Ant Colony System Based Drone Scheduling For Ship Emission Monitoring

Xiaosong Luo
Department of Industrial Engineering
Shanghai Jiao Tong University
Shanghai, China
luo_x_s@sjtu.edu.cn

Zhao-Hui Sun
Department of Industrial Engineering
Shanghai Jiao Tong University
Shanghai, China
zh.sun@sjtu.edu.cn

Siqi Qiu
Department of Industrial Engineering
Shanghai Jiao Tong University
Shanghai, China
siqiqiu1988@163.com

2021 IEEE Congress on Evolutionary Computation (CEC) | 978-1-7281-8393-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CEC45853.2021.9504944

***Abstract*—Emission control area has been set up in many countries to reduce the environmental impact of vessels' emissions. However, the regulations for controlling emissions are frequently violated due to the cost of high-quality fuel. Drones currently have become an accurate and efficient way to monitor the vessels' emissions, which should be properly scheduled to cover more and higher risk of violations when facing a large number of vessels. In this paper, a scheduling model is proposed to simulate the drone scheduling monitoring problem. Due to the movement of vessels over time, the complexity of the model is too large to be solved by classical optimization methods such as CPLEX. An ant colony system algorithm is proposed to solve the scheduling problem of drones. Our method is proved to be more effective and efficient when facing a large number of vessels and drone stations in numerical experiments.**

***Index Terms*—Drone scheduling, Ship emission, Ant colony system, Optimization**

I. Introduction

With the continuous development of the global economy, there has been an increasing need for international cargo transportation. Considering the transportation cost of cargo, the primary tool of transportation is ocean-going container vessel. According to statistical reports, more than 80% of global transportation is carried out by container vessels [1-2]. However, during the navigation of container vessels, a large number of exhaust gases containing carbon dioxide (CO_2), nitrogen oxides (NO_X), sulfur oxides (SO_X), and inhalable particulate matter (PM 2.5) are emitted [3-4]. Considering that a large number of ports are concentrated around densely populated cities, such as Shanghai port and Shenzhen port in China, how to reduce the impact of vessel emission on the health of residents near ports has become an important issue for all countries. For this serious problem, it has been a popular way to set up a vessel emission control area (ECA) for each port and force vessels to use low emission fuels in the area [5].

This work was supported by the 2019 Shanghai Service Industry Guidance Project supported by Shanghai Municipal Development & Reform Commission (Grant No. 2019-01), and the National Natural Science Foundation of China (Grant No. 71632008) for the funding support to this research *(Corresponding author: Zhao-Hui Sun, Email: zh.sun@sjtu.edu.cn).*

Even though the ECA has been established and the relevant rules have been promulgated [6], there are still a large number of violations against relevant regulations due to low emission fuel increases the operating cost of vessels. According to historical data, more than 10% of the cases have violated the regulations [7-8]. To keep regulations in effect, it is necessary to monitor the exhaust gas emission of vessels, and promptly detect and punish those vessels that violate the regulations. Currently, the most common exhaust gas detection methods include vessel technical file verification, hand-held analyzer on-board inspection, and fixed optical or sniffing detector monitoring. However, the above inspecting methods generally have a deficiency of low efficiency or easily affected by the environment [9-10]. Fortunately, with the development of drone technology, it has become possible to fly over vessels with inspection tools and perform exhaust gas monitoring, which not only improves the accuracy of monitoring but also promotes the efficiency of inspection due to the high-speed movement of drones [11-14]. However, in the face of a large number of vessels that need to be monitored, it is essential to schedule the operation of drones properly so they can cover more and higher risk of emission violations. Based on this problem, Xia et al. constructed a time-node model to model this problem and transform it into a complex linear optimization model [14]. On this basis, they proposed the Lagrangian relaxation-based method to solve this problem, which greatly improves the solution speed compared to the original linear optimization solving.

Ant colony system (ACS) is an improved heuristic global optimization algorithm based on ant colony optimization (ACO) [15-16]. This method is widely used to solve various scheduling problems. For example, Montemanni et al. apply ACS to the problem of dynamic vehicle path planning [17], Sun et al. apply ACS to solve the multi-task oriented complex machine layout problem [18], Liu et al. apply the algorithm to the allocation problem of cloud virtual machines [19]. In this problem, the drones are scheduled to visit vessels just like the TSP problem, but there are mainly three different features making this problem quite unique. First, the vessels are moving at uniform speeds. Second, it is not necessary for drones to go through every vessel due to the time window limit. Third, drones are deployed in drone stations and need to

return to drone stations for battery changing under low battery conditions.

The main contribution of this paper is that we come up with corresponding components to transform Xia's model into an ACS solvable problem [14], and then we propose an ACS algorithm to solve the problem. When facing a large number of vessels, our proposed method can gain a high-quality solution in a relevantly short time. The proposed algorithm ensures a feasible solution can be gained quickly in the real application for drone monitoring.

The rest of this paper is organized as follows. In Section II, a brief description of the time-node model and related mathematical models of the drone monitoring problem is presented. In Section III, the framework of the ACS algorithm is introduced. Section III-A introduces the key components in designing ACS, and Section III-B mainly gives the overall algorithm flow of ACS. In Section IV, the comparative experiments between our method and the Lagrangian relaxation-based method demonstrate the feasibility and outperformance of our proposed method. Section V summarizes and forecasts the work of this paper.

II. Description of drone scheduling model

As shown in Fig.1, in the time interval of $[0, T_{max}]$, there are a number of vessels moving towards the ports (indicated by the red "x"). During the time interval, drones take off from the drone station which is indicated by red rectangular to perform ship emission monitoring. The black circles represent the initial position of vessels, the black asterisks represent the termination position of vessels, and the black dashed lines represent the moving track of vessels in the given time interval. Each vessel has a corresponding monitoring weight w_v, which is generated according to historical data. In the given time interval, drones need to fly over vessels for emission monitoring and return to drone stations in time for recharging to prevent battery draining, so it is necessary to generate a proper schedule so that drones can cover more and higher risk of violations.

The time-node model used in our paper refers to the model proposed by Xia et al. [14]. In this model, the drone scheduling problem is simplified by using discrete time points. Assume the discrete-time points form a time point set $T = 1, ..., T_{max}$, and each ship can be monitored in the time point set T_v, which is the time interval between it enters the monitoring zone and reaches the port. The vessel and drone station under different time conditions are represented by vessel time nodes and station time nodes, which are defined as $N = \{(v,t)|v \in V, t \in T_v\}$ and $N_0 = \{(k,t)|k \in K, t \in T\}$ respectively. The number of drones in the drone station at different times is represented by $\{y_{k,t}|k \in K, t \in T\}$, and the upper limitation of the drones' battery power is Q_0. The battery power of the drone when the drone reaches the node i is defined as $\{q_i|i \in (N \cup N_0)\}$. All drones' path can be converted into the arc which connects the ship/station time nodes. The arc between node i and node j can be represented by $x_{i,j}$. And the drones do not have to fly between drone stations, so there are not arcs between station time nodes. The whole path set can be expressed by $G = \{x_{i,j}|i, j \in (N \cup N_0)\}$.

The mathematical model of the problem is shown as follows.

$$F: maximize \sum_{x_{i,j} \in G} x_{i,j} w_{i,j} \tag{1}$$

$s.t.$

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

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

$$q_i - (t_j - t_i) \geq q_j, if\ x_{i,j} = 1, \forall i \in N, \forall j \in N \cup N_0 \tag{4}$$

$$Q_0 - (t_j - t_i) \geq q_j, if\ x_{i,j} = 1, \forall i \in N_0, \forall j \in N \cup N_0 \tag{5}$$

$$\sum_{i \in N \cup N_0} \sum_{j \in N} x_{i,j} \leq 1, \forall v \in V \tag{6}$$

$$x_{i,j} = \begin{cases} 1 & if\ path\ from\ node\ i\ to\ node\ j\ is\ selected \\ 0 & otherwise \end{cases} \tag{7}$$

$$y_{k,t} \in \{0, 1, ..., M\}, \forall k \in K, \forall t \in T \tag{8}$$

$$q_i \geq 0, \forall i \in N \cup N_0 \tag{9}$$

The object function (1) represents that the goal of the model is to maximize the weight of inspection paths. Constrain (2) records the number of drones that depart from the drone station or return to the drone station. Constrain (3) restricts the drone from staying at a certain node after reaching it. Constrains (4-5) record the drones' battery power after they execute different flying paths. Constrain (6) restricts that each vessel can be monitored at most once. Constrain (7) shows that the variable indicating whether the path is selected is a binary value. Constrain (8) is the limitation on the number of drones in the drone station, while M represents the total number of drones in the model. Constrain (9) restricts that at any node the drone arrives at, the remaining battery power should be positive.

Considering the problem model presented above, the ACS solving method is given in the following Section III.

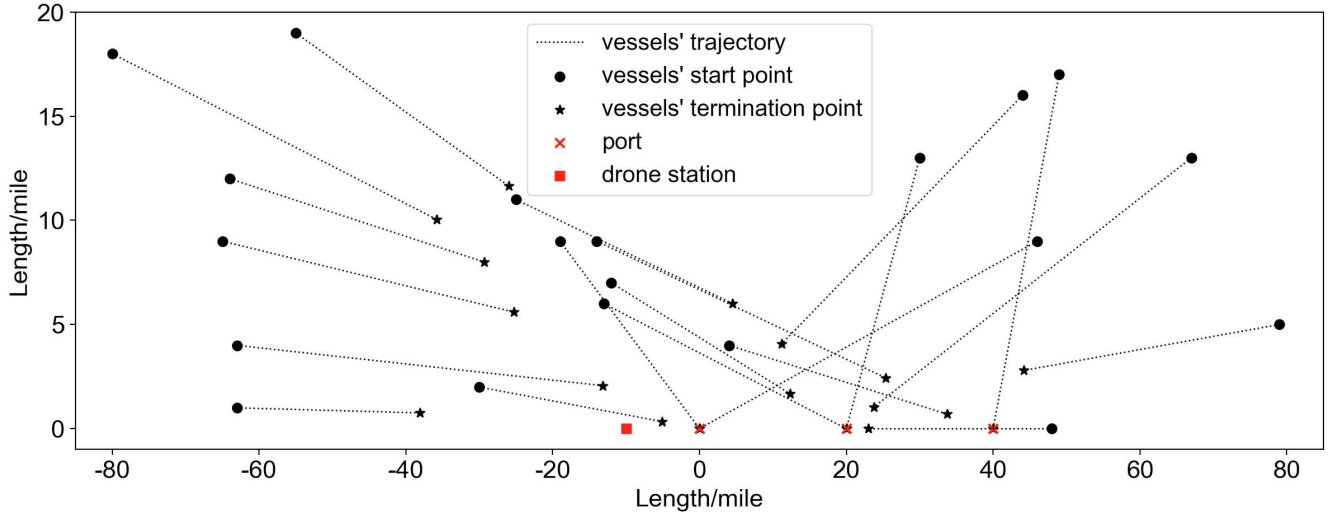

Fig. 1. An example of vessels' trajectory in the time interval.

III. Algorithm Model

A. Key Components in Designing ACS

Node Construction: Each vessel or drone station represents a node in the fully connected graph. The distance between nodes is calculated according to the drones' departure time. The weight of the vessel nodes corresponds to each vessel's inspection weight, and the weight of the drone station is zero.

Pheromone Initialization: The pheromone on the edge between each node is initialized to 1.

Vessel Node Visiting Constrain: Each ant keeps an *unvisited table*, which records all of the unvisited vessel nodes. The vessel node will be deleted once it is visited by the ant.

Drone's Battery Power Constrain: Each ant keeps a *reachable table*, which records all of the vessel nodes within its coverage. Once the ant arrives at a new node, it will update the *reachable table* by calculating if it can fly to the node in the *reachable table* and go back to the nearest drone station from that node with its left battery power. The ant will return to the nearest drone station for recharging if the *reachable table* is empty, and the *reachable table* will update according to the *unvisited table* after recharging.

Time Window Constrain: Once the ant arrives at a new node, it will update the *reachable table* by calculating if it can fly to the node in the reachable table and go back to the nearest drone station from that node within the time window. The ant will return to the nearest drone station if the *reachable table* is empty.

No Staying Constrain: Once the ant arrives at a new node, instead of staying at that node it will start choosing the next target node immediately.

Solution Construction: In each iteration, each ant represents every drone in every drone station in turn to construct the inspection path. The sequence of this process is shown in Fig. 2. For example, the ant first represents drone 1 in the drone station 1 and finishes its path constructing, then the ant represents drone 2 in the drone station 1 and constructs the path, until all the drones' path in the drone station 1 is constructed, the ant will start the same process in the drone station 2.

The realization of this process is given in Algorithm 1. In the algorithm, V represents the number of vessels, K represents the number of drone stations, m represents the number of drones in each drone station, $reachable$ represents the list of vessel nodes which the ant can reach, $unvisited$ represents the list of vessel nodes which the ant has not visited, and nex represents the next target node of the ant which is chosen by wheel selection.

Each ant will randomly select the next target vessel node in its reachable list by wheel selection, and the transition probability from node i to node j is calculated based on equation (10).

$$p(i,j) = \begin{cases} \frac{\tau(i,j)^{\alpha}\eta(i,j)^{\beta}}{\sum_{m \in reachable}\tau(i,j)^{\alpha}\eta(i,j)^{\beta}} & if\ j \in reachable \\ 0 & otherwise \end{cases} \tag{10}$$

$$\eta(i,j) = \frac{w(j)}{d(i,j)} \tag{11}$$

In equation (10), $\tau(i,j)$ means the amount of pheromone between node i and node j, $reachable$ is the $reachable\ table$ of the ant at node i, α is the weight of pheromone, β is the weight of heuristic factor, and $\eta(i,j)$ is the heuristic factor between node i and node j, which can be calculated by equation (11). In equation (11), $w(j)$ is the inspection weight of node j, and $d(i,j)$ represents the distance between node i and node j.

Pheromone Updating Rule: The local pheromone updating is performed after each ant finishing its solution construction, and the global pheromone updating is performed after one iteration based on the global best solution. While the significance of local pheromone updating is to explore a more

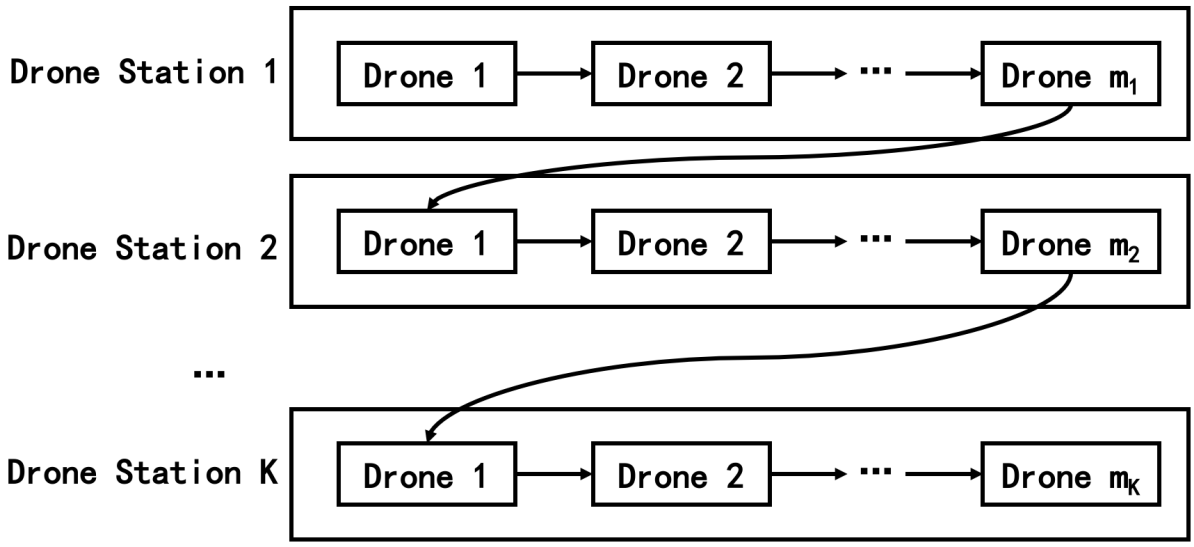

Fig. 2. The Sequence of Drones' Schedule Construction.

Algorithm 1 Route Constructing for Drones

function ROUTE CONSTRUCTING($unvisited$,$reachable$)
  for $i = 1 \rightarrow K$ **do**
    for $j = 1 \rightarrow m$ **do**
      $reachable \leftarrow unvisited$
      $reachable$ update under battery constrain;
      while $reachable$ not empty **do**
        choose next node nex from $reachable$;
        drone move to nex, update parameters;
        remove nex from $reachable$ and $unvisited$;
        $reachable$ update under battery constrain;
        if $reachable$ is empty **then**
          drone return to drone station and recharge;
          $reachable \leftarrow unvisited$
          $reachable$ update under battery constrain;
        end if
        $reachable$ update under time window constrain;
      end while
    end for
  end for
end function

optimal solution, the purpose of global pheromone updating is to reserve the global best solution.

After each ant finishing its construction for the solution, the local updating is performed on each edge of the path according to equation (12).

$$\tau(i,j) = \tau(i,j) * (1-\rho) + \tau_0 * \rho \tag{12}$$

In this equation, ρ $(0 < \rho < 1)$ is the decay factor of local pheromone updating.

In the global pheromone updating process, equation (13) is applied to all edges in the pheromone graph.

$$\tau(i,j) = \tau(i,j) * (1-\epsilon) + \Delta\tau(i,j) * \epsilon \tag{13}$$

$$\Delta\tau(i,j) = \begin{cases} \frac{w_b * c_{nn}}{\sum_{v \in V} w_v} & if\ (i,j) \in global - best - route \\ 0 & otherwise \end{cases} \tag{14}$$

$\Delta\tau(i,j)$ is calculated according to equation (14), and ϵ $(0 < \epsilon < 1)$ is the decay factor of global pheromone updating. In equation (14), w_b is the sum of inspection weight of the global best solution, w_v is the inspection weight of vessel v, and c_{nn} is a constant obtained by the greedy algorithm.

Fitness Function: In this problem, the fitness function is the sum of inspection weight of vessel nodes included in the path, and it can be calculated by equation (15).

$$F = \sum_{v \in A} w_v \tag{15}$$

In this equation, A is the node-set included by ant's path.

B. Complete ACS Algorithm

The complete ACS algorithm is given in Algorithm 2.

Algorithm 2 Complete ACS Algorithm

Initialize the parameters
for $iter = 1 \rightarrow itermax$ **do**
  Each ant constructs route and local pheromone updating
  Global best route updating
  Global pheromone updating
end for

IV. Experiments and Discussions

In this section, the proposed ACS method for drone monitoring will be compared with Xia's Lagrangian relaxation-based method [14]. Considering the Lagrangian relaxation-based method has been able to obtain better results than the direct solution from the CPLEX solver, this paper will not take the direct solution from the CPLEX solver into consideration. All algorithms are implanted in python, and the experiments are performed on an Intel i9-10900K(3.7GHz) Desktop PC with 32 GB RAM.

TABLE I
Parameters of ACS

Parameter	Value	Parameter	Value
α	1	ϵ	0.1
β	4	I	2500
ρ	0.01	N	50

The number set of vessels in the experiment is $V \in \{20, 40, 60, 80, 100, 120, 140, 160, 180, 200\}$. The number set of drone station is $K \in \{1, 2\}$. The experimental area is rectangular with [-85,85] nautical miles wide and [0,20] nautical miles long. The upper bound of the time window is set to $T_{max} = 300$ minutes. The positions of ports are [0,0],[20,0],[40,0], and the positions of drone stations are [0,0],[30,0]. The positions of vessels are randomly generated inside the experimental area, the target ports of vessels are randomly generated in three ports, and the vessels' inspection weight and sailing speed are randomly generated in [5,15] and [5,10] knots respectively. The number of drones in each drone station is $m = 5$, the drones' moving speed is set to 30 knots, and the battery power duration is $Q_0 = 125$ minutes.

The parameters of the ACS are set as shown in Table I. α is the parameter that determines the importance of pheromone, and β is the parameter that determines the importance of heuristic information. ρ is the decay factor of local pheromone updating, and ϵ is the decay factor of global pheromone updating. I is the number of maximum iteration. N is the number of ants in each iteration. Pheromone weight α, heuristic information weight β, and the decay factor of global pheromone updating ϵ are set as the recommendation values of typical ACS. And the number of ants N is increased to strengthen the ability of exploration, while the local factor of global pheromone updating ρ is decreased correspondingly.

In our experiment, the maximum running time is 7200 seconds. However, the Lagrangian relaxation-based method is iteratively running in rounds and the running time is checked after each iteration, so it is possible that the running time exceeds the upper bound. In each experimental condition, ACS is run 20 times. Considering that the Lagrangian relaxation-based method is a certain process, it is run only one time as the comparison result.

According to the experiment results shown in Table II, when the number of vessels and drone stations are large, ACS can achieve a better solution in a shorter time comparing to the Lagrangian relaxation-based method, especially when the number of drone stations is 2 and the number of vessels is larger than 80. The Lagrangian relaxation-based method needs to construct the path set every iteration and the number of feasible paths constructed every round is limited, so as the number of vessels and drone stations becoming larger the efficiency and effect of this method are limited. However, ACS proposed in this paper only considers vessels or drone stations as ants' moving targets, so it is less affected by the number of drone stations and vessels.

When the number of vessels and drone stations is small, ACS cannot guarantee that the result of each operation is better than or equal to that of the Lagrangian relaxation-based method. The average quality of the solution is still comparable and acceptable. And the running speed of ACS still has a very obvious advantage comparing with the Lagrangian relaxation-based method.

In the experiment, ACS's typical changing of the global optimal solution is shown in Fig. 3. As shown in these figures, the global optimal value is improved quickly by the ACS in the early stages of the iteration. When the number of iteration exceeds 500, most of the global optimal solution tends to be stable. Based on such performance, we can consider shortening the number of iterations in real-field applications to increase the speed of path optimization.

Based on the scheduling results obtained by ACS, a visualized scheduling solution can be presented as Fig. 4. The black dots represent the position of drone stations, the red "x" represents the target port of vessels, the black dashed lines represent the vessels' trajectory in the time window, and the blue and red solid lines represent the drones' flying trajectory respectively.

V. Conclusion

In this paper, ACS is applied to the problem of drone scheduling for vessel emission inspection. And a rapid solution for drone scheduling is realized with this algorithm. According to the experimental results, ACS can obtain a better scheduling solution with higher efficiency when facing a large number of vessels and drone stations. And in the case of a small number of vessels and drone stations, ACS is still able to obtain a sub-optimal solution in a shorter time, which means a further improvement in efficiency with little affecting on the inspection quality.

In a further study, the algorithm mechanism and problem mapping methods will be further modified to improve the solution quality and efficiency. And the optimize of drone station location and drones' fleet size will also be considered based on the optimization results.

References

[1] P. Balcombe, J. Brierley. How to decarbonise international shipping: Options for fuels, technologies and policies. Energy Conversion and Management, 2018, 182, 72-88

[2] Y. Xiao, X. Fu, et al. Port investments on coastal and marine disasters prevention: Economic modeling and implications. Transportation Research Part B: Methodological, 78, 2015, 202-221

TABLE II
THE RESULTS OF EXPERIMENTS

$\|V\|$	$\|K\|$	Lagrangian method		ACS			
		F	**Time (s)**	**Ave F**	**Max F**	**Min F**	**Time (s)**
20	1	**111**	5717	99	102	98	**65**
20	2	**150**	2418	**150**	150	150	**101**
40	1	**205**	2584	**205**	205	205	**158**
40	2	**292**	7593	**292**	292	292	**280**
60	1	364	7265	**369**	369	369	**347**
60	2	489	7640	**495**	495	494	**593**
80	1	**449**	7309	448	455	440	**510**
80	2	530	8988	**627**	635	622	**921**
100	1	493	8470	**510**	525	506	**670**
100	2	673	8143	**808**	815	798	**1142**
120	1	361	8782	**586**	600	572	**936**
120	2	729	12988	**949**	965	937	**1903**
140	1	388	12625	**615**	627	602	**1375**
140	2	656	11338	**1051**	1069	1035	**2627**
160	1	398	7459	**761**	775	744	**1613**
160	2	856	14233	**1238**	1248	1232	**3178**
180	1	436	12017	**810**	828	798	**2431**
180	2	891	19457	**1318**	1338	1279	**4290**
200	1	461	10183	**858**	876	837	**2302**
200	2	1032	19680	**1392**	1414	1379	**4650**

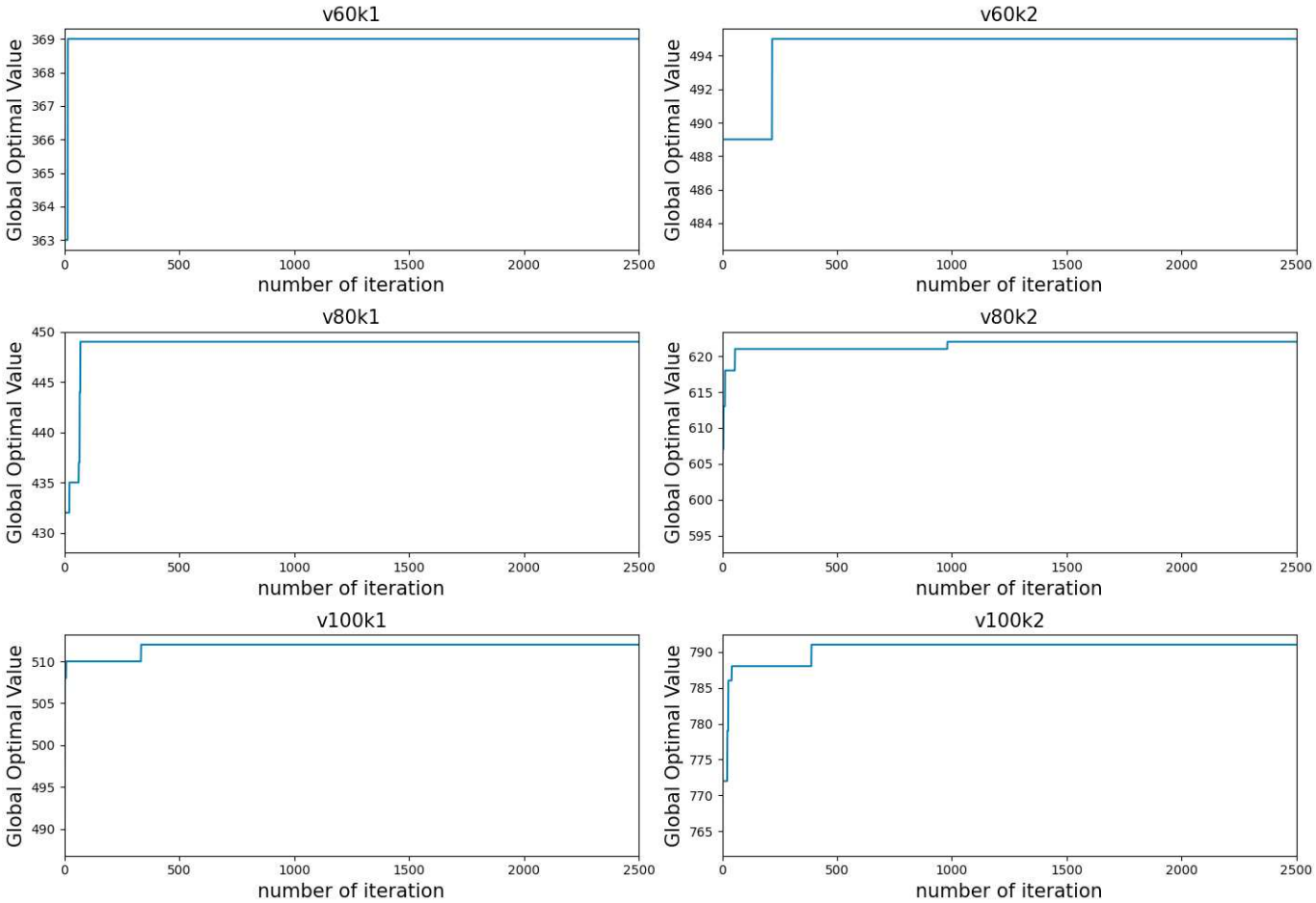

Fig. 3. Typical Changing of Global Optimal Solutions under different cases.

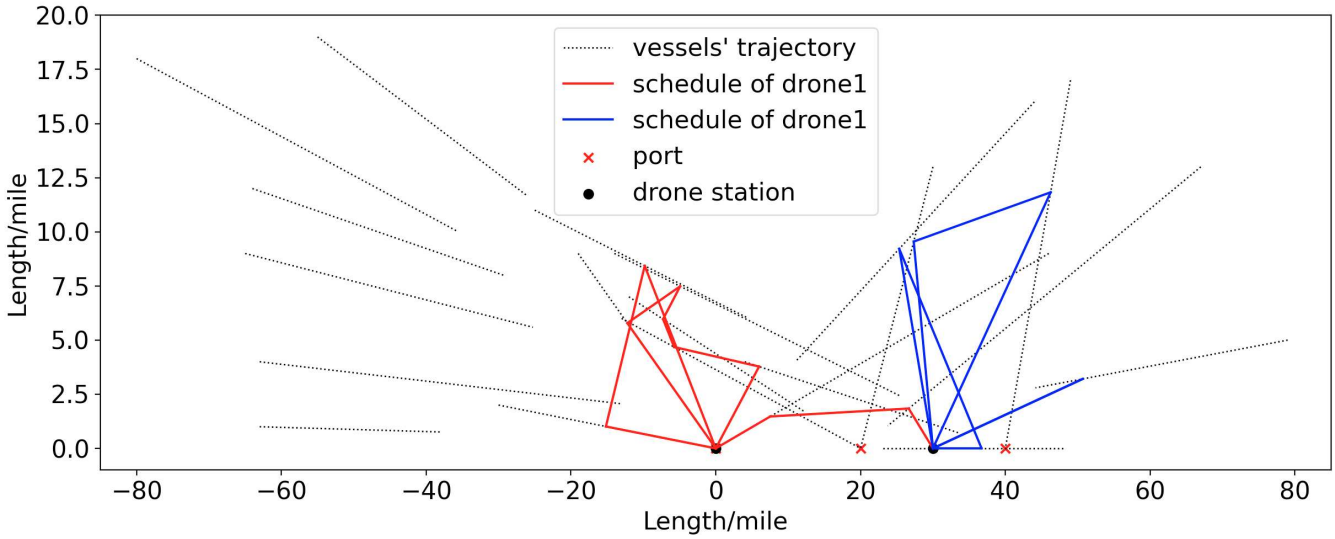

Fig. 4. Visualized Solution for Scheduling Problem with 20 Vessels.

[3] X. Zhang, Y. Zhang. Changes in the SO2 Level and PM2.5 Components in Shanghai Driven by Implementing the Ship Emission Control Policy. Environ. Sci. Technol, 2019, 53, 19, 11580–11587

[4] S. Zheng, Y. Ge. Modeling collusion-proof port emission regulation of cargo-handling activities under incomplete information. Transportation Research Part B: Methodological, 104, 2017, 543-567

[5] J. Chen, T. Zheng. Alternative Maritime Power application as a green port strategy: barriers in China. Journal of Cleaner Production, 2019, 213, 825-837

[6] Implementation plan of ship emission control area in Pearl River Delta, Yangtze River Delta and Bohai Rim (Beijing Tianjin Hebei). Ministry of Transportation of the People's Republic of China, 2015,177

[7] Reducing sulfur emissions from ships. OECD, 2018

[8] Q. He, X. Zhang, et al. Speed optimization over a path with heterogeneous arc costs. Transportation Research Part B: Methodological, 104, 2017, 198-214

[9] Fung. Enforcement of fuel switching regulations – practices adopted in the US, EU and other regions, and lessons learned for China (2016). NRDC

[10] Q. Meng, Y. Du, et al. Shipping log data based container ship fuel efficiency modeling. Transportation Research Part B: Methodological, 83, 2016, 207-229

[11] J. Chen, S. Wang. A Modelling Framework of Drone Deployment for Monitoring Air Pollution from Ships. Intelligent Interactive Multimedia Systems and Services, 2018, 281-288

[12] S. Beryozkina, N. AI-Shakhs, et al. Real-life Application of the Emission Monitoring System by Using a Drone. International Conference on Environment and Electrical Engineering, 2020

[13] M. Degner, H. Ewald, et al. Mobile gas sensing system for detection of combustion pollutants – suitable for drone based measurements. International Conference on Sensing Technology, 2018

[14] J. Xia, K. Wang. Drone scheduling to monitor vessels in emission control areas. Transportation Research Part B, 2019, 174-196

[15] M. Dorigo, L. Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 1, NO. 1, 1997

[16] L. Gambardella, E. Taillard. MACS-VRPTW: A Multiple Colony System For Vehicle Routing Problems With Time Windows. New Ideas in Optimization, 1999, 63-76

[17] R. Montemanni, L. Gambardella. Ant Colony System for a Dynamic Vehicle Routing Problem. Journal of Combinatorial Optimization, 2005, 10, 327–343

[18] Z.H. Sun, D. Liang et al. Multi-task processing oriented production layout based on evolutionary programming mechanism. Applied Soft Computing, VOL.98, 2021

[19] X. Liu, Z. Zhan. An Energy Efficient Ant Colony System for Virtual Machine Placement in Cloud Computing. IEEE Transactions on Evolutionary Computation, VOL.22, 2016