Soft Computing (2024) 28:6637–6656
https://doi.org/10.1007/s00500-023-09513-x

APPLICATION OF SOFT COMPUTING

# An energy-saving path planning method using adaptive ant colony system for marine surface vehicles

Liangxiong Dong[1] · Xinhua Gan[2]

Accepted: 23 November 2023 / Published online: 28 December 2023

**Abstract**
Given the need for ship emission reduction and the energy constraints of unmanned surface vehicles, it is imperative to develop an efficient ship path planning method that incorporates energy-saving considerations. In this paper, considering the challenges posed by the complexity of the marine environment, an adaptive ant colony system is proposed for ship energy-saving path planning. Firstly, a ship energy consumption estimation model is presented by analyzing the influence of wind, wave, and current on energy consumption. Secondly, two process evaluation indexes are designed for the quality and diversity of the paths in the optimization process. Based on these indexes, three improvements are presented to enhance the performance of the ant colony system. These improvements include (a) an adaptive state transition rule adjusted dynamically based on the information entropy, (b) the number of ants contributing to the global updating rule and the corresponding update value are controlled dynamically based on the quality of the path, and (c) an improved recombination mechanism. Besides, an improved turning point optimization method and B-spline curve are introduced for path post-processing. Finally, the effectiveness of the proposed method is verified by simulation experiments.

**Keywords** Path planning · Ant colony system · Energy saving · Marine surface vehicle

## 1 Introduction

Over the past decade, ship autonomy technology has made significant advancements due to extensive research on intelligent algorithms such as deep learning and bionic algorithms (Chen et al. 2021; Xiao et al. 2021; Tang et al. 2021; Öztürk et al. 2022; Xing et al. 2022). In the pursuit of achieving full autonomy for ships, path planning algorithms play a crucial role. The path planning algorithm aims to optimize the economy and safety of the ship while performing tasks (Niu et al. 2020). One aspect that has garnered significant attention from scholars is the economy of the path, with a focus on reducing carbon emissions and operating costs (Barreiro et al. 2022; Du et al. 2022a). In particular, for Unmanned Surface Vehicles (USVs) with limited energy, the energy required for the path determines the feasibility of the task. To guide the ship to the destination safely with lower energy consumption in the intricate marine environment, an advanced path planning algorithm considering energy consumption is urgently needed.

Path planning algorithms can be categorized into two types based on the availability of environmental information: global path planning algorithms and local path planning algorithms (Zhao et al. 2023). The global path planning algorithm primarily focuses on ensuring the optimality of the path using known information. On the other hand, the local path planning algorithm is specifically designed to handle unknown risks that may arise during navigation. As the global path planning significantly influences the energy consumption required for ship mission execution, this article will primarily concentrate on researching global path planning algorithms.

For the ship path planning problem, many algorithms and their variants have been used, such as Dijkstra's

✉ Xinhua Gan
ganxinhua@zjou.edu.cn

Liangxiong Dong
dongliangxiong@163.com

[1] School of Marine Engineering, Guangzhou Maritime University, Guangzhou 510725, China

[2] School of Naval Architecture and Maritime, Zhejiang Ocean University, Zhoushan 316022, China

algorithm (Zyczkowski and Szlapczynski 2023), A* algorithm (Song et al. 2019), genetic algorithm (GA; Wang et al. 2020), ant colony optimization (ACO; Dong et al. 2021), particle swarm optimization (PSO; Du et al. 2022b), and pigeon-inspired optimization (PIO; Duan and Qiao 2014). Dijkstra's algorithm and A* algorithm are commonly used graph-based methods for calculating the shortest path between two points. These algorithms are highly efficient for small-scale path planning tasks. However, they may require significant computational time when dealing with complex environments (Gu et al. 2023). Compared to graph-based methods, intelligent algorithms like GA, ACO, and PSO demonstrate better adaptability in handling complex path planning tasks. Among them, the ACO is widely utilized for solving path planning problems by virtue of its self-organization, parallelism, and positive feedback (Liu et al. 2017; Hou et al. 2022; Yu et al. 2022). The ACO algorithm is generally effective, but a typical negative feature is its tendency to fall into a local optimum (Stodola et al. 2022). To this end, researchers have improved and extended the fundamentals of the algorithm with various techniques.

In ACO, ants choose nodes randomly through a roulette selection process based on node information. While this approach enhances the global search ability, it may also impact the convergence speed of the algorithm. Thus, a pseudo-random-proportional rule is designed (Dorigo and Gambardella 1997), and compared to the traditional state transition rule it prefers nodes with low cost and a high amount of pheromones, thereby accelerating the convergence of the algorithm. Based on this, the researchers made improvements to the control parameter by adjusting it according to various factors such as the quality of the solution (Luo et al. 2020), the number of iterations (Wu et al. 2023), and the diversity of the solution (Zhang et al. 2021a). To address the drawbacks caused by the uniform distribution of initial pheromones, some methods of unevenly distributing pheromones are usually adopted to improve the performance of ACO. These methods include providing extra pheromones to the semicircular region where the target point is located (Tang and Xin 2022), adding additional pheromones to the elliptical region between the target point and the starting point (He and Fan 2021), and adaptively allocating pheromones according to the distance of the node from the starting point and the target point (Luo et al. 2020; Zhang et al. 2021b; Wu et al. 2023). To enhance the guidance of previous ants to subsequent ants, Miao et al. (2021) performed additional rewards and penalties for the optimal and worst paths after one iteration, respectively. Liu et al. (2017) devised a new pheromone diffusion strategy to improve the search performance of the algorithm, where the pheromone diffuses along the direction of the potential field force. In addition to optimizing the fundamentals of ACO, some studies have embarked on further optimization of the solutions, such as comparing and recombining different solutions to obtain a better solution (Gao et al. 2020; Hou et al. 2022) or discarding some redundant nodes (Yang et al. 2019). Based on the above analysis, the primary challenges to enhancing the performance of ACO in global path planning can be summarized as follows: (1) striking a balance between global exploration and convergence speed, and (2) enhancing solution utilization and post-processing.

The energy consumption of ships during navigation is intricately linked to the marine environment. Therefore, it is necessary to obtain the relationship model between ship energy consumption and the marine environment. In the study conducted by Xia et al. (2019), a calculation method for ship energy consumption is proposed using the ship dynamics model. On the other hand, Lee et al. (2011, 2015) calculated the energy consumption of ships by considering the work done to overcome environmental resistance. Besides, calculations using either the fitted formula (Zhou 2020) or main engine power and sailing resistance (Du et al. 2022a; Han et al. 2022) are also commonly used.

The above-mentioned research has achieved some achievements, however, it should be noted that there is limited research on ship energy-saving path planning. Developing an algorithm that can effectively plan a global path for energy-saving while considering ship motion characteristics remains a challenging task. For example, in the process of ship energy-saving path planning, the ship energy consumption model is required to take into account both accuracy and efficiency. Additionally, most of the existing algorithms still have some weaknesses in complex marine environments, such as easy to fall into local optimum and excessive turning points. To this end, this paper proposes an adaptive ant colony system (AACS) for energy-saving path planning of ships. Firstly, inspired by previous research, a ship energy consumption estimation model considering the influence of wind, wave, and current on ship energy consumption is designed. This work aims to save energy by leveraging the marine environment. Therefore, the proposed ship energy consumption estimation model primarily calculates the energy expended by the ship to counteract external disturbances. Secondly, our proposed algorithm improves on the fundamentals of ACS based on the diversity of the planning process and the quality of the solution. Given the diversity of the planning process, the proportion of algorithms exploitation and exploration is adaptively adjusted. And the number of ants involved in the global updating and recombination of solutions is based on the quality of the solution in the iteration. Also, to further reduce the energy consumption and redundant nodes, an improved turning point optimization method and B-spline curve are designed. The contributions of this work are summarized as follows:

(1) A ship energy consumption estimation model for evaluating and guiding energy-saving path planning is established. The model can effectively reflect the influence of wind, wave, and current on energy consumption during ship navigation.

(2) By introducing the process evaluation method and the improved recombination mechanism into ACS, an adaptive ant colony system is proposed to address the problem of ship energy-saving path planning. AACS effectively balances global search and convergence speed, while also leveraging valuable components of suboptimal solutions to enhance the performance of the algorithm.

(3) A post-processing smoothing approach including an improved turning point optimization method and a B-spline smoothing method is proposed to reduce the energy consumption of the path and improve the smoothness.

The rest parts of this study are structured as follows: In Sect. 2, the environmental modeling and the energy consumption estimation model are described. In Sect. 3, the improving methods and process of AACS are explained. In Sect. 4, the comparative experimental results are discussed. Conclusions are given in Sect. 5.

## 2 Related model of ship path planning problem

### 2.1 Grid environment model

Environment modelling is a technique for simulating real application scenarios. It is a prerequisite for path planning and provides environmental information for algorithm implementation (Liu et al. 2023). In this article, mainly considering the position changes of the ship, it is assumed that the marine environment is a two-dimensional space. In terms of modeling two-dimensional space, the grid method is a common method (Chen et al. 2021). The grid method decomposes the space $S$ into white grids and black grids, which represent navigable areas $S_f$ and non-navigable areas $S_o$, respectively (Zhao et al. 2023). The white grid can store various kinds of environmental information to further simulate the actual environment. To ensure the safety of ship navigation, the proposed model considers errors in map data and ship positioning information. It expands the non-navigable area outwards, and the expanded part is treated as gray grids, representing the non-navigable risk area $S_r$. The composition of the space is shown in Eq. (1). Figure 1 represents the process of gridding a virtual environment into a grid map after expanding the obstacles. Besides, the grid coding method used in the model primarily involves serial index and coordinates. In the cost calculation process, it is often necessary to convert the serial index into coordinates. The conversion method is shown in Eq. (2).

$$S{=}S_f + S_o + S_r \tag{1}$$

$$\begin{cases} x = \begin{cases} b(\mathrm{mod}(index, N_x) - 0.5), \mathrm{mod}\,(index, N_x) \neq 0 \\ b(N_x - 0.5), \mathrm{mod}(index, N_x) = 0 \end{cases} \\ y = b\left(N_y + 0.5 - \mathrm{ceil}(index/N_x)\right) \end{cases} \tag{2}$$

where $b$ is the size of the grid; $N_x$ and $N_y$ are the number of grids in row and column direction respectively.

### 2.2 Energy consumption estimation model

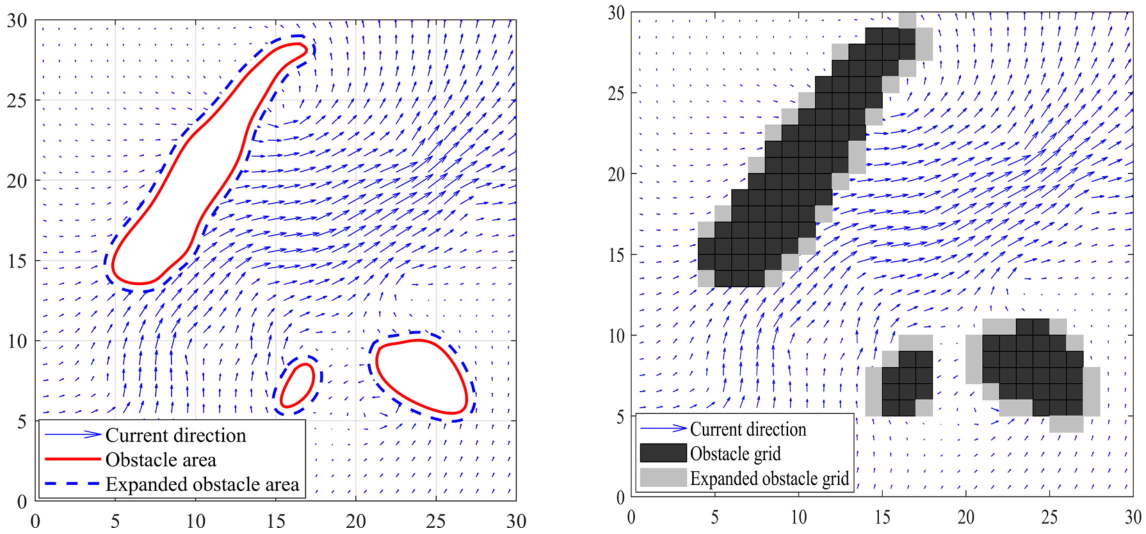
For planning of ship's energy-saving route, it is necessary to obtain the estimation model of the energy consumption required by the ship moving between grids. To this end, this study presents a ship energy consumption estimation model, inspired by the approach proposed by Lee et al. (2011). This method assumes that the resistance of the ship and the thrust of the propulsion system are balanced, thereby simplifying the calculation of energy consumption to the calculation of work required to overcome resistance. The coefficients in the energy consumption estimation model are given in Table 1.

The main sources of resistance to a ship while sailing are wind, wave, and current. In practical application problems, the wave forces acting on ships are divided into the first-order wave forces and the higher-order wave forces (Zhang et al. 2022). Among these, second-order wave forces are the main factor influencing the position of the ship. This paper assumes that the second-order wave force is a constant value and applies the calculation method proposed by Daidola to determine the wave drift force acting on the ship in regular waves. The calculation method can be expressed as:

$$\begin{bmatrix} F_{wave\_x} \\ F_{wave\_y} \end{bmatrix} = \frac{\rho_{sea} g L \xi^2}{2} \begin{bmatrix} C_{xw} \cos(\delta) \\ C_{yw} \sin(\delta) \end{bmatrix} \tag{3}$$

where wave drift force coefficients $C_{xw}$ and $C_{yw}$ can be calculated using the regression formula derived by Daidola (Sun 2021):

$$\begin{cases} C_{xw} = 0.05 - 0.2\left(\frac{\lambda}{L}\right) + 0.75\left(\frac{\lambda}{L}\right)^2 - 0.51\left(\frac{\lambda}{L}\right)^3 \\ C_{yw} = 0.46 + 6.83\left(\frac{\lambda}{L}\right) - 15.65\left(\frac{\lambda}{L}\right)^2 + 8.44\left(\frac{\lambda}{L}\right)^3 \end{cases} \tag{4}$$



(a) Expanded virtual map

(b) Grid map after gridding

**Fig. 1** Environmental gridding

**Table 1** Coefficients in the energy consumption estimation model

| Coefficients | Definition or meaning (Unit) |
|---|---|
| $\rho_{\text{sea}}$ | Density of sea water (kg/m$^3$) |
| $\rho_{air}$ | Density of air (kg/m$^3$) |
| $L$ | LOA of the ship (m) |
| $L_{ij}$ | Distance between grid i and j (m) |
| $g$ | Gravitational acceleration (m/s$^2$) |
| $\xi$ | Average wave amplitude (m) |
| $\lambda$ | Wave length (m) |
| $C_{xw}$ | Wave drift force coefficient in x direction |
| $C_{yw}$ | Wave drift force coefficient in y direction |
| $\delta$ | Wave encounter angle (deg) |
| $u$ | Forward speed of the ship |
| $u_{rw}$ | Forward speed of the ship relative to the wind (m/s) |
| $v_{rw}$ | Sway speed of the ship relative to the wind (m/s) |
| $u_{rc}$ | Forward speed of the ship relative to the current (m/s) |
| $v_{rc}$ | Sway speed of the ship relative to the current (m/s) |
| $A_{ax}$ | Maximum longitudinal projections of aerodynamic area (m$^2$) |
| $A_{ay}$ | Maximum transverse projections of aerodynamic area (m$^2$) |
| $S_{wet}$ | Wetted surface area (m$^2$) |
| $D$ | Draft (m) |
| $C_D$ | Drag coefficient |
| $C_T$ | Total resistance coefficient |
| $C_x^{wind}$ | Wind pressure coefficient in the x-direction |
| $C_y^{wind}$ | Wind pressure coefficient in the y-direction |
| $F_x^{each}$ | Force exerted by each environmental factor in the x-direction of the ship (N) |
| $F_y^{each}$ | Force exerted by each environmental factor in the y-direction of the ship (N) |
| $v_y^{each}$ | Speed of each environmental factor in the y-direction relative to the ship (m/s) |

For the wind, the approximate equation shown in Eq. (5) is used to calculate the disturbance force it exerts on the ship.

$$\begin{bmatrix} F_{wind\_x} \\ F_{wind\_y} \end{bmatrix} = \frac{\rho_{air}}{2}\begin{bmatrix} C_x^{wind} u_{rw}|u_{rw}|A_{ax} \\ C_y^{wind} v_{rw}|v_{rw}|A_{ay} \end{bmatrix} \tag{5}$$

Based on the slender body assumption, the resistance caused by the current when the ship is sailing can be expressed as follows (Zhang et al. 2022; Lee et al. 2011).

$$\begin{bmatrix} F_{current\_x} \\ F_{current\_y} \end{bmatrix} = \frac{\rho_{sea}}{2}\begin{bmatrix} S_{wet} C_T u_{rc}|u_{rc}| \\ D C_D L v_{rc}|v_{rc}| \end{bmatrix} \tag{6}$$

When the next node $j$ to be moved by the ship is determined, the energy consumed $E_{total}$ of the ship from $i$ to $j$ can be calculated according to the following.

$$\begin{bmatrix} E_x^{each}(i,j) \\ E_y^{each}(i,j) \end{bmatrix} = \frac{L_{ij}}{2}\begin{bmatrix} F_x^{each}(i) + F_x^{each}(j) \\ \left(F_y^{each}(i) v_y^{each}(i) + F_y^{each}(j) v_y^{each}(j)\right)\frac{1}{u} \end{bmatrix} \tag{7}$$

$$E_{total}(i,j) = \sum E_x^{each}(i,j) + \sum E_y^{each}(i,j) \tag{8}$$

In the process of edge cost calculation, there will inevitably be negative energy consumption values. For state transition rule, negative values should be avoided. In the y-direction, the ship always needs to do work to offset the impact of the environment, so it is reasonable to take the absolute value of $E_y^{each}$. And in the x-direction, since the modification of the edge cost is easy to impair the physical meaning, the negative energy consumption value is often set to 0.

## 3 Design of the adaptive ant colony system

### 3.1 Process evaluation method

Ant colony system (ACS) is a swarm intelligence heuristic algorithm used to search for optimal solutions. As a well-known variant of ACO, ACS has good search ability, but it still has the disadvantage of being prone to falling into local optimum (Zhang et al. 2021b). To address this, two process evaluation indexes are designed and introduced into ACS.

(1) Process diversity index

The diversity of candidate solutions reflects the current global search ability of the algorithm, but few studies have analyzed and utilized it. In this paper, information entropy is introduced as a process diversity index to quantify the diversity in the optimization process. Without loss of generality, it is assumed that $r$ different paths are obtained by $e$ ants, where the number of ants searching for path $i$ is $a$. And the probability of path $i$ being searched and the information entropy of this iteration can be calculated according to Eqs. (9) and (10).

$$P_i(t) = \frac{a}{e} \tag{9}$$

$$H(t) = -\sum_{i=1}^{r} P_i(t) \ln P_i(t) / \ln(e) \tag{10}$$

where $P_i(t)$ represents the probability that path $i$ is searched in the t-th iteration;$H(t)$ is the information entropy in the t-th iteration.

(2) Path quality index

Under the influence of the local updating rule, ACS will obtain more different solutions in a single iteration compared to ACO. In order to assess different solutions effectively, it is crucial to establish an evaluation index. Therefore, a path quality index is proposed in this work, mapping different solutions to [0, 2]. When the energy consumption of path $i$ is lower than the current optimal solution, path $i$ is mapped to an interval greater than 1, and the degree of exceeding 1 is negatively correlated with the energy consumption; otherwise, path $i$ is mapped to an interval smaller than 1. The path quality index is designed as follows.

$$\varsigma(i) = 1 + \frac{2}{\pi}\arctan\left(\frac{E_{\min} - E(i)}{E_{aver} - E_{\min} + 1}\right) \tag{11}$$

where $\varsigma(i)$ is the mapping value of path $i$;$E(i)$ is energy consumption of path $i$;$E_{\min}$ is the energy consumption of the current optimal path, and $E_{aver}$ is the average energy consumption of all the complete paths in this iteration.

### 3.2 Adaptive ant colony system

#### 3.2.1 Improved heuristic information function

To achieve the planning of energy-saving paths, it is necessary to combine the heuristic information function with the energy consumption estimation model mentioned in Sect. 2.2. However, the heuristic information function based only on the energy consumption value tends to lead the algorithm into a deadlock state. Therefore, in this article, a novel heuristic information function is designed by combining energy consumption, distance information of the goal point and distance information of the next point. The function is calculated as follows:

$$\omega_{ij} = \frac{k_1}{E(j)} + \frac{1}{k_2 d_{jz} + k_3 d_{ij}} \tag{12}$$

where $d_{jz}$ and $d_{ij}$ are the Euclidean distances from the candidate node $j$ to the goal point $z$ and the current point $i$, respectively; And $k_1$, $k_2$ and $k_3$ are the corresponding coefficients respectively.

### 3.2.2 Adaptive state transition rule

Heuristic algorithms are very sensitive to parameters and, therefore, these algorithms generally lack adaptability to different application scenarios (Stodola et al. 2022). As an essential step of ACS, the state transition rule plays a significant role in the optimization process. However, the control parameter of the state transition rule in ACS is usually set manually as fixed values and cannot be dynamically adjusted according to the present state. For enhancing the algorithm's ability to perform global searches, an adaptive state transition rule based on the process diversity index is proposed, which can be calculated according to Eqs. (13) and (14).

$$\varepsilon(t) = \begin{cases} q_0 + (H(t) - \varphi)\,,\ H(t) > \varphi \\ q_0\,,\ H(t) \le \varphi \end{cases} \tag{13}$$

$$s = \begin{cases} \mathop{\arg\max}\limits_{j \in allowed_k} [\tau_{ij}(t)]^{\alpha} [\omega_{ij}(t)]^{\beta}\,,\ if\ q \le \varepsilon(t) \\ \dfrac{[\tau_{ij}(t)]^{\alpha} [\omega_{ij}(t)]^{\beta}}{\sum\limits_{s \in allowed_k} [\tau_{is}(t)]^{\alpha} [\omega_{is}(t)]^{\beta}}\,,\ if\ q > \varepsilon(t) \\ 0\,,\ else \end{cases} \tag{14}$$

where $\tau_{ij}$ is the pheromone value of the edge $ij$;$\varepsilon(t)$ is a dynamic control parameter in the t-th iteration; $q$ is a random number between [0,1]; $q_0$ is the initial value of the control parameter; $\alpha$ and $\beta$ are the coefficients that adjust the influence of pheromone and heuristic information on the node selection, respectively;$\varphi$ is a constant and its value with $q_0$ will be determined by statistical experiments in Sect. 4.1.

### 3.2.3 Adaptive global updating rule

For ACS, once all ants have complete their search, pheromone is updated on the edges of the most energy-saving path according to

$$\tau(t+1) = (1-\rho)\cdot\tau(t) + \rho\cdot\Delta\tau^{\min} \tag{15}$$

$$\Delta\tau^{\min} = \frac{Q}{E_{\min}} \tag{16}$$

where $\rho$ is a pheromone decay parameter;$\Delta\tau^{\min}$ is the pheromone increment on the best path; $Q$ is the update intensity of pheromones; $E_{\min}$ is the energy consumption of the best path.

The traditional global updating rule, as shown in Eq. (15), can effectively converge the ant colony to the vicinity of the sub-optimal solution. However, it is also easy to fall into local optimum. To overcome the above shortcoming, an adaptive global updating rule based on the path quality index is proposed.

Specifically, the repetitive parts in the effective solution will be removed after each iteration, and the remaining parts will be evaluated and mapped to [0,2] according to Eq. (11). In order to ensure good global search ability in the early stage and continuously strengthen the convergence capabilities with iterations, this paper proposes a dynamic threshold $v$ to regulate the number of pheromone update objects. When the mapping value of the solution is greater than the threshold value, the corresponding pheromone will be rewarded, otherwise no operation will be performed. The adaptive global updating rule is defined as:

$$\tau(t+1) = \begin{cases} (1-\rho)\cdot\tau(t) + \rho\cdot\left(\sum\limits_{k=1}^{f}\Delta\tau_k(t) + \Delta\tau^{\min}\right)\,,\ if\ f > 0 \\ (1-\rho)\cdot\tau(t) + \rho\cdot\Delta\tau^{\min}\,,\ \text{otherwise} \end{cases} \tag{17}$$

$$\Delta\tau_k(t) = \frac{Q\cdot(f - g_k + 1)\cdot\varsigma(k)^2}{f\cdot k\cdot E_k} \tag{18}$$

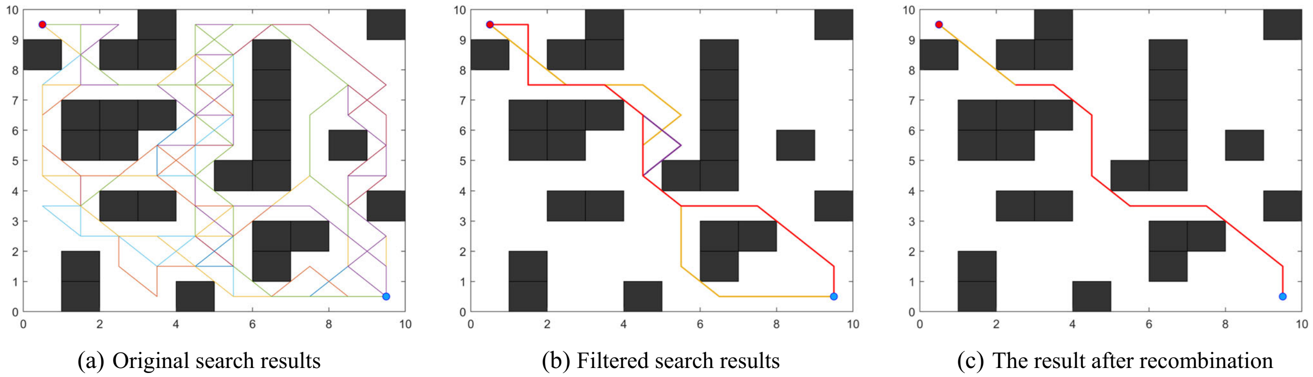
$$v = v_0 - (v_0 - 1)\cdot\left(\frac{2t}{T}\right)^2 \tag{19}$$

where $f$ is the number of effective solutions after filtering; $g_k$ is the ranking of the effective solution in this iteration and $v_0$ is the initial value of the dynamic threshold.

### 3.2.4 Improved recombination mechanism

The difference between ACS and ACO is that ACS does not require all ants to concentrate on a certain solution, but rather allows ants to search in the neighborhood of a certain solution. Based on this principle, a large number of different solutions will be obtained after each search. Although the vast majority of these solutions are not optimal, they probably contain a fraction of the optimal solution (Gao et al. 2020; Hou et al. 2022). And this part is not well utilized by ACS. In order to fully utilize the suboptimal solution, an improved recombination mechanism is proposed in this paper.

Figure 2 is a simple example to illustrate this mechanism. Figure 2a displays the effective paths obtained by the ant colony after completing a search. Among these paths, there are paths that contain parts of the global optimal path, as well as numerous paths with low quality. It is evident that recombination with inferior paths generally offers only limited improvement. To enhance search efficiency, the dynamic threshold is introduced to filter the recombined objects. Only paths that are larger than the current threshold are selected for recombination with the current



(a) Original search results (b) Filtered search results (c) The result after recombination

**Fig. 2** Example of the improved recombination mechanism

optimal path. Figure 2b shows the effective path after filtering. The red path is the current optimal path. Normally, the current optimal path is the optimal path that the algorithm searches for in the iterative process. By recombining the current optimal solution with the filtered effective path, the quality of the current optimal path is improved, as shown in Fig. 2c.

## 3.3 Post-processing smoothing approach

### 3.3.1 Improved turning point optimization method

After discretizing the environment by the grid method, the ant's search is constrained to eight directions. Given this characteristic, the paths obtained through planning usually have many unnecessary turns. These turns not only increase the energy cost of the ship in practice but even make the ship impossible to track. Therefore, an improved turning point optimization method is proposed to reduce the number of unnecessary turns in the path.

The proposed method differs from the traditional method in that it strikes a balance between path smoothness and energy consumption. The specific implementation process is as follows: For the original path, the set of all nodes as

$$N = \{N_1, N_2, \ldots, N_{n-1}, N_n\} \tag{20}$$

Starting from $N_1$, find the nodes in the set $N$ that can be directly connected and do not collide with the obstacle. Without loss of generality, let the set of these nodes as

$$N^* = \{N_2, N_3, \ldots, N_{i-1}, N_i\} \tag{21}$$

If it requires less energy to go directly from $N_1$ to $N_i$ than the original path, the intermediate points are removed, and $N_i$ is used as the starting point to continue optimizing until the end point is reached. Otherwise, the nodes are judged sequentially from far to near until a suitable node is found. Besides, if none of these nodes satisfy the condition or if $N^*$ is empty, the process is repeated using the next point $N_2$ as the starting point.

As shown in Fig. 3, Path 1 represents the result obtained using the proposed method, while Path 2 represents the result obtained using a common method (Ma et al. 2023). Although Path 2 is smoother and shorter compared to Path 1, it requires 3.44% and 4.05% more energy than the original path and Path 1, respectively. In contrast, Path 1 sacrifices some of its smoothness to save more energy.

### 3.3.2 Smoothing method using B-spline curve

To further improve the smoothness of the path to meet the navigation requirements of the ship, B-spline curve is introduced in this paper. Since the navigation space of the ship is discretized, the planned path consists of $n$ nodes, and these $n$ nodes can be used as control points $P_i$ of a B-spline curve. The formula of the B-spline curve is shown in Eq. (22).

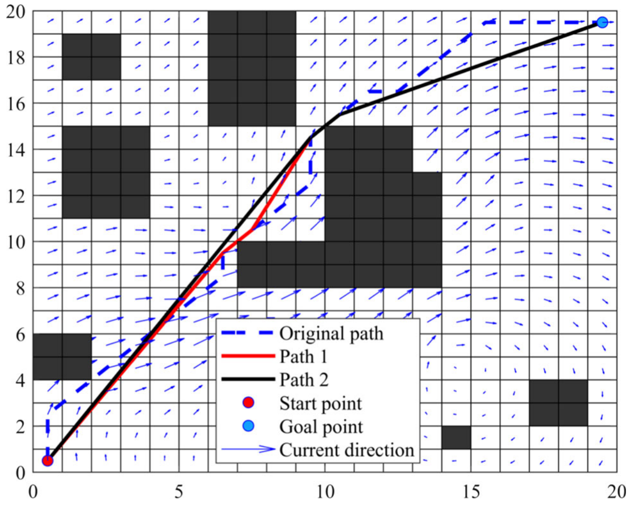


**Fig. 3** Example of the improved turning point optimization method

$$P(u) = \sum_{i=1}^{n+1} P_i N_{i,k}(u) \tag{22}$$

where $u$ is the knot vector, and $N_{i,k}(u)$ is the B-spline based function defined by the following DeBoor-Cox recursion formulas (Yang et al. 2019)

$$N_{i,1}(u) = \begin{cases} 1\,, & \mathrm{x}_i \le u < x_{i+1} \\ 0\,, & otherwise \end{cases} \tag{23}$$

$$N_{i,k}(u) = \frac{(u - x_i)N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}} \tag{24}$$

where $x_i$ is the knot values.

### 3.4 Process of adaptive ant colony system

After introducing the above improving strategies specifically, a complete adaptive ant colony system is formed within the following steps:

Step 1: Load the marine environment model for path planning and initialize the parameters.

Step 2: Calculate the control parameter for this iteration according to Eq. (13). Then, each ant selects the next node according to Eq. (14). After completing a move, the pheromone is locally updated. Repeat this process until the ant reaches the target point or falls into a deadlock state.

Step 3: Once all ants have completed their search, all the duplicated parts of the set for effective paths will be removed and the remaining parts will be evaluated according to Eq. (11). And the dynamic threshold at time $t$ is calculated using Eq. (19). The remaining part is filtered based on the dynamic threshold.

Step 4: The filtered effective paths are optimized based on an improved recombination mechanism and the current optimal path is updated.

Step 5: Update pheromones according to Eqs. (17) and (18). And the diversity of paths in this iteration is evaluated using Eqs. (9) and (10).

Step 6: Steps 2–5 will be repeated continuously until the stopping conditions are satisfied.

Step 7: The optimal path is post-processed based on the improved turning point optimization method and the smoothing method using B-spline curves.

Step 8: The optimal path is outputted.

The flowchart of the energy-saving path planning method is shown in Fig. 4.

## 4 Experimental results and discussions

To verify the effectiveness of AACS in solving the problem of energy-saving path planning for ships, a series of experiments are conducted in different environments. Considering that ant colony algorithm is a probabilistic optimization algorithm, to ensure the reliability of the experimental results, each group of experiments is repeated 60 times. The simulation configuration used in this work is the following: Windows 10 64-bit; processor, Intel(R) Core(TM) i7-12700KF clocked at 3.6 GHz; memory, 32 GB; simulation software, Matlab R2018b.

### 4.1 Parameter determination

The selection of parameters is a critical and indispensable step for ant colony algorithm, which directly affects the performance of the algorithm. However, since there is no accepted and effective method for parameter selection, most studies use statistical experiments to determine the parameters of the algorithm. Therefore, in order to determine the suitable parameters for AACS, comprehensive statistical experiments are performed in this section. For brevity, only the determination processes for $q_0$, $\varphi$ and $v_0$ are introduced in the environment of Fig. 5. Among them, $q_0$ is taken in the range of [0.1,0.9] with an interval of 0.1, $\varphi$ is taken in the range of [0.6,1] with an interval of 0.05, and $v_0$ is taken in the range of [0,1] with an interval of 0.05. Other critical parameters in the test are set as shown in Table 2, and they are determined in a similar process.

To adaptively adjust the proportion of exploration and exploitation, parameters $q_0$ and $\varphi$ are used in Eq. (13) to change the control parameter. Different values of $q_0$ and $\varphi$ are set up for combined testing, and the experimental results are shown in Fig. 6. According to Eq. (10), the maximum value of $H(t)$ is 1. Therefore, when $\varphi = 1$, the control parameter $\varepsilon(t) = q_0$; that is, the traditional state transition rule, which can be used as an experimental comparison benchmark. From the experimental results, it can be found that the value of $q_0$ has a large impact on both the energy consumption and the planning efficiency. In detail, as $q_0$ increases, the energy consumption usually decreases and then increases. And the magnitude of the decrease becomes larger rapidly with the increase of $\varphi$. In terms of planning efficiency, the efficiency tends to improve as $q_0$ increases, but there are small fluctuations in the process. Meanwhile, the overall efficiency tends to decrease as $\varphi$ increases. Moreover, compared with the traditional state transition rule, it can be found that both energy consumption and planning efficiency are improved after introducing the improvement. In general, the

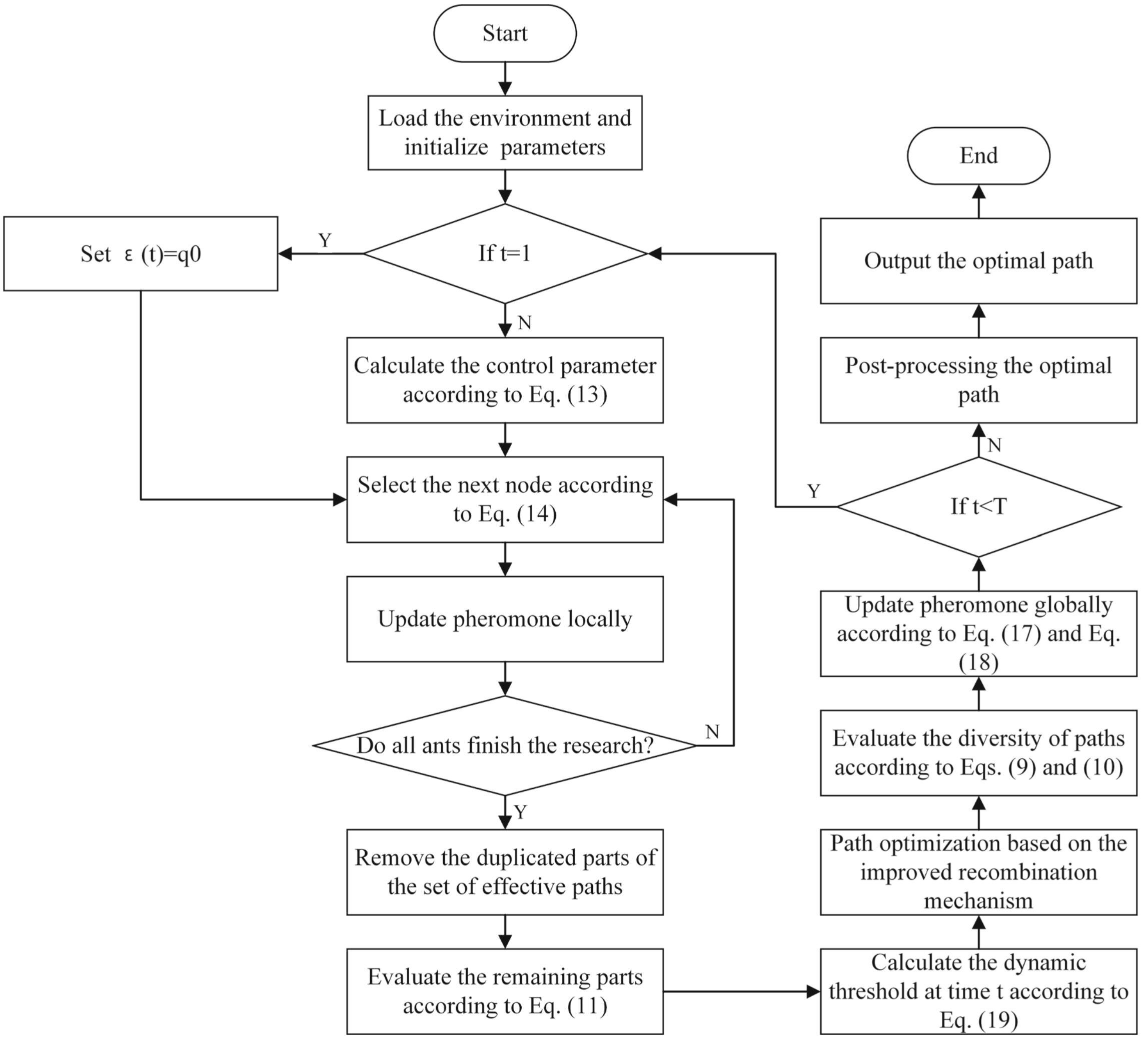

Start
Load the environment and initialize parameters
If t=1
Y
Set ε (t)=q0
N
Calculate the control parameter according to Eq. (13)
Select the next node according to Eq. (14)
Update pheromone locally
Do all ants finish the research?
N
Y
Remove the duplicated parts of the set of effective paths
Evaluate the remaining parts according to Eq. (11)
Calculate the dynamic threshold at time t according to Eq. (19)
Path optimization based on the improved recombination mechanism
Evaluate the diversity of paths according to Eqs. (9) and (10)
Update pheromone globally according to Eq. (17) and Eq. (18)
If t<T
Y
N
Post-processing the optimal path
Output the optimal path
End

**Fig. 4** Flowchart of AACS

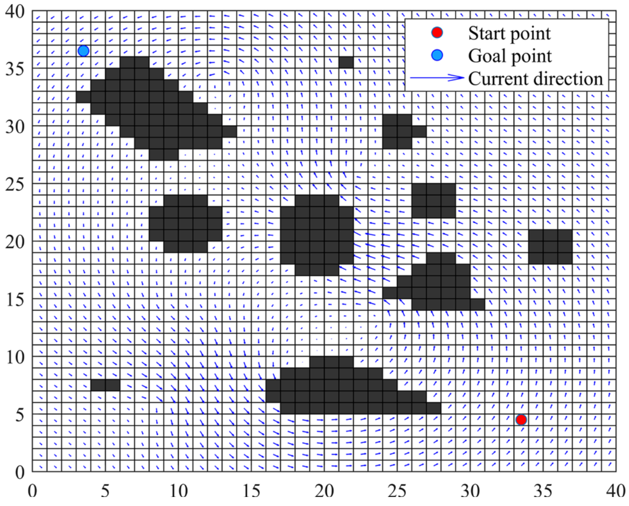


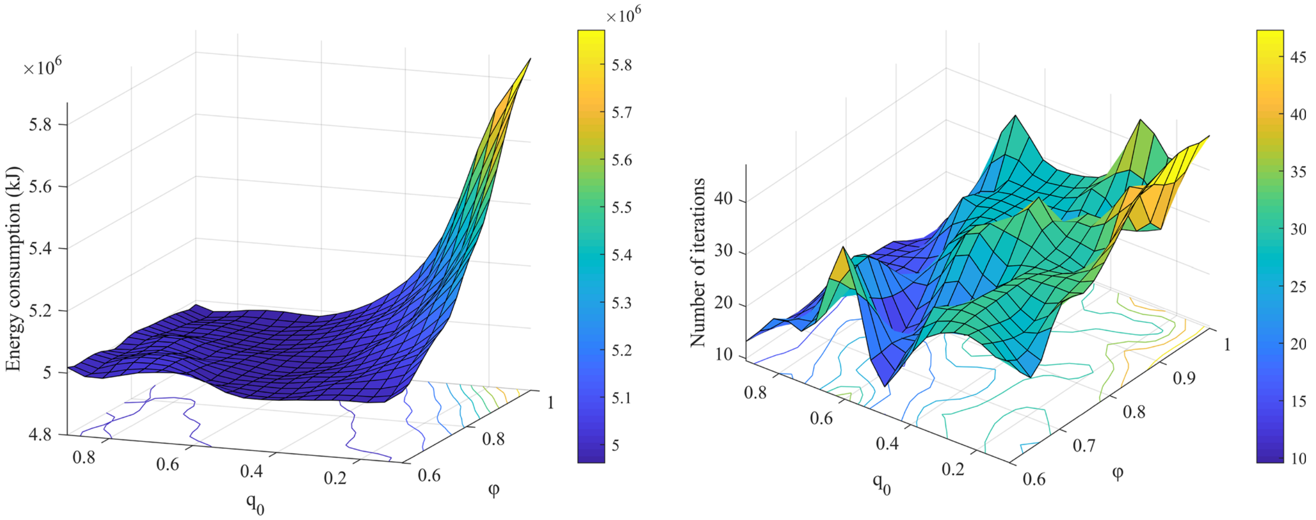
**Fig. 5** Simulation environment in statistical experiment

recommended value ranges based on the present work are $q_0 \in [0.4, 0.6]$ and $\varphi \in [0.65, 0.75]$.

The optimal number of ants for contribute to the global updating rule has been a widely researched topic. In ant system all the ants contribute, while in the ACS only the best one does. Obviously, there are intermediate possibilities. Therefore, the parameter $v_0$ is used in Eq. (19) to dynamically control the number of these ants. The experimental results of $v_0$ are shown in Fig. 7. On the whole, the performance of the algorithm improves as $v_0$ decreases. When $v_0 \leq 0.6$, the performance of the algorithm varies less with $v_0$. However, when $v_0 > 0.6$, the performance of the algorithm degrades rapidly with the increase of $v_0$. Based on the above, it can be found that the number of ants contributing to the global updating rule within a reasonable range should be set more in the early stage and slowly reduced with iterations. Therefore, the recommended value ranges is $v_0 \in [0.05, 0.25]$.

**Table 2** Parameters setting of AACS in statistical experiment

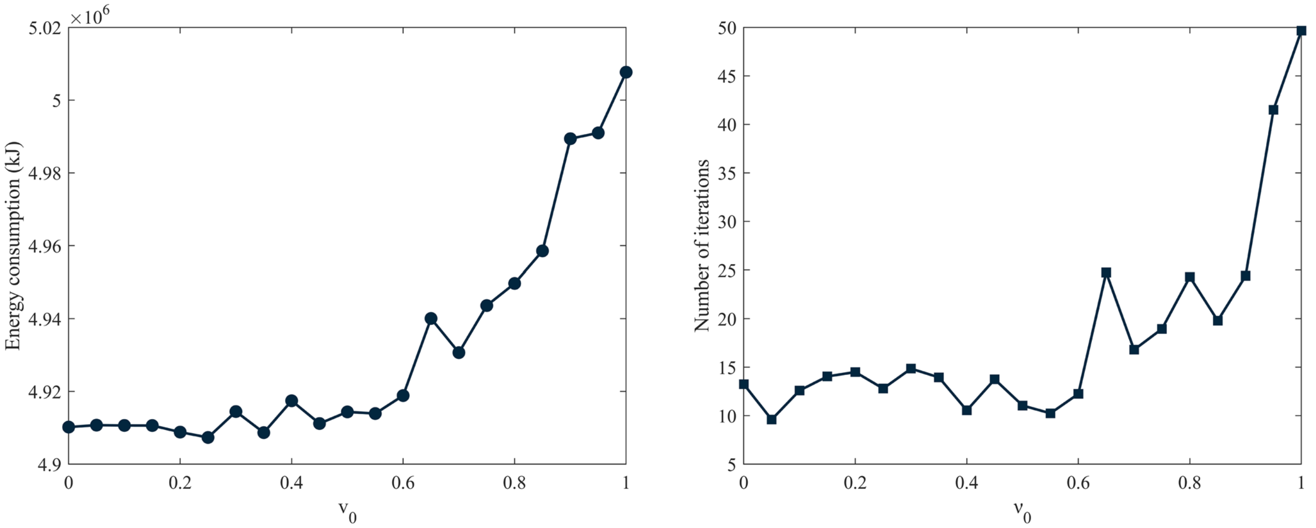
| Parameter | $m$ | $T$ | $Q$ | $\alpha$ | $\beta$ | $\rho$ | $\rho_{local}$ | $k_1$ | $k_2$ | $k_3$ | $k$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Value | 50 | 100 | 50 | 1.1 | 6 | 0.2 | 0.2 | 125 | 1 | 0 | 0.00001 |



(a) Energy consumption varying with $q_0$ and $\varphi$

(b) Number of iterations varying with $q_0$ and $\varphi$

**Fig. 6** Results of statistical experiments on $q_0$ and $\varphi$



(a) Energy consumption varying with $v_0$

(b) Number of iterations varying with $v_0$

**Fig. 7** Results of statistical experiments on $v_0$

## 4.2 Experimental simulation in static environment

To verify the adaptability and effectiveness of AACS in static environments, three different maps are created for simulation, and the size of a unit grid is 1 × 1 km. Since the influence of the marine environment is not taken into account, the path length is the most important factor in determining the energy consumption of the ship. In this section, the path length is used as the energy consumption index to evaluate the results obtained by all algorithms. In addition, two novel improved ACO which are IACA (Luo et al. 2020) and IAACO (Miao et al. 2021), A* algorithm, ACS, PSO, and an improved pigeon-inspired optimization (IPIO; Xia et al. 2022) are used for comparison. The parameters of each algorithm are set as follows:

- AACS: $m = 50$, $T = 100$, $Q = 50$, $\alpha = 1.1$, $\beta = 6$, $\rho = 0.2$, $\rho_{local} = 0.2$, $k_1 = 0$, $k_2 = 1$, $k_3 = 1$, $k = 1$, $q_0 = 0.5$, $\varphi = 0.75$, $v_0 = 0.25$.
- IAACO: $m = 50$, $T = 100$, $Q = 2.5$, $\alpha = 1$, $\beta = 7$, Environment 1: $\delta_0 = 0.15$, $\kappa = 0.2$; Environment 2: $\delta_0 = 0.5$, $\kappa = 0.15$; Environment 3: $\delta_0 = 0.5$, $\kappa = 0.2$.
- IACA: $m = 50$, $T = 100$, $Q = 100$, $\alpha = 1.1$, $\beta = 7$, $\rho = 0.2$, $\lambda = 0.2$, Environment 1: $q_0 = 0.3$, $N_0 = 5$; Environment 2: $q_0 = 0.3$, $N_0 = 8$; Environment 3: $q_0 = 0.7$, $N_0 = 8$;
- ACS: $m = 50$, $T = 100$, $Q = 50$, $\alpha = 1.1$, $\beta = 6$, $\rho = 0.2$, $\rho_{local} = 0.2$, $q_0 = 0.7$.
- PSO: $m = 100$, $T = 100$, $c_1 = 2$, $c_2 = 2$, $w = 0.9$.
- IPIO: $m = 100$, $Nc_1 = 75$, $Nc_2 = 100$, $R = 0.2$, $F = 0.5$, $\gamma = 0.5$, $CR = 0.3$.

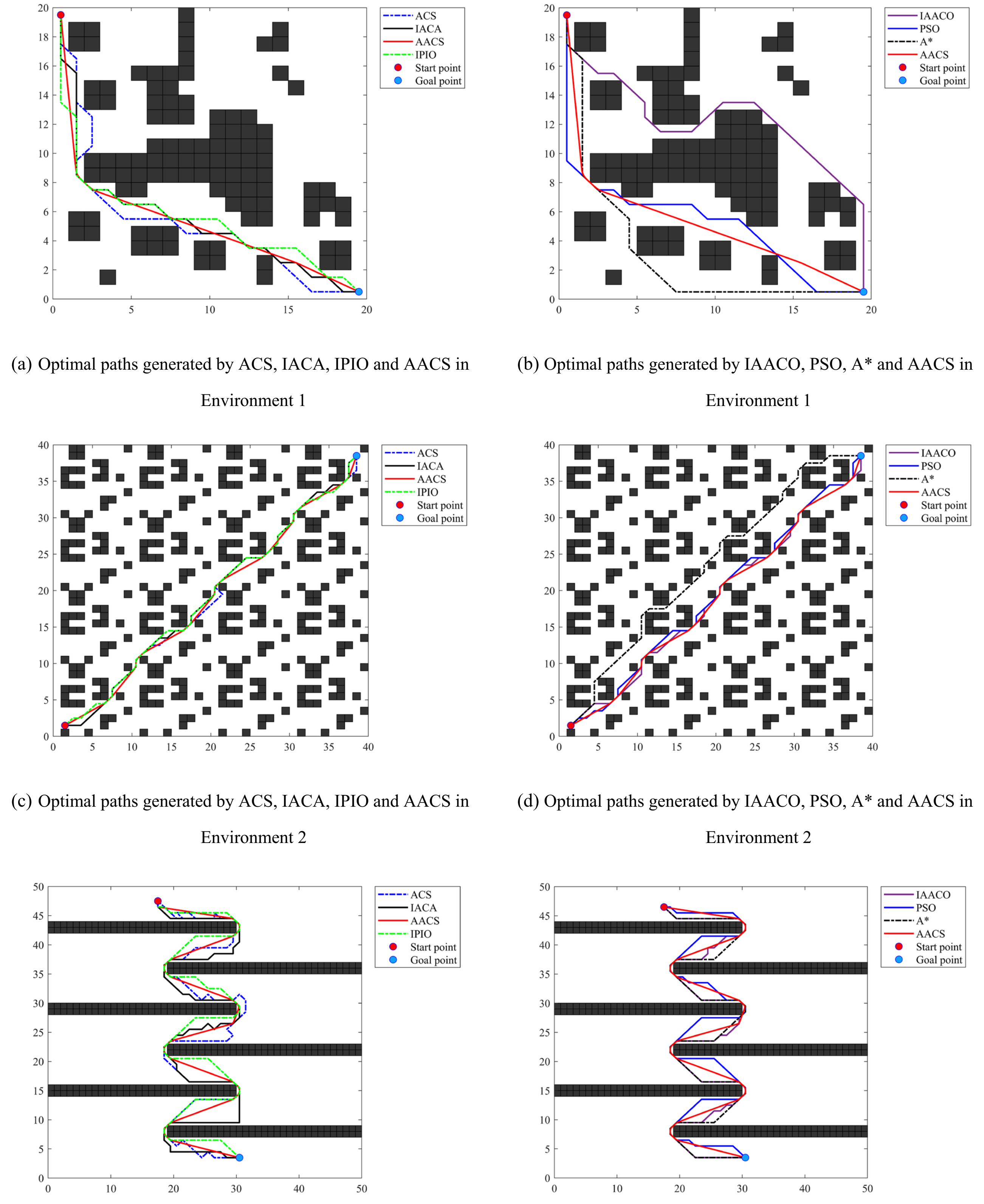
The simulation results are presented in Table 3, which primarily compares the optimal results and average results obtained by each algorithm. Additionally, the convergence performance of each variant of ACO is also compared to validate the effectiveness of the improvement. Figure 8 displays the optimal paths obtained by each algorithm in three different environments. The box-whisker plot of the path length for each algorithm is shown in Fig. 9.

According to Table 3 and Fig. 9, the results achieved by AACS are superior to other methods in terms of both optimal and average results. This indicates that AACS exhibits better planning performance in static environments. And as shown in Fig. 8, it is clear that the path produced by the proposed AACS has less number of turns and shorter length. Different from other comparison algorithms, the A* algorithm's path construction process is not random as it consistently chooses the node with the lowest cost in the Open list. In addition to the A* algorithm, AACS and IPIO demonstrate better stability than other methods in various environments. In terms of convergence speed, ACS requires the least number of iterations in Environment 1, while AACS has a faster convergence rate than other ACO variants in the remaining environment. However, ACS always falls into the local optimum in Environment 1. Although the local updating rule in ACS can effectively improve the ability to jump out of the local optimum, the objects of the global updating rule in ACS are too limited. In contrast, the other three ACO variants have more update objects in addition to the optimal solution in their global updating rule, which results in better global search capabilities. But IACA and IAACO update the pheromones of all complete paths, which can lead to

**Table 3** Comparison of simulation results in static environment

| Environment | Algorithm | Optimal path length (km) | Average of path length (km) | Average number of iterations |
|---|---|---|---|---|
| # 1 | A* | 33.899 | 33.899 | – |
| | PSO | 32.728 | 32.906 | – |
| | IPIO | 32.728 | 32.826 | – |
| | ACS | 33.556 | 33.759 | **14.950** |
| | IAACO | 32.728 | 33.777 | 29.017 |
| | IACA | 32.728 | 33.263 | 29.910 |
| | AACS | **30.814** | **30.965** | 21.728 |
| # 2 | A* | 58.184 | 58.184 | – |
| | PSO | 57.012 | 58.175 | – |
| | IPIO | 57.012 | 57.709 | – |
| | ACS | 57.012 | 58.171 | 66.883 |
| | IAACO | 57.012 | 57.684 | 53.283 |
| | IACA | 57.012 | 57.523 | 39.810 |
| | AACS | **54.122** | **54.179** | **9.525** |
| # 3 | A* | 106.326 | 106.326 | – |
| | PSO | 106.326 | 108.297 | – |
| | IPIO | 106.326 | 107.021 | – |
| | ACS | 122.225 | 140.462 | 57.100 |
| | IAACO | 107.497 | 115.117 | 78.133 |
| | IACA | 114.669 | 130.601 | 92.450 |
| | AACS | **100.386** | **104.863** | **32.433** |

The best result for each section is highlighted in bold



(a) Optimal paths generated by ACS, IACA, IPIO and AACS in Environment 1

(b) Optimal paths generated by IAACO, PSO, A* and AACS in Environment 1

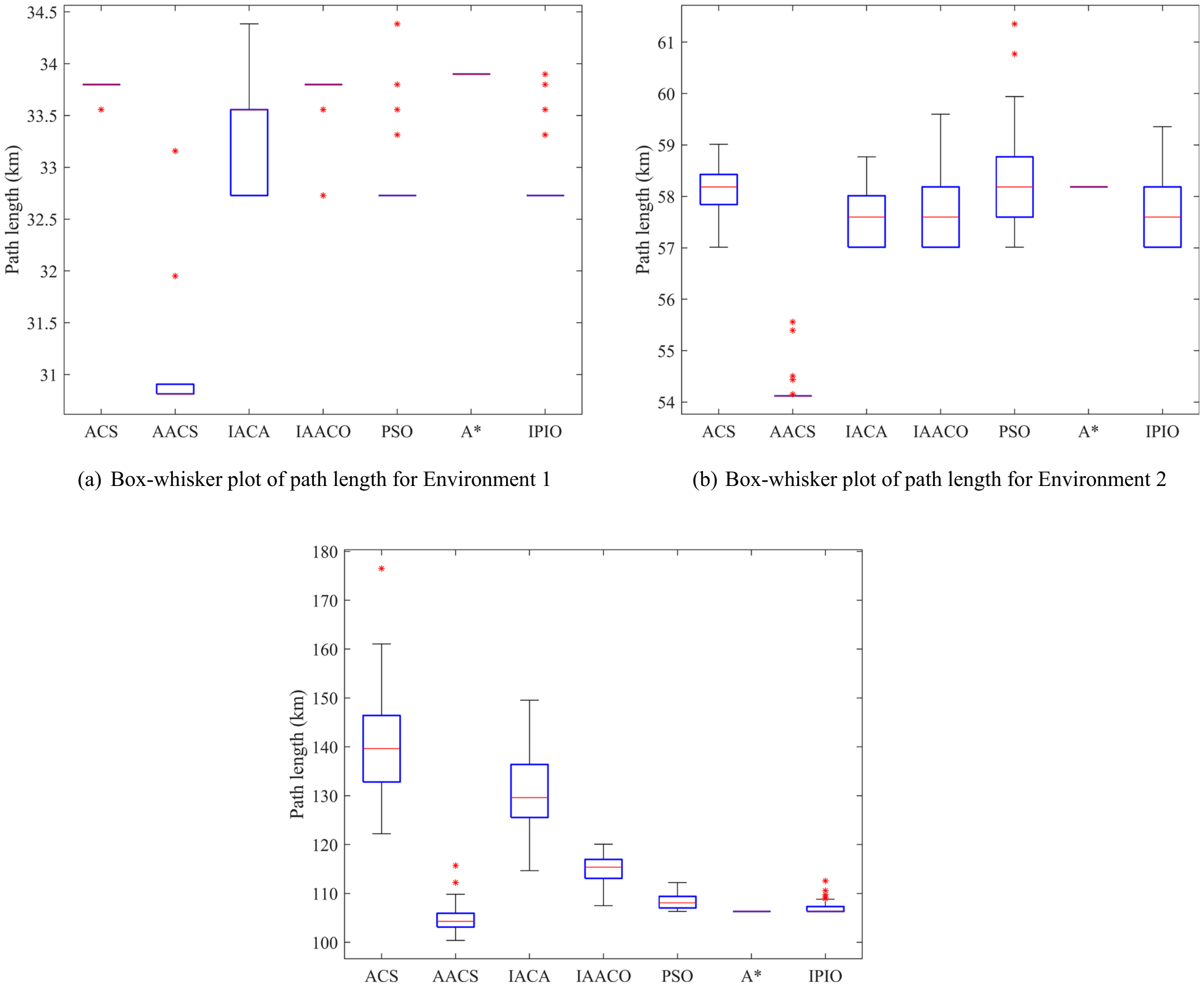
(c) Optimal paths generated by ACS, IACA, IPIO and AACS in Environment 2

(d) Optimal paths generated by IAACO, PSO, A* and AACS in Environment 2

(e) Optimal paths generated by ACS, IACA, IPIO and AACS in Environment 3

(f) Optimal paths generated by IAACO, PSO, A* and AACS in Environment 3

**Fig. 8** Comparison of path planning results in static environment



(a) Box-whisker plot of path length for Environment 1

(b) Box-whisker plot of path length for Environment 2

(c) Box-whisker plot of path length for Environment 3

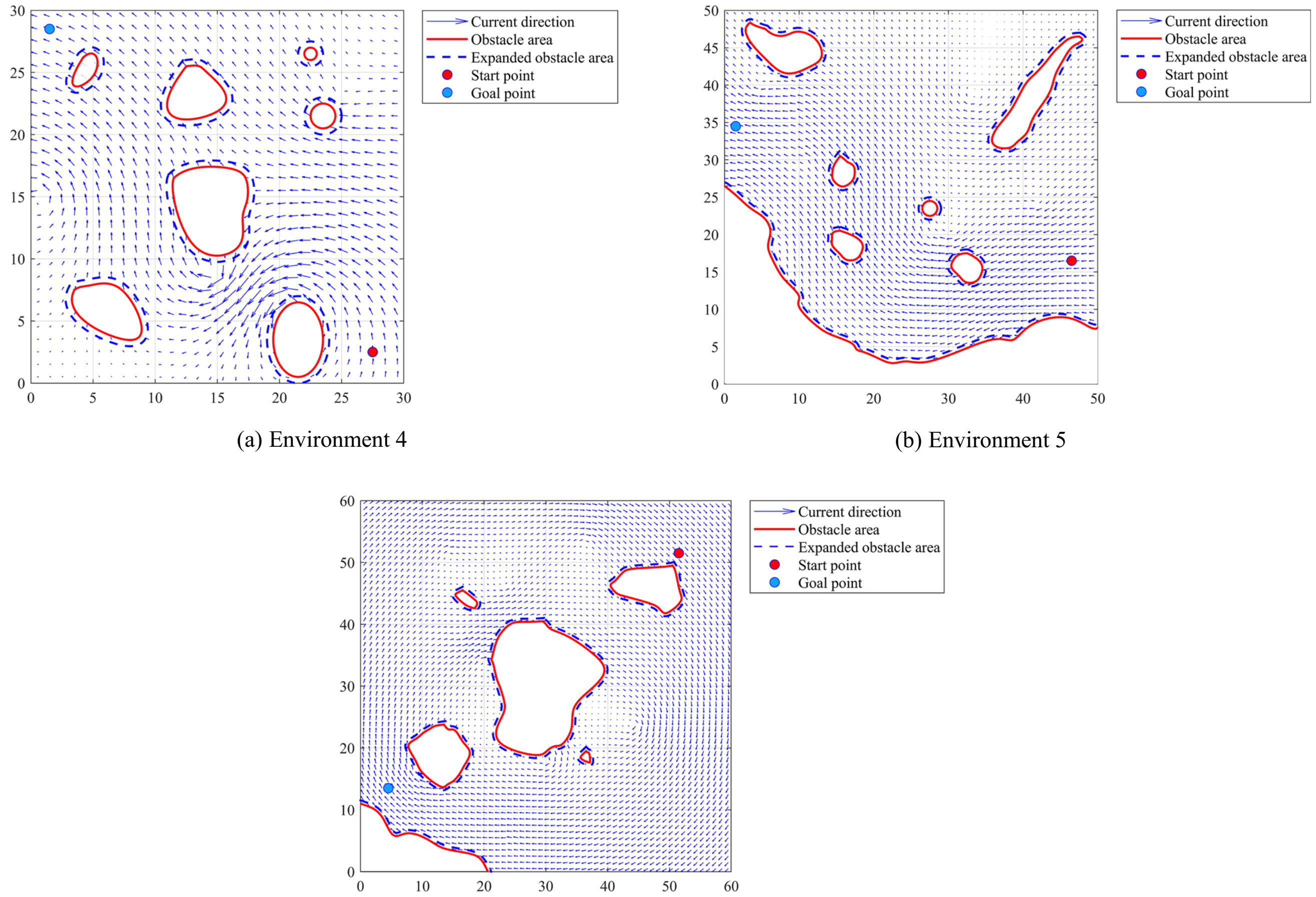
**Fig. 9** Comparison of the solution quality in static environment

slow convergence in complex scenarios such as Environment 3.

### 4.3 Experimental simulation in virtual marine environment

In this section, the performance of AACS is further validated in three scenarios that consider environmental influences. The three scenarios are shown in Fig. 10. Moreover, ACS, A* algorithm, PSO and IPIO are used for comparison. The specific configuration of different scenarios is shown in Table 4. And the speed of the ship is set to 8 m/s. The parameters that need to be varied in the simulation experiment are shown in Table 5.

Figure 11 displays the best path generated by each method. As shown in Fig. 11, the energy-saving path considering the environmental impact is more inclined to follow or reverse the direction of the environmental load to save energy. Furthermore, the implementation of the post-processing smoothing approach by AACS results in a path with fewer turning points and smoother transitions at those points. This reduces the difficulty of the ship tracking path. Based on Fig. 12 and Table 6, it can be observed that the optimal path planned by AACS consumes the least energy, whereas the optimal path planned by ACS consumes the most energy. Moreover, except for the A* algorithm, AACS exhibits the best stability in different environments. Compared with other comparison algorithms, ACS has the worst average result in Environment 4, while PSO has the worst stability in the remaining environment. This comparison result not only proves the superiority of AACS compared with other algorithms, but also verifies the



(a) Environment 4

(b) Environment 5

(c) Environment 6

**Fig. 10** Virtual marine environments for the simulations

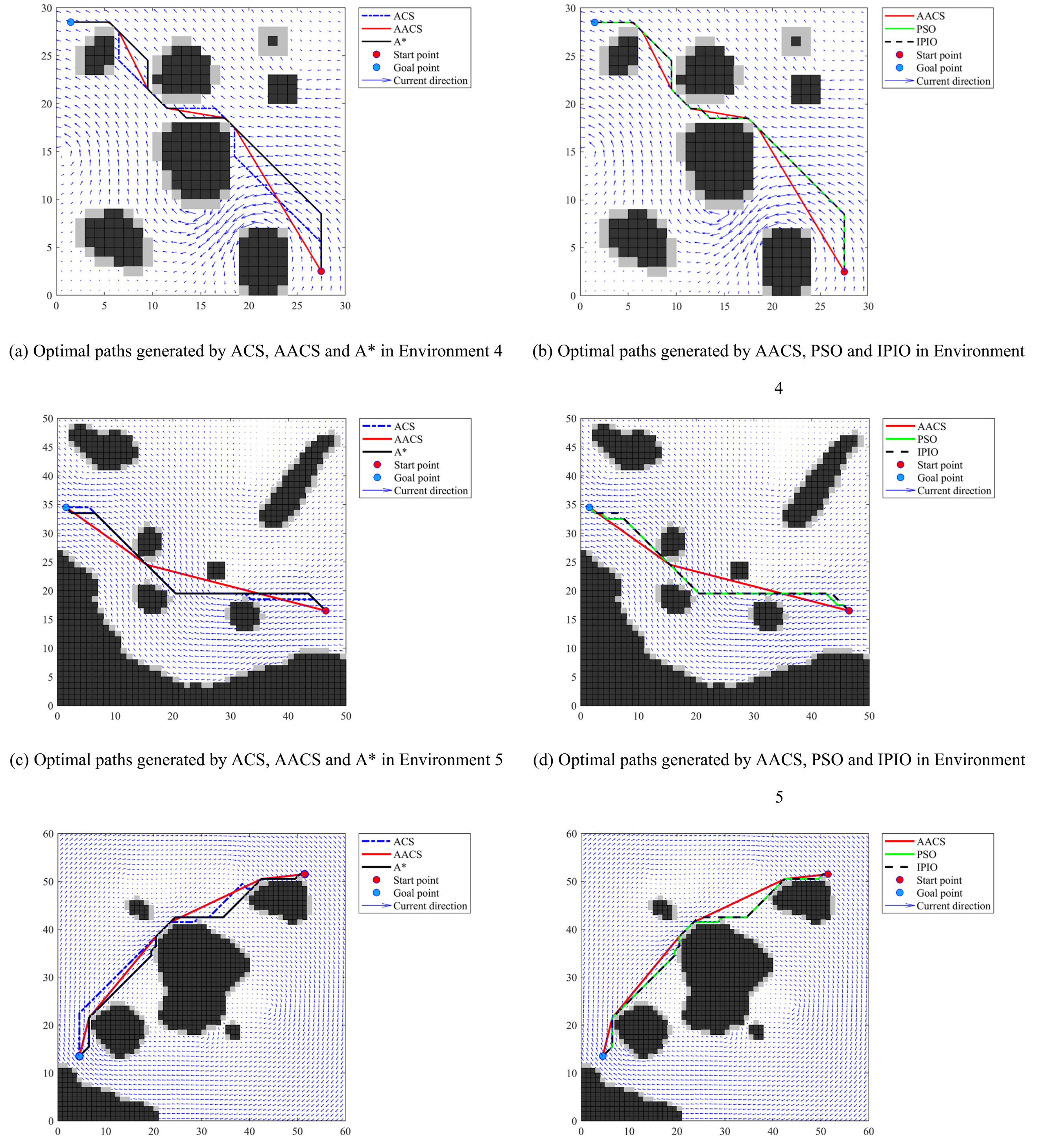
**Table 4** The configurations of virtual marine environment

| Environment | Current speed (m/s) | Current direction (deg) | Wind speed (m/s) | Wind direction (deg) | Wave direction (deg) | Wave length (m) | Amplitude of waves (m) | Wave speed (m/s) |
|---|---|---|---|---|---|---|---|---|
| # 4 | 0.08 | 2D geostrophic current | 5 | 30 | 40 | 70 | 0.5 | 8 |
| # 5 | 0.08 | 2D geostrophic current | 5 | 170 | 160 | 70 | 0.5 | 8 |
| # 6 | 0.08 | 2D geostrophic current | 5 | 270 | 270 | 70 | 0.5 | 8 |

**Table 5** Parameters setting in virtual marine environment

| Environment | $k_1$ | $k_2$ | $k_3$ | $k$ |
|---|---|---|---|---|
| # 4 | 200 | 1 | 0 | 0.000004 |
| # 5 | 1000 | 1 | 0 | 0.000005 |
| # 6 | 100 | 1 | 0 | 0.000003 |

effectiveness of the proposed improvement strategy. In addition, AACS requires fewer iterations to obtain the optimal solution in these environments compared to ACS. The main reason is that the sub-optimal solutions are effectively utilized by the adaptive global updating rule as well as the improved recombination mechanism.



(a) Optimal paths generated by ACS, AACS and A* in Environment 4

(b) Optimal paths generated by AACS, PSO and IPIO in Environment 4

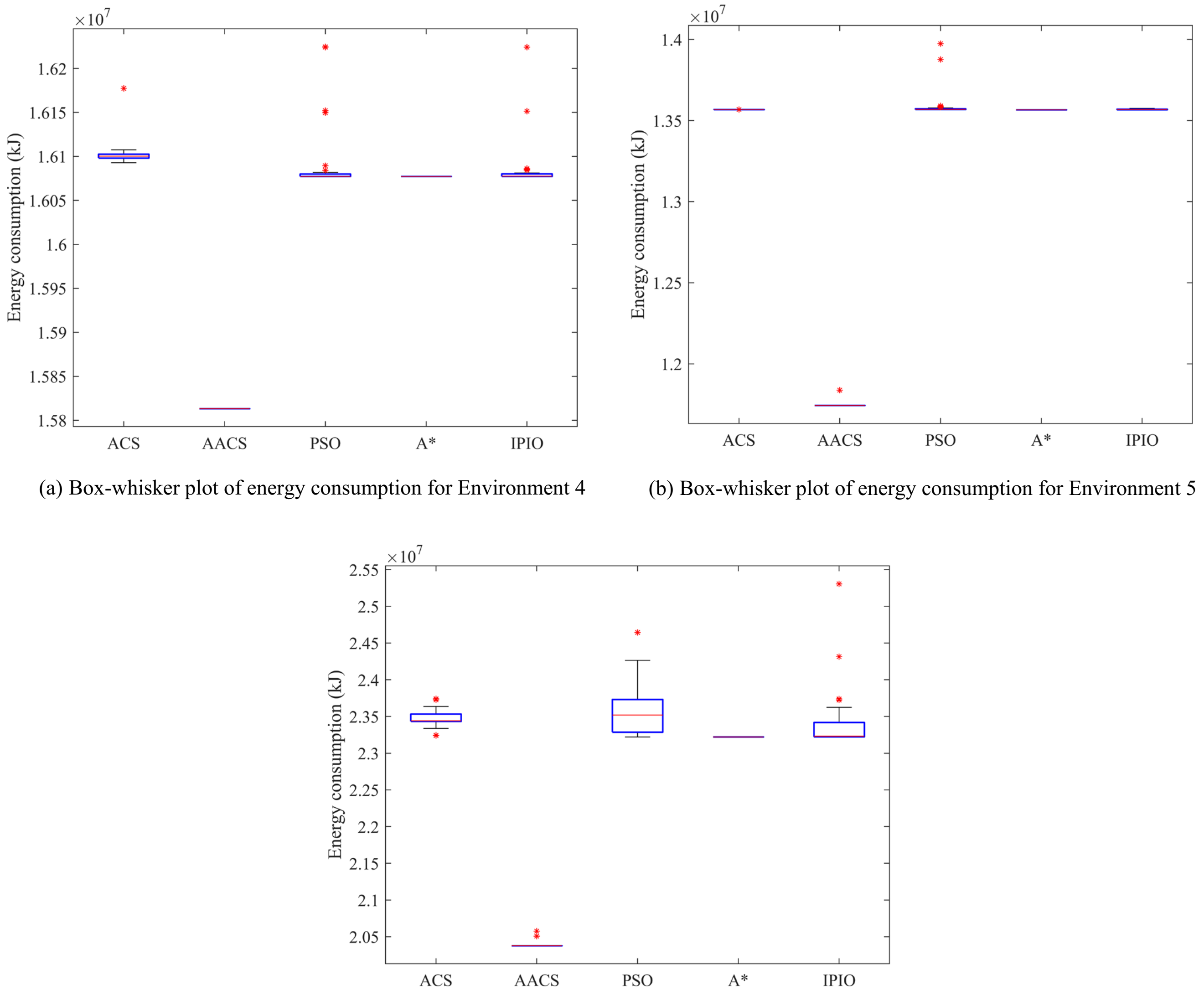
(c) Optimal paths generated by ACS, AACS and A* in Environment 5

(d) Optimal paths generated by AACS, PSO and IPIO in Environment 5

(e) Optimal paths generated by ACS, AACS and A* in Environment 6

(f) Optimal paths generated by AACS, PSO and IPIO in Environment 6

**Fig. 11** Comparison of path planning results in virtual marine environment



(a) Box-whisker plot of energy consumption for Environment 4

(b) Box-whisker plot of energy consumption for Environment 5

(c) Box-whisker plot of energy consumption for Environment 6

**Fig. 12** Comparison of the solution quality in virtual marine environment

### 4.4 Experimental simulation in actul environment

In this section, a public database of Danish ferry is used to further verify the feasibility of the proposed method. The database contains data related to the case ship between February 16th and April 12th, 2010, spanning almost two months (Petersen et al. 2012). These data are publicly available at http://cogsys.dtu.dk/propulsionmodelling/. To verify the performance of our method in the actual environment, 11 different paths in March are selected for comparison. The sea area in which the case ship operates is discretized into 60 × 60 grids with each grid having a length of 1.053 km. For the safety of navigation, the unnavigable area in the sea area is extended by 200 m. In addition, it is assumed that the ship sailed at a constant speed and the ship speed was obtained by averaging the data. As the main focus of this work is on saving energy by taking advantage of environmental factors during the voyage, only the part of the path that is in the open sea is retained and evaluated based on the energy consumption estimation model.

To differentiate between AACS with post-processing smoothing approach introduced and AACS without post-processing smoothing approach, we will refer to the former as AACS*. Figure 13a and b display the comparison between the planned path by AACS* and the actual path taken. As the redundant nodes in the path are removed using the post-processing smoothing approach, the path planned by AACS* is closer to the real path and can better meet the motion characteristics of the ship. The results depicted in Fig. 13c indicate that the optimal paths planned

**Table 6** Comparison of simulation results in virtual marine environment

| Environment | Algorithm | Optimal path energy consumption (kJ) | Average of path energy consumption (kJ) | Average number of iterations |
|---|---|---|---|---|
| # 4 | A* | 16,077,208.159 | 16,077,208.159 | – |
| | PSO | 16,077,208.159 | 16,092,952.185 | – |
| | IPIO | 16,077,208.159 | 16,087,195.109 | – |
| | ACS | 16,092,826.930 | 16,101,573.110 | 62.417 |
| | AACS | **15,813,263.762** | **15,813,263.762** | **16.783** |
| # 5 | A* | 13,565,727.638 | 13,565,727.638 | – |
| | PSO | 13,565,727.638 | 13,582,472.826 | – |
| | IPIO | 13,565,727.638 | 13,568,311.478 | – |
| | ACS | 13,566,382.920 | 13,567,112.460 | 66.650 |
| | AACS | **11,743,353.700** | **11,744,926.180** | **12.017** |
| # 6 | A* | 23,220,567.245 | 23,220,567.245 | – |
| | PSO | 23,220,567.245 | 23,554,860.320 | – |
| | IPIO | 23,220,567.245 | 23,370,167.395 | – |
| | ACS | 23,238,631.370 | 23,480,726.880 | 62.483 |
| | AACS | **20,377,078.956** | **20,395,486.194** | **18.883** |

The best result for each section is highlighted in bold

by AACS in most scenarios without implementing the post-processing smoothing approach consume more energy than the actual paths. The main reason for this phenomenon is the low resolution of the grid map. Although increasing the resolution of the grid map is an effective way to solve this problem, it results in a lot of additional time costs. In contrast, the post-processing smoothing approach is more appropriate. As can be seen in Fig. 13, the optimal path after using the post-processing smoothing approach requires approximately 3.21% less energy than the actual path of the case ship.

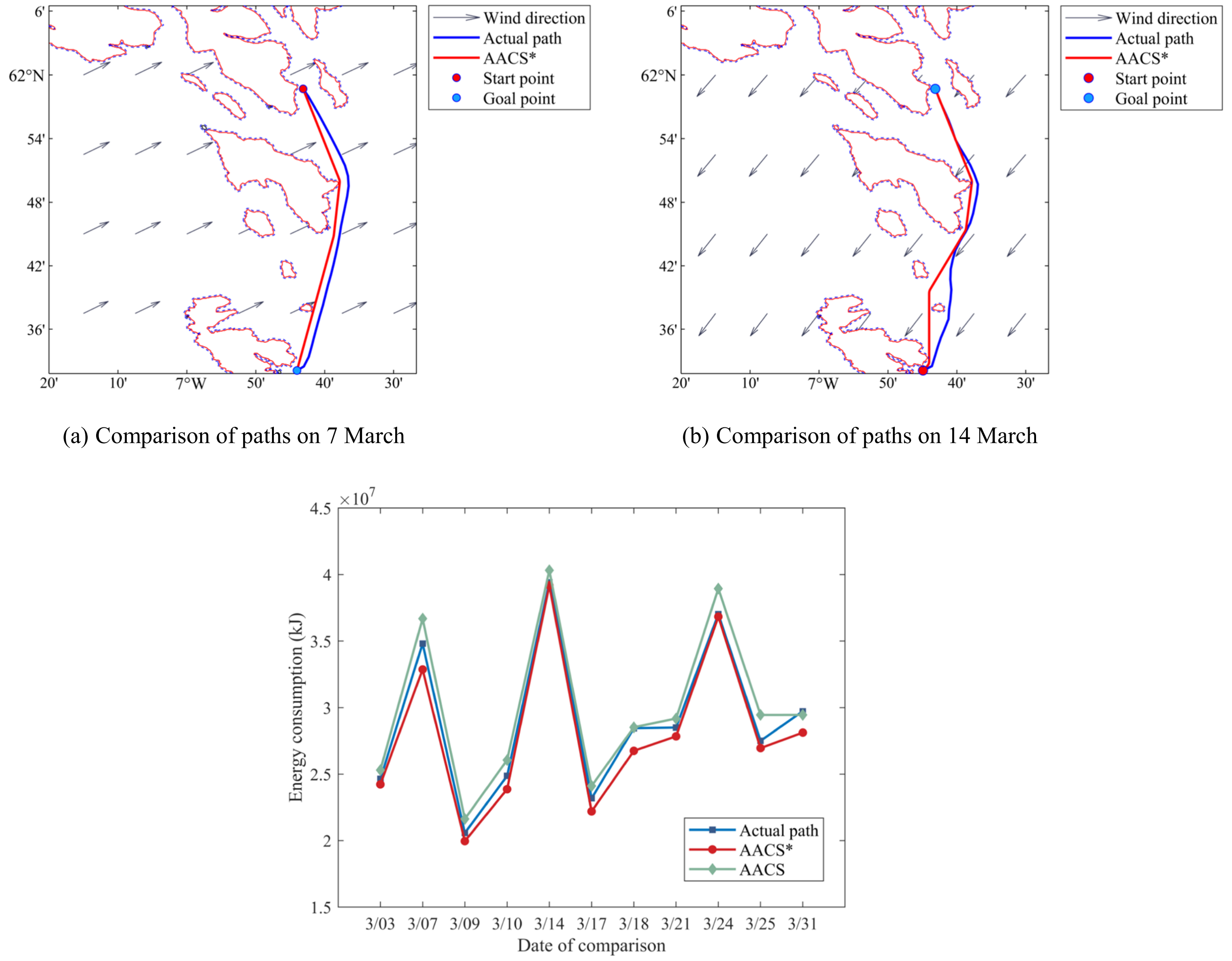
## 5 Conclusion

To adapt to the energy consumption requirements of ship technology development on the path, this paper conducts an insightful study on the problem of ship energy-saving path planning. By designing a ship energy consumption estimation model and introducing it into the heuristic information function of the algorithm, the proposed method can better exploit environmental factors to save energy. Also, this improved method not only evaluates the optimization process but also applies the evaluation results to the various optimization stages of the algorithm. In the stage of path construction, an adaptive state transition rule is designed according to the diversity of solutions, which effectively improves the search performance of the algorithm. During the pheromone update stage, the number of ants contributing to global updating is dynamically adjusted. Furthermore, an improved recombination mechanism is proposed to improve the planning efficiency of the algorithm. In order to further smooth the path and improve its quality, an improved turning point optimization method and the smoothing method using B-spline curves are introduced. Finally, simulation experiments have been conducted in different scenarios. The simulation results not only demonstrate the superiority of the proposed method over comparison algorithms but also verify its feasibility in practical applications. However, it needs to be acknowledged that this study still has some limitations that should be addressed in future work.

(1) This paper primarily focuses on energy consumption in the presence of stable environmental disturbances. However, during long-term navigation missions, changes in the environment over time cannot be ignored. Thus, future research should also consider the problem of energy consumption optimization in time-varying environments.

(2) By introducing path post-processing smoothing technology, the smoothness and continuity of the path are guaranteed. In order to further consider the dynamic characteristics of ships, ship dynamics models should be introduced in future studies.

(3) The lack of experiments in the study limits its persuasiveness when addressing practical application problems. In future research, we aim to establish an experimental platform and conduct experimental verification.



(a) Comparison of paths on 7 March

(b) Comparison of paths on 14 March

(c) Comparison of the simulation results

**Fig. 13** Results of experimental simulation in actual environment

**Funding** This research was supported by Zhejiang Provincial Natural Science Foundation of China (No. LY20E090002), Zhoushan City Science and Technology Planned Project (No. 2018C21018), and Zhejiang University Students' Science and Technology Innovation Activity Plan (No. 2022R411C053).

**Data availability** The datasets generated during the current study are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** The authors declare that they have no competing interests.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## References

Barreiro J, Zaragoza S, Diaz-Casas V (2022) Review of ship energy efficiency. Ocean Eng 257:111594. https://doi.org/10.1016/j.oceaneng.2022.111594

Chen Y, Bai G, Zhan Y et al (2021) Path planning and obstacle avoiding of the USV based on improved ACO-APF hybrid algorithm with adaptive early-warning. IEEE Access 9:40728–40742. https://doi.org/10.1109/ACCESS.2021.3062375

Dong L, Li J, Xia W, Yuan Q (2021) Double ant colony algorithm based on dynamic feedback for energy-saving route planning for ships. Soft Comput 25:5021–5035. https://doi.org/10.1007/s00500-021-05683-8

Dorigo M, Gambardella L (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66. https://doi.org/10.1109/4235.585892

Du W, Li Y, Zhang G et al (2022a) Energy saving method for ship weather routing optimization. Ocean Eng 258:111771. https://doi.org/10.1016/j.oceaneng.2022.111771

Du W, Li Y, Zhang G et al (2022b) Ship weather routing optimization based on improved fractional order particle swarm optimization.

Ocean Eng 248:110680. https://doi.org/10.1016/j.oceaneng.2022.110680

Duan H, Qiao P (2014) Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. Int J Intell Comput Cybern 7:24–37. https://doi.org/10.1108/IJICC-02-2014-0005

Gao W, Tang Q, Ye B et al (2020) An enhanced heuristic ant colony optimization for mobile robot path planning. Soft Comput 24:6139–6150. https://doi.org/10.1007/s00500-020-04749-3

Gu Q, Zhen R, Liu J, Li C (2023) An improved RRT algorithm based on prior AIS information and DP compression for ship path planning. Ocean Eng 279:114595. https://doi.org/10.1016/j.oceaneng.2023.114595

Han Q, Yang X, Song H, Du W (2022) Multi-objective ship path planning using non-dominant relationship-based WOA in marine meteorological environment. Ocean Eng 266:112862. https://doi.org/10.1016/j.oceaneng.2022.112862

He Y, Fan X (2021) Application of improved ant colony optimization in robot path planning. Comput Eng Appl 57(16):276–282

Hou W, Xiong Z, Wang C, Chen H (2022) Enhanced ant colony algorithm with communication mechanism for mobile robot path planning. Robot Auton Syst 148:103949. https://doi.org/10.1016/j.robot.2021.103949

Lee T, Chung H, Myung H (2011) Multi-resolution path planning for marine surface vehicle considering environmental effects. In: OCEANS 2011 IEEE - Spain, Santander, Spain, 2011, pp 1–9. https://doi.org/10.1109/Oceans-Spain.2011.6003669

Lee T, Kim H, Chung H, Bang Y, Myung H (2015) Energy efficient path planning for a marine surface vehicle considering heading angle. Ocean Eng 118–131. https://doi.org/10.1016/j.oceaneng.2015.07.030

Liu J, Yang J, Liu H et al (2017) An improved ant colony algorithm for robot path planning. Soft Comput 21:5829–5839. https://doi.org/10.1007/s00500-016-2161-7

Liu C, Wu L, Xiao W et al (2023) An improved heuristic mechanism ant colony optimization algorithm for solving path planning. Knowl-Based Syst 271:110540. https://doi.org/10.1016/j.knosys.2023.110540

Luo Q, Wang H, Zheng Y, He J (2020) Research on path planning of mobile robot based on improved ant colony algorithm. Neural Comput Appl 32:1555–1566. https://doi.org/10.1007/s00521-019-04172-2

Ma X, Li C, Tan Y, Mei H (2023) Research of artificial fish swarm optimization algorithm of path planning for mobile robot. Mech Sci Technol Aerosp Eng. https://doi.org/10.13433/j.cnki.1003-8728.20230103

Miao C, Chen G, Yan C, Wu Y (2021) Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. Comput Ind Eng 156:107230. https://doi.org/10.1016/j.cie.2021.107230

Niu H, Ji Z, Savvaris A, Tsourdos A (2020) Energy efficient path planning for Unmanned Surface Vehicle in spatially-temporally variant environment. Ocean Eng 196:106766. https://doi.org/10.1016/j.oceaneng.2019.106766

Öztürk Ü, Akdağ M, Ayabakan T (2022) A review of path planning algorithms in maritime autonomous surface ships: navigation safety perspective. Ocean Eng 251:111010. https://doi.org/10.1016/j.oceaneng.2022.111010

Petersen JP, Winther O, Jacobsen DJ (2012) A machine-learning approach to predict main energy consumption under realistic operational conditions. Ship Technol Res 59:64–72. https://doi.org/10.1179/str.2012.59.1.007

Qian X (2018) Research on nonlinear model predictive control for dynamic positioning ship. Dissertation, Dalian Maritime University.

Song R, Liu Y, Bucknall R (2019) Smoothed A* algorithm for practical unmanned surface vehicle path planning. Appl Ocean Res 83:9–20. https://doi.org/10.1016/j.apor.2018.12.001

Stodola P, Otřísal P, Hasilová K (2022) Adaptive Ant Colony Optimization with node clustering applied to the travelling salesman problem. Swarm Evol Comput 70:101056. https://doi.org/10.1016/j.swevo.2022.101056

Sun C (2021) Research on cooperative positioning control of multiple dynamic positioning vessels. Dissertation, Harbin Engineering University

Tang X, Xin S (2022) Improved ant colony algorithm for mobile robot path planning. Comput Eng Appl 58(5):287–295

Tang J, Liu G, Pan Q (2021) A review on representative swarm intelligence algorithms for solving optimization problems: applications and trends. IEEE/CAA J Autom Sinica 8:1627–1643. https://doi.org/10.1109/JAS.2021.1004129

Wang L, Zhang Z, Zhu Q, Ma S (2020) Ship route planning based on double-cycling genetic algorithm considering ship maneuverability constraint. IEEE Access 8:190746–190759. https://doi.org/10.1109/ACCESS.2020.3031739

Wu L, Huang X, Cui J et al (2023) Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot. Expert Syst Appl 215:119410. https://doi.org/10.1016/j.eswa.2022.119410

Xia G, Han Z, Zhao B et al (2019) Global path planning for unmanned surface vehicle based on improved quantum ant colony algorithm. Math Probl Eng 2019:1–10. https://doi.org/10.1155/2019/2902170

Xia X, Luo J, Wang X (2022) Improved pigeon-inspired optimization algorithm of fuzzy variation operator. J Henan Univ Sci Technol (natural Science) 43:39–44. https://doi.org/10.15926/j.cnki.issn1672-6871.2022.03.007

Xiao Z, Xu X, Xing H et al (2021) RTFN: a robust temporal feature network for time series classification. Inf Sci 571:65–86. https://doi.org/10.1016/j.ins.2021.04.053

Xing H, Xiao Z, Zhan D et al (2022) SelfMatch: robust semisupervised time-series classification with self-distillation. Int J Intell Syst 37:8583–8610. https://doi.org/10.1002/int.22957

Yang H, Qi J, Miao Y et al (2019) A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization. IEEE Trans Ind Electron 66:8557–8566. https://doi.org/10.1109/TIE.2018.2886798

Yu J, You X, Liu S (2022) Dynamically induced clustering ant colony algorithm based on a coevolutionary chain. Knowl-Based Syst 251:109231. https://doi.org/10.1016/j.knosys.2022.109231

Zhang S, Pu J, Si Y (2021a) An adaptive improved ant colony system based on population information entropy for path planning of mobile robot. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3056651

Zhang S, Pu J, Si Y, Sun L (2021b) Path planning for mobile robot using an enhanced ant colony optimization and path geometric optimization. Int J Adv Rob Syst 18:172988142110192. https://doi.org/10.1177/17298814211019222

Zhang S, Sang H, Sun X et al (2022) A multi-objective path planning method for the wave glider in the complex marine environment. Ocean Eng 264:112481. https://doi.org/10.1016/j.oceaneng.2022.112481

Zhao L, Bai Y, Paik JK (2023) Global-local hierarchical path planning scheme for unmanned surface vehicles under dynamically unforeseen environments. Ocean Eng 280:114750. https://doi.org/10.1016/j.oceaneng.2023.114750

Zhou W (2020) Research on ship route optimization based on energy consumption management. Dissertation, Zhejiang Ocean University

Zyczkowski M, Szlapczynski R (2023) Collision risk-informed weather routing for sailboats. Reliab Eng Syst Saf 232:109015. https://doi.org/10.1016/j.ress.2022.109015

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.