping: Options for fuels, technologies and policies," *Energy Convers. Manage.*, vol. 182, pp. 72–88, Feb. 2019.

[3] A. Ng and J. Liu, *Port-Focal Logistics and Global Supply Chains*. London, U.K.: Palgrave Macmillan, 2014.

[4] Y.-B. Xiao, X. Fu, A. K. Y. Ng, and A. Zhang, "Port investments on coastal and Marine disasters prevention: Economic modeling and implications," *Transp. Res. B, Methodol.*, vol. 78, pp. 202–221, Aug. 2015.

[5] X. Zhang *et al.*, "Changes in the SO_2 level and $PM_{2.5}$ components in Shanghai driven by implementing the ship emission control policy," *Environ. Sci. Technol.*, vol. 53, no. 19, pp. 11580–11587, Oct. 2019, doi: 10.1021/acs.est.9b03315.

[6] S. Zheng, Y.-E. Ge, X. Fu, Y. Nie, and C. Xie, "Modeling collusion-proof port emission regulation of cargo-handling activities under incomplete information," *Transp. Res. B, Methodol.*, vol. 104, pp. 543–567, Oct. 2017.

[7] D. Wu *et al.*, "Pollutants emitted from typical Chinese vessels: Potential contributions to ozone and secondary organic aerosols," *J. Cleaner Prod.*, vol. 238, Nov. 2019, Art. no. 117862.

[8] T. Uyanık, Ç. Karatuğ, and Y. Arslanoğlu, "Machine learning approach to ship fuel consumption: A case of container vessel," *Transp. Res. D, Transp. Environ.*, vol. 84, Jul. 2020, Art. no. 102389.

[9] T. Kirschstein and F. Meisel, "GHG-emission models for assessing the eco-friendliness of road and rail freight transports," *Transp. Res. B, Methodol.*, vol. 73, pp. 13–33, Mar. 2015.

[10] J. J. Corbett, J. J. Winebrake, E. H. Green, P. Kasibhatla, V. Eyring, and A. Lauer, "Mortality from ship emissions: A global assessment," *Environ. Sci. Technol.*, vol. 41, no. 24, pp. 8512–8518, Dec. 2007, doi: 10.1021/es071686z.

[11] Q. Zhang, Z. Zheng, Z. Wan, and S. Zheng, "Does emission control area policy reduce sulfur dioxides concentration in shanghai?" *Transp. Res. D, Transp. Environ.*, vol. 81, Apr. 2020, Art. no. 102289.

[12] Z. Tan, H. Liu, S. Shao, J. Liu, and J. Chen, "Efficiency of Chinese ECA policy on the coastal emission with evasion behavior of ships," *Ocean Coastal Manage.*, vol. 208, Jul. 2021, Art. no. 105635.

[13] Y.-T. Chang, H. Park, S. Lee, and E. Kim, "Have emission control areas (ECAs) harmed port efficiency in europe?" *Transp. Res. D, Transp. Environ.*, vol. 58, pp. 39–53, Jan. 2018.

[14] L. Li, S. Gao, W. Yang, and X. Xiong, "Ship's response strategy to emission control areas: From the perspective of sailing pattern optimization and evasion strategy selection," *Transp. Res. E, Logistics Transp. Rev.*, vol. 133, Jan. 2020, Art. no. 101835.

[15] L. Zhen, M. Li, Z. Hu, W. Lv, and X. Zhao, "The effects of emission control area regulations on cruise shipping," *Transp. Res. D, Transp. Environ.*, vol. 62, pp. 47–63, Jul. 2018.

[16] Y.-T. Chang, Y. Roh, and H. Park, "Assessing noxious gases of vessel operations in a potential emission control area," *Transp. Res. D, Transp. Environ.*, vol. 28, pp. 91–97, May 2014.

[17] *Prevention of Air Pollution From Ships*. Accessed: Nov. 11, 2020. [Online]. Available: https://www.imo.org/en/OurWork/Environment/Pages/Air-Pollution.aspx

[18] (2018). *Implementation Plan for Ship Air Pollutant Emission Control Area in China*. [Online]. Available: https://xxgk.mot.gov.cn/2020/jigou/haishi/202008/t20200828_3457437.html

[19] Q. He, X. Zhang, and K. Nip, "Speed optimization over a path with heterogeneous arc costs," *Transp. Res. B, Methodol.*, vol. 104, pp. 198–214, Oct. 2017.

[20] Q. Meng, Y. Du, and Y. Wang, "Shipping log data based container ship fuel efficiency modeling," *Transp. Res. B, Methodol.*, vol. 83, pp. 207–229, Jan. 2016.

[21] Z. F. Pan, L. An, and C. Y. Wen, "Recent advances in fuel cells based propulsion systems for unmanned aerial vehicles," *Appl. Energy*, vol. 240, pp. 473–485, Apr. 2019.

[22] Z. Ji *et al.*, "Determination of the safe operation zone for a turbine-less and solid oxide fuel cell hybrid electric jet engine on unmanned aerial vehicles," *Energy*, vol. 202, Jul. 2020, Art. no. 117532.

[23] G. Bakioglu and A. O. Atahan, "AHP integrated TOPSIS and VIKOR methods with pythagorean fuzzy sets to prioritize risks in self-driving vehicles," *Appl. Soft Comput.*, vol. 99, Feb. 2021, Art. no. 106948.

[24] P. Liu, Y. Ma, and Y. Zuo, "Self-driving vehicles: Are people willing to trade risks for environmental benefits?" *Transp. Res. A, Policy Pract.*, vol. 125, pp. 139–149, Jul. 2019.

[25] F. Zhou, S. Pan, W. Chen, X. Ni, and B. An, "Monitoring of compliance with fuel sulfur content regulations through unmanned aerial vehicle (UAV) measurements of ship emissions," *Atmos. Meas. Techn.*, vol. 12, no. 11, pp. 6113–6124, Nov. 2019.

[26] T. F. Villa, R. A. Brown, E. R. Jayaratne, L. F. Gonzalez, L. Morawska, and Z. D. Ristovski, "Characterization of the particle emission from a ship operating at sea using an unmanned aerial vehicle," *Atmos. Meas. Techn.*, vol. 12, no. 1, pp. 691–702, Jan. 2019.

[27] H. Yuan, C. Xiao, Y. Wang, X. Peng, Y. Wen, and Q. Li, "Maritime vessel emission monitoring by an UAV gas sensor system," *Ocean Eng.*, vol. 218, no. 1, Dec. 2020, Art. no. 108206.

[28] W. Xu *et al.*, "Approximation algorithms for the generalized team orienteering problem and its applications," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 176–189, Feb. 2021.

[29] R. El-Hajj, D.-C. Dang, and A. Moukrim, "Solving the team orienteering problem with cutting planes," *Comput. Oper. Res.*, vol. 74, pp. 21–30, Oct. 2016.

[30] X. Xu, J. Li, and M. Zhou, "Delaunay-triangulation-based variable neighborhood search to solve large-scale general colored traveling salesman problems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1583–1593, Mar. 2021.

[31] X. Meng, J. Li, X. Dai, and J. Dou, "Variable neighborhood search for a colored traveling salesman problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1018–1026, Apr. 2018.

[32] X. Meng, J. Li, M. C. Zhou, X. Dai, and J. Dou, "Population-based incremental learning algorithm for a serial colored traveling salesman problem," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 2, pp. 277–288, Feb. 2018.

[33] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," *Eur. J. Oper. Res.*, vol. 209, no. 1, pp. 1–10, 2011.

[34] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *Eur. J. Oper. Res.*, vol. 255, no. 2, pp. 315–332, 2016.

[35] N. Bianchessi, R. Mansini, and M. G. Speranza, "A branch-and-cut algorithm for the team orienteering problem," *Int. Trans. Oper. Res.*, vol. 25, no. 2, pp. 627–635, Mar. 2018.

[36] L. Ke, L. Zhai, J. Li, and F. T. S. Chan, "Pareto mimic algorithm: An approach to the team orienteering problem," *Omega*, vol. 61, pp. 155–166, Jun. 2016.

[37] C. Archetti, M. G. Speranza, Á. Corberán, J. M. Sanchis, and I. Plana, "The team orienteering arc routing problem," *Transp. Sci.*, vol. 48, no. 3, pp. 442–457, Aug. 2014.

[38] J. Xia, K. Wang, and S. Wang, "Drone scheduling to monitor vessels in emission control areas," *Transp. Res. B, Methodol.*, vol. 119, pp. 174–196, Jan. 2019.

[39] Z. Cao, S. Gong, M. Zhou, and K. Liu, "A self-braking symbiotic organisms search algorithm for bi-objective reentrant hybrid flow shop scheduling problem," in *Proc. IEEE 14th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2018, pp. 803–808.

[40] Z. C. Cao, C. R. Lin, M. C. Zhou, and R. Huang, "Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 825–837, Apr. 2019.

[41] A. Che, P. Wu, F. Chu, and M. C. Zhou, "Improved quantum-inspired evolutionary algorithm for large-size lane reservation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 12, pp. 1535–1548, Dec. 2015.

[42] P. Wu, F. Chu, A. Che, and M. Zhou, "Bi-objective scheduling of fire engines for fighting forest fires: New optimization approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1140–1151, Apr. 2018.

[43] W. Deng, J. Xu, H. Zhao, and Y. Song, "A novel gate resource allocation method using improved PSO-based QEA," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–9, Oct. 1, 2020, doi: 10.1109/TITS.2020.3025796.

[44] B. C. Mohan and R. Baskaran, "A survey: Ant colony optimization based recent research and implementation on several engineering domain," *Expert Syst. Appl.*, vol. 39, no. 4, pp. 4618–4627, Mar. 2012.

[45] Y. Feng *et al.*, "Target disassembly sequencing and scheme evaluation for CNC machine tools using improved multiobjective ant colony algorithm and fuzzy integral," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 12, pp. 2438–2451, Dec. 2019.
[46] G. Gao, Y. Mei, Y.-H. Jia, W. N. Browne, and B. Xin, "Adaptive coordination ant colony optimization for multipoint dynamic aggregation," *IEEE Trans. Cybern.*, early access, Jan. 5, 2021, doi: 10.1109/TCYB.2020.3042511.
[47] Z.-H. Sun, D. Liang, Z. Zhuang, L. Chen, and X. Ming, "Multitask processing oriented production layout based on evolutionary programming mechanism," *Appl. Soft Comput.*, vol. 98, Jan. 2021, Art. no. 106896.
[48] X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. L. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 113–128, Feb. 2018.
[49] D. Liang, Z.-H. Zhan, and J. Zhang, "An adaptive ant colony system for public bicycle scheduling problem," in *Proc. Neural Inf. Process.*, L. Cheng, A. C. S. Leung, and S. Ozawa, Eds. Cham, Switzerland: Springer, 2018, pp. 417–429.
[50] L. Ke, C. Archetti, and Z. Feng, "Ants can solve the team orienteering problem," *Comput. Ind. Eng.*, vol. 54, no. 3, pp. 648–665, Apr. 2008.
[51] *IBM CPLEX Optimizer*. Accessed: Mar. 4, 2021. [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer
[52] D. P. Bertsekas, "Nonlinear programming," *J. Oper. Res. Soc.*, vol. 48, no. 3, p. 334, 1997.
[53] *The Fastest Solver: Gurobi*. Accessed: Feb. 10, 2021. [Online]. Available: https://www.gurobi.com/
[54] M. Schlüter, J. A. Egea, and J. R. Banga, "Extended ant colony optimization for non-convex mixed integer nonlinear programming," *Comput. Oper. Res.*, vol. 36, no. 7, pp. 2217–2229, Jul. 2009.
[55] P.-Y. Yin and J.-Y. Wang, "Ant colony optimization for the nonlinear resource allocation problem," *Appl. Math. Comput.*, vol. 174, no. 2, pp. 1438–1453, Mar. 2006.

Zhao-Hui Sun (Member, IEEE) is currently pursuing the Ph.D. degree in industrial intelligent system with the Department of Industrial Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China.

He has authored/coauthored over ten research papers in refereed international journals and conferences. His current research interests include evolutionary optimization, neuroergonomics, brain-inspired intelligence, cognitive computing, complex networks, non-parametric machine learning, operations research, knowledge automation, and their applications in industrial or medical problems. He has been serving as a reviewer for many top-tier international journals and conferences in his research field.

Xiaosong Luo (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, where he is currently pursuing the master's degree in industrial engineering with the Department of Industrial Engineering, School of Mechanical Engineering. His current research interests include evolutionary computation and production plan and scheduling.

Edmond Q. Wu (Member, IEEE) received the Ph.D. degree in control theory and application from Southeast University, Nanjing, China, in 2009. He is an Associate Professor at the Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai Jiao Tong University, Shanghai, China. He is also with the Science and Technology on Avionics Integration Laboratory, China National Aeronautical Radio Electronics Research Institute, Shanghai. His research interests include deep learning, cognitive modeling, and industrial information processing. He is currently an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

Tian-Yu Zuo is currently pursuing the M.S. degree with the School of Automation, Nanjing University of Information Science & Technology, Nanjing, China. His current research interests include evolutionary algorithms, swarm intelligence, and their applications in real-world optimization problems.

Zhi-Ri Tang (Student Member, IEEE) received the bachelor's and master's degrees in microelectronics from Wuhan University, Wuhan, China, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in computer science. His research interests include machine learning and cognitive computing.

Zilong Zhuang received the bachelor's degree from Beihang University in 2016. He is currently pursuing the Ph.D. degree with the Department of Industrial Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include artificial intelligence, computational intelligence, and their applications in industry.