Autonomous Intelligent Systems

ORIGINAL ARTICLE Open Access

Multiple unmanned ship coverage and exploration in complex sea areas

Feifei Chen[1,2] and Qingyun Yu[1*]

Abstract

This study addresses the complexities of maritime area information collection, particularly in challenging sea environments, by introducing a multi-agent control model for regional information gathering. Focusing on three key areas—regional coverage, collaborative exploration, and agent obstacle avoidance—we aim to establish a multi-unmanned ship coverage detection system. For regional coverage, a multi-objective optimization model considering effective area coverage and time efficiency is proposed, utilizing a heuristic simulated annealing algorithm for optimal allocation and path planning, achieving a 99.67% effective coverage rate in simulations. Collaborative exploration is tackled through a comprehensive optimization model, solved using an improved greedy strategy, resulting in a 100% static target detection and correct detection index. Agent obstacle avoidance is enhanced by a collision avoidance model and a distributed underlying collision avoidance algorithm, ensuring autonomous obstacle avoidance without communication or scheduling. Simulations confirm zero collaborative failures. This research offers practical solutions for multi-agent exploration and coverage in unknown sea areas, balancing workload and time efficiency while considering ship dynamics constraints.

Keywords: Multi-agent control model, Regional coverage, Collaborative exploration, Agent obstacle avoidance

1 Introduction

Maritime area information collection encompasses gathering diverse data in the marine environment, essential for scientific research, search and rescue operations, landing and beaching activities, sea area warnings, and ship information [1]. This process holds immense importance in developing and utilizing marine resources, safeguarding the marine environment, rescuing marine accidents, and maintaining national territorial waters security [2]. However, the marine environment's complexity and variability, influenced by factors like waves, currents, meteorological conditions, and marine life, pose significant challenges. Despite advances in ocean information collection equipment, including ocean observation ships, satellite remote sensing, and drones, the vastness and ever-changing nature of the ocean continue to affect detection accuracy, posing challenges to the reliability of ocean detection equipment.

Current research in complex sea areas has primarily focused on enhancing the autonomy and efficacy of unmanned ships, specifically in areas of path planning, dynamic task allocation, and collaborative fleet operations [3]. While recent studies have employed various computational methods such as artificial intelligence, machine learning, and heuristic approaches to refine the precision and dependability of these unmanned systems, several critical challenges persist [4]. The ocean's unpredictable weather patterns and fluctuating sea states can profoundly impact the stability and performance of unmanned ships [5]. Furthermore, integrating these vessels into dense maritime traffic, which includes both manned and unmanned crafts, raises intricate navigational and safety concerns [6]. The necessity for robust communication systems to facilitate timely and accurate data exchange between multiple agents and control centers adds another layer of complex-

*Correspondence: qingyunyu@tongji.edu.cn
[1] Department of Control Science and Engineering, College of Electronics and Information Engineering, Tongji University, Shanghai, 200092, China
Full list of author information is available at the end of the article

Springer

ity to their widespread implementation. Overcoming these obstacles is pivotal in advancing the capabilities of unmanned ships and fully unlocking their potential in complex sea operations [7].

To tackle the aforementioned challenges, this study presents a multi-unmanned ship coverage detection system tailored for complex sea areas. Our comprehensive approach incorporates three key strategies: regional coverage, collaborative exploration, and intelligent agent collaborative collision avoidance. By adopting these methods, we aim to enhance the efficiency and safety of unmanned ship operations in dynamic and challenging maritime environments. In the subsequent sections of this paper, we delve into the specifics of our proposed framework, present the results of our research, and discuss their implications. Ultimately, our goal is to contribute to the evolving field of unmanned ship technology and pave the way for more effective and safer maritime operations.

2 Previous works

Maritime area information collection is the process of obtaining various information in the marine environment, including marine physics, chemistry, biology, and other aspects. It is of great significance to the development and utilization of marine resources and the protection of the marine environment [8]. Modern ocean information collection technology has developed to a relatively mature stage, including ocean observation ships, satellite remote sensing, drones, etc. However, the marine environment is complex and changeable. It is affected by the interaction of waves, currents, tides, meteorological conditions, water temperature, salinity, marine life, and other factors, making the marine environment increasingly complex. In addition, factors such as seabed topography, seabed sediments, seabed organisms, and marine pollution will also cause changes in the marine environment, which will have a greater impact on the detection accuracy of traditional platform detection loads. In order to cope with the complex and changeable marine environment, on the one hand, high target positioning accuracy can be obtained by observing targets at close range; on the other hand, using multiple agents to coordinately complete target detection in designated complex sea areas can effectively improve detection accuracy and reduce safety risks.

This article conducts multi-agent dynamic planning for a designated irregular closed sea area. Each agent is required to achieve effective coverage of the entire area and ensure that it can detect all static targets in the area. It also discovers and detects multiple static targets with different attributes. During the process, it is possible to control single or multiple agents to approach a static target at a specified angle, speed, and direction to complete the observation action. To solve this problem, it is necessary to realize three main tasks: regional coverage path planning, exploration task allocation, and collaborative obstacle avoidance of multi-agent.

2.1 Area coverage path planning

Area coverage path planning refers to using mobile agents to traverse the target environment area within their physical contact range or within their sensor sensing range and try to meet optimization goals such as short task completion time, few repeated paths, or small untraversed areas. Full coverage applications of intelligent agents appear in various aspects such as military, agriculture, industry, commerce, disaster relief, and urban life, such as automatic demining, crop harvesting, air traffic inspection, etc. Compared with single-agent coverage path planning, in addition to considering the traversal and non-duplication of paths, it also needs to consider the balance of task allocation, the challenge of resource conflict, the uncertainty of the number of agents and the task environment, and the uncertainty of the agent negotiation and conflict, prior information dependence, algorithm adaptability, and scalability, etc. The work of [9] extended the single-agent spiral spanning tree coverage algorithm to multiple agents and proposed multi-agent spanning tree coverage (MSTC) with backtracking optimization, which improved the efficiency and robustness of agent coverage. At present, multi-agent coverage path planning algorithms can be divided into three categories: no planning, centralized planning and discrete planning. The random path planning algorithm [10] does not require map information, and the agent uses random reactive motion planning. Achieving ideal coverage results at the expense of time or investing a large number of agents. The centralized planning algorithm allocates areas under the condition of obtaining global environmental information, and the agents each complete the coverage tasks in the allocated areas, which can produce more efficient paths and greatly shorten the duration of completing the coverage tasks. However, there are many limitations in practical applications. The discrete programming algorithm does not have a centralized task allocation mechanism, and each agent exchanges information through explicit or implicit communication to collaborate to complete coverage. The discrete programming algorithm is an online, real-time planning algorithm that can adapt to coverage in unknown environments, but its effect is closely related to the algorithm itself.

Much work has been devoted to improving the performance of area coverage path planning. In order to take advantage of the power of the centralized framework, the authors of [11] proposed a backward horizon centralized online homogeneous multi-agent planner based on goal assignment, which ensures complete coverage of the unknown workspace and has advantages in coverage completion time. In [12], it proposed a new multi-agent continuous coverage strategy, aiming to optimize coverage performance and obtain a more equal coverage path for each agent, while ensuring the minimization of obstacle avoidance and coverage cycles. The work of [13] provided a new multi-agent coverage path planning algorithm (MACoPP) to optimize the load balance between agents by considering the initial state of the agents while minimizing the completion of non-convex area monitoring. The authors of [14] proposed a path planner formulated in continuous time to improve the reliability probability measure of area coverage. The study of [15] proposed a complete coverage path planning (CCPP) algorithm based on energy compensation and obstacle vectoring (ECOV). This algorithm has the advantages of a low path coverage repetition rate and high coverage rate and can be adapted to complex map models. The authors of [16] proposed a coverage method based on an iterative heuristic algorithm to minimize the number of coverage lines and thereby reduce the number of turns in the path. The authors of [17] proposed an agent-decentralized control strategy based on graphical neural networks, using communication between agents to display and share information between adjacent agents for control decisions, achieving higher coverage quality.

Due to the strong task practicality of area coverage path planning, many works have proposed algorithms for specific tasks or scenarios. Many works [18, 19] have studied the coverage path planning problem of autonomous or heterogeneous unmanned aerial vehicles (UAVs) over a limited area. The study of [20] aimed at finding the optimal path for a set of UAVs with limited power constraints to fully cover. For the task of multiple non-overlapping convex polygon regions, a Branch-and-Bound (BnB)-based method was developed to find the (nearly) optimal path and a genetic algorithm-based method was developed to effectively solve large-scale problems under different objectives. In [21], it proposed a new multi-agent coverage path planning algorithm which can improve the adaptability and robustness of the system in unknown complex environments through interaction and collaboration between agents. The authors of [22] proposed a multi-UAV coverage path planning (CPP) framework for detecting large-scale and complex three-dimensional structures. Aiming at the problem of full coverage of multi-agent space when the density function on the domain is initially unknown, the study of [23] asked a group of mobile agents to learn the spatial field on a domain at the same time and distribute themselves in space to optimize the coverage path.

Multi-agent area coverage needs to face challenges such as agent negotiation and conflict, optimization of coverage performance, and high computing power and time costs. For the sea area scenario of this competition, this work uses the optimal area allocation and coverage path optimization based on cellular discrete decomposition and simulated annealing (SA) algorithm for the polygonal sea area to be explored. In the initial stage, the agent performs coverage and exploration work at the same time. Until the coverage task is completed, all agents switch to the exploration working mode. This path planning strategy optimizes the implementation effect of the task, can complete basic full coverage, and has certain advantages in indicators such as coverage completion time.

2.2 Multi-agent task allocation

When a multi-agent system handles full coverage actions, it needs to go through several stages, including task decomposition, task assignment, task scheduling, etc. These three stages are collectively called task distribution. Multi-agent task allocation (MATA) plans the best task execution sequence for each agent in the system while satisfying various constraints. The system execution cost is completing all the tasks required by the system in the smallest possible case is essentially a combinatorial optimization problem. The MATA task allocation scheme should cope with the three factors of acceleration ratio, resource conflict (such as path conflict), uncertainty in the number of agents (such as task completion time with deadline), or environmental uncertainty (such as environmental probabilistic transition). At present, the main methods to solve multi-agent task allocation include linear programming method, market mechanism-based method, heuristic search algorithm, etc.

Allocation methods based on mathematical programming can convert multi-agent task planning problems into mathematical operations through linear programming algorithms. Although the optimal solution can be obtained, it has high time and space complexity. When the problem scale increases At this time, the difficulty of solving also increases sharply, and the time consumption increases exponentially. The heuristic allocation method [24] gradually approaches the optimal solution by continuously iterating an initial solution. It has the advantages of strong adaptability and fast search speed, but it generally obtains a local optimal solution. The method based on the market mechanism [25] allows multiple agents to bid or negotiate based on market conditions to achieve task allocation, mainly including auction algorithms and contract network algorithms. In general, the task allocation strategy based on heuristic algorithms has better results in solving MATA problems in complex environments and is the best strategy to be applied to actual task allocation.

The work of [26] aimed at the problem of multi-UAV collaborative search and rescue. They considered the problem that the UAV performing the task could not establish a connection with the control center. By predicting the network topology, they set up relay tasks to distribute and repair the connection to ensure network connectivity during task execution., combined with the auction algorithm to achieve real-time task allocation. The authors of

[27] aimed at the MRTA problem under limited communication range and resource constraints, using a solution strategy based on multi-hop networks combined with distributed auctions to solve the real-time optimal allocation problem under resource and power consumption. In [28], it aimed at the MATA problem where tasks have execution time and time windows. Each agent is determined by battery life and the upper limit of the execution task. They proposed a method based on dynamic programming combined with a distributed auction algorithm to realize multi-agent distributed real-time tasks. distribute. In response to dynamic disturbances or failures, the study of [29] introduced dynamic iterative task allocation graph search (DITAGS) to achieve resilient task scheduling and motion planning problems in dynamic environments involving heterogeneous teams. The authors of [30] considered the problem of assigning tasks to a team of heterogeneous mobile agents. The study of [31] proposed a heuristic-based approach to generate task reassignment suggestions to handle emergencies and complete tasks faster by multi-agent teams operating in challenging environments. The work of [32] designed two group-based distributed auction algorithms to implement multi-agent dynamic transportation task allocation problems, and quantified the potential of the algorithm in terms of the number of agents used and the capacity of the agents. The authors of [33] innovated real-time multi-agent task allocation by proposing a hypergraph MRS model and a hypergraph search algorithm to coordinate heterogeneous multi-agent systems to meet the frequent reconstruction caused by dynamic production needs. Analyzed and solved challenges caused by frequent production process adjustments. The work of [34] proposed a game theory multi-agent task allocation framework that enables a large group of agents to optimally allocate tasks in a dynamically changing environment. The algorithm enables the agent to learn and asymptotically achieve optimal smooth task allocation.

Due to the complexity and uncertainty of crowd intelligence tasks and collaborative agents, most task allocation mechanisms have problems such as a narrow scope of application, low matching accuracy, and low allocation efficiency. This work aims at optimizing the efficiency of task allocation and scheduling. We design a cost function related to distance and use an improved greedy algorithm to design a task allocation strategy to dynamically allocate exploration tasks to each agent in stages, that is, to its nearest task point., to shorten task allocation time and overall running time.

2.3 Multi-agent collaborative obstacle avoidance

The obstacle is one of the core technologies of multi-agent coverage path planning, that is, planning a collision-free and minimal-cost path between the starting point and the endpoint. In addition to completing tasks efficiently, intelligent agents must also comply with external constraints to ensure the safety of the navigation process and not cause immeasurable losses due to collisions with other agents or various obstacles. In the early days of research, most scholars focused on static, simple situations with a single obstacle, and achieved good results. However, the real environment is complex, dynamic, and even unknown. How to operate in a dynamically complex, unknown environment? Medium and high-efficiency obstacle avoidance and path planning are still a hot and difficult research topic. Research on obstacle avoidance problems mainly focuses on the following aspects: 1) Obstacle avoidance with different agent speeds and sizes; 2) Treatment of regular and irregular obstacles: Regular obstacles gener- ally refer to obstacles that can be modeled with mathematical models. Since irregular obstacles cannot be modeled directly, they are generally handled by vertex approximation methods; 3) Obstacle avoidance in static and dynamic environments; 4) Single agent and multi-agent: a single agent does not need to consider the interaction between agents. Multi-agent systems need to consider the non-stationarity of the environment. It can be divided into centralized and distributed. The centralized system is coordinated in real-time through a central controller, while each agent in the distributed system makes independent decisions based on onboard sensors, which is suitable for deploying a large number of agents with a low computing budget.

Traditional obstacle avoidance methods include artificial potential field method, optimal interactive collision avoidance method, ant colony algorithm, etc. The artificial potential field method is based on the principle that the target generates gravitational force and the obstacle generates repulsive force. By establishing a potential field in the environment, the resultant force generated by the target and obstacles acts on the agent, thereby guiding the agent's motion state [35]. The work of [36] proposed the discrete artificial potential field (DAPF) algorithm, which implements a dynamic method by constructing a discrete potential field. The path length planned by the algorithm is more reasonable and the running time is shortened. However, this algorithm requires a known global map. It can be used only after downloading. The optimal interactive collision avoidance method (ORCA) transforms the dynamic collision avoidance problem into a quadratic linear programming problem and solves it in the convex region of the velocity plane [37]. In [38], applied the probabilistic velocity obstacle (PVO) method to dynamic occupancy grids and proposed a method to estimate collision probability with uncertainty in the position, shape, and speed of obstacles, occlusion, and limited sensor range. directly affects calculations. Experi- ments have proven that the agent can safely navigate obstacles at a constant linear speed, but it lacks the ability to handle obstacles at nonlinear speeds. These algorithms can handle many obstacle avoidance problems, but they may fall into local optima, and they still need to be further improved in terms of adaptability, computational efficiency, autonomy, and stability. In recent years, artificial intelligence algorithms have also been widely used in collaborative obstacle avoidance problems in multi-agent environments. Based on the powerful environment perception capabilities of deep neural networks, supervised learning was first applied. Many works use airborne sensors to input motion data into neural networks, and convolutional neural networks are commonly used to extract environmental features and train to output simple obstacle avoidance instructions [39]. In addition, deep reinforcement learning (DRL) has also received a lot of attention [40, 41] and has begun to be widely used in the field of obstacle avoidance. Inspired by a recognition neuron of locusts, the authors of [42] proposed an artificial neural network model and optimized the model parameters to enhance the adaptability of the model. The model was combined with a visual sensor to realize the intelligent control of the agent. Dynamic obstacle avoidance.

This work uses the optimal interaction speed obstacle method to handle the obstacle avoidance task. In the optimal interaction speed obstacle method, each agent assumes half of the obstacle avoidance responsibility, so it is only suitable for homogeneous agents to avoid obstacles. Faced with situations such as heterogeneous agents, stationary obstacles, and asymmetric agents, the evenly divided responsibility allocation mechanism cannot respond flexibly and limits navigation performance. This work improves this shortcoming. For different obstacle avoidance situations, the obstacle avoidance responsibilities are flexibly allocated based on priority classification, and the sum of the responsibilities of each pair of colliding bodies is ensured to be 1. This improvement expands the two situations of dynamic obstacle avoidance and static obstacle avoidance and realizes the optimization of the hierarchical responsibility allocation mechanism.

3 Methodology

3.1 Problem statement

The ocean environment is complex and changeable, which has a great impact on the detection accuracy of traditional platform detection loads. Under normal circumstances, target observation at close range can achieve higher target positioning accuracy. In the process of discovering and exploring multiple static targets with different attributes, it is necessary to control single or multiple agents to approach static targets at a specified angle with a steady speed and direction. The target completes the observation action to ensure that the target's high-precision position, speed, image or video information can be obtained. Using intelligent agents to assist in target detection in designated complex sea areas can effectively improve detection accuracy and reduce safety risks. To carry out full area coverage and target detection within a designated irregular closed sea area, the following three problems need to be solved:

Area coverage (Problem 1): It is necessary to use multi-agent to complete effective coverage of the entire area in the irregular closed sea area to discover all static targets, and the agent cannot enter the no-navigation zone.

Collaborative exploration (Problem 2): The agent needs to complete the exploration tasks of all static targets in a specified way, where the attributes of the static targets will change dynamically.

Agent obstacle avoidance (Problem 3): The agent needs to maintain safe navigation and cannot enter the threat radius of other agents or static targets. It needs to avoid collisions. For the above problems, it is necessary to design a multi-agent control model for regional information collection. Through the model, dynamic task allocation and path planning can be performed for multi-agents, so that multi-agents can collaboratively complete the tasks of regional coverage and static target detection. The following detailed analysis is made for the three sub-problems.

For Problem 1, it can be transformed by considering the ship as a particle with a certain detection radius and detection angle range (only regarded as a particle when covering the path planning) and considering the static target and the area within the threat radius as obstacles. A multi-agent collaborative full-coverage path planning problem in a known closed plane area. Specifically, the goal of this problem is a minimax problem, so that the agent with the longest exploration time takes the shortest time, that is, the task completion time is the shortest. For this problem, it is necessary to design a region allocation and path planning algorithm so that the agent can complete the coverage task of the entire two-dimensional area in the shortest possible working time.

For Problem 2, considering the detection of static targets as a task, it is necessary to design an internal decision-making algorithm for the agent to enable the agent to choose between the coverage task and the detection task. To further consider the multi- agent scenario under shared observation conditions, it is also necessary to design a multi-agent scheduling algorithm. Through the agent's internal decision-making algorithm and scheduling algorithm, the agent can complete the coverage and exploration tasks as much as possible.

For Problem 3, static targets and other agents are regarded as static obstacles and dynamic obstacles. It is necessary to design an obstacle avoidance algorithm to ensure that the agent remains as safe as possible during navigation. For intelligent agents, autonomous obstacle avoidance should be a basic ability, so the obstacle avoidance algorithm should be distributed without the need for central server scheduling.

3.2 Proposed model

To address the three sub-problems, this paper adopts regional coverage strategy, collaborative exploration strategy, and intelligent agent collaborative collision avoidance

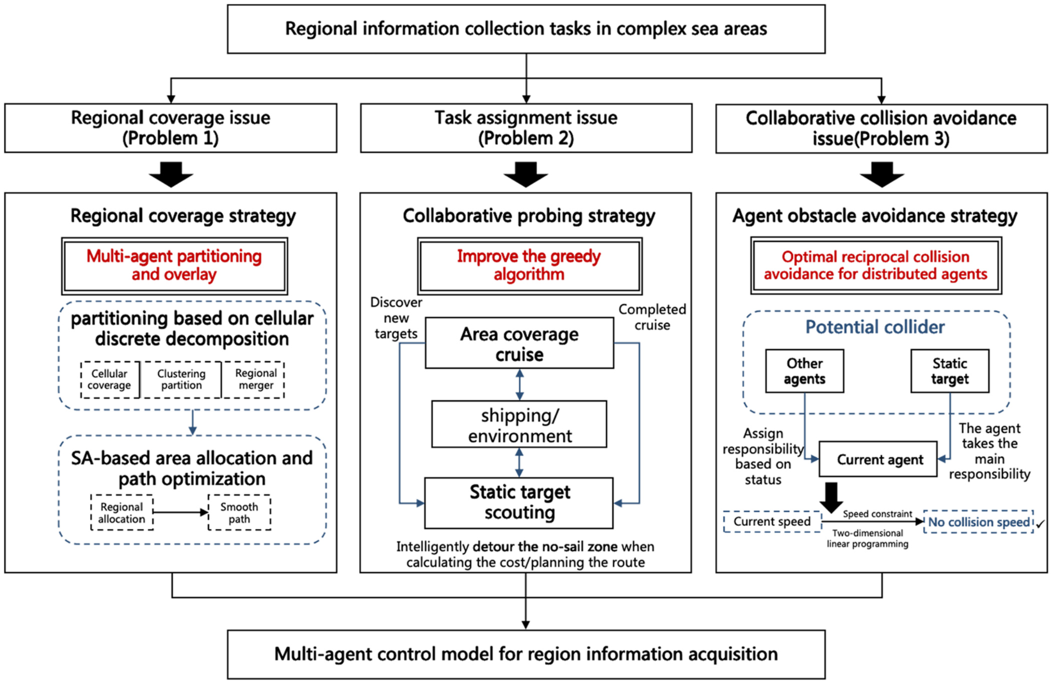
Regional information collection tasks in complex sea areas
Regional coverage issue (Problem 1)
Task assignment issue (Problem 2)
Collaborative collision avoidance issue(Problem 3)
Regional coverage strategy
Multi-agent partitioning and overlay
partitioning based on cellular discrete decomposition
Cellular coverage
Clustering partition
Regional merger
SA-based area allocation and path optimization
Regional allocation
Smooth path
Collaborative probing strategy
Improve the greedy algorithm
Discover new targets
Completed cruise
Area coverage cruise
shipping/ environment
Static target scouting
Intelligently detour the no-sail zone when calculating the cost/planning the route
Agent obstacle avoidance strategy
Optimal reciprocal collision avoidance for distributed agents
Potential collider
Other agents
Static target
Assign responsibility based on status
The agent takes the main responsibility
Current agent
Current speed
Speed constraint
No collision speed ✓
Two-dimensional linear programming
Multi-agent control model for region information acquisition

Figure 1 Multi-unmanned ship coverage detection system in complex sea areas

strategy respectively to form a multi-unmanned ship coverage detection system in complex sea areas, as shown in Fig. 1.

1) *Coverage Problem*

In addressing the coverage problem, several challenges arise, including heterogeneous multi-agent systems, arbitrarily shaped areas, and the distribution of no-navigation zones. To tackle these issues, we adopt a two-step strategy: first, we discretize the target area into a honeycomb-like grid, and then optimize the path through all cellular points. However, given the vastness of the site and the high number of agents involved in practical scenarios, finding the optimal solution can be computationally intensive and time-consuming, posing challenges for engineering applications.

Therefore, we decompose the problem into two smaller optimization tasks: regional allocation and intra-regional path planning. After clustering the honeycomb points, we utilize the simulated annealing algorithm for regional allocation. Subsequently, leveraging the genetic algorithm, we smoothly plan the path within each region by balancing the longest path, the shortest path, and the number of path turns. This approach ensures efficient planning within smaller areas, achieving a balance between computational cost and time. Additionally, by considering the boundaries of no-navigation zones and regional borders, we fine-tune the agent's navigation path by relocating nearby coverage task points, thereby maintaining optimal coverage while ensuring efficient detours.

2) *Exploration Problem*

When addressing the exploration problem, we encounter several complexities, such as dynamic environmental changes, numerous exploration constraints, and heterogeneous agents. To overcome these difficulties, we employ an optimal cost-greedy exploration strategy. Upon discovering an agent, we immediately initiate an exploration, record the static target, and continue with the coverage tasks. As agents complete their assigned coverage duties, we dynamically adjust their detection targets by considering factors like travel time to static targets, exploration intervals, and the number of required probes. This real-time cost estimation enables intelligent agents to collaborate efficiently in detecting static targets.

3) *Collision Avoidance Problem*

To enhance collaborative effectiveness and address collision avoidance between agents and static targets during navigation and exploration, we adopt the optimal reciprocal collision avoidance (ORCA) algorithm. We introduce the concept of agent navigation priority, giving the highest priority to agents in the exploration state. This allows agents with lower priorities to bear more speed variations. Even in the absence of communication, ship speeds can be controlled through predefined consensus to prevent collisions. This approach ensures collision avoidance among

Table 1 Description of symbols

S	Mission area (excluding no-navigation zones)
S_i	Task sub-area
N_A	The number of agents
N_O	The number of static targets
$A = \{A_1, A_2, \ldots, A_i, \ldots, A_{N_A}\}$	Agent collection
$O = \{O_1, O_2, \ldots, O_i, \ldots, O_{N_O}\}$	Static target set
v_{sail}	Agent attributes: Maximum sailing speed
v_{detect}	Agent properties: Maximum exploration speed
r_{detect}	Agent properties: Agent detection target distance
r_{threat}	Agent attributes: Agent threat range
t_{detect}	The time for an agent to complete continuous exploration of the same target
P_i	The agent A_i's navigation path in the task area

agents without compromising the completion of exploration activities.

By incorporating these strategies, we aim to enhance the efficiency and effectiveness of multi-agent systems in various scenarios, including coverage, exploration, and collision avoidance.

3.3 Assumptions

Some of the details of this model are assumed as follows:

1) Agents share global observation information and the synchronization of information is without delay. For example, as long as a static target enters the maximum detection target distance of an agent, the attributes of the static target can be synchronized to all agents immediately. The body is knowable.

2) It is assumed that there are no unexpected situations during the navigation and exploration process of the agent, and it can run according to the set state and speed.

3.4 Notations

The notations of the model are shown in Table 1.

3.5 The area coverage model

Regarding the area coverage problem in unknown sea areas, since the boundaries of the task area are known, the area coverage of multi-agent can be divided into two sub-problems, namely the area division problem and the single-agent coverage path problem. First, the task area is assigned to each agent through a reasonable regional division. Each agent is responsible for a sea area and completes the coverage path solution of a single agent in the sea area. Combining the solutions of the two sub- problems can solve the problem of effective coverage of the entire area by multi-agents in irregular closed sea areas.

When dividing the area, the task area S needs to be divided into sub-areas S_i.

$$S = U_{i=1}^{N_A} S_i. \tag{1}$$

We can determine a series of points in the task area and use these points to draw circles with a certain radius so that the task area is completely covered by a series of circles. The work of [43] proposed a minimum circle complete coverage method. The hexagonal mesh structure is formed after the circle centers are connected. Figure 2 shows a schematic diagram of the minimum circle covering a rectangular area.

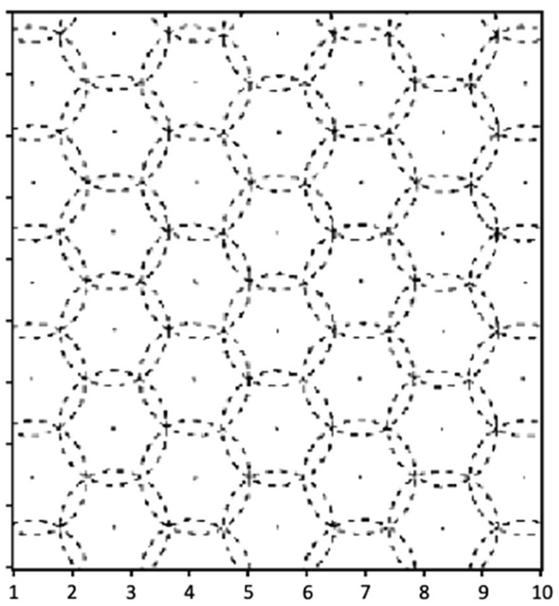

Figure 2 An illustration of minimum circle covering rectangular area

Then, assuming that the task area S_i is assigned to agent Ai for coverage, the path $P_i = \{P_i^j, \text{j} = 1, 2, \ldots, \}$ of the agent A_i should satisfy the following:

$$Si = U_j D\left(P_i^j, r_{\text{detect}}\right), \tag{2}$$

where P_i^j is the point that constitutes the path P_i, $D(P, r) = \{Q \| \|Q - P\| < r\}$ representing a circular area with the point P as the center and r as the radius.

Since the information inside the mission area is unknown before coverage begins, obstacle avoidance of static targets in the area and detection while covering cannot be considered in advance when dividing the area. In order to achieve efficient regional coverage, when the divided area is assigned to a single agent for coverage, each agent must plan a path as short as possible.

In summary, the multi-agent area coverage problem can be formulated as a constrained optimization problem:

$$\text{minmax}\big(|P_1|, |P_2|, \dots, |P_A|\big), \tag{3}$$

$$\text{s.t.} \quad S = U_{i=1}^{N_A} S_i = U_{i=1}^{N_A} U_i D\big(P_i^j, r_{\text{detect}}\big). \tag{4}$$

However, considering that in practical applications, effective coverage and coverage time in the entire area often require balance and trade-offs, and full coverage is not a strong constraint, it is more reasonable to express this problem with the following multi-objective optimization model:

$$\max\big|U_{i=1}^{N_A} S_i\big| = \big|U_{i=1}^{N_A} U_j D\big(P_i^j, r_{\text{detect}}\big)\big|. \tag{5}$$

For the aforementioned coverage path problem, optimal area allocation and coverage path optimization based on a simulated annealing (SA) algorithm are used. Considering that the initial positions and speeds of the agents are different, the area that each agent is responsible for covering should be allocated according to certain optimal criteria. Therefore, the shortest time for an agent to traverse all cellular points starting from its respective starting point is considered as the goal, and a heuristic algorithm is used for point allocation and path planning of each agent. In terms of specific implementation, a simulated annealing algorithm is used, and the solution is defined as a sequence of cellular points that an agent passes through. The sequence elements are the numbers of each cellular point. The total number of elements in each sequence is the cellular points that the corresponding agent passes through. The sequence of the inner elements represents the order of the honeycomb points on the path. Through SA optimization, a solution that minimizes the target value is finally obtained, that is, the optimal coverage path of each agent.

3.6 The collaborative exploration model

The problem of collaborative exploration of static targets, it is decomposed into two problems: internal decision-making of agents and scheduling between agents. The internal decision-making problem of the agent refers to the fact that the agent needs to make local decisions while performing the coverage task, and decide whether it needs to complete the exploration task of the observed static target nearby. Since there is a time interval for the agent to detect the same static target in the review phase, it is a good strategy to detect a new static target every time it is discovered in the coverage phase, which can greatly reduce the workload of the exploration phase. Scheduling between agents refers to: after the agents complete their own coverage tasks, they enter the collaborative exploration stage. If there are still unexplored static targets in the area at this time, the central server needs to schedule the agents to the corresponding static target points to complete the exploration mission.

First, assuming that the agent already knows the attributes of all static targets in the task area, and all agents only perform exploration tasks, then for such a problem, the goal is to complete more effective explorations in a shorter time. Since a static target may have multiple exploration tasks, all the tasks to be explored in the task area are recorded as $\text{TASK} = \{\text{TASK}_1, \text{TASK}_2, \dots, \text{TASK}_j, \dots\}$, including a total of tasks $\sum_i^{N_O} n_i$. The set of exploration tasks completed by the agent A_i is recorded as task_i.

The time it takes to complete the exploration includes the time it takes for the agent to travel from its current position to the exploration starting point, and the waiting time t_{detect} (if any) for the same agent to continuously explore the same static target. Then the time objective of this problem can be expressed as:

$$\begin{cases} \min\max\big(\frac{|P_i|}{V_{\text{sail}}} + t_{\text{detect}}\big), & i = 1, 2, \dots, N_A, \\ \text{s.t.} \quad U_{i=1}^{N_A} \text{task}_i = \text{TASK}. \end{cases} \tag{6}$$

However, considering that in practical applications, complete exploration of all targets and exploration time often require balance and trade-offs, and full exploration is not a strong constraint, it is more reasonable to express the problem with the following multi-objective optimization model:

$$\begin{cases} \min\max\big(\frac{|P_i|}{V_{\text{sail}}} + t_{\text{detect}}\big), & i = 1, 2, \dots, N_A, \\ \max |U_{i=1}^{N_A} \text{task}_i|. \end{cases} \tag{7}$$

The premise of the above model is that the agent completes area coverage and static target detection respectively. In the actual environment, the agent may perform detection at the same time during the navigation process, that is, multiple agents navigate in a location-based environment, and the environment becomes partial over time. It can be seen that in this case, the multi-agent must continue to carry out area coverage and static target detection at the same time. The optimization model of this comprehensive problem can be established as:

$$\begin{cases} \min\max\big(\frac{|P_i|}{V_{\text{sail}}} + t_{\text{detect}}\big), & i = 1, 2, \dots, N_A, \\ \max |U_{i=1}^{N_A} \text{task}_i|, \\ \max |U_{i=1}^{N_A} S_i| = |U_{i=1}^{N_A} U_j D(P_i^j, r_{\text{detect}})|. \end{cases} \tag{8}$$

In view of the above multi-objective optimization model, this paper proposes an improved greedy algorithm to

Algorithm 1 The improved greed algorithm for exploration

```
for agent $A_i$ do
    if $A_i$ has completed the coverage task then
        traverse all static targets in the system, calculate $A_i$'s cost of detecting each target, and select the static target with the smallest detection cost $O_x$;
        if the number of probes required $m$ for $O_x$ is less than the number of agents selected for probing $n$ then
            select the agent $A_j$ with the highest current exploration cost among these agents; if $A_j$'s exploration action has not yet started, reassign the static target to it;
        else
            let the agent $A_i$ go and explore $O_x$;
        End if
    Else
        let the agent $A_i$ go and explore $O_x$;
    End if
End for
```

achieve scheduling between agents. This method is a dynamic scheduling method that can make real-time and dynamic decisions for agents. In the improved greedy algorithm in the exploration phase, the central server implements scheduling between agents by assigning agents to perform exploration tasks at the static target point with the smallest exploration cost. The calculation of the exploration cost is based on the distance between the agent and the static target and the remaining distance of the target. interval time. Specifically, the central server will estimate the time it takes to explore the target based on the distance between the agent and the static target, compare it with the remaining interval time of the target, take the larger value between the two, and traverse the system. From all discovered static target points, the static target with the lowest cost is selected and assigned to the agent to explore.

The innovation of this improved greedy algorithm is that the "static target grabbing" part is added to the algorithm. Specifically, after the central server assigns a static target O to agent A, it will update the cost of agent A''s detection of the static target O in real-time. When allocating static targets to other agents, if an agent A' detects the static target O and the cost is much less than that of the agent A. The central server will schedule the agent A' to explore the static target O according to the situation and assign a new static target to the agent A. The specific process of the improved greedy algorithm is shown in Algorithm 1.

3.7 The agent obstacle avoidance model

Since the current multi-agent is performing tasks in a large-scale sea scene, the key issue that needs to be considered is the efficient regional information collection (coverage and exploration) multi-agent control method in the overall task area. Therefore, the agent does not avoid obstacles. Considering the optimization models of the previous two sections, it is only regarded as the local adjustment and planning that needs to be carried out after the agent solves the path. The following is a problem modeling of the obstacle avoidance sub-problem of the agent.

For the agent obstacle avoidance problem, the distributed optimal reciprocal collision avoidance (ORCA) method is adopted in this study. The dynamic collision avoidance problem is transformed into a quadratic linear programming problem and solved in the convex area of the velocity plane. This algorithm is the first method that can ensure the local collision-free motion of a large number of agents in a cluttered workspace, but it is not considered the problem of static obstacles appearing in the environment. This model limits the speed direction of a plane. As long as the agent's speed falls within this half-plane, the collision can be avoided.

For the two agents, recorded as A_1, A_2, their positions can be expressed as coordinates p_1, p_2, and their speeds can be expressed as v_1, v_2. Based on the position and speed of A_2, the agent A_1 plans to derive speed obstacles and collision avoidance speed offset vectors. The mathematical expression of the speed obstacle and collision avoidance speed offset vector is as follows:

$$VO^{\tau}_{A_1|A_2} = \{v \mid \exists t \in (0,\tau), tv \in D(p_2 - p_1, r_{\text{threat}})\}. \tag{9}$$

Regarding the minimum offset vector u of $VO^{\tau}_{A_1|A_2}$, it is represented as:

$$u = \left(\underset{v \exists VO^{\tau}_{A_1|A_2}}{\operatorname{argmin}} \left\| v - \left(v_1^- - v_2^-\right) \right\| \right) - \left(v_1^- - v_2^-\right)). \tag{10}$$

Denote $\overrightarrow{n}$ as the outer normal vector of the speed obstacle area at the collision avoidance speed offset vector point. For the obstacle avoidance offset vectors of all surrounding obstacles (agents other than itself), the agent A_1 generates allowed velocity half-planes, and intersects all velocity

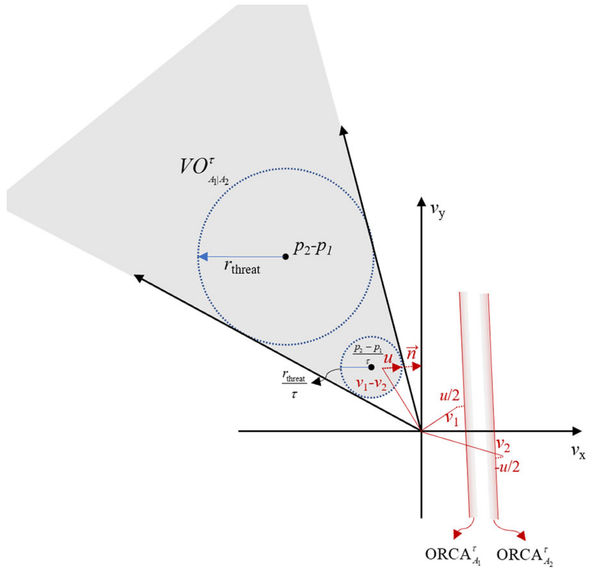

Figure 3 Schematic diagram of ORCA obstacle avoidance algorithm selecting the obstacle avoidance speed of agents A_1 and A_2

half-planes to obtain the final optional velocity set:

$$\mathrm{ORCA}^{\tau}_{A_1} = \left\{v \middle| \left(v - \left(v_1^{-} + \alpha \times u_i\right)\right) \cdot \overrightarrow{n_i} \geq 0, \right. \\ \left. i = 2, 3, \ldots, N_O\right\}, \tag{11}$$

in which, α represents the responsibility distribution coefficient, which is set to 0.5 when avoiding obstacles between agents, that is, both parties avoiding obstacles share the same responsibility for speed changes in pairs.

The schematic diagram of the above ORCA obstacle avoidance algorithm is shown in Fig. 3.

A hierarchical responsibility allocation mechanism is implemented for both dynamic obstacle avoidance and static obstacle avoidance situations. The same ORCA obstacle avoidance strategy is used for static obstacles, but the obstacle avoidance responsibility is assigned to the agent, which improves the agent's obstacle avoidance flexibility. Within the final set of optional speeds, the agent will choose the speed solution closest to the current speed, which is the final collision avoidance speed.

When an individual is in a high-density scene, its ORCA half-plane intersection at a certain moment is likely to be an empty set. At this time, a speed with the smallest penalty is usually selected as the new individual speed. The penalty size is often selected as the maximum value of the Euclidean distance from the speed to the half-plane dividing line, and the agent will choose the speed with the smallest value of the penalty term.

4 Experiments and analysis

4.1 Performance evaluation

1) *Regional effective coverage.*

Regional effective coverage = (actual coverage area of the agent - overlapping coverage area of the agent) / area. According to the regional coverage rate of 100%, it is worth 10 points, 80% is worth 3 points, and 0% is worth 0 points. A linear piecewise function is constructed to calculate the score.

2) *Effective detection of static targets.*

When the agent enters the detection range of a static target, it counts as one valid detection, and the number of times the target attribute needs to be detected is –1. The effective detection rate of static targets = 1 - the remaining number of detections of static targets / the total number of detections of static targets = the number of correct detections / the total number of detections of static targets. A static target effective detection rate of 100% is worth 10 points, 80% is worth 6 points, and 0% is worth 0 points. A linear piecewise function is constructed to calculate the score.

3) *Correct detection of static targets.*

The agent completes the target approach detection action according to the specified angle, and the number of correct detections is counted. The correct detection rate of the static target = the number of correct detections/the total number of detection executions. A 100% correct detection rate for static targets is worth 10 points, and a 0% rate is worth 0 points. A linear function is constructed to calculate the score.

4) *Collaborative effectiveness.*

When each agent enters the target threat area or the safe area of other agents, a collaboration failure will be counted. According to the total number of collaborative failures, a linear piecewise function is constructed to calculate the final score. 0 times is worth 10 points, 5 times is worth 6 points, and 10 times or more is worth 0 points.

4.2 The running interface

The algorithm is simulated and tested on the Galaxy Kylin V10 SP1 system, and various indicators and scores are calculated. Figure 4 shows the visual running interface of the algorithm. In the current state, the ships have cooperated to complete the full coverage of the area and the detection of static targets. In the figure, the gray polygonal area represents the target sea area; the red polygonal area represents the no-navigation zone; the graphic composed of red and blue circles represents the unmanned ship, in which the dark blue double-sector area represents the detection range, and the red circle in the center represents its threat range; The graphic composed of a red circle and a blue sector represents a static target, in which the dark blue sector represents its detection range and the red circle represents its threat range.

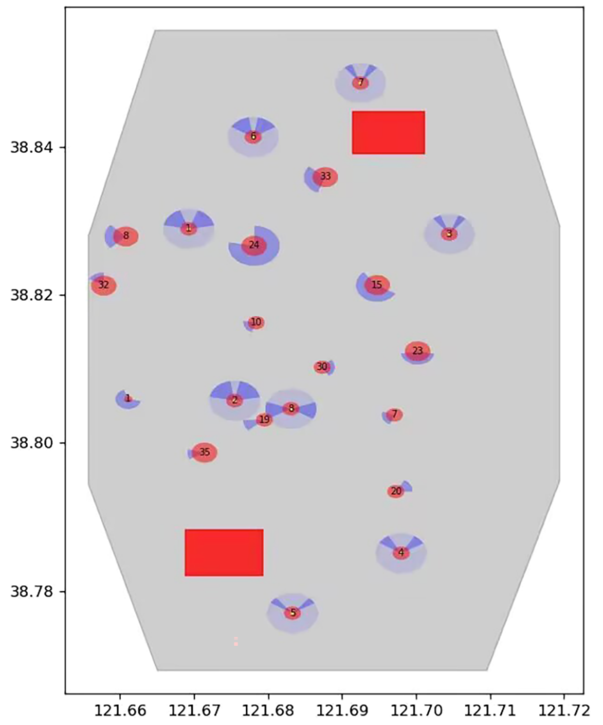

Figure 4 The interface of algorithm visual running

4.3 Experiment settings

The experimental scenario includes the longitude and latitude of the boundary point of the mission area, the set of no-navigation zones (including the longitude and latitude of each no-navigation zone boundary point), the number of the shipping agent, the initial position longitude and latitude, the detection angle range, the detection radius, the threat radius, the maximum sailing speed, and the maximum Probing speed. The status information of the agent includes the agent number, current timestamp, longitude and latitude, navigation speed, heading angle, acceleration, angular velocity, attitude information, and discovered static target information (in list form, if none, it is an empty list). If a static target appears within the detection range of an agent, the information in the static target list includes the static target's number, latitude and longitude, detectable angle, remaining number of detections, threat radius, detection radius, and detection interval.

An example scenario without a no-go zone is shown in Fig. 5, which contains information about all static targets. The red area is the threat area of the agent and static targets, which can be used to represent the collision volume or attack range; the blue area is the detection range of the agent, the green area is the detection radius coverage of the agent, and the black dotted line represents the static target. The angles that can be explored, the numbers below the static target and the agent are their numbers, and the number in the center of the static target is the initial number of times to be explored.

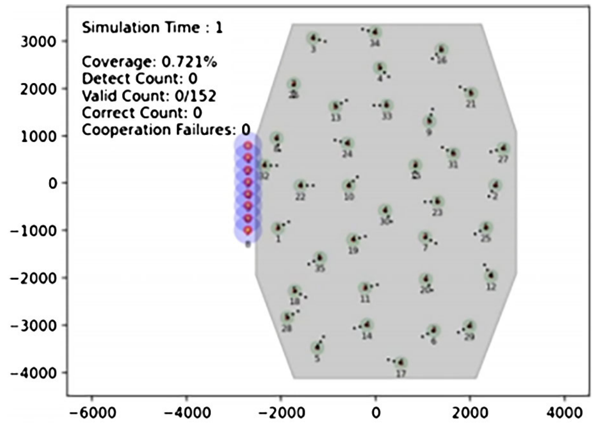

Figure 5 The simulation scenario

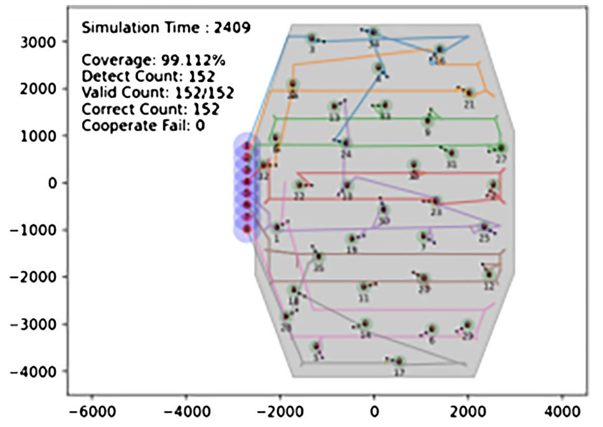

Figure 6 Results of the simulation

4.4 Results

The threat radius of all agents is 100 meters, the detection radius is 300 meters, the maximum sailing speed is 20 knots, the maximum detection speed is 10 knots; the threat radius of all static targets is 100 meters, the detection radius is 200 meters, The total number of explorations to be performed is displayed in the center of the static target, its number is displayed below, and its exploration angle is displayed with a black dotted line. The interval between two explorations by the same agent is 120 seconds. Based on the simulation results (Fig. 6), it can be seen that:

1) *Regional effective coverage.*

After simulation testing, the regional coverage rate is 99.67%, and the regional effective coverage index score is 9.94 points. Since this algorithm also adds new path points to the corner areas that may be missed based on the cellular points, regional omissions should not occur. Analysis suggests that the reasons are as follows: The ship is subject to dynamic constraints and cannot turn on the spot. It may produce a certain path deviation when turning, which is inconsistent with the planned path, resulting in a small area being covered and missing.

2) *Effective detection of static targets.*

Effective detection is the process in which the agent completes the detection action towards the target, from outside the detection range to within the detection range. After simulation testing, the total number of times the static target needs to be probed is 136 times, and the number of correct probes performed is 136 times. The effective detection rate of static targets is 100%, and the effective detection index score of static targets is 10 points.

3) *Correct detection of static targets.*

If the effective detection meets the time, angle, and speed constraints, it will be recorded as a correct detection. After simulation testing, the total number of probe executions was 136, and the correct number of probes was 136. The correct detection rate of static targets is 100%, and the correct detection index score of static targets is 10 points.

4) *Collaborative effectiveness.*

During the simulation test, the agent will not enter the threat range of other agents or static targets, and the number of collaboration failures is 0. The collaborative effectiveness index score is 10 points.

In a word, almost all indicators reached full wind, proving that our algorithm can achieve almost 100% coverage of the scene in this scenario and complete the correct exploration of all targets.

4.5 Discussion

The area division algorithm and coverage algorithm proposed and adopted in this article can be extended to two-dimensional operating sea areas with different peripheral boundaries, different areas, and heterogeneous multi-agents. This algorithm uses honeycomb hexagons to cover the entire two-dimensional task area, obtains the center point set of all hexagons, and adjusts the discrete point distribution of the task boundary area to ensure complete coverage while reducing time consumption as much as possible. Suitable for a variety of task area shapes; after converting the task area into several discrete points, based on the agent's initial position, coverage radius, navigation speed and other conditions, use the clustering method to convert the discrete points into several sub-areas, and then use heuristics. The simulated annealing algorithm allocates sub-regions to each agent; at the same time, considering the actual kinematic model of the agent, the movement path of the agent in the sub-region is smoothly optimized to reduce the number of turns and the energy consumption of the actual ship, making it easier to transfer the model to Introduce practical applications.

The improved greedy collaborative exploration algorithm proposed in this article can ensure the load balance of the agent as much as possible and reduce the task completion time. On the one hand, during the coverage process, the agent immediately explores the static target every time it finds it, reducing the number of tasks in the exploration phase; when the agent completes the assigned coverage task, it will enter the collaborative exploration phase, and the center. The server uses an improved greedy algorithm to continuously select the static target with the smallest exploration cost and assign it to the agent. In addition, after the central server allocates a static target, it updates the cost required for the agent to detect the target in real-time. Once it is found that the number of agents going to detect the static target is more than the number of explorations required for the static target if other agents detect the target, the cost is smaller, and the central server will adjust the current allocation results, always achieve the detection of static targets at the minimum cost, reduce the task completion time, be suitable for various unknown scenarios, and ensure the load balance of the agent.

The underlying collision avoidance algorithm ORCA used in this article has a small amount of calculation and can be deployed on unmanned ships in a distributed manner. The ORCA algorithm is a distributed collision avoidance algorithm with a wide range of application scenarios. Through this algorithm, the optimal collision avoidance speed of the agent under different speed targets (such as maintaining the current speed, maximizing the speed as much as possible, etc.) can be solved. This algorithm only needs to know the position, speed, and other information of the agent itself and potential collision objects and can use linear programming to solve the problem in polynomial time. According to the different states of the agent when performing tasks, corresponding responsibilities can be set for the agent's obstacle avoidance. Agents who perform important tasks bear lower responsibilities when avoiding obstacles. Responsibilities can be flexibly allocated in various tasks to achieve the generalization of the general model. In general engineering scenarios, the algorithm only needs to be estimated or measured by the agent to estimate or measure parameters such as position and speed required by the algorithm, and the algorithm can be directly deployed in a distributed manner.

5 Conclusion

This study has successfully developed and evaluated a comprehensive multi-agent system for regional coverage and static target exploration in complex maritime environments. Through the implementation of a multi-objective optimization model, and employing strategies such as simulated annealing for area coverage and a modified greedy algorithm for task allocation, we have achieved near-perfect effectiveness in both regional coverage and target detection, as demonstrated by our simulation results. The proposed model has proven robust and adaptable, suitable for various marine conditions, and capable of optimizing both time efficiency and resource allocation among multiple unmanned ships. This adaptability is crucial for practical applications ranging from environmental monitoring to maritime rescue operations, where dynamic conditions and unpredictable obstacles are common.

The proposed algorithms, though effective in simulations, have yet to undergo rigorous real-world testing, which is crucial to assessing their robustness and adaptability to the unpredictable dynamics of the marine environment. Additionally, the study's focus on static target exploration necessitates the integration of dynamic target tracking and response mechanisms to enhance its practical applications. Furthermore, the heterogeneity of unmanned ships and their kinematic constraints, as well as the communication requirements and limitations, remain understudied areas that future research must address. Therefore, real-world testing, the incorporation of dynamic target tracking, consideration of a wider range of ship types, and the development of robust communication protocols represent avenues for further improving the system's capabilities, robustness, and adaptability for maritime exploration. Future work should explore the integration of more advanced artificial intelligence techniques to further enhance the autonomy and decision-making capabilities of unmanned vessels. Additionally, expanding the system to incorporate varying types of maritime vehicles and testing in real-world scenarios will be vital to validate and refine our proposed models. Through this research, we contribute to the field of autonomous maritime exploration, offering a scalable and efficient solution to navigate and monitor complex sea areas, which is essential for advancing marine science and technology.

Acknowledgements
Although Professor Li, Postdoctoral Zhao and Doctor Tang did not become the authors of the paper, I am very grateful for their contributions to this paper. Thanks to Li Li for confirming the structure of the paper and giving guiding suggestions; Hui Zhao participated in the revision of the paper and improved the quality of the paper.

Author contributions
Feifei Chen was responsible for modeling and writing the paper. Qingyun Yu designed experiments and writing. All authors read and approved the final manuscript.

Funding
This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 72171172 and 92367101; the Aeronautical Science Foundation of China under Grant No. 2023Z066038001; the National Natural Science Foundation of China Basic Science Research Center Program under Grant No. 62088101; Shanghai Municipal Science and Technology Major Project under Grant No. 2021SHZDZX0100; Chinese Academy of Engineering, Strategic Research and Consulting Program, under Grant No. 2023-XZ-65.

Data availability
Data will be available when needed.

Declarations

Ethics approval and consent to participate
The authors declare that this article does not contain any experiments, and therefore does not have any problems of ethical approval or consent to participate. All authors are informed and have given their consent to this submission.

Consent for publication
All authors have approved the manuscript and declare that this is an original contribution and that the material in this paper is not considered for publication elsewhere.

Competing interests
The authors declare no competing interests.

Author details
[1]Department of Control Science and Engineering, College of Electronics and Information Engineering, Tongji University, Shanghai, 200092, China.
[2]Shanghai Municipal Economic and Information Technology Commission, Shanghai, 200125, China.

Received: 28 April 2024 Revised: 20 May 2024 Accepted: 23 May 2024
Published online: 05 July 2024

References

1. T. Balch, The case for randomized search, in *2000 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, San Francisco, 2000), pp. 213–215
2. V.P. Tran, M.A. Garratt, K. Kasmarik et al., Dynamic frontier-led swarming: multi-robot repeated coverage in dynamic environments. IEEE/CAA J. Autom. Sin. **10**(3), 646–661 (2023)
3. J. Lu, B. Zeng, J. Tang et al., Tmstc*: a path planning algorithm for minimizing turns in multi-robot coverage. IEEE Robot. Autom. Lett. (2023)
4. S. Schwartz, An overview of graph covering and partitioning. Discrete Math. **345**(8), 112884 (2022)
5. W. Yu, Z. Liu, B.X. New, LP relaxations for minimum cycle/path/tree cover problems. Theor. Comput. Sci. **803**, 71–81 (2020)
6. W. Yu, Z. Liu, X. Bao, New approximation algorithms for the minimum cycle cover problem. Theor. Comput. Sci. **793**, 44–58 (2019)
7. J. Tang, M.H. Mixed, Integer Programming for Time-Optimal Multi-Robot Coverage Path Planning with Heuristics. IEEE Robot. Autom. Lett. **8**(10), 6491–6498 (2023)
8. M. Zhu, A. Hahn, Y.Q. Wen, F. Zhang, Maritime unmanned vehicle cruise path planning for maritime information collection. IFAC-PapersOnLine **49**(23), 103–108 (2016)
9. N. Hazon, G.A. Kaminka, Redundancy, efficiency and robustness in multi-robot coverage, in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (IEEE, Los Alamitos, 2005), pp. 735–741
10. A. Auger, B. Doerr, *Theory of randomized search heuristics. Foundations and recent developments* (World Scientific, Singapore, 2011). https://doi.org/10.1142/7438
11. R. Mitra, I. Saha, Scalable online coverage path planning for multi-robot sys- tems, in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, Los Alamitos, 2022), pp. 10102–10109
12. Y. Tang, R. Zhou, G. Sun, B. Di, R. Xiong, A novel cooperative path plan- ning for multirobot persistent coverage in complex environments. IEEE Sens. J. **20**(8), 4485–4495 (2020)
13. L. Collins, P. Ghassemi, E.T. Esfahani, D. Doermann, K. Dantu, S. Chowd-Hury, Scalable coverage path planning of multi-robot teams for monitoring non-convex areas, in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, Los Alamitos, 2021), pp. 7393–7399
14. M. Li, A. Richards, M. Sooriyabandara, Asynchronous reliability-aware multi-uav coverage path planning, in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, Los Alamitos, 2021), pp. 10023–10029
15. L. Gao, W. Lv, X. Yan, Y. Han, Complete coverage path planning algorithm based on energy compensation and obstacle vectorization. Expert Syst. Appl. **203**, 117495 (2022)
16. I. Vandermeulen, R. Groß, A. Kolling, Turn-minimizing multirobot coverage, in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE, Los Alamitos, 2019), pp. 1014–1020

17. W. Gosrich, S. Mayya, R. Li, J. Paulos, M. Yim, A. Ribeiro, V. Kumar, Coverage control in multi-robot systems via graph neural networks, in *2022 International Conference on Robotics and Automation (ICRA)* (IEEE, Los Alamitos, 2022), pp. 8787–8793
18. J. Chen, C. Du, Y. Zhang, P. Han, W. Wei, A clustering-based coverage path planning method for autonomous heterogeneous uavs. IEEE Trans. Intell. Transp. Syst. **23**(12), 25546–25556 (2021)
19. J. Chen, F. Ling, Y. Zhang, T. You, Y. Liu, X. Du, Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system. Swarm Evol. Comput. **69**, 101005 (2022)
20. J. Xie, J. Chen, Multiregional coverage path planning for multiple energy con- strained uavs. IEEE Trans. Intell. Transp. Syst. **23**(10), 17366–17381 (2022)
21. L. Jiao, Z. Peng, L. Xi, S. Ding, J. Cui, Multi-agent coverage path planning via proximity interaction and cooperation. IEEE Sens. J. **22**(6), 6196–6207 (2022)
22. W. Jing, D. Deng, Y. Wu, K. Shimada, Multi-UAV coverage path planning for the inspection of large and complex structures, in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, Los Alamitos, 2020), pp. 1480–1486
23. K. Nakamura, M. Santos, N.E. Leonard, Decentralized learning with limited communications for multi-robot coverage of unknown spatial fields, in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, Los Alamitos, 2022), pp. 9980–9986
24. W. Zhu, L. Li, L. Teng, W. Yonglu, Multi-uav reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double- chromosome encoding. Chin. J. Aeronaut. **31**(2), 339–350 (2018)
25. R. Zlot, A. Stentz, M.B. Dias, S. Thayer, Multi-robot exploration controlled by a market economy, in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 3 (IEEE, Los Alamitos, 2002), pp. 3016–3023
26. S.S. Ponda, L.B. Johnson, A.N. Kopeikin, H.-L. Choi, J.P. How, Distributed planning strategies to ensure network connectivity for dynamic heterogeneous teams. IEEE J. Sel. Areas Commun. **30**(5), 861–869 (2012)
27. D.-H. Lee, S.A. Zaheer, J.-H. Kim, A resource-oriented, decentralized auction algorithm for multirobot task allocation. IEEE Trans. Autom. Sci. Eng. **12**(4), 1469–1481 (2014)
28. L. Luo, N. Chakraborty, K. Sycara, Distributed algorithms for multirobot task assignment with task deadline constraints. IEEE Trans. Autom. Sci. Eng. **12**(3), 876–888 (2015)
29. G. Neville, S. Chernova, H. Ravichandar, D-itags: a dynamic interleaved approach to resilient task allocation, scheduling, and motion planning. IEEE Robot. Autom. Lett. **8**(2), 1037–1044 (2023)
30. X. Luo, M.M. Zavlanos, Temporal logic task allocation in heterogeneous multirobot systems. IEEE Trans. Robot. **38**(6), 3602–3621 (2022)
31. S. Al-Hussaini, J.M. Gregory, S.K. Gupta, Generating task reallocation suggestions to handle contingencies in human-supervised multi-robot missions. IEEE Trans. Autom. Sci. Eng. **1**, 367–381 (2024)
32. X. Bai, A. Fielbaum, M. Kronmüller, L. Knoedler, J. Alonso-Mora, Group-based distributed auction algorithms for multi-robot task assignment. IEEE Trans. Autom. Sci. Eng. **20**(2), 1292–1303 (2022)
33. Z. Nie, K.-C. Chen, Hypergraphical real-time multirobot task allocation in a smart factory. IEEE Trans. Ind. Inform. **18**(9), 6047–6056 (2021)
34. S. Park, Y.D. Zhong, N.E. Leonard, Multi-robot task allocation games in dynamically changing environments, in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, Los Alamitos, 2021), pp. 8678–8684
35. E. Gilbert, D. Johnson, Distance functions and their application to robot path planning in the presence of obstacles. IEEE J. Robot. Autom. **1**(1), 21–30 (1985)
36. A. Lazarowska, Discrete artificial potential field approach to mobile robot path planning. IFAC-PapersOnLine **52**(8), 277–282 (2019)
37. J. Van Den Berg, S.J. Guy, M. Lin, D. Manocha, Reciprocal n-body collision avoidance, in *Robotics Research: The 14th International Symposium ISRR* (Springer, Berlin, 2011), pp. 3–19
38. C. Fulgenzi, A. Spalanzani, C. Laugier, Dynamic obstacle avoidance in uncer- tain environment combining pvos and occupancy grid, in *Proceedings 2007 IEEE International Conference on Robotics and Automation* (IEEE, Los Alamitos, 2007), pp. 1610–1616
39. M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, C. Cadena, From perception to decision: a data-driven approach to end-to-end motion planning for autonomous ground robots, in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, Los Alamitos, 2017), pp. 1527–1533
40. Z. Sui, Z. Pu, J. Yi, S. Wu, Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration. IEEE Trans. Neural Netw. Learn. Syst. **32**(6), 2358–2372 (2020)
41. H. Shi, L. Shi, M. Xu, K.-S. Hwang, End-to-end navigation strategy with deep reinforcement learning for mobile robots. IEEE Trans. Ind. Inform. **16**(4), 2393–2402 (2019)
42. L. Salt, D. Howard, G. Indiveri, Y. Sandamirskaya, Parameter optimization and learning in a spiking neural network for uav obstacle avoidance targeting neuromorphic processors. IEEE Trans. Neural Netw. Learn. Syst. **31**(9), 3305–3318 (2019)
43. Y. Guo, Z. Qu, Coverage control for a mobile robot patrolling a dynamic and uncertain environment, in *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)*, vol. 6 (IEEE, Los Alamitos, 2004), pp. 4899–4903

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.