



Research paper

Path planning for Multi-USV target coverage in complex environments

Jing Luo ^{*}, Yumin Su*Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin, 150001, China*

ARTICLE INFO

Keywords:

Unmanned surface vehicles
 Locking sweeping method
 Greedy allocation strategy
 ant colony optimization method
 Target coverage task

ABSTRACT

This research introduces a path planning methodology aimed at coordinating multiple unmanned surface vehicles (USVs) to accomplish target coverage task in obstacle-rich environments. Leveraging the locking sweeping method (LSM), our proposed method constructs task target point distance fields that integrates environmental constraints, alongside a task target point distance matrix. During the task allocation phase, we design a cost assessment function to assess the generated allocation solutions. And a greedy allocation strategy is employed to determine the optimal number of USVs required for mission completion, ensuring evenly distribution of target task points among them. Subsequently, we improve the ant colony optimization (ACO) method by redesigning the heuristic function and pheromone update rules, considering environmental constraints and task execution sequence constraints. This refinement facilitates the generation of optimized task execution sequences and safe navigation paths in obstacle environments. The effectiveness of the proposed method is validated through multiple sets of simulation experiments and compared with existing methods. The results demonstrate the practicality and efficacy of the method in addressing the challenges of USVs target coverage task in obstacle environments.

1. Introduction

In recent years, with the rapid development and widespread application of surface unmanned vehicle (USV) technology, USVs have been playing an increasingly important role in fields such as marine surveying, environmental monitoring, and maritime transportation (Zhou et al., 2024). It can be foreseen that in the near future, with advancements in sensor and robotics technologies, USVs will be able to effectively execute tasks in various harsh and remote environments, thereby improving task efficiency and reducing unnecessary human casualties and labor costs. However, compared to other types of unmanned platforms such as autonomous robots, autonomous cars, and unmanned aerial vehicle (UAV), the level of autonomy in USVs is still relatively low. One solution to address this issue is the simultaneous cooperative deployment of multiple USVs. Compared to single surface USV systems, multi-USV systems operating collaboratively offer greater flexibility and maneuverability, are less affected by dynamic environmental changes, and can operate across larger marine areas (Wang et al., 2021; Miao et al., 2022; Huang et al., 2024).

The multi-USV coverage problem involves deploying multiple USVs to ensure comprehensive coverage of a given marine area, efficiently and effectively executing tasks such as data collection, monitoring, or

surveying (MahmoudZadeh et al., 2022). This problem requires each USV to cover specific target points within the area by optimizing their deployment and coordination to minimize total travel distance and energy consumption while ensuring that each target point is covered exactly once (Ma and Hao, 2021). This involves solving complex issues such as task allocation, path planning, and coordination among the USVs.

To address the multi-USV task coverage problem, it is essential to initially construct a suitable mathematical model and subsequently select appropriate methods for solving it based on this model. In modeling the multi-USV task coverage, it is crucial to meet constraints related to USV performance and coordination, as well as factors such as environmental constraints and task execution sequencing. This complexity renders it a multi-constraint and challenging NP problem (Ma and Chen, 2023; Ma and Han, 2023). Consequently, the multi-task coverage problem is analogous to the multiple traveling salesman problem (MTSP), where, given a specific list of destinations, the task involves determining optimal path planning for multiple USVs to ensure each target point is covered once, aiming to find the shortest and most efficient paths. Presently, exact methods like mathematical modelling (Wang et al., 2007; Odili et al., 2020), linear programming (Alighanbari, 2004; Afonso et al., 2020) and dynamic programming (Goemans, 1995;

^{*} Corresponding author.

E-mail address: luojing@hrbeu.edu.cn (J. Luo).

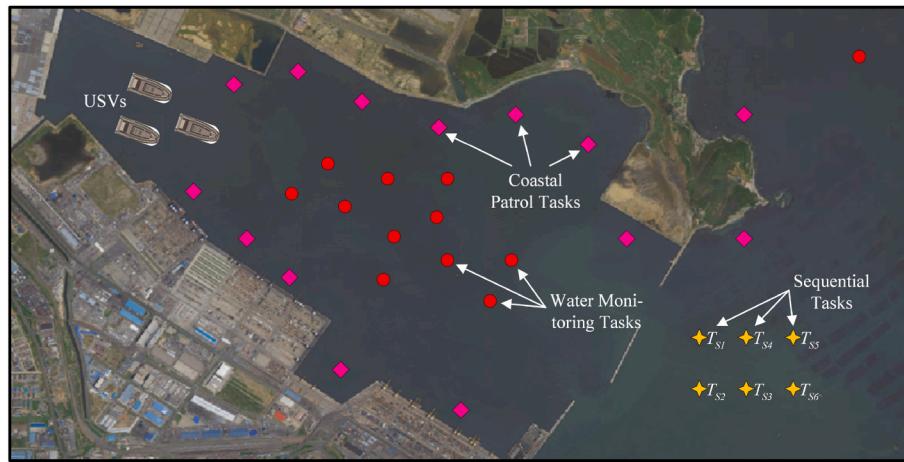


Fig. 1. Multi-USV are conducting the mission.

Scholz et al., 2016) are extensively employed in solving the MTSP problem. However, these exact methods can only yield optimal solutions for small-scale MTSP problems, making it challenging to find the optimal solution within a limited timeframe when dealing with large-scale problems. As the scale of the problem increases and artificial intelligence technology advances, more intelligent optimization methods have emerged to tackle this challenge. For instance, genetic algorithm (GA), self-organizing map (SOM), and ant colony optimization (ACO) methods are increasingly utilized in this regard.

Wang et al. (Yang and Xan, 2021) proposed a method based on quantum GA to solve the multi-drone task allocation problem. Park et al. (2021) developed a task allocation system for USVs missions, employing GA to solve the defined optimization problem model. Jia et al. (2018) proposed a metaheuristic method based on improved GA to address cluster task pre-allocation problems with complex kinematic constraints, resource constraints, and time constraints. Xu et al. (2018) improved GA to resolve potential deadlock issues when solving task pre-allocation problems for cluster execution with temporal constraints. Sun et al. (2017) addressed the multi-robot target coverage problem by introducing a locking mechanism based on the SOM method. Tang et al. (2020) proposed a hybrid method combining SOM with the adaptive dynamic window approach (DWA) to address task allocation and collaborative search problems for multiple agents in dynamic environments. Tang et al. (Tan et al., 2022) integrated SOM with the tasks treatment list (TTL) method to solve the multi-type target coverage problem for heterogeneous USVs. Liu et al. (Liu and Bucknall, 2018) introduced collision avoidance potential fields during neuron updates to endow neurons with collision avoidance capabilities in obstacle environments. Building upon this, considering the energy and communication constraints of USVs, they further improved the SOM method to solve multi-USV coverage problems (Liu et al., 2019a). Yao et al. (2023) proposed a hierarchical two-layer framework based on the SOM method to address collaborative path planning problems for multi-USV systems in spatially varying sea current environments. Gao et al. (2021) proposed a heuristic method based on ACO to achieve reconnaissance task allocation for multiple drones, considering the characteristics of heterogeneous targets. Qizilbash et al. (2020) introduced a motion planner based on the ACO method to address task allocation problems for industrial warehouse robots. Li et al. (2020) presented a cooperative mission allocation method for multiple drones based on the ACO method. Yang et al. (2019) implemented an efficient two-layer ACO method for autonomous robot navigation. Horbulin et al. (2020) proposed a new optimization method combining taboo search, exhaustive search, and ACO to solve path planning problems for drones in dynamic replenishment scenarios. Liu et al. (2019b) combined the ACO method with genetic operators, effectively expanding the search range of

solutions.

However, most of the above target coverage methods are conducted in scenarios without obstacles, while USV mission scenarios typically involve various obstacles. In such scenarios, the Euclidean distance between target points cannot accurately represent the true distance cost between points. Additionally, the above research seldom considers the special task types of USVs. Some tasks of USVs have temporal sequential requirements, such as scanning mission. Moreover, in obstacle scenarios, the aforementioned target coverage methods usually need to be combined with path planning methods to optimize paths, and they cannot directly output collision-free paths suitable for safe navigation.

Inspired by the above research, we propose a distance matrix based on the locking sweeping method (LSM) (Luo et al., 2024a) to represent the distance cost between target points in obstacle environments. The distance field obtained during the calculation process can be utilized in the subsequent path planning phase to directly generate collision-free paths between points, and the boundaries of obstacles can be adjusted according to the safety requirements of USVs. Based on the aforementioned research, a two-layered collaborative target coverage framework for multiple USVs is proposed to effectively address the problem of covering a set of target points within obstacle-laden areas. At a higher level, a greedy allocation strategy is employed to determine the optimal number of USVs for mission execution and the allocation solutions. Subsequently, at a lower level, an improved ACO method is applied to generate collision-free paths which meet the mission requirements. Additionally, a cost assessment function is designed to assess the generated coverage solutions. The ultimate objective is to assign task points to each USV in the optimal sequence to minimize the overall cost.

The remainder of this research has the following structure: Section 2 defines the problem and constructs the cost assessment function to evaluate the result. Section 3 introduces the method of the study. Section 4 presents the numerical simulations, and the conclusions and future work are presented in Section 5.

2. Problem formulation

In this section, we will first describe the application scenario of multi-USV mission planning, and then design a new cost function based on the application scenario.

2.1. Environment description

In this scenario, multiple task points within the area show in Fig. 1 have been designated as water monitoring locations and coastal patrol locations. To inspect these locations, multiple USVs can be deployed for timely detection and monitoring.

Throughout this research, the USVs are denoted by the set $\mathbf{U} = \{U_1, U_2, \dots, U_N\}$, $N \in \mathbf{N}^*$, and the task points are denoted by the set $\mathbf{T} = \{T_1, T_2, \dots, T_M\}$, $M \in \mathbf{N}^*$. The USVs have crucial task of achieving complete coverage of all task points within the designated area, with the strict requirement that each point is visited only once. Furthermore, it is essential for the USVs to return to its initial position upon successfully completing the mission. The tasks set T_n of U_n is denoted by the set $T_n = \{T_0, T_1, \dots, T_K\}$, $K \in \mathbf{N}^*$. T_0 denotes the initial position of U_n .

We will also suppose the inclusion of certain special sequential task points, such as mine sweeping tasks, within the mission. In such cases, it is required that the USVs cover these task points in a specific sequential order. The yellow task points in Fig. 1 show a collection of sequential task points, denoted as set $T_S = \{T_{S1}, T_{S2}, T_{S3}, T_{S4}, T_{S5}, T_{S6}\}$. These task points need to be covered in sequence, that is, T_{S1} is the current target point of U_i . When T_{S1} is covered, the next target point of U_i is T_{S2} , and so on, until all sequential task points in the set T_S are visited. The final visit sequence is $T_{S1} \rightarrow T_{S2} \rightarrow T_{S3} \rightarrow T_{S4} \rightarrow T_{S5} \rightarrow T_{S6}$.

2.2. Cost assessment

To enhance the efficiency of multiple USVs in mission and optimize mission resources, this research optimizes the navigation trajectory of multi-USV through the construction of a new cost function. Firstly, some definitions are given as follows.

Definition 1 (Time cost). Once the mission commences, each USV is required to visit all mission points within its designated set and ultimately return to the initial position. The mission concludes when all USVs have successfully returned to their initial positions. Here, we define the time cost as the duration it takes for the USV to traverse all task points on its designated route and return to the initial position from the point of departure. The final time cost is determined by selecting the maximum value among the individual time costs of all USVs, representing the total time consumed from the mission's initiation to its completion. Therefore, the time cost C_{final_t} can be derived as follows:

$$C_t(n) = \sum_{i=0}^K \sum_{j=0}^K \frac{|T_i T_j|}{v_n} f_{ij}^n \quad \forall n \in \{1, 2, \dots, N\} \quad (1)$$

$$f_{ij}^n = \begin{cases} 1, & \text{if } U_n \text{ will traverse from } T_i \text{ to } T_j \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

$$C_{final_t} = \max(C_t(1), C_t(2), \dots, C_t(N)) \quad (3)$$

where $|T_i T_j|$ represents the distance cost between T_i and T_j . v_n is the linear velocity of U_n . f_{ij}^n is a BOOL variable, indicating whether the U_n will traverse from T_i to T_j . $C_t(n)$ represents the time cost of U_n .

Definition 2 (Energy cost). The energy cost is defined as the cumulative energy consumed by each USV throughout its mission. As USVs only consume energy during navigation, the energy cost C_e primarily relies on the mission time of each USV, which is given by

$$C_e = \sigma_1 \sum_{n=1}^N C_t(n) \quad (4)$$

where σ_1 represents the energy consumption per unit time of a USV.

Definition 3 (Resource cost). The deployment of each additional USV in a mission requires additional resources. Maintaining a reasonable control over the number of USVs deployed is crucial in optimizing task coverage outcomes. Additionally, if an individual USV completes its mission prematurely, it will also cause a waste of resources. So, the resource cost C_r is defined as

$$C_r = \sigma_2 \cdot N + \sigma_3 \cdot \chi \quad (5)$$

$$\chi = \frac{1}{N} \sum_{i=1}^N \left| \frac{C_t(n) \cdot N}{\sum_{i=1}^N C_t(i)} - 1 \right| \quad (6)$$

where σ_2 represents the resource consumption generated by each additional deployed USV. N is the number of deployed USVs. χ represents the temporal consistency performance for coverage solutions. σ_3 is the waste of resources caused by the decline in temporal consistency performance. Additionally, the following constraints must be adhered to during the calculation process to ensure the validity of the results:

$$C_t(n) \cdot v_n \leq d_{limit} \quad \forall n \in \{1, 2, \dots, N\} \quad (7)$$

$$\sum_{n=1}^N \sum_{i=0}^M f_{ij}^n = 1 \quad \forall j \in \{1, 2, \dots, M\} \quad (8)$$

$$\sum_{i=1}^M f_{0i} = N \quad (9)$$

$$N \leq N_{limit} \quad (10)$$

$$\left| \frac{C_t(n) \cdot N}{\sum_{i=1}^N C_t(i)} - 1 \right| \leq \varpi \quad \forall n \in \{1, 2, \dots, N\} \quad (11)$$

where d_{limit} is the maximum range of the USVs. N_{limit} represents the maximum number of USVs that can be deployed. ϖ is task load limit parameter.

Remark 1. The endurance constraint, denoted as Eq. (7), imposes a limitation on the maximum achievable range of USVs. This constraint guarantees that USVs navigate within a maximum range d_{limit} from their initial point. It governs the feasibility and effectiveness of USVs in successfully accomplishing their intended missions within the prescribed endurance limitations.

Remark 2. During the mission, it is required that each mission point can only be visited once by an USV. The constraints in Eq. (8) is adopted to ensuring that the USV U_n covers every task point in the task set T_n exactly once.

Remark 3. In various mission scenarios, the required number of deployed USVs in the optimal coverage solution may vary. The optimal number of USVs is denoted as N in Eq. (9). Nevertheless, it is essential to note that N cannot exceed the maximum number of deployed USVs, represented by N_{limit} in Eq. (10).

Remark 4. Throughout the mission, it is crucial to ensure that the completion times for each USV closely align, minimizing resource wastage resulting from individual USV completing tasks too quickly. This constraint is represented by Eq. (11), denoting the task load constraint. ϖ is employed to regulate the task load of USVs, ensuring that each USV can handle tasks of comparable number and complexity.

Definition 4 (Cost function). The objective of the coverage task is to minimize total cost while ensuring the completion of coverage for each task target. The total cost encompasses time cost, energy cost, and resource cost. By optimizing the total cost, we can enhance task efficiency and reduce overall consumption. The cost function can be expressed as follows:

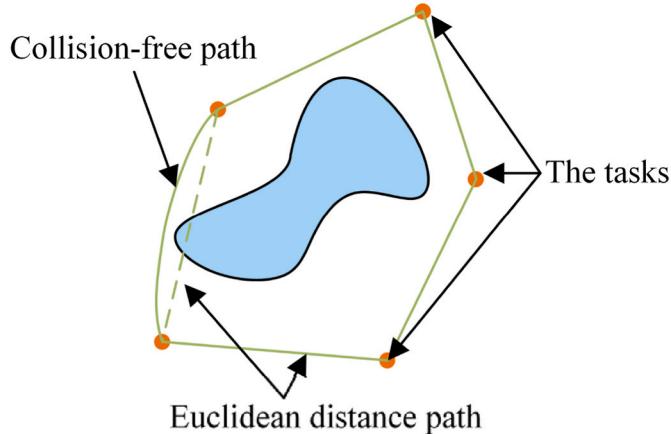


Fig. 2. Example of task allocation.

$$C_{total} = \alpha \cdot C_{final_t} + \beta \cdot C_e + \gamma \cdot C_r \quad (12)$$

$$\alpha + \beta + \gamma = 1 \quad (13)$$

where α, β, γ are the weight parameters, which are utilized to control the preference for the time cost, energy cost, and resource cost during the generation of coverage solutions.

Remark 5. Based on the above definition, our ultimate objective is to minimize mission cost C_{total} and generate feasible task coverage solutions for various task scenarios, taking into account user-configured cost preferences. Subsequently, we employ a path planning method to generate safe collision-free paths for each USV.

3. Method design

In this section, we address the problem discussed in the previous section by dividing it into three stages: 1) constructing the distance matrix, 2) task allocation and 3) path planning. First, we employ the locked scanning method to construct a distance potential field for task points, taking into account obstacles' influence. This enables us to generate an actual distance cost matrix between task points. Second, we utilize a greedy allocation strategy to determine the number of USVs required and assign task points to each USV. Third, we propose the IACO method to optimize the coverage solutions and specify the best path planning solutions to minimize mission cost.

3.1. Adjustable distance matrix based on locking sweeping method

In the majority of task assignment methods, the Euclidean distance is commonly utilized to estimate the distance cost between two task points (Li et al., 2020; Yang et al., 2019). However, these methods disregard the influence of obstacles within the environment. In the realm of USVs, the mission area often includes diverse obstacles. Ensuring the consistent avoidance of entering the safety range of obstacles along the planned task path becomes essential when obstacles are present between two adjacent task points. Consequently, the actual distance cost between task points exceeds the Euclidean distance, as shown in Fig. 2. To address this challenge, this section proposes a method for constructing the distance cost based on the LSM. This method aims to account for the influence of obstacles and enhance the accuracy of distance estimation.

The LSM as a numerical method, which is employed to describe the propagation scenario of wave fronts on interfaces by numerically solving

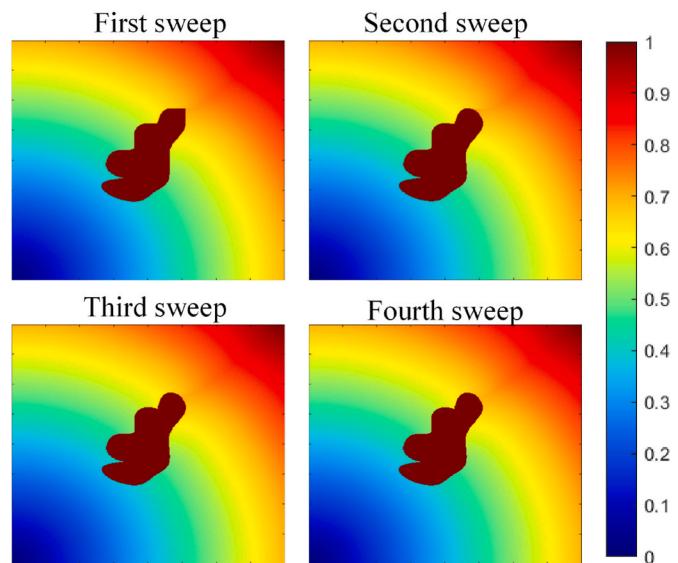


Fig. 3. Example of potential field solution process based on LSM.

the viscosity solution of the Eikonal equation (Bak et al., 2010):

$$|\nabla A(x,y)| \cdot F(x,y) = 1 \quad \forall (x,y) \in \Omega \quad (14)$$

where Ω represents the metric space, and (x,y) represents the point in Ω . $F(x,y)$ represents the propagating velocity related to position (x,y) . $A(x,y)$ represents the arrival-time from the initial point to (x,y) .

The LSM calculates the arrival-time field by systematically sweeping the entire configuration space in a specific order, determining the potential value at each point through recursive tracing back to the initial point. This value represents the local interface arrival-time, where a greater distance between a point and the initial point corresponds to a larger arrival-time. Points within the obstacle area are deemed unreachable, yielding an infinite arrival-time for them. The potential field solution process as shown in Fig. 3 and the details on LSM can be found in previous studies (Luo et al., 2024a).

Directly employing potential fields to indicate distance could introduce navigation safety risks for USVs. This is particularly pertinent in areas adjacent to coastlines, where traffic congestion is more probable and shallower waters amplify the potential danger. Moreover, ensuring adequate clearance between USVs and coastlines is imperative due to potential inaccuracies in map data. This measure is essential for optimizing the safety of autonomous navigation. To tackle these challenges, this research proposes the utilization of a two-step LSM for deriving distance-indicating fields for mission points.

Step 1 (Coastline expanding): Given the LSM's capacity to indicate distance information through the potential field it creates, emitting multiple waves from the obstacle boundary can generate a new potential field D_{o_init} that accurately represents the proximity of the local point to the obstacle. Then, we can acquire a boundary-expanded map D_o , which can be adaptively adjusted as follows:

$$D_{o_init} = LSM(\Omega_o, D_o) \quad (15)$$

$$D_o(x,y) = \begin{cases} 0, & \text{if } D_{o_init}(x,y) > \varsigma \\ \frac{\varsigma - D_{o_init}(x,y)}{\varsigma}, & \text{otherwise.} \end{cases} \quad (16)$$

where Ω_o is a set of points in the obstacle area, and $LSM(\cdot)$ represents the LSM process. ς is a propagation scale limit parameter.

Remark 6. The ς can be adjusted based on the navigation and safety requirements of the USVs. It serves to regulate the scope of coastline expansion. In essence, within the potential field D_{o_init} , any potential

Input: task points set (T), obstacle points set (Ω_o), distance field (D), metric space (Ω)

1. Initialize the distance field: $D(x, y) = \infty \forall (x, y) \in \Omega$;
2. Let $D(x, y) = 0 \forall (x, y) \in \Omega_o$;
3. Calculate the obstacle distance field D_{o_init} according to Eq. (15);
4. Set the propagation scale limit parameter ς ;
5. Generate the boundary-expanded map D_o according to Eq. (16);
6. **for** $m = 1$ to M **do**
7. Calculate the distance field D_m for the point T_m according to Eq. (17);
8. **end for**
9. Calculate the construct adjustable distance matrix ϑ according to Eq. (18);
10. **return** D and ϑ

values exceeding ς are reset to 0, while those below ς are normalized to the range $[0, 1]$, thereby reducing the dimensionality of the potential field. Fig. 4a–c illustrates a straightforward application example. This feature becomes especially critical when the USV needs to navigate in coastal environments influenced by tides.

Step 2 (Distance matrix): Based on the above boundary-expanded map, we can utilize the task point T_m as the starting point and employ the LSM method to systematically sweep the entire metric space, thereby obtaining the distance field D_m for the task point T_m , as follows:

$$D_m = LSM(T_m, D_o) \quad \forall m \in \{1, 2, \dots, M\} \quad (17)$$

where T_m is the m th task point.

Remark 7. Utilizing Eq. (17), by traversing the task point set T , the distance field for each task point can be computed. Fig. 4d illustrates a distance field example for a task point. We can determine the distance weights between other task points and it. The distance weights consider the influence of obstacles in the scene, reflecting the actual navigation distance cost of the USVs. This feature improves the practical performance of the method.

Remark 8. The distance field demonstrates the feature that the distance field value at the initial point is the global minimum (Luo et al., 2024a). Once the task allocation results are obtained, this feature assists in generating a safe collision-free path between two adjacent task points without the need for additional path planning methods, thereby enhancing the overall allocation efficiency.

Finally, the distance matrix ϑ of task points can be defined as follows:

$$\vartheta = \begin{bmatrix} 0 & D_1(T_2) & D_1(T_3) & \cdots & D_1(T_M) \\ D_2(T_1) & 0 & D_2(T_3) & \cdots & D_2(T_M) \\ D_3(T_1) & D_3(T_2) & 0 & \cdots & D_3(T_M) \\ \vdots & \vdots & \vdots & 0 & \vdots \\ D_M(T_1) & D_M(T_2) & D_M(T_3) & \cdots & 0 \end{bmatrix} \quad (18)$$

where $D_i(T_j)$ represents the distance weight from points T_i to T_j .

On this foundation, the overall procedure of constructing adjustable distance matrix is presented in Algorithm 1.

Algorithm 1. Construct Adjustable Distance Matrix.

3.2. Greedy allocation strategy

The core concept behind employing a greedy allocation strategy for task assignment is to consistently assign the target point with the smallest distance cost to the USV with the smallest range. This method aims to minimize the time required to complete the mission until all target points are allocated. For different missions, deploying varying numbers of USVs typically results in different coverage solutions. Each coverage solution can yield the corresponding mission cost based on Eq. (12). Deploying an appropriate number of USVs can effectively reduce the cost of USVs performing missions and enhance the mission efficiency of USVs. This optimization aids in refining the coverage solution and improving the efficiency and accuracy of path planning in the next step.

The pseudocode for the greedy allocation strategy is shown in Algorithm 1. We include the distance field D of target points and the distance matrix ϑ calculated in the previous section as inputs. First, we need to determine the minimum number of USVs required to complete the mission. The detail process is outlined in steps 2–14. Assign target points to a USV one by one based on distance cost until reaching its range limit, then switch to other USV until all target points are assigned. The number of USVs at this point represents the minimum required to complete the mission N_{min} . Then, calculate the cost of the solution as the minimum cost C_{min} in step 15. From steps 16 to 26, we generate task coverage solutions based on greedy allocation strategy, ranging from deploying the minimum to the maximum number of USVs. In each allocation round, we select the USV U_s with the shortest range as the winner. We then allocate the target point T_c with the minimum distance cost to the winner U_s and update the U_s 's position to the T_c , as shown in steps 19–23. This process continues until all target points are covered. Steps 25–26 yield coverage solutions for various numbers of USVs, accompanied by their corresponding total costs. In other words, for n USVs, we can obtain their coverage solution P_n and the solution cost C_n . Finally, from steps 27 to 32, we select the optimal coverage solution P_n and the required number of deployed USVs N_o by comparing the total costs of different coverage plans, and then take them as the output of Algorithm 2.

Algorithm 2. Greedy Allocation Strategy.

Input: USVs set (U), task points coverage status set (S), task points set (T), the maximum range of

the USVs (d_{limit}), the distance field of task points (D), the distance matrix (ϑ)

1. Initialize the initial position of all USVs and coverage status of task points;

2. Let $S(T_m) = 0 \forall m \in \{1, 2, \dots, M\}$, $n = 1$;

3. **while** $S = 0$ **do**

4. Select the task point T_c closest to the position of USV U_n based on D and ϑ ;

5. Update the range d_n of USV U_n ;

6. **if** $d_n \leq d_{limit}$ **then**

7. Assign target point T_c to USV U_n ;

8. Update the position of USV U_n to T_c ;

9. Update the target point status $S(T_c)$ to 1;

10. **else**

11. $n = n + 1$;

12. **end if**

13. **end while**

14. Let the minimum number of USVs required to complete the mission N_{min} to n ;

15. Calculate the cost C_{min} based on Eq. (12);

16. **for** $n = N_{min}$ to N **do**

17. Let $S(T_m) = 0 \forall m \in \{1, 2, \dots, M\}$;

18. **while** $S = 0$ **do**

19. Select the USV U_s with the shortest range;

20. Select the task point T_c closest to the position of USV U_s based on D and ϑ ;

21. Assign target point T_c to USV U_s ;

22. Update the position of USV U_s to T_c ;

23. Update the target point status $S(T_c)$ to 1;

24. **end while**

25. Get the coverage solution P_n for n USVs;

26. Calculate the cost C_n based on Eq. (12);

27. **if** $C_n \leq C_{min}$ **then**

28. Update the C_{min} to C_n ;

29. Update the optimal number of USVs N_o to n ;

30. Update the optimal coverage solution P_o to P_n ;

31. **end if**

32. **end for**

33. **return** N_o and P_o

3.3. Ant colony optimization method

The design intention and characteristics of the ACO method are particularly well-suited for solving discrete combinatorial optimization problems like the TSP. This method simulates the gradual optimization process of ants finding the shortest path through the pheromone mechanism, making it highly effective for addressing the specific

requirements of the TSP. Consequently, the ACO method is generally considered an appropriate and effective choice for solving TSP problems. Therefore, this research optimizes the ACO method to tackle the problem of multi-USV target coverage in obstacle environments. First, the ACO method as applied to the TSP problem is introduced.

The ACO method is a metaheuristic method inspired by the foraging behaviour of ants in nature. The foraging process of the ant colony relies

on a positive feedback mechanism. This mechanism operates as follows: as ants forage, they deposit pheromones along their path to guide subsequent ants. Subsequently, the method computes the transition probability based on the concentration of pheromones and a heuristic function. This allows each ant to establish a list of viable movements and select one based on predefined state transition rules. The essence of the method lies in the transition probability and pheromone update mechanisms.

In the TSP, the basic framework of the ACO method is as follows: Multiple ants are deployed to construct paths individually and leave pheromones along these paths. For each ant, when it needs to access the next target point, the transition probability between target points is calculated based on the pheromone concentration and heuristic function. Subsequently, the ant selects its next target point using the roulette wheel selection. The transition probability p_{ij}^k from target point T_i to target point T_j for k th ant can be expressed as

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij})^U \cdot (\eta_{ij})^V}{\sum_{G \in \mathcal{T}_k(i)} (\tau_{iG})^U \cdot (\eta_{iG})^V}, & \text{if } j \in \mathcal{T}_k(i) \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

where τ and η represent the pheromone concentration and the heuristic function, respectively. U and V are the pheromone factor and the heuristic function factor. $\mathcal{T}_k(i)$ represents the set of target points that the k th ant needs to cover at position T_i . The heuristic function η is defined as follows:

$$\eta_{ij} = \frac{1}{|T_i T_j|} \quad (20)$$

where η_{ij} is the reciprocal of Euclidean distance between T_i and T_j .

Upon completion of visits to all target points by all ants, the pheromone concentration on each path in the t th iteration is updated using Eq. (21).

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij} \quad \forall \rho \in (0, 1) \quad (21)$$

where ρ is the pheromone evaporation coefficient. $\Delta\tau_{ij}$ represents the increment of pheromone on the path (T_i, T_j) from T_i to T_j in this iteration, and it can be expressed as

$$\Delta\tau_{ij} = \sum_{k=1}^{\mathcal{A}} \Delta\tau_{ij}^k \quad (22)$$

where \mathcal{A} is the number of ants. $\Delta\tau_{ij}^k$ represents the amount of pheromone left on the path (T_i, T_j) by the k th ant in this iteration, and it can be expressed as

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{if } k \text{th ant traverse from } T_i \text{ to } T_j \\ 0, & \text{otherwise.} \end{cases} \quad (23)$$

where Q is a positive constant, and L_k represents the length of the path traveled by the k th ant.

3.4. Improved ant colony optimization method considering mission scenarios

In practical applications, USVs are typically deployed across diverse mission scenarios. These USVs must not only possess the capability to

fulfill missions but also prioritize navigation safety as the cornerstone of mission success. To enhance the practicality and cost-effectiveness of the ACO method, we propose an improved ant colony optimization (IACO) method that takes into account various mission scenarios.

In the heuristic function model, combined with the distance matrix created in the previous section, a novel heuristic function is introduced to optimize the total distance cost of the coverage solution. Its rule is that whenever ants transfer between task points, the heuristic function tends to guide ants to move towards target points with smaller actual distance cost along collision-free paths. Throughout the iterative process, the influence of environmental obstacles blocking paths is fully considered. The heuristic function will be expressed as

$$\eta_{ij} = \frac{1}{\vartheta(i, j)} \quad (24)$$

where $\vartheta(i, j)$ represents the element located in the i th row and j th column of the distance matrix, depicting the actual distance cost of the ant from T_i to T_j .

For special sequential task points, we can adjust the transition probabilities by setting the heuristic function of points, encouraging ants to visit these special points in sequence. For example, the point $T_i \in T_S$ and it has the subsequent task point. Then, the heuristic function η_{ij} will be expressed as

$$\eta_{ij} = \begin{cases} 1, & \text{if } T_i \text{ is the preceding point of } T_j \quad \forall T_j \in T \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

where T_S is the set of sequential task points, and T is the set of all task points.

Furthermore, to enhance the search efficiency and convergence speed of the ACO method, we optimized the pheromone update rule. In each iteration, we store the current best solution, and if there exists a solution in the subsequent iteration that is better than the stored best solution, we strengthen the pheromones on this path, accelerating the accumulation of pheromones on the optimal path. The new pheromone update rule can be expressed as

$$\tau_{ij}(t+1) = \begin{cases} (1 - \rho) \cdot \tau_{ij}(t) + (1 + \vartheta) \cdot \Delta\tau_{ij} & \text{if } L_k(t) < L_{\min} \quad \forall \rho \in (0, 1), \\ (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}, & \text{otherwise.} \end{cases} \quad (26)$$

where $\vartheta \in (0, 1)$ is a gain parameter. $L_k(t)$ represents the length of the path traveled by the k th ant in the t th iteration. L_{\min} the saved length of the optimal path from the previous iteration. And if $L_k(t) < L_{\min}$, then $L_{\min} = L_k(t)$.

After the loop ends, we can obtain the optimal task execution sequence for each USV. At this point, we need to generate safe guidance paths to direct the USVs to visit each target point within the task set. When there are no obstacles within the mission scenario, connecting adjacent task points one by one suffices to generate the final path for USV task execution. However, when obstacles are present between two adjacent task points, the path calculation process is as follows:

$$\text{path}_{ij} = \begin{cases} \text{grad}(D_i, T_j) & \text{if } \mathfrak{I}_{ij} = 1 \\ \text{connect } T_i \text{ and } T_j & \text{otherwise} \end{cases} \quad (27)$$

where \mathfrak{I}_{ij} is a BOOL variable, indicating whether there are obstacles between adjacent points T_i and T_j . $\text{grad}()$ represents the gradient descent method process, utilized for computing the collision-free path from point T_j to T_i along the gradient descent direction of the distance field D_i .

On this foundation, the overall procedure of IACO method is presented in Algorithm 3.

Algorithm 3. IACO method.

```

Input: task points set of  $U_n$  ( $T_n$ ), distance field ( $D$ ), distance matrix ( $\mathcal{D}$ ), number of ants ( $N_{ant}$ ),
number of iterations ( $N_{iter}$ )

1. Initialize the initial position of all ants;

2. Calculate the heuristic function that takes into account the effects of obstacles according to Eq. (24)-

(25);

3. for  $t = 1$  to  $N_{iter}$  do;
4.     Set all ants on the initial position;
5.     for  $k = 1$  to  $N_{ant}$  do;
6.         Calculate the movement path of the  $k$ th ant according to Eq. (19);
7.         Calculate the length of path  $L_k(t)$ ;
8.     end for
9.     Calculate the optimal path length in the  $t$ th iteration  $L_{best}(t)$ ;
10.    Update the path pheromone according to Eq. (26);
11.    if  $L_{best}(t) \leq L_{min}$  then
12.        Update the  $L_{min}$  to  $L_{best}(t)$ ;
13.        Update the optimal path  $path$ ;
14.    end if
15. end for
16. Optimize  $path$  based on Eq. (27) to generate the final collision-free safe path  $path_{final}$ ;
17. return  $path_{final}$ 
```

4. Numerical simulation

In this section, we conducted two sets of numerical simulations to illustrate the feasibility and superiority of the proposed method. In the first set of simulations, varying numbers of USVs were deployed to cover tasks within a scenario featuring obstacles. The aim was to evaluate the effectiveness of the greedy allocation strategy and the method's ability to handle obstacles. Subsequently, in the second set of simulations, the optimization capabilities of different coverage methods were compared in a scenario with task execution sequence constraints. The results show the validity of our method.

4.1. Performance evaluation of target coverage method in obstacle environment

In the second set of simulations, we randomly deploy 60 target points within a square area measuring 500×500 pixels. Several irregular obstacles are distributed randomly throughout the task area. The initial position of the USVs is at the center of the area, with coordinates (250, 250). A total of $N = 5$ USVs is available to perform coverage tasks. The maximum travel distance of the USVs is 2000 pixels, with a velocity of 1 pixel per second. To ensure the navigation safety of USVs during task execution, the obstacle boundary expansion coefficient is set to $\zeta = 0.1$. Additionally, the weight coefficients (α, β, γ) in the cost function are set to 0.4, 0.2, and 0.4 respectively. The tuning of other parameters in

method has been well studied in (Tan et al., 2022; Li et al., 2023), which include a number of invaluable empirical parameter tuning works. According to the provided recommendations in these works, the param-

ters configured as shown in Table 1.

First, we verify the effectiveness of the greedy allocation strategy. We conducted task coverage with the minimum to maximum number of USVs in the same scenario, as shown in Fig. 5a–c. During this phase of coverage solutions calculation, the influence of obstacles was not considered, and the distance cost between task points was represented by the Euclidean distance between the two points. To ensure that all target points in the area are covered, it was calculated that a minimum of 3 USVs are needed. The task coverage results for 3 USVs is shown in Fig. 5a. The maximum number of USVs that can be deployed is 5. Then, we generated two coverage results for deploying 4 and 5 USVs respectively, as shown in Fig. 5b and c. Additionally, considering the obstacles obstructing navigation within the scenario, we optimized the coverage results using path planning method to generate collision-free paths for safe navigation during task execution. By evaluating the cost of results using a cost assessment function, it was determined that deploying 4 USVs incurred the lowest task coverage cost. Some relevant data from the simulation results are shown in Table 2.

Second, we verify IACO's ability to handle obstacles. Compared to previous methods, we incorporated the distance matrix from Algorithm 1 to represent the distance cost between target points, fully considering the impact of obstacles during the coverage solution calculation process. Using the greedy allocation strategy from Algorithm 2, we determined that the minimum number of USVs required for mission completion remains 3. Therefore, we generated task allocation solutions for $N = 3$, $N = 4$, and $N = 5$ respectively. Subsequently, we optimized the task coverage results based on Algorithm 3, which allowed for the direct generation of paths ensuring safe navigation. The results are shown in

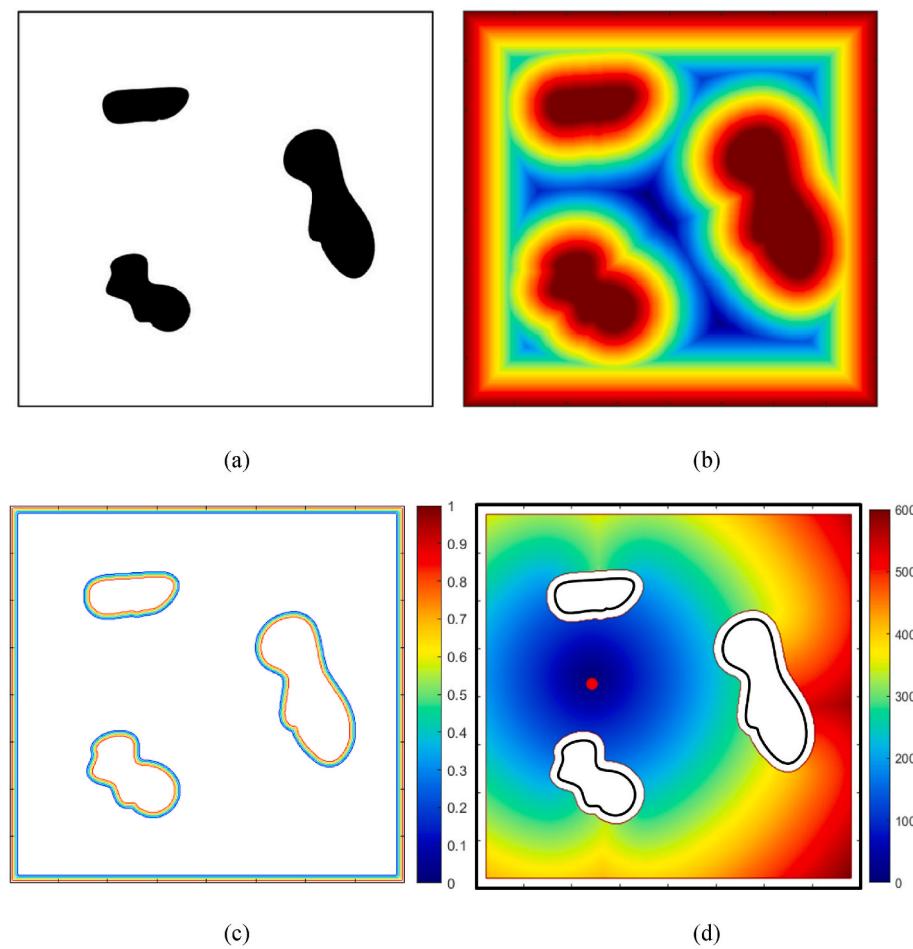


Fig. 4. Example of the distance field solution process based on LSM for a task point. (a) The environment including multiple islands; (b) obstacle distance map; (c) expanded coastline map with $\xi = 0.1$; (d) the resulting distance field for a task point.

Fig. 5d-f. After evaluating the resulting cost using the cost assessment function, it is determined that the optimal number of USVs required to complete the mission is 4. Some relevant data from the simulation results are presented in **Table 2**.

From **Table 2**, when deploying 3 USVs to complete the mission, the time cost is highest while the energy consumption is lowest. Conversely, deploying 5 USVs results in the lowest time cost but the highest energy consumption. Therefore, it can be inferred that as the number of deployed USVs increases, task points are evenly distributed among them, leading to a decrease in the time cost for completing the mission but an increase in the overall energy consumption. The results in **Fig. 5a-c**, where obstacle effects were not considered during task planning, incurred additional energy cost during subsequent path optimization. The mission time cost for 5 USVs in **Fig. 5c** exceeded that of 4 USVs in **Fig. 5b**. However, the overall trend adheres to the aforementioned regular pattern. Hence, both an excess and a shortage of USVs would lead to higher cost. Optimal cost is achieved when deploying an appropriate number of USVs, namely $N = 4$. Thus, it can be concluded that proposing the optimal number of USVs required for mission during

the greedy allocation process contributes to the generation of better task coverage solutions.

By further analysing the results in **Fig. 5** and **Table 2**, superiority provided by the proposed method is evident. When deploying 3 USVs, the planning outcomes are illustrated in **Fig. 5a** and d. In **Fig. 5a**, there is an obstacle blocking the connection between task points 13 and 14 for USV1, resulting in a higher actual distance cost. Therefore, in **Fig. 5d**, the task point allocation for USV1 and USV3 during the greedy allocation phase is adjusted. Additionally, although the task point set of USV2 remains unchanged, the influence of obstacles is considered during path planning. The task execution sequence and path planning result have been optimized, resulting in a reduced actual energy cost for USV2 compared to the outcome in **Fig. 5a**. Similarly, when $N = 4$ and $N = 5$, the total cost of coverage solutions based on IACO method is lower than those based on the initial method. It is evident that the proposed method accounts for environmental factors in both the task point allocation and path planning stages, yielding results more aligned with practical requirements.

4.2. Different methods comparison

To validate the effectiveness of the proposed method, we compared it with some enhanced methods, namely ACO-VP (Li et al., 2023), ISOM (Liu et al., 2019a), and ACO-TTL (Tan et al., 2022) methods. Here, we continue to consider an example of an obstacle scenario with 60 target task points in the area. Unlike previous setups, four target task points in the scenario are designated as sequential task points, with the execution sequence being $T_{S1} \rightarrow T_{S2} \rightarrow T_{S3} \rightarrow T_{S4}$.

Table 1
The design parameters of method.

Indexes	Items
Cost function parameters	$\alpha = 0.4, \beta = 0.2, \gamma = 0.4$ $\sigma_1 = 1, \sigma_2 = 50, \sigma_3 = 2000, \varpi = 0.15$
IACO parameters	$U = 1, V = 5, \rho = 0.1$ $Q = 1, N_{iter} = 100, N_{any} = 50$

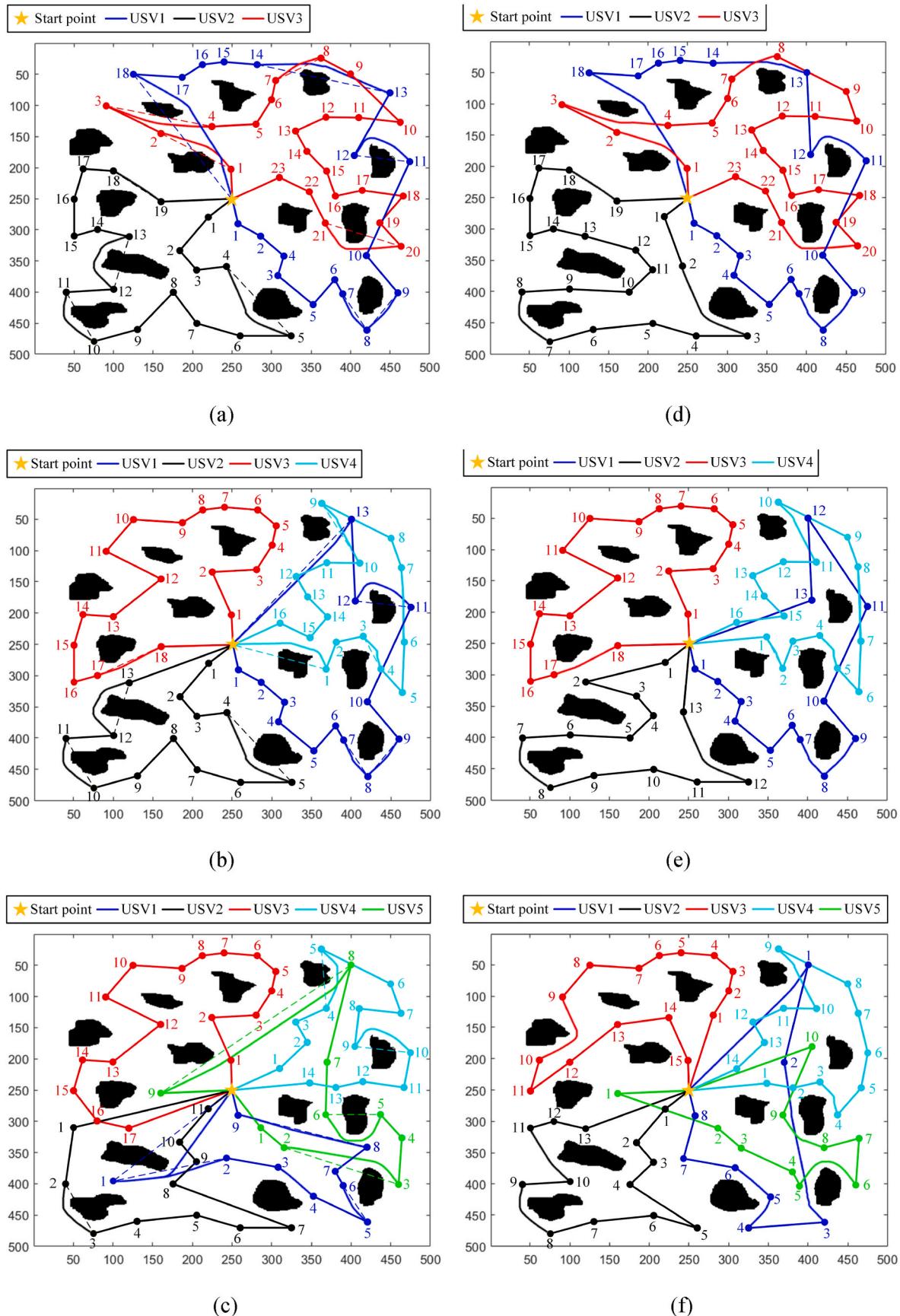
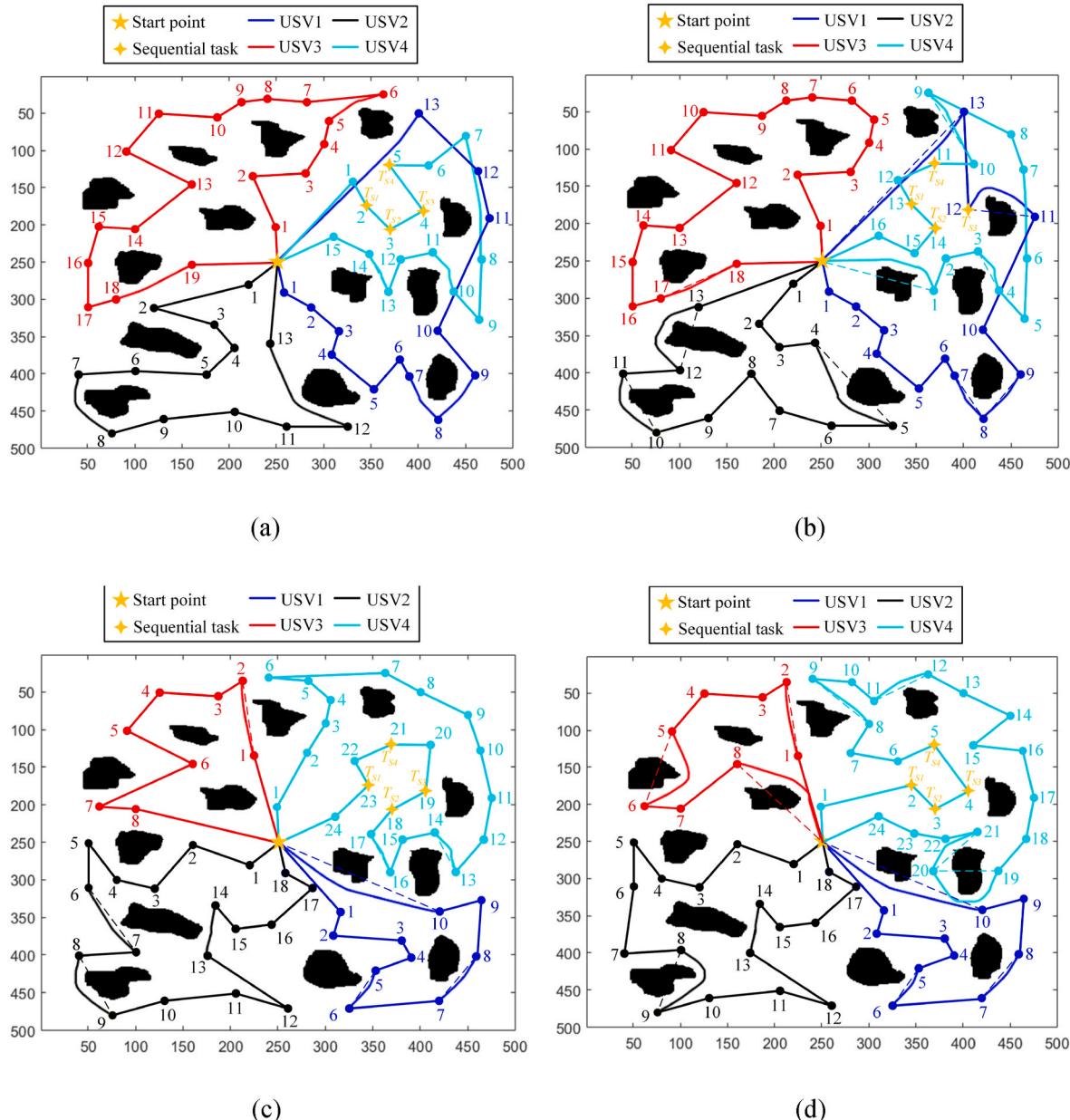


Fig. 5. Target coverage for USVs with 60 task points. (a) Paths based on the initial method of three USVs; (b) paths based on the initial method of four USVs; (c) paths based on the initial method of five USVs; (d) paths based on the proposed method of three USVs; (e) paths based on the proposed method of four USVs; (f) paths based on the proposed method of five USVs.

Table 2

Results data of target coverage with 60 target points in obstacle environment.

Method	Number of USVs	Path length					Time cost	Energy cost	Total cost
		USV1	USV2	USV3	USV4	USV5			
Initial method	3	1485	1328	1475	Un-deployed	Un-deployed	1485	4288	1550
	4	1161	1078	1054	1116	Un-deployed	1161	4409	1453
	5	957	1019	1024	966	1181	1181	5147	1649
The proposed method	3	1446	1318	1482	Un-deployed	Un-deployed	1482	4246	1539
	4	1135	1056	1054	1102	Un-deployed	1135	4347	1427
	5	1076	979	986	917	974	1076	4932	1546

**Fig. 6.** Comparative results. (a) IACO; (b) ACO-VP; (c) ISOM; (d) SOM-TTL.

Firstly, the proposed method is employed to compute the coverage plan. The calculation reveals that deploying 4 USVs results in the lowest cost for task execution. Therefore, to better highlight the performance advantage of the proposed method, the coverage solutions of other methods are also based on deploying 4 USVs. The final comparison results are depicted in Fig. 6. In Fig. 6a, adjustments were made to

optimize the coverage plan because of the sequential task points in the scenario. These adjustments were aimed at adhering to the constraints of task execution sequence without significantly compromising the time consistency performance and total cost of the original solution, as seen in the results from Fig. 5d the task execution sequences for USV1, USV3, and USV4 have been altered. The time consistency performance

Table 3

Comparison results data of different methods.

Method	Path length				Sequential task	Time cost	Energy cost	Temporal consistency performance	Total cost
	USV1	USV2	USV3	USV4					
The proposed method	1091	1056	1170	1036	Finished	1170	4353	3.88%	1450
ACO-VP	1161	1078	1054	1116	Unfinished	1161	4409	3.29%	1453
ISOM	838	1198	774	1325	Unfinished	1325	4133	12.8%	1563
SOM-TTL	838	1211	791	1464	Finished	1464	4304	24.3%	1721

indicator for the coverage solution is 3.88%, with a total cost of 1450.

The ACO-VP method is an enhanced version of the ACO method tailored for addressing multi-agent target point coverage tasks, as shown in the computation results in Fig. 6b. Throughout the computation process, it prioritizes the task consistency performance of each USV. However, its computation of distance cost relies on the Euclidean distance between nodes, limiting its effectiveness in environments with obstacles. Furthermore, it falls short in meeting the constraints of task execution sequence. The ISOM method employs obstacle repulsion fields during neuron movement to prevent collisions, illustrated in the computation results depicted in Fig. 6c. Nonetheless, it allocates task points to USVs based on regions during task assignment, resulting in significant variations in task loads among USVs, the coverage solution has poor time consistency performance. This issue becomes more prominent when task target points are unevenly distributed in the region. Moreover, the ISOM method lacks the capability to handle sequential task points. The SOM-TTL method integrates the tasks treatment list (TTL) method to fulfill the requirements of task execution sequence constraints, as shown in the computation results in Fig. 6d. However, it neglects the impact of obstacles during the computation process, rendering it unsuitable for real-world applications involving USVs. Some relevant data from the simulation results are presented in Table 3.

From Table 3, the proposed method in this research and the ACO-VP method prioritize temporal consistency performance during the task allocation stage. Their aim is to achieve higher temporal consistency performance by sacrificing energy cost, resulting in values of 3.88% and 3.29%, respectively. Although ISOM method and SOM-TTL method boast lower energy costs, their generated solutions exhibit poorer temporal consistency performance, at 12.8% and 24.3%, respectively. USV1 and USV3 complete the mission too quickly, leading to significant resource wastage. Overall, the task allocation method proposed in this research demonstrates superior performance in obstacle environments, with the lowest total mission cost being 1450 and the capability to complete tasks with specific sequential target points. Note that unlike other methods, the proposed method can directly generate safe navigation paths without the need for additional path planning methods to optimize the results.

5. Conclusions and future work

This research proposes a path planning method for coordinating multiple USVs to complete target coverage task in obstacle environments. Since USVs task execution scenarios typically involve various obstacles, this method designs a task point distance matrix based on the LSM method. This matrix can output distance costs between target points considering environmental constraints, and the generated target point distance potential fields can support the subsequent generation of safe collision-free navigation paths in the path planning stage. Building on this research, the task allocation stage employs a greedy allocation strategy to determine the optimal number of USVs for task execution and the task allocation solutions. A new cost function is designed based on time cost, energy cost, and resource cost to evaluate task allocation solutions. Considering special task execution sequence constraints and environmental constraints, the heuristic function and pheromone update function of the ACO method are redesigned in the path planning

stage. The improved method can directly generate collision-free safe paths that meet mission requirements. We conduct multiple sets of simulation experiments to validate the effectiveness of the proposed method.

In the future, we will focus on enhancing the practicality of the proposed method. The method currently only addresses the offline planning phase, overlooking the interference of dynamic obstacles during the mission execution of USVs. This could be improved by integrating the method with previous research on collision avoidance methods (Su et al., 2023). Additionally, the generated paths only consider environmental constraints. When constructing the distance matrix, the influence of sea currents in the environment can be taken into account (Luo et al., 2024b).

CRediT authorship contribution statement

Jing Luo: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yumin Su:** Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Afonso, R., Maximo, M., Galvao, R., 2020. Task allocation and trajectory planning for multiple agents in the presence of obstacle and connectivity constraints with mixed-integer linear programming. *Int. J. Robust Nonlinear Control* 30 (14), 5464–5491.
- Alighanbari, M., 2004. Task Assignment Algorithms for Teams of UAVs in Dynamic Environments. Massachusetts Institute of Technology.
- Bak, S., McLaughlin, J., Renzi, D., 2010. Some improvements for the fast sweeping method. *SIAM J. Sci. Comput.* 32 (5), 2853–2874.
- Gao, S., Wu, J., Ai, J., 2021. Multi-UAV reconnaissance task allocation for heterogeneous targets using grouping ant colony optimization algorithm. *Soft Comput.* 25, 7155–7167.
- Goemans, M.X., 1995. Worst-case comparison of valid inequalities for the TSP. *Math. Program.* 69 (1), 335–349.
- Horbulin, V.P., Hulyantytskyi, L.F., Sergienko, I.V., 2020. Optimization of UAV team routes in the presence of alternative and dynamic depots. *Cybern. Syst. Anal.* 56 (4), 195–203.
- Huang, B., Zhu, C., Xu, Y., 2024. Energy tradeoff-oriented quasi-optimal distributed affine formation maneuver control for electric marine surface vehicles. *IEEE Transactions on Transportation Electrification* (in press).
- Jia, Z., Yu, J., Ai, X., 2018. Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm. *Aero. Sci. Technol.* 76, 112–125.
- Li, Y., Zhang, S., Chen, J., Jiang, T., Ye, F., 2020. Multi-UAV cooperative mission assignment algorithm based on ACO method. In: International Conference on Computing, Networking and Communications (ICNC), pp. 304–308.
- Li, J., Xiong, Y., She, J., 2023. UAV path planning for target coverage task in dynamic environment. *IEEE Internet Things J.* 10 (20), 17734–17745.
- Liu, Y., Bucknall, R., 2018. Efficient multi-task allocation and path planning for unmanned surface vehicle in support of ocean operations. *Neurocomputing* 275, 1550–1566.
- Liu, Y., Song, R., Bucknall, R., Zhang, X., 2019a. Intelligent multi-task allocation and planning for multiple unmanned surface vehicles (USVs) using self-organizing maps and fast marching method. *Inf. Sci.* 496, 180–197.

- Liu, Y., Ma, J., Zang, S., Min, Y., 2019b. Dynamic path planning of mobile robot based on improved ant colony optimization algorithm. In: Proc. Int. Conf. Netw. Commun. Comput., pp. 248–252. Luoyang, China.
- Luo, J., Zhang, Y.H., Zhuang, J.Y., 2024a. Intelligent task allocation and planning for unmanned surface vehicle (USV) using self-attention mechanism and locking sweeping method. *J. Mar. Sci. Eng.* 12 (1), 179.
- Luo, J., Zhuang, J., Jin, M., 2024b. An energy-efficient path planning method for unmanned surface vehicle in a time-variant maritime environment. *Ocean Eng* 301, 117544.
- Ma, D.F., Chen, X., 2023. Neural network model-based reinforcement learning control for AUV 3-d path following. *IEEE Transactions on Intelligent Vehicles* 9 (1), 893–904.
- Ma, W.H., Han, Y.Y., 2023. Ship route planning based on intelligent mapping swarm optimization. *Comput. Ind. Eng.* 176, 108920, 2023.
- Ma, W.H., Hao, S.F., 2021. Scheduling decision model of liner shipping considering emission control area regulations. *Appl. Ocean Res.* 106, 102416.
- MahmoudZadeh, S., Abbasi, A., Yazdani, A., 2022. Uninterrupted path planning system for Multi-USV sampling mission in a cluttered ocean environment. *Ocean engineering* 254, 111328.
- Miao, R., Wang, L., Pang, S., 2022. Coordination of distributed unmanned surface vehicles via model-based reinforcement learning methods. *Appl. Ocean Res.* 122, 103106.
- Odili, J.B., Noraziah, A., Sidek, R.M., 2020. Swarm intelligence algorithms' solutions to the travelling salesman's problem. *Mater. Sci. Eng.* 769 (1), 012030.
- Park, J., Kim, S., Noh, G., 2021. Mission planning and performance verification of an unmanned surface vehicle using a genetic algorithm. *Int. J. Nav. Archit. Ocean Eng.* 13, 575–584.
- Qizilbash, A., Henkel, C., Mostaghim, S., 2020. Ant colony optimization based multi-robot planner for combined task allocation and path finding. In: Proceedings of the 17th International Conference on Ubiquitous Robots (UR). IEEE, pp. 487–493.
- Scholz, A., Henn, S., Stuhlmann, M., 2016. A new mathematical programming formulation for the single-picker routing problem. *Eur. J. Oper. Res.* 253 (1), 68–84.
- Su, Y.M., Luo, J., Zhuang, J.Y., 2023. A constrained locking sweeping method and velocity obstacle based path planning method for unmanned surface vehicles in complex maritime traffic scenarios. *Ocean Eng* 279, 113578.
- Sun, W., Zhang, F., Xue, M., Hu, W., Li, L., 2017. An SOM-based algorithm with locking mechanism for task assignment. In: Proceedings of the IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM). IEEE, pp. 36–41.
- Tan, G., Zhuang, J., Zou, J., 2022. Multi-type task allocation for multiple heterogeneous unmanned surface vehicles (USVs) based on the self-organizing map. *Appl. Ocean Res.* 126, 103262.
- Tang, H., Lin, A., Sun, W., Shi, S., 2020. An improved SOM-based method for multi-robot task assignment and cooperative search in unknown dynamic environments. *Energies* 13 (12), 3296–3314.
- Wang, L.Y., Zhang, J., Li, H., 2007. An improved genetic algorithm for tsp. In: 2007 International Conference on Machine Learning and Cybernetics, vol. 2, pp. 925–928.
- Wang, S., Ma, F., Yan, X., Wu, P., Liu, Y., 2021. Adaptive and extendable control of unmanned surface vehicle formations using distributed deep reinforcement learning. *Appl. Ocean Res.* 110, 102590.
- Xu, G.T., Liu, L., Long, T., 2018. Cooperative multiple task assignment considering precedence constraints using multi-chromosome encoded genetic algorithm. In: 2018 AIAA Guidance, Navigation, and Control Conference, vol. 1859.
- Yang, W.Z., Xan, Y., 2021. Multi-UAV task assignment based on quantum genetic algorithm. *J. Phys. Conf. Ser.* 1824 (1), 012010.
- Yang, H., Qi, J., Miao, Y., Sun, H., 2019. A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization. *IEEE Trans. Ind. Electron.* 66 (11), 8557–8566.
- Yao, P., Lou, Y., Zhang, K., 2023. Multi-USV cooperative path planning by window update based self-organizing map and spectral clustering. *Ocean Eng* 275, 114140.
- Zhou, B., Huang, B., Su, Y., 2024. Interleaved periodic event-triggered communications based distributed formation control for cooperative unmanned surface vessels. *IEEE Transact. Neural Networks Learn. Syst.* 10, 123–135.