

Article

IB-DARP: An Algorithm for Multi-Vessel Collaborative Task and Path Planning

Yuhao Wang  and Liang Luo *

School of Naval Architecture, Ocean and Energy Power Engineering, Wuhan University of Technology,
Wuhan 430063, China; anlxqag@gmail.com

* Correspondence: luoliang@whut.edu.cn

Abstract

This paper presents IB-DARP (Iteration Balancing—Divide Areas Routing Problem), an enhanced multi-vessel cooperative mission and path planning method designed to address the limitations of traditional approaches, including uneven task allocation, workload imbalance, and path conflicts. The proposed method integrates four key mechanisms to improve planning robustness and computational efficiency. A historical data mining mechanism is first employed to extract stable navigation patterns from accumulated vessel trajectories and construct a high-confidence maritime route network. Based on this network, a precomputation mechanism significantly reduces planning-stage computational complexity by calculating essential inter-node distances in advance. A heading-aware partitioning mechanism further decomposes the multi-vessel planning problem into tractable single-vessel subproblems, while an iterative auction–equilibrium mechanism dynamically adjusts task assignments to enhance global load balance and suppress conflicts. To evaluate the effectiveness of IB-DARP, comprehensive ablation studies and large-scale scenario experiments were conducted, demonstrating its advantages in mission allocation, conflict mitigation, and cooperative path optimization. The results confirm that IB-DARP provides a scalable and efficient solution for multi-vessel cooperative mission planning in complex maritime environments.

Keywords: improved DARP; path planning; task allocation; multi-vessel cooperation

1. Introduction

Path planning serves as a core component of autonomous navigation systems. Its primary objective is to identify an optimal or feasible motion path for an agent—such as a robot, unmanned aerial vehicle, or unmanned surface vessel—from a start point to a goal within complex environments, while satisfying specific constraints such as obstacle avoidance, compliance with physical laws, and optimization of time or energy consumption. With ongoing advancements in research, path planning methodologies have demonstrated a clear trend of evolution from standalone algorithms toward diversified, integrated approaches. For instance, reference [1] combines classical graph search with random sampling techniques; reference [2] incorporates fuzzy logic into the ant colony optimization algorithm; while reference [3] proposes an enhanced reinforcement learning algorithm, R-SAC, integrated with a residual network architecture to improve performance. By merging the strengths of different algorithms and cross-pollinating ideas, these methods significantly enhance the quality of planned paths and their applicability in complex real-world scenarios.



Academic Editor: Xinqiang Chen

Received: 4 December 2025

Revised: 8 January 2026

Accepted: 8 January 2026

Published: 12 January 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

In practical applications, two main paradigms have emerged. The first paradigm centers on dynamic perception and online decision-making, emphasizing continuous interaction with the environment during execution to adapt to dynamic changes and uncertainties. This approach prioritizes real-time responsiveness and local optimization, making it particularly suitable for scenarios with high real-time demands such as dynamic obstacle avoidance and human–robot collaborative environments. For instance, reference [4] leverages reinforcement learning to enable rapid autonomous decision-making for search and rescue paths through dynamic environmental interaction. Reference [5] employs a Q-learning algorithm that incorporates wind direction and obstacle avoidance requirements into path planning. Reference [6] utilizes deep reinforcement learning to achieve continuous obstacle avoidance and path planning in unknown environments. Reference [7] develops a collision-avoidance path planning method for multi-ship encounter scenarios by explicitly considering ship maneuverability and compliance with COLREGs. Furthermore, studies such as [8–11] also focus on path planning in dynamic environments, further enriching research in this direction.

The second paradigm in path planning is characterized by predefined decomposition and static allocation, making it well-suited for static scenarios where environmental information is complete and task structures are well-defined.

In static path planning, “decompose-then-allocate” methods dominate, decomposing global problems into structured subproblems solved via search or optimization. Representative advances include graph- and search-based approaches such as quadtree-enhanced visibility graphs [12], improved Lazy-PRM [13], spatio-temporal A* for multi-vessel avoidance [14], and TEA-based multi-robot coordination [15]; sampling-based planners improving RRT with multi-objective costs, pruning, and B-spline smoothing [16,17]; grid- and PDE-based methods employing anisotropic and hybrid fast marching [18,19]; potential-field and swarm-intelligence techniques including safety-potential-field modeling [20], APF–ACO fusion [21], pheromone diffusion [22], adaptive ACO [23], and PSO hybrids [24–26]; meta-heuristic and bio-inspired methods integrating tabu search, GA, and neural models [27–30]; and learning-based approaches such as double DQN [31], TD3–APF [32], and GNN-driven multi-agent reinforcement learning [33].

In contrast, another line of research adheres to the “allocate-then-decompose” paradigm, wherein the entire task area is first allocated to individual agents, followed by separate path planning and task decomposition within each assigned subregion.

Under the “allocate-then-decompose” paradigm, researchers have developed various strategies tailored to specific applications. Representative studies include graph-theoretic coverage for UAVs using convex hull partitioning and TSP transformation [34], time-adjusted load balancing for multi-AUV systems considering sonar and currents [35], and Morse decomposition for multi-AUV search and rescue to optimize region segmentation [36]. Additional contributions include ACO adaptations for underwater gliders [37], acoustic model–PSO integration [38], hybrid grid modeling with dynamic partitioning [39], predator-prey inspired Q-learning [40], and energy-aware Voronoi-based path generation [41].

Despite their methodological differences, “allocate-then-decompose” approaches share a common prerequisite: before considering path planning, tasks or operational regions must be partitioned according to either a reasonably defined explicit allocation criterion or an implicit one that can be easily extracted. Such allocation criteria are typically problem-dependent and closely tied to specific task characteristics, which makes them difficult to generalize across different operational scenarios.

Notably, Kapoutsis et al. [42] addressed this issue by proposing the DARP (Divide Areas Routing Problem) framework, in which region allocation emerges implicitly through

an iterative balancing process driven by robots' initial positions, rather than relying on manually designed allocation metrics. This framework represents an important conceptual shift from metric-driven partitioning to self-organized allocation, and has subsequently inspired a series of DARP-based extensions and variants.

Building upon this concept, Zhang et al. [43] enhanced DARP for multi-robot agricultural operations by combining boustrophedon coverage, A*, and DFS-based validation, achieving full coverage and efficient path planning. Apostolidis et al. [44] designed a multi-UAV platform combining grid optimization, DARP, and MST-based path generation, validated in both simulation and real missions for high coverage and low mission time. Cai et al. [45] proposed a multi-AUV coverage method using Morse decomposition and SAR-A*, with customized backtracking to balance workload, reduce turns, and prioritize probable target areas, demonstrating superior load distribution and mission efficiency in experiments.

In contrast to task–path coupled planning methods, many existing multi-vessel cooperative approaches primarily focus on execution-level coordination or engineering-oriented scheduling. For example, the Divide–Merge framework [46] addresses cooperative operation and maintenance scheduling through predefined task decomposition followed by route optimization, while Distributed-Integrated MPC [47] emphasizes coordinated control and formation maintenance under the assumption of known reference paths. Although effective within their respective application scopes, these methods rely on externally specified task assignments or preplanned routes, and therefore do not explicitly address the problem of task allocation emerging jointly with path planning.

Consequently, when applied to the large-scale maritime scenarios considered in this study—characterized by sparse task distributions and strong route constraints—existing approaches still struggle to adequately address the tight coupling between task allocation and path planning. To tackle this challenge, we propose IB-DARP, an enhanced DARP-based framework designed to jointly handle task assignment and route generation in an integrated manner, providing a unified foundation for multi-vessel collaborative planning in complex maritime environments.

The remainder of this paper is organized as follows. Section 2 introduces the research foundation and mathematical model. Section 3 presents the workflow and computational principles of IB-DARP. Section 4 reports simulation experiments and performance analysis. Section 5 concludes the paper and discusses future research directions.

2. Mathematical Model for Integrated Vessel Mission and Path Planning

2.1. Problem Description

This study addresses a tightly coupled multi-vessel task allocation and route planning problem in large-scale maritime operations. The objective is to assign sparsely distributed global tasks to individual vessels while generating feasible and efficient sailing routes that respect task sequences, heterogeneous vessel capabilities, operational time windows, and spatio-temporal dependencies under practical computational constraints.

Under traditional DARP-based frameworks, task or region partitioning is achieved through initial-position-driven balancing over discretized spatial grids, as conceptually illustrated in Figure 1. Such formulations are effective for structured coverage problems with uniform spatial domains, but become unsuitable for large-scale maritime scenarios where tasks are not evenly distributed and navigation is governed by constrained route networks rather than homogeneous spatial regions. As a result, directly extending conventional DARP formulations to maritime operations fails to adequately capture the coupling between task allocation and route feasibility.

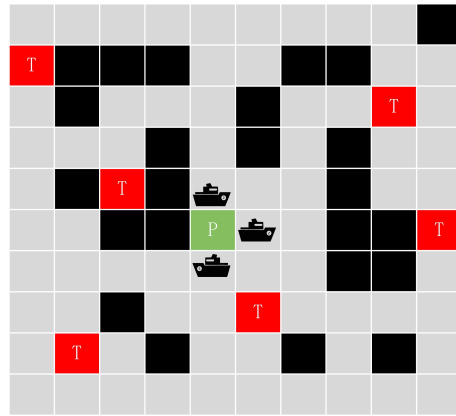


Figure 1. Spatial partitioning scheme in traditional DARP-based frameworks. The red regions (T) denote abstracted task areas, the green points (P) represent ports, and the black regions indicate obstacle areas.

2.2. Mathematical Model

2.2.1. Notations and Definitions

To depict the distribution of ports, task points, and vessel routes, we employ a marine vector background map from Natural Earth [48] at 1:110 million resolution, capturing global land–sea contours. As shown in Figure 2, navigable areas are marked in blue, obstacles in white, and ports and major harbors with green markers.

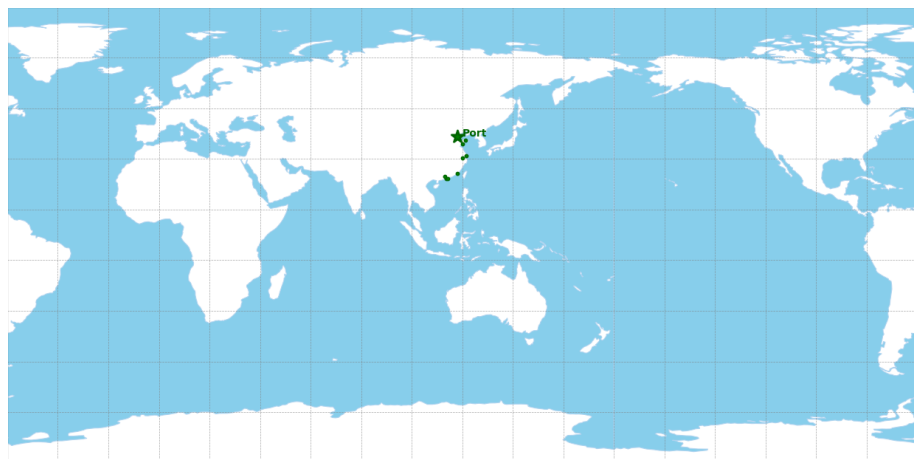


Figure 2. Schematic diagram of the planning area and port.

Given the dense distribution of major coastal ports in China, a representative core port is selected as the starting point for multi-vessel task allocation to reduce model complexity. Other port locations are incorporated later during path planning and voyage calculation to balance model simplicity with operational realism.

To characterize the state of each vessel, the vessel set is defined as S ,

$$S = \{S_1, S_2, \dots, S_m\} \quad (1)$$

where S_m denotes the m -th vessel. The task set is defined as T ,

$$T = \{T_1, T_2, \dots, T_i\} \quad (2)$$

where T_i represents the i -th task. Each task is assigned a corresponding weight from the set W ,

$$W = \{w_1, w_2, \dots, w_i\}, \quad (3)$$

where w_i indicates the importance of task T_i , reflecting its contribution to the overall objective.

To specify functional information such as vessel departure and arrival, the port set is defined as

$$P = \{P_1, P_2, \dots, P_j\}, \quad (4)$$

where P_j denotes the j -th port.

Vessel operations during task execution are subject to constraints from the geographical environment and other factors. Consequently, the adjacency relationship is defined as an undirected graph

$$G = \{V, E\}, \quad V = T \cup P, \quad (5)$$

where the node set V represents the collection of task and port points, and the edge set E denotes the reachability relationships between tasks and between ports and tasks. An edge $(V_m, V_n) \in E$ exists if and only if $m \neq n, m, n \in [1, n + j]$, indicating that a vessel can traverse between points V_m and V_n .

These adjacency relationships underpin the subsequent task allocation and path planning, with edge weight determination and computational results detailed in later sections.

2.2.2. Objective Function

Under all operational constraints, a multi-objective cost model is introduced to directly guide the route planning process for each TSP instance generated by the IB-DARP task allocation. This formulation evaluates candidate routes by jointly considering navigation distance, task execution cost, and task coverage effectiveness.

The overall optimization objective is conceptually expressed as

$$\min F = \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3, \quad (6)$$

This formulation serves as a conceptual description of the optimization goals rather than a scalar objective to be explicitly evaluated. In practice, these objectives are embedded into the heuristic evaluation functions and decision rules of the route planning algorithm, guiding the trade-offs among competing criteria during solution construction. The concrete implementation of this multi-objective guidance is detailed in Section 3.2.2.

(1) Minimization of Total Vessel Travel Distance

Since vessel travel distance directly affects mission duration and energy consumption, minimizing the total sailing distance is a fundamental objective. For a given TSP route π_k , the travel distance is defined as

$$f_1(\pi_k) = \sum_{(i,j) \in \pi_k} d_{ij}, \quad (7)$$

where d_{ij} denotes the network distance between two consecutive nodes i and j along the route π_k .

(2) Minimization of task execution cost

To guide path selection during route planning, a task execution cost is defined to evaluate transitions between successive task nodes. This cost reflects task priority, incremental travel distance, and temporal feasibility, encouraging the construction of routes that are both feasible and efficient with respect to task execution. The cumulative task execution cost of a route is defined as

$$f_2(\pi_k) = \sum_{(i,j) \in \pi_k} c_{ij}, \quad (8)$$

where the transition cost from node i to node j is given by

$$c_{ij} = \gamma_1 \hat{w}_j + \gamma_2 \hat{d}_{ij} + \gamma_3 \hat{t}_{ij}, \quad (9)$$

where w_j denotes the weight of task j , d_{ij} represents the incremental travel distance from node i to node j , and t_{ij} denotes the temporal cost associated with task execution, including waiting penalties for early arrival and infeasibility penalties for late arrival. The normalized terms $\hat{w}_j$, $\hat{d}_{ij}$, and $\hat{t}_{ij}$ ensure dimensional consistency, while the weighting coefficients γ_1 , γ_2 , and γ_3 balance the relative contributions of these factors.

(3) Maximization of Task Coverage Ratio

Under time and resource constraints, prioritizing the execution of high-value tasks is essential. Task coverage effectiveness is quantified as the ratio between the total weight of tasks actually executed along a route and the total weight of tasks assigned to the corresponding region,

$$f_3(\pi_k) = \frac{\sum_{i \in V(\pi_k)} w_i}{\sum_{i \in T_{R_k}} w_i}, \quad (10)$$

where T_{R_k} denotes the set of tasks assigned to region R_k , and $V(\pi_k)$ represents the set of tasks visited by route π_k .

2.2.3. Constraints

(1) Unique Task Allocation

To avoid redundant execution, each task is assigned to at most one vessel, ensuring unique allocation,

$$\sum_{s \in S} x_{is} \leq 1, \quad \forall i \in T. \quad (11)$$

(2) Task Time Window

Task execution must adhere to predefined time windows. A vessel must arrive at the task area within the permitted interval; otherwise, the task is considered unaccomplished,

$$e_i \leq a_i^s \leq l_i + M(1 - x_{is}), \quad \forall i \in T, s \in S, \quad (12)$$

where a_i^s denotes the arrival time of vessel s at task i , $[e_i, l_i]$ is the task time window, and M is a sufficiently large constant. If a_i^s falls outside this interval, the task is deemed failed ($x_{is} = 0$).

(3) Path Continuity Sequencing

The task execution path of a vessel must be continuous, meaning each task node must have exactly one predecessor and one successor. A binary sequencing variable z_{ij}^s indicates whether vessel s travels from node i to node j ,

$$\sum_{j \in V} z_{ij}^s = x_{is}, \quad \forall i \in V, s \in S, (i, j) \in E, \quad (13)$$

$$\sum_{j \in V} z_{ji}^s = x_{is}, \quad \forall i \in V, s \in S, (i, j) \in E. \quad (14)$$

To ensure temporal feasibility, the arrival time at the next task point should be no earlier than the completion time of the current task plus travel time,

$$a_j^s \geq a_i^s + \tau_i + t_{ij}^s - M(1 - z_{ij}^s), \quad \forall i, j \in V, s \in S, (i, j) \in E, \quad (15)$$

where τ_i is the execution duration of task i , and t_{ij}^s is the travel time of vessel s from i to j .

3. The Proposed IB-DARP

To overcome the limitations of traditional task allocation and path planning models and effectively reduce the high computational complexity inherent in the mTSP, this section

proposes an Iterative Balancing DARP (IB-DARP) algorithm. By organically integrating four core mechanisms, the algorithm achieves an efficient and reliable cooperative solution.

(1) Historical data mining mechanism

This mechanism analyzes accumulated navigation data to extract frequent and high-confidence sailing trajectories, while incorporating expert knowledge. It establishes a reliable topological foundation for task allocation and path planning.

(2) Precomputation and cost estimation mechanism based on high-confidence routes

It precalculates sailing distances between key nodes and establishes an efficient query mechanism. This transforms complex online path planning into rapid network distance queries, significantly reducing the complexity and time overhead of inter-task voyage calculations.

(3) Heading-aware adaptive partitioning mechanism

This constructs directionally consistent task partitions based on the azimuth characteristics of task points relative to the port. It decomposes the complex multi-vessel cooperative planning problem into multiple parallel single-vessel path planning subproblems, effectively controlling the overall problem scale.

(4) Iterative auction-based dynamic load balancing mechanism

This establishes dynamic correlations among the path planning subproblems of different vessels, simulating an auction process to enable orderly task migration between vessels. Through multiple rounds of auctions, it continuously optimizes the global task allocation scheme, ensuring balanced workload distribution among vessels and near-optimal total sailing cost.

These mechanisms are organically integrated to form the complete framework of the IB-DARP algorithm. The overall algorithmic framework is illustrated in Figure 3.

3.1. Route Network Construction and Task Mapping

In large-scale maritime environments, analysis of historical navigation data reveals frequently traveled and reliable routes, including port approaches and major shipping corridors. By extracting and processing these trajectories, a high-confidence baseline route network $G_b = (V, E_b)$ can be constructed to support subsequent path planning and optimization.

3.1.1. Construction of High-Confidence Route Network

A nearly two-decade archive of AIS data forms the foundation of this study. The historical AIS records are first cleaned to remove trajectory discontinuities, invalid records, positional deviations exceeding the map resolution for non-port navigation points, and stationary position records with unchanged timestamps. The spatial coverage of the processed trajectory data is illustrated in Figure 4.

Maritime routes are typically nonlinear, unevenly distributed, and spatially structured, forming elongated and curved clusters with strongly varying densities. Under such conditions, centroid-based clustering methods are unsuitable, as they assume compact cluster shapes and require predefined cluster numbers. Therefore, HDBSCAN is adopted in this study, as it can robustly identify arbitrarily shaped clusters, adapt to density variations, and automatically suppress noise without prior specification of the cluster count. These properties make HDBSCAN particularly suitable for maritime trajectory data characterized by strong density heterogeneity between open-sea regions and coastal or port areas. The specific HDBSCAN parameter settings used in this study are listed in Table 1.

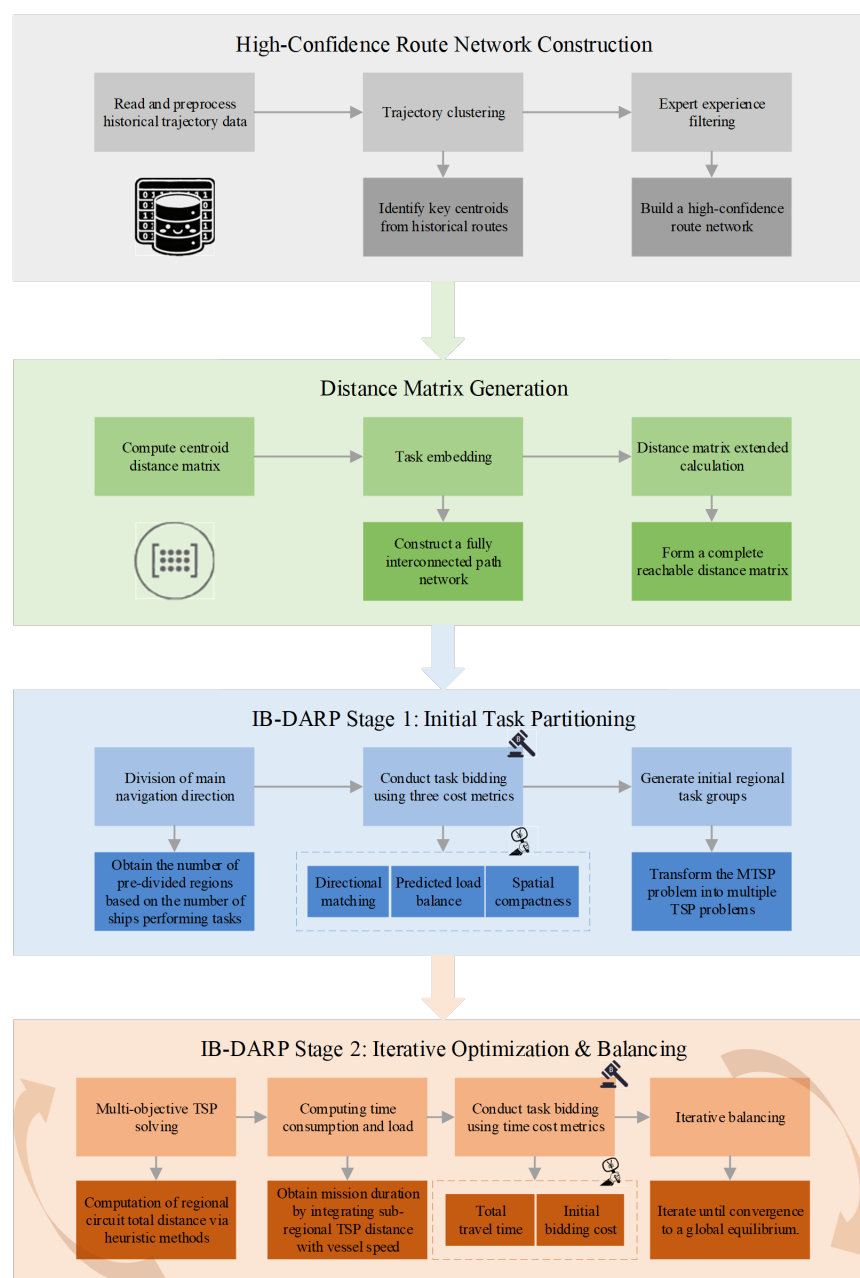


Figure 3. Framework of the IB-DARP algorithm.

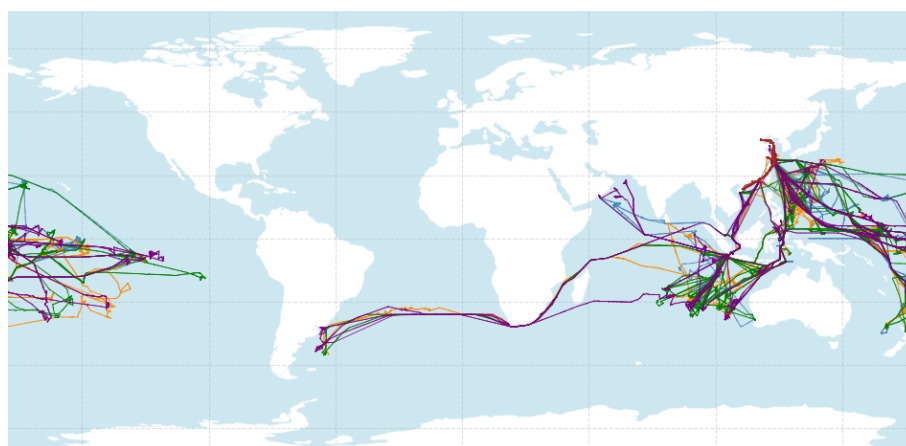
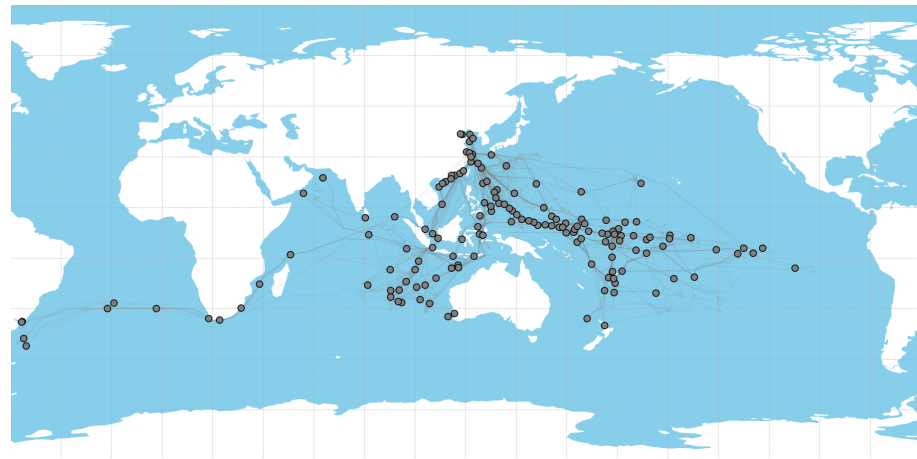


Figure 4. Spatial distribution of raw AIS trajectory data.

Table 1. HDBSCAN parameter settings used in this study.

Parameter	Min_Cluster_Size	Min_Samples	Metric	Cluster_Selection_Method
Setting	200	5	Euclidean	EOM (default)

Representative route centers are selected directly from historical AIS trajectories rather than using geometric centroids, ensuring that all extracted centers are located on navigable paths. As illustrated in Figure 5, these centers collectively form a continuous and executable global route structure.

**Figure 5.** Extracted navigation patterns based on HDBSCAN clustering.

In coastal and port-adjacent areas, where navigational complexity is high, a dense distribution of route centers is required to capture fine-grained route structures. In contrast, navigation in wide open-water regions is less constrained, reducing the need for densely distributed centers. However, density-based clustering methods inherently reflect spatial data density rather than planning requirements and therefore cannot adaptively regulate the number of centers across different regions. As a result, directly retaining all clustering-derived centers introduces substantial redundancy and weakens the global structural representation needed for effective task allocation and path planning.

Therefore, a region-adaptive center reduction strategy is incorporated on top of the HDBSCAN-based trajectory clustering to suppress redundant route centers induced by non-uniform spatial distributions, while preserving global route coverage and topological connectivity. V denotes the set of route centers obtained from HDBSCAN clustering, where each center represents a candidate node on the global navigation network.

$$V = \{V_1, \dots, V_n\}, \quad V_n = (\lambda_n, \phi_n). \quad (16)$$

The reduction procedure jointly considers spatial proximity, historical route consistency, and center occurrence frequency within a unified framework. Specifically, when the geodesic distance between two centers is smaller than the merging threshold $d_{\text{merge}} = 80$ nmi and both centers appear on the same historical route segment, only the center closer to the midpoint of that segment is retained, thereby merging locally dense centers. Subsequently, within a neighborhood of radius $r = 240$ nmi around each center, the maximum number of retained centers is limited to $k_{\text{max}} = 3$. If this limit is exceeded, centers with lower occurrence frequencies in historical trajectories are preferentially removed. Finally, to prevent structural discontinuities caused by over-reduction, if historical trajectories exist within a neighborhood but no center is retained, the center with the highest occurrence frequency along the corresponding route segment is reintroduced.

After center reduction, high-confidence route segments $E_b^{(n,m)}$, as illustrated in Figure 6, are reconstructed by recovering the original waypoint sequences between adjacent route centers along historical AIS trajectories.

$$E_b^{(n,m)} = \{V_n, p_1, \dots, p_k, V_m\}, \quad (17)$$

where V_n and V_m denote two connected route centers, and $\{p_1, \dots, p_k\}$ represents the intermediate waypoints extracted from the corresponding historical trajectory.

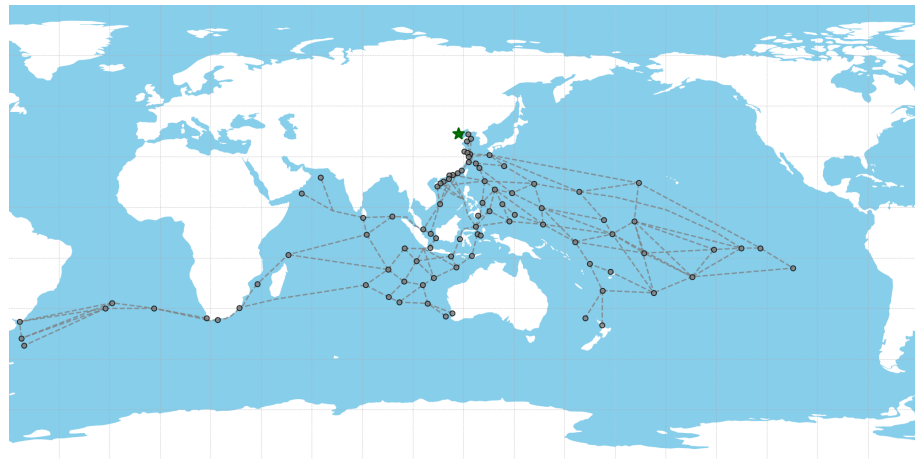


Figure 6. Schematic diagram of a high-confidence route network.

Based on the constructed high-confidence route network, the Haversine formula is employed to calculate the pairwise geographical distances between port P and each centroid V_m , as well as between any two centroids V_m and V_n .

$$f_{\text{Haversine}}(V_n, V_m) = R \cdot c, \quad (18)$$

where $c = 2 \cdot \arctan 2(\sqrt{a}, \sqrt{1-a})$, $a = \sin^2(\frac{\Delta\phi}{2}) + \cos(\phi_n) \cdot \cos(\phi_m) \cdot \sin^2(\frac{\Delta\lambda}{2})$, $\Delta\phi = \phi_m - \phi_n$, $\Delta\lambda = \lambda_m - \lambda_n$ represent the differences in latitude and longitude, respectively. R is the Earth's mean radius, taken as 6371 km.

On the basis of obtaining the geographical distances between nodes, the Dijkstra algorithm is further introduced. Using the high-confidence route network G_b as the underlying graph, the shortest path distance $d_{(n,m)}$ between any two nodes V_n and V_m is computed.

$$d_{n,m} = \min_{E_{b,n,m} \in \rho} \sum_{(V_i, V_j) \in E_{b,n,m}} f_{\text{Haversine}}(V_i, V_j), \quad (19)$$

where ρ denotes the set of all feasible paths from V_n to V_m .

Based on the aforementioned high-confidence route network and shortest path computation, a centroid matrix D_{center} can be constructed. This matrix integrates the shortest path distances between all centroids V and the port P .

$$D_{\text{center}} = \begin{bmatrix} 0 & d_{1,2} & \cdots & d_{1,i+1} \\ 0 & 0 & \cdots & d_{2,i+1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ddots & d_{i,i+1} \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{(i+1) \times (i+1)} \quad (20)$$

where $d_{n,m}$ represents the shortest path distance between nodes V_n and V_m . Since the constructed route graph is undirected and satisfies $d_{n,m} = d_{m,n}$, D_{center} can be simplified to an upper triangular matrix form.

3.1.2. Dynamic Route Network Generation and Update

Since task points may not lie directly on the predefined high-confidence route network (as shown in Figure 7), a dynamic task-connected network $G_d = (T, E_d)$ is constructed to integrate randomly distributed task points into the static base network, enabling rapid adaptation to real-time task locations while leveraging the existing network structure.

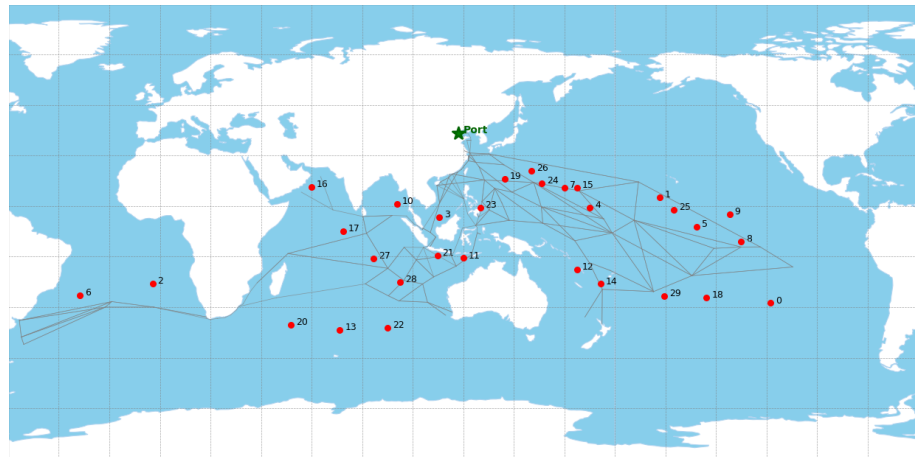


Figure 7. Spatial distribution of task points.

To achieve efficient integration of task points with the high-confidence route network, this paper proposes an attraction-guided model based on the visibility graph method. By introducing the route attraction force F_p from nearby high-confidence routes to the starting task point T_s , and the goal attraction force $F_{t(\text{goal})}$ from the target task point T_e , the generated path is guided to effectively utilize the high-confidence route while maintaining directionality, ultimately converging stably to the target location. The combined force expression is as follows:

$$F_t = \alpha \cdot F_p + \beta \cdot F_{t(\text{goal})} \quad (21)$$

where F_p represents the attraction force of the high-confidence route, which draws the path toward and aligns it with existing routes; $F_{t(\text{goal})}$ denotes the goal attraction force, guiding the path to converge toward the task point.

Under the combined effect of these dual attraction forces, task points and route nodes are progressively generated and connected, forming a fully connected route set E_d , which can be expressed as

$$E_d = \{T_s, \dots, I_{p_n}\} \cup E_{b_{n,m}} \cup \{I_{p_m}, \dots, T_e\}, \quad \forall T \in \{T_1, T_2, \dots, T_i\} \quad (22)$$

where I_p represents the intersection point between the high-confidence route and the temporary connecting path, while $\{T_s, \dots, I_{p_n}\}$ and $\{I_{p_m}, \dots, T_e\}$ denote the temporary connecting paths between the task point and the high-confidence route, respectively. The connection principle of this model is illustrated in Figure 8.

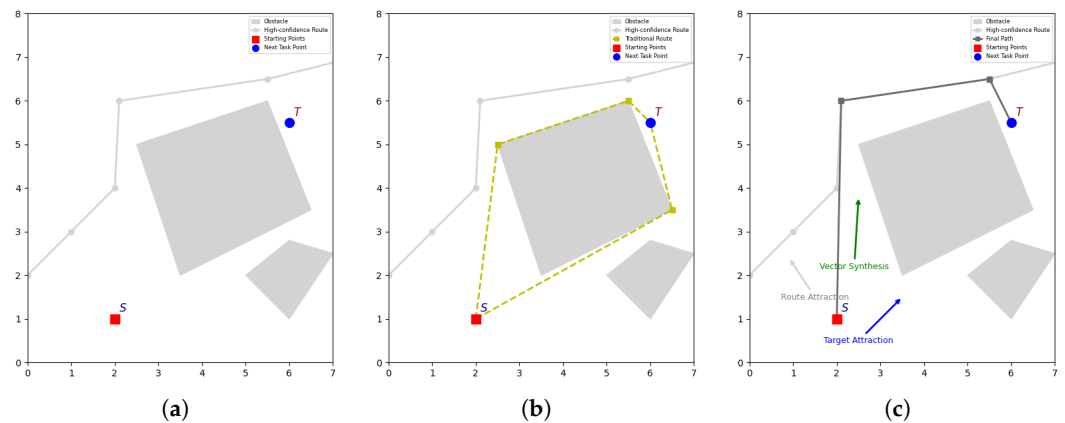


Figure 8. Schematic of the dual-attraction visibility graph Model. (a) Initial State. (b) Traditional connection method. (c) Visualization approach utilized in this study.

The pseudocode of the attraction-guided model is presented in Algorithm 1, which defines the connection criteria, candidate route selection, attraction-force computation, and collision-free validation for integrating task points into the high-confidence route network.

Algorithm 1 Attraction-Guided Route-Aware Visibility Graph

Input: Start point T_s , end point T_e , high-confidence route segments E_b , maximum number of connections $N = 3$, maximum connection distance $d_c = 800$ nmi, weights $\alpha = 0.7$, $\beta = 0.3$

Output: Fully connected route set E_d

```

1: Initialize  $E_d \leftarrow \emptyset$ 
2: if  $f_{\text{Haversine}}(T_s, T_e) \leq d_c$  then
3:   if the straight-line path ( $T_s \rightarrow T_e$ ) is collision-free then
4:      $E_d \leftarrow (T_s, T_e)$ 
5:     return  $E_d$ 
6:   end if
7: end if
8: Identify up to  $N$  high-confidence route segments in  $E_b$ 
9: for all candidate route segments  $r_n$  near  $T_s$  do
10:   Project  $T_s$  onto  $r_n$  to obtain intersection point  $I_{p_0}$ 
11:   Compute route attraction force  $\mathbf{F}_p = (I_{p_0} - T_s) / \|I_{p_0} - T_s\|$ 
12:   Compute goal attraction force  $\mathbf{F}_t^{\text{goal}} = (T_e - T_s) / \|T_e - T_s\|$ 
13:   Compute combined attraction force  $\mathbf{F}_t = \alpha \mathbf{F}_p + \beta \mathbf{F}_t^{\text{goal}}$ 
14:   Update intersection point  $I_{p_n}$  along direction  $\mathbf{F}_t$ 
15:   if the path ( $T_s \rightarrow I_{p_n}$ ) is collision-free then
16:     Accept  $I_{p_n}$  and break
17:   end if
18: end for
19: Repeat the above procedure symmetrically from  $T_e$  to obtain  $I_{p_m}$ 
20: Construct  $E_d = (T_s \rightarrow I_{p_n}) \cup E_b^{(n,m)} \cup (I_{p_m} \rightarrow T_e)$ 
21: return  $E_d$ 

```

Figure 9 shows the connections between selected task points and high-confidence routes, with generated paths exhibiting smooth transitions, structural continuity, and compliance with real-world navigation constraints.

Based on the dynamic fully connected network G_d , a distance matrix D_{network} is computed to support the subsequent IB-DARP task allocation and path planning. Figure 10 illustrates this matrix, where each element represents the route distance between the corresponding task points.

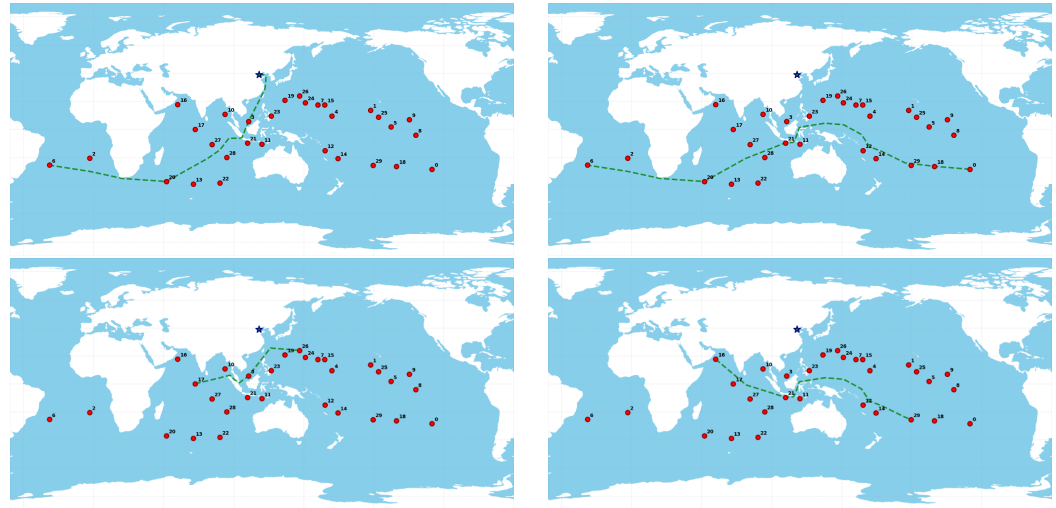


Figure 9. Dynamic task fully connected route (partial view).

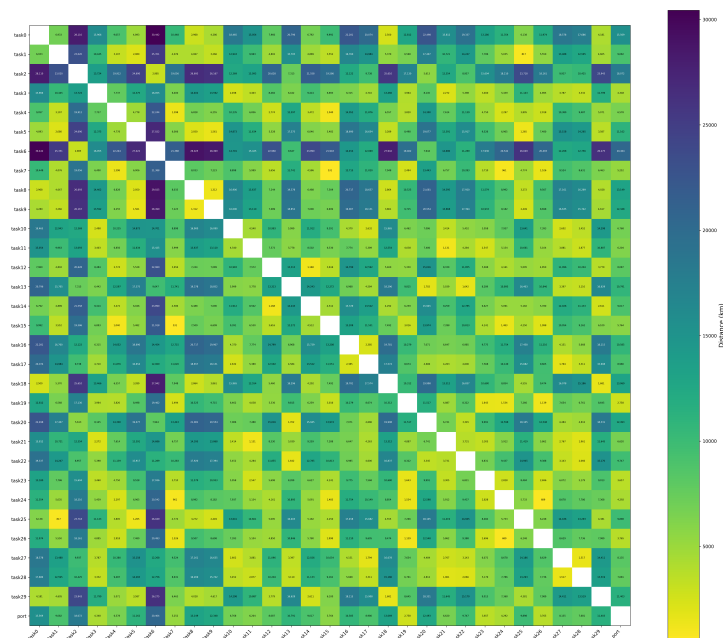


Figure 10. Schematic diagram of the dynamic fully connected distance matrix, where each entry denotes the sailing distance between two nodes (km).

3.2. IB-DARP Based Task Allocation and Path Planning

3.2.1. Initial Iteration: Task Region Division Mechanism

The traditional DARP algorithm requires five key elements: target point uniqueness, path completeness, load balancing, region connectivity, and initial position differentiation. For maritime missions, these are refined as follows: Task uniqueness is ensured by converting executed tasks into route nodes; path completeness and connectivity are maintained via the fully connected high-confidence route network; load balancing uses a cost function integrating heading consistency, spatial compactness, and workload, with iterative optimization based on current total time; for vessels departing from the same port, a polar-coordinate-based heading substitution mechanism establishes initial partitions aligned with operational requirements.

Based on these improvements, a task region pre-partitioning method is first defined. According to the coordinates of tasks executed at each iteration, the task range θ_R can be pre-partitioned. The range of θ_R is defined as the angular span from 0° (due east) counterclockwise, covering the first task T_{first} to the last task T_{end} .

$$\begin{aligned}\theta_s &= \frac{180^\circ}{\pi} \cdot \tan^{-1} \left(\frac{T_{\text{firstlat}} - P_{\text{lat}}}{T_{\text{firstlon}} - P_{\text{lon}}} \right), \\ \theta_e &= \frac{180^\circ}{\pi} \cdot \tan^{-1} \left(\frac{T_{\text{endlat}} - P_{\text{lat}}}{T_{\text{endlon}} - P_{\text{lon}}} \right), \\ \theta_R &= [\theta_s, \theta_e]\end{aligned}\quad (23)$$

Based on the number of vessels m involved in each mission execution, a preliminary heading-based partition can be established, yielding pre-partition axes $\theta_p = \{\theta_1, \dots, \theta_m\}$ and corresponding pre-partitioned regions $R_m = \{R_1, \dots, R_m\}$. Here, any pre-partition axis θ_k ($k = 1, \dots, m$) is expressed as

$$\theta_k = \theta_s + \frac{2k-1}{2m}(\theta_e - \theta_s), \quad k = 1, 2, \dots, m \quad (24)$$

The regional partitioning is defined as follows:

$$\begin{cases} R_1 = \{\theta \in [\theta_s, \theta_e] \mid \theta_s \leq \theta \leq \theta_1\}, \\ R_j = \{\theta \in [\theta_s, \theta_e] \mid \theta_{j-1} \leq \theta \leq \theta_j\}, \quad j = 2, \dots, m-1, \\ R_m = \{\theta \in [\theta_s, \theta_e] \mid \theta_{m-1} \leq \theta \leq \theta_e\} \end{cases} \quad (25)$$

With complete initialization, IB-DARP proceeds to the initial iteration phase, employing a spatially informed auction mechanism for task allocation. Tasks are assigned in descending order of distance from the port, starting with the farthest point, to reduce uncertainties from random assignment,

$$T_{\text{sorted}} = \text{sort}[T, D_{\text{network}}(T_i, P) \downarrow] \quad (26)$$

Under a fair competitive environment, each region competes to acquire tasks with the objective of minimizing a comprehensive cost function, which consists of the following three dimensions:

(1) Heading angle matching degree

$$C_{\text{direction}}(i, j) = \frac{|\theta_i - \theta_j|}{\theta_j - \theta_s}, \quad (27)$$

where θ_i represents the directional angle of the current auction task T_i , and θ_j denotes the angle of the pre-partition axis for region R_j .

(2) Estimated load balancing

$$C_{\text{load}}(i, j) = \frac{|R_j| - n/m}{n/m} \quad (28)$$

Normalized:

$$C_{\text{load}}^{\text{norm}}(j) = \max \left(0, \min \left(1, \frac{C_{\text{load}}(i, j) + 1}{2} \right) \right), \quad (29)$$

where $|R_j|$ is the number of tasks in region R_j , $m = |S|$ is the number of regions, $n = |T|$ is the total number of tasks.

(3) Spatial compactness

$$C_{\text{compactness}}(i, j) = \frac{\max(\Theta_j \cup \{\theta_i\}) - \min(\Theta_j \cup \{\theta_i\})}{\theta_e - \theta_s}, \quad (30)$$

where $\Theta_j = \{\theta_i \mid T_i \in R_j\}$ represents the set of angles of existing tasks within region R_j , and $\theta_e - \theta_s$ denotes the total pre-partitioned area.

For task T_i and region R_m , the final comprehensive cost function is defined as

$$C_{\text{total}}^{\text{initial}}(i, j) = \omega_d \cdot C_{\text{direction}}(i, j) + \omega_l \cdot C_{\text{load}}^{\text{norm}}(j) + \omega_c \cdot C_{\text{compactness}}(i, j) \quad (31)$$

where $\omega_d + \omega_l + \omega_c = 1$.

The auction decision mechanism follows the minimum cost principle: for each sorted task $T_i \in T_{\text{sorted}}$, it is allocated to the region with the lowest comprehensive cost,

$$R_j = \arg \min_{R_k \in R_m} C_{\text{total}}(i, j) \quad (32)$$

After all task points have been assigned to regions, the mean direction vector $\vec{V}_j$, for tasks within each region R_j is calculated as

$$\vec{V}_j = \frac{1}{|R_j|} \sum_{T_i \in R_j} (T_{\text{lon}_i} - P_{\text{lon}}, T_{\text{lat}_i} - P_{\text{lat}}) \quad (33)$$

where P represents the port's latitude and longitude coordinates, and $\vec{V}_j$ denotes the mean direction vector of task points within region R_j .

Based on this vector, the angles of the pre-partition axes are further optimized. The calculation formula for any optimized axis θ_k can be expressed as

$$\theta_k = \frac{180^\circ}{\pi} \cdot \text{atan2}(V_{k,\text{lat}}, V_{k,\text{lon}}), \quad k = 1, 2, \dots, m \quad (34)$$

The set of optimized axes for the regions, $\theta_o = \{\theta_1, \dots, \theta_m\}$, can be obtained, where θ_o defines the final directional reference for regional partitioning.

The final regional partitioning satisfies both completeness and mutual exclusivity,

$$\bigcup_{m=1}^m R_m = T, \quad R_i \cap R_j = \emptyset \quad (i \neq j) \quad (35)$$

The specific partitioning is determined by the optimized axis angles,

$$\begin{cases} R_1 = \{ T_i \in T \mid \theta_s \leq \angle(\vec{PT}_i) \leq \theta_1 \}, \\ R_j = \{ T_i \in T \mid \theta_{j-1} \leq \angle(\vec{PT}_i) \leq \theta_j \}, \quad j = 2, \dots, m-1, \\ R_m = \{ T_i \in T \mid \theta_{m-1} \leq \angle(\vec{PT}_i) \leq \theta_e \}. \end{cases} \quad (36)$$

where $\angle(\vec{PT}_i)$ represents the directional angle between the task point and the port, and all directional angles contained within the range R_m fall within the partitioned region.

This allocation relationship can be expressed as a mapping function Φ ,

$$\Phi : T \rightarrow R, \quad \Phi(T_i) = R_m \iff T_i \in R_m, \quad (37)$$

where Φ is a mapping from the task set T to the region set R , and $\Phi(T_i) = R_m$ indicates that task T_i is allocated to region R_m if and only if $T_i \in R_m$.

Following the aforementioned procedure, the initial task distribution can be calculated. Figure 11 illustrates two common scenarios encountered during the planning process.

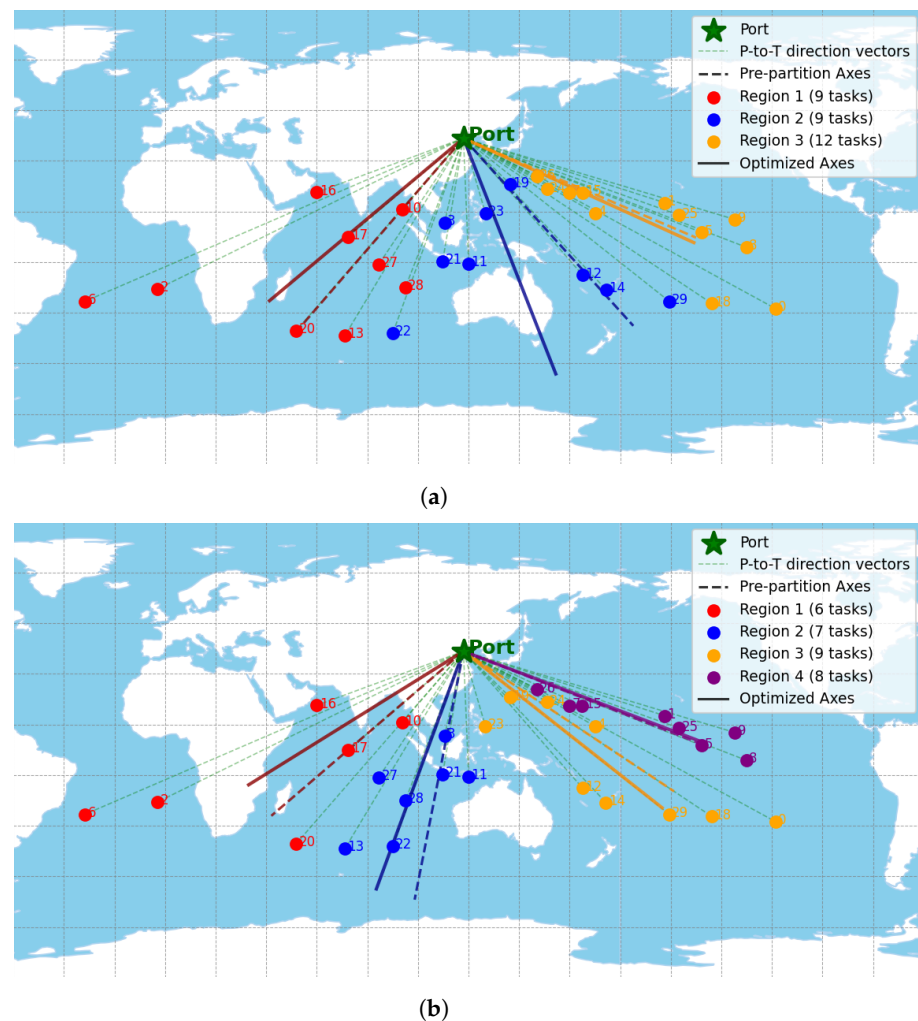


Figure 11. Schematic diagram of first iteration task allocation results. (a) Initial Task Distribution Results for $m = 3$. (b) Initial Task Distribution Results for $m = 4$.

3.2.2. Balancing Iteration: Vessel Load Balancing Mechanism

Based on the initial task partitioning, the original mTSP is decomposed into multiple regional TSP subproblems. At this stage, a multi-objective optimization model is applied exclusively to TSP-level route planning, jointly considering travel distance, task execution cost, and task coverage effectiveness.

The resulting routes provide quantitative estimates of travel distance and execution time, which are then used by the IB-DARP iterative mechanism to evaluate workload imbalance and guide subsequent task migration and balancing. The detailed TSP solution procedure is summarized in Algorithm 2.

Based on the initial task partitioning, each regional TSP subproblem is solved using Algorithm 2 to obtain an optimized visiting sequence. Let the ordered task set of region R_k after TSP sequencing be denoted as $T_{R_k}^{\text{sorted}}$. The total length of the resulting route can be directly computed from the dynamic fully connected distance matrix D_{network} as

$$L(\pi_k) = D_{\text{network}}(P, T_{R_k,1}^{\text{sorted}}) + \sum_{j=1}^{l_k-1} D_{\text{network}}(T_{R_k,j}^{\text{sorted}}, T_{R_k,j+1}^{\text{sorted}}) + D_{\text{network}}(T_{R_k,l_k}^{\text{sorted}}, P), \quad (38)$$

where $l_k = |T_{R_k}^{\text{sorted}}|$ denotes the number of tasks in region R_k .

Algorithm 2 Multi-Objective Path Planning for a TSP Instance

Input: Task set of region R_k : T_{R_k} , port node P , dynamic fully connected distance matrix D_{network} , task time windows $\{[e_i, l_i] \mid i \in T_{R_k}\}$, task weights $\{w_i \mid i \in T_{R_k}\}$, objective weights ($\gamma_1 = 0.4, \gamma_2 = 0.3, \gamma_3 = 0.3$), ACO parameters ($N_{\text{ant}} = 200, N_{\text{iter}} = 200, \alpha_{\text{ACO}} = 1.2, \beta_{\text{ACO}} = 3.4, \rho = 0.5, Q = 100$)

Output: Optimized TSP route π_k , route length $L(\pi_k)$

- 1: Initialize pheromone matrix $\tau(i, j)$ uniformly for all $i, j \in T_{R_k} \cup \{P\}$
- 2: Precompute normalization bounds: $w_{\max} \leftarrow \max_{i \in T_{R_k}} w_i, d_{\max} \leftarrow \max_{i, j} D_{\text{network}}(i, j), t_{\max} \leftarrow \max_{i \in T_{R_k}} (l_i - e_i)$
- 3: **for** iter = 1 to N_{iter} **do**
- 4: **for** ant = 1 to N_{ant} **do**
- 5: Initialize route $\pi \leftarrow [P]$, current node $i \leftarrow P$, current time $t \leftarrow 0$
- 6: Initialize visited task set $V \leftarrow \emptyset$, accumulated weight $W \leftarrow 0$
- 7: **while** $V \neq T_{R_k}$ **do**
- 8: Candidate set $C \leftarrow T_{R_k} \setminus V$
- 9: **for all** $j \in C$ **do**
- 10: $d_{ij} \leftarrow D_{\text{network}}(i, j), t_{\text{arr}} \leftarrow t + \text{travel_time}(d_{ij})$
- 11: **if** $t_{\text{arr}} > l_j$ **then**
- 12: $c_{ij} \leftarrow +\infty$
- 13: **else**
- 14: $t_{\text{wait}} \leftarrow \max(0, e_j - t_{\text{arr}})$
- 15: $\hat{w}_j \leftarrow w_j / w_{\max}, \hat{d}_{ij} \leftarrow d_{ij} / d_{\max}, \hat{t}_{\text{wait}} \leftarrow t_{\text{wait}} / t_{\max}$
- 16: $c_{ij} \leftarrow \gamma_1 \hat{w}_j + \gamma_2 \hat{d}_{ij} + \gamma_3 \hat{t}_{\text{wait}}$
- 17: **end if**
- 18: $\eta(i, j) \leftarrow 1 / (\epsilon + c_{ij})$
- 19: **end for**
- 20: Select next node j^* from C with probability $p(i, j) \propto \tau(i, j)^{\alpha_{\text{ACO}}} \eta(i, j)^{\beta_{\text{ACO}}}$
- 21: Append j^* to π , update $V \leftarrow V \cup \{j^*\}, W \leftarrow W + w_{j^*}$
- 22: Update $t \leftarrow t + \text{travel_time}(d_{ij^*}) + t_{\text{wait}}(j^*), i \leftarrow j^*$
- 23: **end while**
- 24: Append return to port: $\pi \leftarrow \pi + [P]$
- 25: Compute route distance $f_1(\pi)$ and coverage ratio $f_3(\pi)$
- 26: Store solution candidate $(\pi, f_1(\pi), f_3(\pi))$
- 27: **end for**
- 28: Select iteration-best route π^* by lexicographic preference:
 - (i) feasibility under time windows,
 - (ii) maximum f_3 ,
 - (iii) minimum f_1
- 29: Update pheromone: $\tau \leftarrow (1 - \rho)\tau$
- 30: **for all** edges $(u, v) \in \pi^*$ **do**
- 31: $\tau(u, v) \leftarrow \tau(u, v) + Q / (\epsilon + f_1(\pi^*))$
- 32: **end for**
- 33: **end for**
- 34: **return** $\pi_k \leftarrow \pi^*, L(\pi_k) \leftarrow f_1(\pi_k)$

Meanwhile, in practical applications, vessels exhibit certain speed variations. Let the set of optimal vessel speeds be denoted as $\mathbf{v} = \{v_1, v_2, \dots, v_m\}$. By sorting the regions in descending order of path length and the vessels in descending order of speed, a vessel-region allocation mapping is established.

$$\Phi : R_{(i)} \leftrightarrow S_{(i)}, \quad \text{for } i = 1, 2, \dots, m \quad (39)$$

where $R_{(i)}$ denotes the region with the i -th largest path length, and $S_{(i)}$ represents the vessel with the i -th highest speed.

This allows for the determination of the total travel duration for each vessel, which serves as the load metric used in the subsequent IB-DARP migration and balancing process. Specifically, the travel time of vessel k is defined as:

$$\mathcal{T}_k = \frac{L(\pi_k)}{v_k \times 1.852} \quad (40)$$

Using the travel time $\mathcal{T}_k$ as the load metric, the rationality of the allocation scheme is evaluated. The average load and maximum imbalance degree are calculated as follows:

$$\bar{\mathcal{T}} = \frac{1}{m} \sum_{k=1}^m \mathcal{T}_k, \quad \Delta = \max_k \mathcal{T}_k - \min_k \mathcal{T}_k \quad (41)$$

To prevent the algorithm from falling into short-cycle oscillations due to repeated migration of the same tasks, a task-based prohibition mechanism is introduced. A historical task set $\mathcal{H}_{\text{task}}$ is defined to record tasks migrated during the most recent N iterations. When selecting a migration task, it must be chosen from the highest-load region R_{max} while satisfying $\mathcal{T}_{\text{migrate}} \notin \mathcal{H}_{\text{task}}$. If all candidate tasks are prohibited, the algorithm terminates.

$$\mathcal{H}_{\text{task}} = \{T_{\text{migrate}}^{(t-1)}, T_{\text{migrate}}^{(t-2)}, \dots, T_{\text{migrate}}^{(t-N)}\} \quad (42)$$

where N is a predefined window size, indicating that after a task is migrated, it cannot be migrated again within the subsequent N iterations. If $|\mathcal{H}_{\text{task}}| > N$, the earliest added task is removed. Initially, $\mathcal{H}_{\text{task}} = \emptyset$.

When the balance condition $\Delta > \alpha \cdot \bar{\mathcal{T}}$ is not satisfied, the iterative optimization mechanism is triggered. First, the vessel with the highest current load is identified,

$$S_{\text{max}} = \arg \max_k T_k \quad (43)$$

and a migration task is selected from its corresponding region R_{max} :

$$\text{Candidates} = \{T \in R_{\text{max}} \mid T \notin \mathcal{H}_{\text{task}}\} \quad (44)$$

The determination of the migration task follows the following rule: if the candidate set is non-empty, select the task with the highest comprehensive cost from the initial auction phase.

$$\mathcal{T}_{\text{migrate}} = \begin{cases} \arg \max_{T \in \text{Candidates}} C_{\text{total}}^{\text{initial}}(i = \text{max}, j = \text{max}), & \text{if Candidates} \neq \emptyset, \\ \emptyset (\text{Termination triggered}), & \text{otherwise} \end{cases} \quad (45)$$

After selecting the migration task, the system reallocates it to the suboptimal region with the lowest comprehensive cost, excluding the original region,

$$R_{\text{sub}} = \arg \max_{k \neq \text{max}} C_{\text{total}}^{\text{initial}}(i = \text{migrate}, j = k) \quad (46)$$

After each migration is completed, the historical task set is updated: $\mathcal{H}_{\text{task}} \cup \{\mathcal{T}_{\text{migrate}}\}$. To prevent the algorithm from falling into short-cycle oscillations due to repeated migration of the same tasks, a task-based prohibition mechanism is introduced. A historical task set $\mathcal{H}_{\text{task}}$ is defined to record tasks that have been migrated in the recent N iterations.

During each migration task selection, candidates are filtered from the highest-load region R_{max} , excluding any tasks present in the prohibition set $\mathcal{H}_{\text{task}}$. The selected task $\mathcal{T}_{\text{migrate}}$ must satisfy $\mathcal{T}_{\text{migrate}} \notin \mathcal{H}_{\text{task}}$.

Subsequently, TSP path planning is reperformed for the two affected regions R_{max} and R_{sub} where changes occurred. After planning, region-vessel rematching is conducted, and this process repeats until load balance is achieved as follows:

- (1) Load balance is satisfied ($\Delta < \alpha \cdot \bar{T}$).
- (2) The maximum iteration count K_{max} is reached.
- (3) No migratable tasks remain (Candidates = $\emptyset$).

Although the algorithm involves multiple TSP solutions, the computational complexity is significantly reduced since each iteration only requires optimizing two local regions, thus maintaining high computational efficiency overall.

4. Simulation Experiments

This chapter validates the proposed algorithm through comparative experiments using historical real-world operational data and different algorithm configurations. The focus is on evaluating the algorithm's performance in terms of load balancing, task coverage, and path planning quality.

4.1. Experimental Design

To systematically evaluate IB-DARP in multi-vessel task allocation and path planning, experiments were conducted to assess parameter sensitivity and scalability. Baseline parameters are listed in Table 2. Experimental environment: All experiments were conducted on a hardware platform equipped with a 13th Gen Intel(R) Core(TM) i9-13900KF @ 3.00 GHz processor.

Table 2. Algorithm parameter configuration.

Algorithm Category	Algorithm Name	Key Parameter	Parameter Value
Proposed Method	IB-DARP	(α, N, K_{max})	(0.15, 3, 50)
Underlying Solver	ACO	$(N_{ant}, N_{iter}, \alpha_{ACO})$ (β_{ACO}, ρ, Q)	(200, 200, 1.2) (3.4, 0.5, 100)

4.1.1. Ablation Study

A systematic ablation study was conducted to quantify the contribution of each IB-DARP module. Key components—including heading angle consistency, load balancing, and spatial compactness—were individually disabled to form baseline variants. Experiments were performed on 20 representative scenarios with 50 task points and three vessels to ensure comparability and representativeness.

- (1) Without heading angle matching degree

In this set of experiments, the heading angle consistency constraint was disabled, leaving only the estimated load balancing and spatial compactness modules active. The parameter settings are summarized in Table 3.

Table 3. Parameter settings for ablation study: without heading angle matching degree.

Experiment ID	Heading Angle Matching Degree	Estimated Load Balancing	Spatial Compactness
1	0	0.8	0.2
2	0	0.6	0.4
3	0	0.4	0.6
4	0	0.2	0.8

The experimental results are illustrated in Figure 12.

When the heading angle consistency constraint is disabled, load balancing is still achievable, but paths exhibit crossovers, detours, and unnecessary loops, underscoring

the constraint's key role in maintaining trajectory continuity and structural integrity. High weight on load balancing produces uniform initial allocations requiring larger iterative adjustments, extending computation time, while high spatial compactness emphasizes nearby tasks, showing locally greedy behavior with limited speed improvement. In this experiment, computation time ranged from 26 to 31 s, maximum load imbalance slightly increased, and total sailing time was 2113 h. The absence of heading guidance expands the search space, favoring shorter geometric paths but causing irregular, operationally unrealistic trajectories, as illustrated in Figure 13.

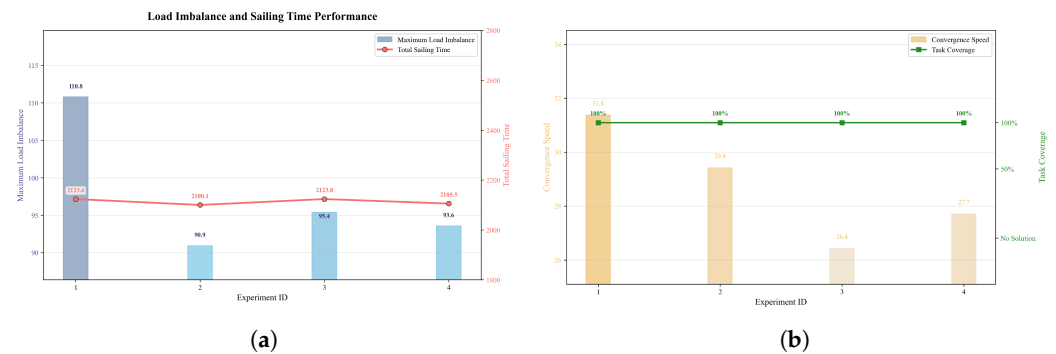


Figure 12. Experimental results illustration without heading angle matching degree. (a) Load imbalance (h) and total sailing time (h) under different parameter settings. (b) Convergence speed(s) and task coverage under different parameter settings.

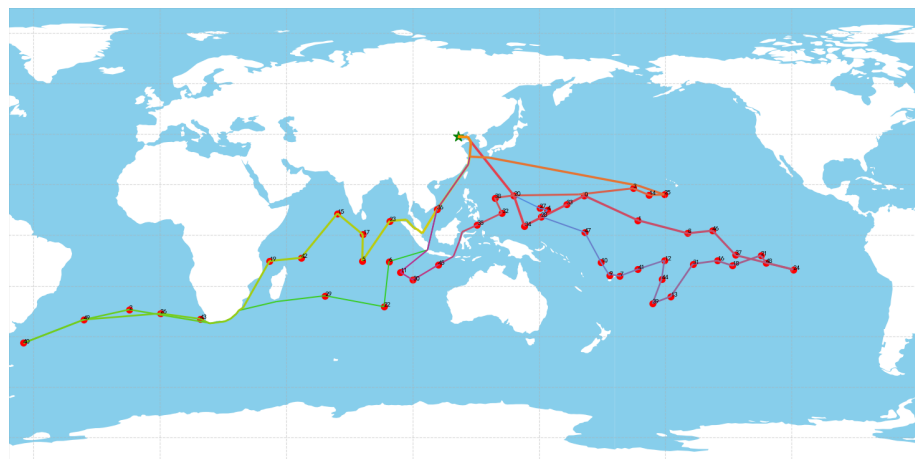


Figure 13. Illustration of typical path-crossing results.

(2) Without estimated load balancing

In this set of experiments, the load balancing strategy was disabled, while the heading angle matching degree and spatial compactness constraints remained active. The parameter settings are summarized in Table 4.

Table 4. Parameter settings for ablation study: without estimated load balancing.

Experiment ID	Heading Angle Matching Degree	Estimated Load Balancing	Spatial Compactness
1	0.8	0	0.2
2	0.6	0	0.4
3	0.4	0	0.6
4	0.2	0	0.8

The experimental results are illustrated in Figure 14.

With the load balancing component removed, the remaining constraints—heading angle consistency and spatial compactness—sustain the basic spatial structure and rationality of paths. Experimental results show computation time ranging from 21 to 32 s, a slight increase in maximum load imbalance, and minimal changes in total sailing time, indicating that load balancing primarily fine-tunes allocations rather than governs global structure. The spatial distribution of task points and spatial compactness dominate load balance, and occasional local path crossovers do not compromise overall trajectory feasibility. Thus, under the current scenario and dataset, load balancing is a weak contributor, becoming critical only with highly uneven or clustered task distributions, as illustrated in Figure 15.

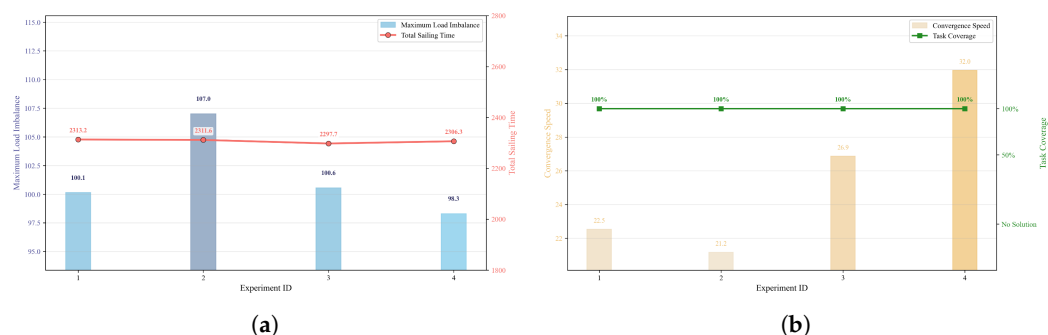


Figure 14. Experimental results illustration without estimated load balancing. (a) Load imbalance(h) and total sailing time(h) under different parameter settings. (b) Convergence speed(s) and task coverage under different parameter settings.

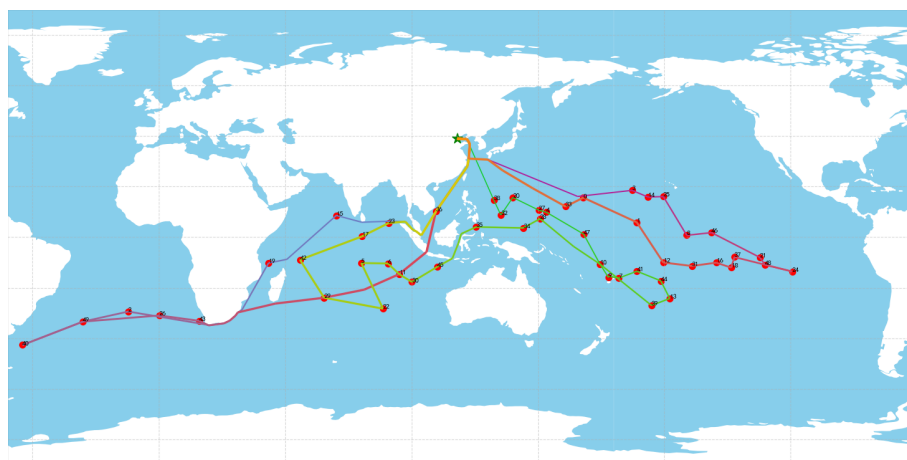


Figure 15. Illustration of Typical Results without estimated load balancing.

(3) Without spatial compactness

In this set of experiments, the spatial compactness constraint was disabled, while the heading angle matching degree and estimated load balancing modules remained active. The parameter settings are summarized in Table 5.

Table 5. Parameter settings for ablation study: without spatial compactness.

Experiment ID	Heading Angle Matching Degree	Estimated Load Balancing	Spatial Compactness
1	0.8	0.2	0
2	0.6	0.4	0
3	0.4	0.6	0
4	0.2	0.8	0

The experimental results are illustrated in Table 6.

Removal of the spatial compactness constraint severely degrades system performance. Increased weighting of estimated load balancing sharply reduces computational efficiency, while task allocation becomes spatially dispersed, losing regional clustering and resulting in frequent cross-region routing, disordered sequences, and poor path quality. Quantitatively, non-convergence probability rises to 0.95 in Experiment 1, with successful runs requiring 89–114 s, 4 to 6 times longer than the baseline. Mechanistically, spatial compactness suppresses task dispersion and excessive competition for distant tasks, stabilizing allocation boundaries and reducing the solution space. Without it, the algorithm undergoes numerous corrective iterations, leading to high computational costs and convergence difficulties. These findings confirm spatial compactness as a critical component in IB-DARP for maintaining path coherence, stabilizing the search space, and ensuring efficient, reliable convergence.

Table 6. Convergence performance and computation time without spatial compactness.

Experiment ID	Non-Convergence Probability (<120 s)	Avg. Computation Time of Successful Runs
1	0.95	113.94
2	0.6	100.95
3	0.4	89.74
4	0.4	92.15

Based on the above analysis, this study further conducted comparative experiments with different weight combinations to optimize the parameter configuration of each module in IB-DARP, aiming to identify the optimal balance for overall performance. The experimental settings are summarized in Table 7.

Table 7. Parameter settings for weight combination experiments.

Experiment ID	Heading Angle Matching Degree	Spatial Compactness	Estimated Load Balancing
1	0.8	0.1	0.1
2	0.6	0.3	0.1
3	0.5	0.4	0.1
4	0.6	0.2	0.2
5	0.5	0.3	0.2
6	0.1	0.8	0.1
7	0.3	0.6	0.1
8	0.4	0.5	0.1
9	0.2	0.6	0.2
10	0.3	0.5	0.2

The experimental results are illustrated in the following Figure 16.

The results confirm the robustness of IB-DARP, with stable computation time, total sailing time, and maximum load imbalance. Parameter analysis indicates that prioritizing spatial compactness and heading angle alignment ensures path continuity, regional clustering, and navigational rationality, while appropriately weighting estimated load balancing mitigates task allocation deviations. The optimal configuration, with heading angle matching 0.6, spatial compactness 0.3, and estimated load balancing 0.1, yields the most stable and balanced planning performance across scenarios.

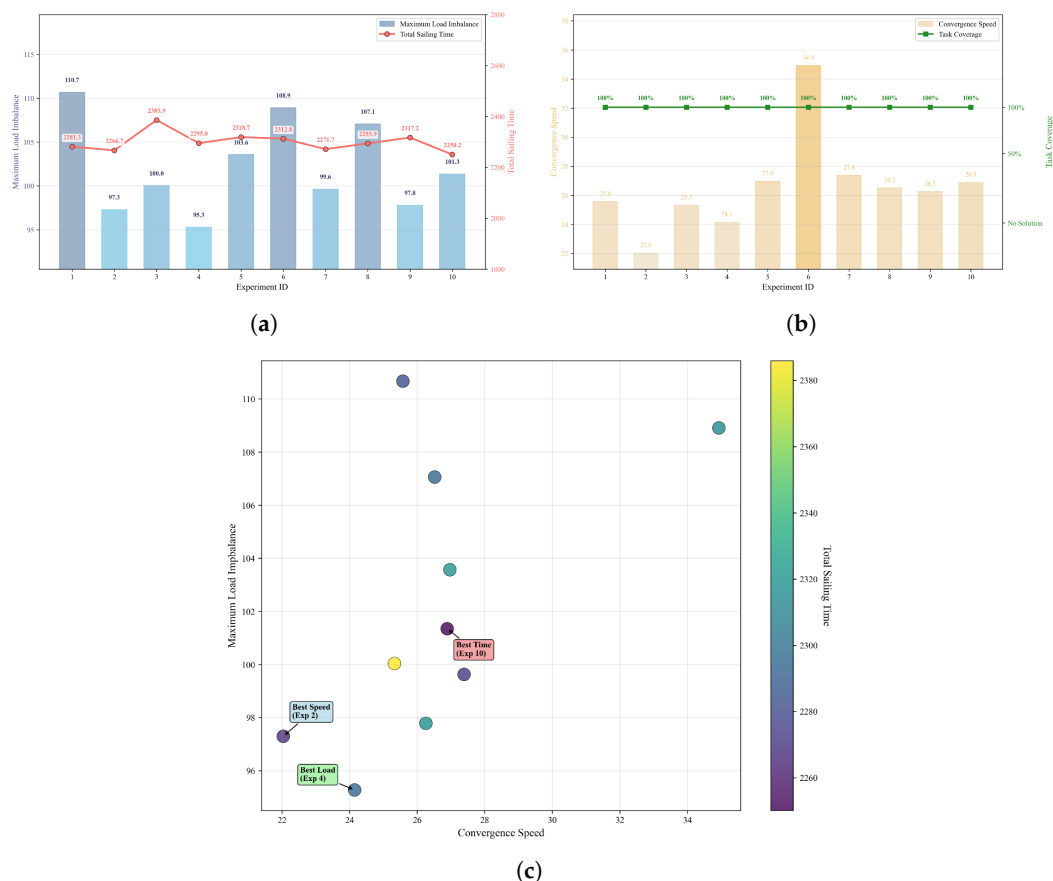


Figure 16. Experimental results illustration under parameter settings of weight combinations. (a) Load imbalance (h) and total sailing time (h) under different parameter settings. (b) Convergence speed (s) and task coverage under different parameter settings. (c) Performance comparison among experiments, including load imbalance (h), total sailing time (h), and convergence speed (s).

4.1.2. Scalability Experiment Design

This experiment is conducted using real historical operational data. By constructing test scenarios with varying task scales and fleet configurations, we evaluate the generalization capability of the proposed algorithm. The detailed settings of the comprehensive scenarios are shown in Table 8.

Table 8. Comprehensive scenario settings.

Scenario ID	Number of Tasks	Number of Ships	Number of Task Spatial Clusters
S1	30	3	20
		4	20
S2	50	3	20
		4	20
S3	70	3	20
		4	20
		5	20
S4	90	3	20
		4	20
		5	20

For each scenario defined in Table 8, the proposed IB-DARP algorithm is executed multiple times, and the results in Table 9 report the averaged performance metrics across repeated runs. This provides a quantitative summary of convergence performance under different task scales and fleet configurations.

Experimental results demonstrate that the algorithm achieves stable and superior performance across all representative scenarios. It consistently maintains a high task coverage rate and rapid convergence, while keeping the workload imbalance ratio below the anticipated 15% threshold (defined as the maximum difference in total sailing time among vessels normalized by the average sailing time). These findings indicate that IB-DARP exhibits excellent adaptability and robustness in large-scale task planning scenarios.

Table 9. Averaged convergence performance of IB-DARP over multiple runs under different scenarios defined in Table 8.

Scenario ID	Number of Ships	Task Coverage	Convergence Time (s)
S1	3	100%	16.45
S1	4	100%	18.47
S2	3	100%	21.34
S2	4	100%	26.56
S3	3	100%	24.10
S3	4	100%	29.98
S3	5	100%	42.08
S4	3	100%	30.87
S4	4	100%	37.37
S4	5	100%	53.25

4.1.3. Algorithm Performance Comparison

In practical maritime task planning scenarios, most metaheuristic algorithms require problem-specific adaptation and cannot be directly applied to the considered setting. To enable a fair and implementable comparison, we select representative baseline methods with reasonable configurations, including a GA-based multi-route planner, a multi-colony ACO approach, and a clustering-based method combining HDBSCAN with a heuristic TSP solver. Under the same representative scenario (S2 with 50 tasks and 4 vessels), these methods are quantitatively compared with the proposed IB-DARP in terms of task coverage, computational efficiency, and load balance, as summarized in Table 10. All results are obtained from 30 independent experimental runs.

Table 10. Performance comparison under Scenario S2.

Method	Task Coverage	Convergence Time (s)	Maximum Load Imbalance
GA	100%	119.89	14.87%
Multi-colony ACO	100%	330.18	13.24%
k-means + ACO	100%	8.17	36.43%
IB-DARP	100%	27.08	10.91%

Genetic-algorithm-based methods are able to reach feasible solutions; however, their performance is highly sensitive to constraint design. Overly strict planning constraints significantly increase convergence cost, whereas relaxed constraints often result in irregular, intersecting, or operationally infeasible routes, limiting practical reliability. Multi-colony ACO can generate more realistic and coherent routes by jointly considering task assignment and path construction, but the need to optimize multiple routes under complex constraints leads to a substantial computational burden and slow convergence. Clustering-based approaches (e.g., k-means + ACO) offer high computational efficiency, yet they are strongly dependent on the initial spatial distribution of tasks and tend to suffer from severe load imbalance when task densities vary across regions.

In contrast, IB-DARP achieves a better balance between solution quality and efficiency by decoupling task allocation and route optimization through an iterative balancing mechanism. This enables stable load distribution and realistic routing with moderate computational cost. Moreover, since the TSP solver within IB-DARP is modular, more advanced heuristics can be integrated in future work to further improve convergence speed without altering the overall framework.

4.2. Demonstration of IB-DARP Planning Results

To intuitively illustrate the planning capability of IB-DARP, this section presents representative planning results obtained from typical test scenarios. As shown in Figure 17.

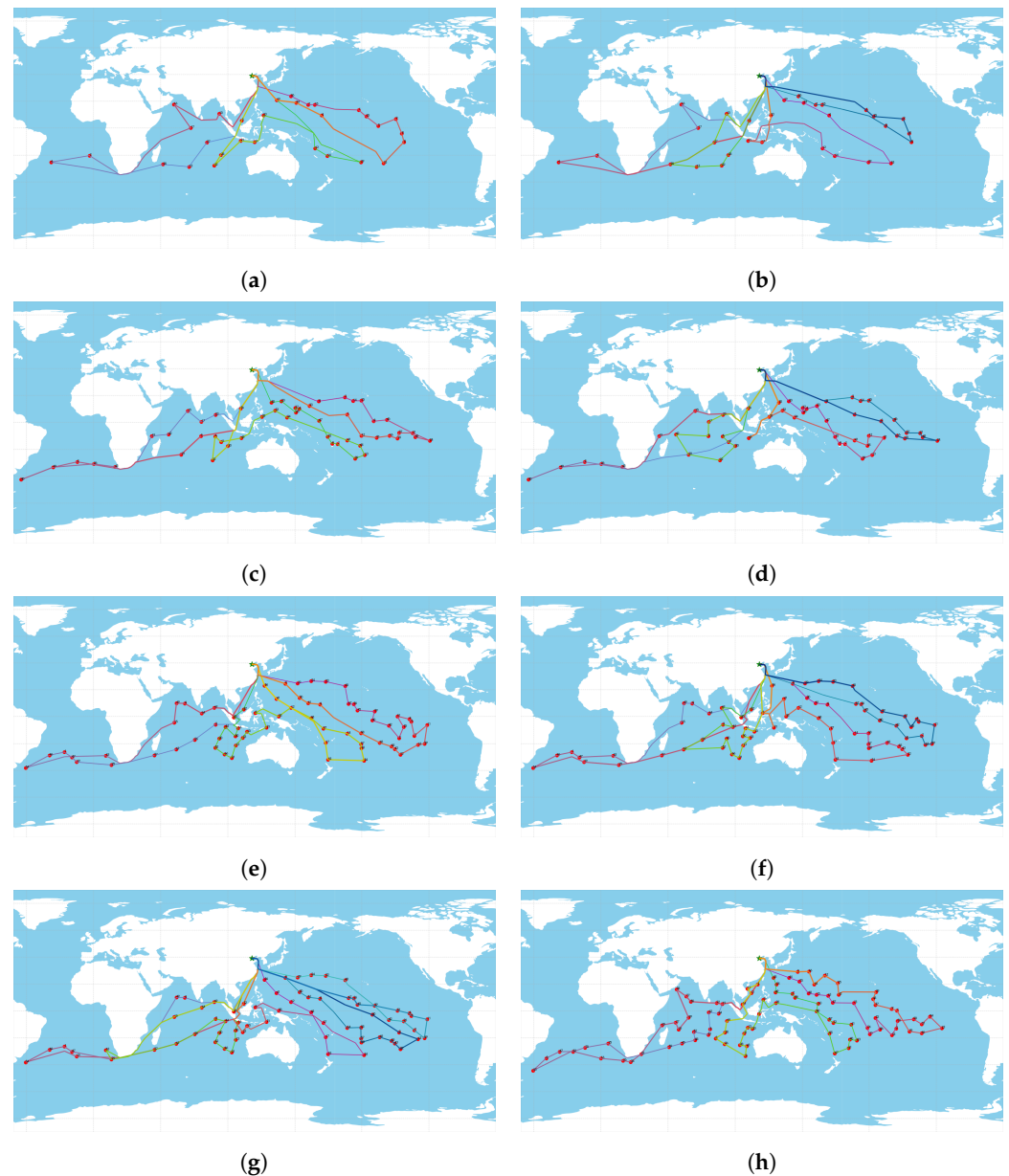


Figure 17. Cont.

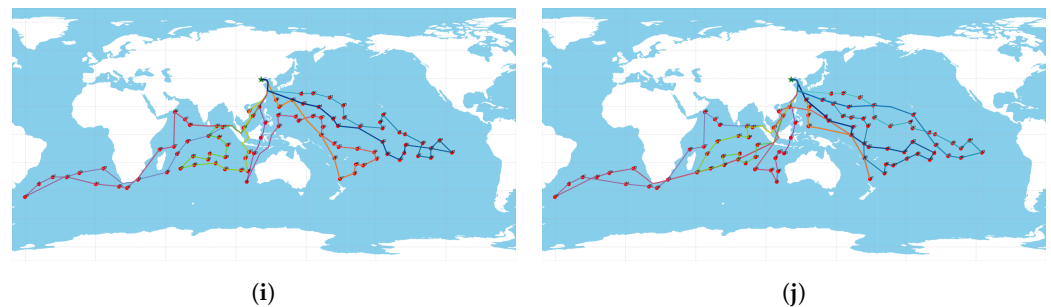


Figure 17. Illustration of planning results. (a) Task allocation and routing outcome in Scenario S1 with 3 vessels. (b) Task allocation and routing outcome in Scenario S1 with 4 vessels. (c) Task allocation and routing outcome in Scenario S2 with 3 vessels. (d) Task allocation and routing outcome in Scenario S2 with 4 vessels. (e) Task allocation and routing outcome in Scenario S3 with 3 vessels. (f) Task allocation and routing outcome in Scenario S3 with 4 vessels. (g) Task allocation and routing outcome in Scenario S3 with 5 vessels. (h) Task allocation and routing outcome in Scenario S4 with 3 vessels. (i) Task allocation and routing outcome in Scenario S4 with 4 vessels. (j) Task allocation and routing outcome in Scenario S4 with 5 vessels.

5. Conclusions

This work proposes IB-DARP, a direction-aware and load-balanced framework for multi-vessel cooperative task planning. By integrating angular partitioning, auction-based assignment, and iterative neighborhood migration, the method achieves structured task allocation and improved workload balance. Experimental results show that IB-DARP maintains high solution quality, ensures full task coverage, and significantly reduces computation time in large-scale scenarios. The framework offers an effective and scalable approach for intelligent scheduling in maritime cooperative operations.

Future Work

The current IB-DARP framework is developed under static or quasi-static assumptions and provides a structured task-level planning solution once vessels and mission requirements are clearly defined. Future work will focus on extending this framework toward more realistic dynamic maritime operation scenarios by progressively incorporating dynamic and predictive capabilities. In particular, environmental and operational factors such as vessel speed variability, payload conditions, and meteorological–hydrodynamic information (e.g., wind, waves, and currents) will be considered to support energy-aware and predictive route planning. Moreover, the modular design of IB-DARP allows the underlying TSP solver to be further enhanced or replaced, and data-driven models, including deep learning–based approaches (e.g., transformer- or graph-based architectures), will be explored to enable dynamic decision-making and adaptive task reallocation during mission execution, bridging the gap between pre-mission planning and online cooperative control.

Author Contributions: Conceptualization, Y.W.; Methodology, Y.W.; Software, Y.W.; Validation, Y.W.; Formal analysis, Y.W.; Investigation, Y.W.; Resources, Y.W.; Data curation, Y.W.; Writing—original draft, Y.W.; Writing—review & editing, Y.W.; Visualization, Y.W.; Supervision, L.L.; Project administration, L.L.; Funding acquisition, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the “The key technology research and engineering demonstration of green and intelligent ships in the Xijiang River” (Grant GuikeAA23023013) of the Guangxi Science and Technology Major Program. This research was also financially supported by the “Research and application of key technologies for green standard ship design and manufacturing” (Grant GuikeAA23062037) of the Guangxi Science and Technology Major Program.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Han, S.; Wang, L.; Wang, Y.; He, H. A dynamically hybrid path planning for unmanned surface vehicles based on non-uniform Theta* and improved dynamic windows approach. *Ocean Eng.* **2022**, *257*, 111655. [\[CrossRef\]](#)
2. Lyridis, D.V. An improved ant colony optimization algorithm for unmanned surface vehicle local path planning with multi-modality constraints. *Ocean Eng.* **2021**, *241*, 109890. [\[CrossRef\]](#)
3. Yang, J.; Ni, J.; Li, Y.; Wen, J.; Chen, D. The Intelligent Path Planning System of Agricultural Robot via Reinforcement Learning. *Sensors* **2022**, *22*, 4316. [\[CrossRef\]](#)
4. Ai, B.; Jia, M.; Xu, H.; Xu, J.; Wen, Z.; Li, B.; Zhang, D. Coverage path planning for maritime search and rescue using reinforcement learning. *Ocean Eng.* **2021**, *241*, 110098. [\[CrossRef\]](#)
5. Silva Junior, A.G.d.; Santos, D.H.d.; Negreiros, A.P.F.d.; Silva, J.M.V.B.d.S.; Gonçalves, L.M.G. High-Level Path Planning for an Autonomous Sailboat Robot Using Q-Learning. *Sensors* **2020**, *20*, 1550. [\[CrossRef\]](#)
6. Guo, S.; Zhang, X.; Zheng, Y.; Du, Y. An Autonomous Path Planning Model for Unmanned Ships Based on Deep Reinforcement Learning. *Sensors* **2020**, *20*, 426. [\[CrossRef\]](#)
7. He, Y.; Li, Z.; Mou, J.; Hu, W.; Li, L.; Wang, B. Collision-avoidance path planning for multi-ship encounters considering ship manoeuvrability and COLREGs. *Transp. Saf. Environ.* **2021**, *3*, 103–113. [\[CrossRef\]](#)
8. Chen, P.; Pei, J.; Lu, W.; Li, M. A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance. *Neurocomputing* **2022**, *497*, 64–75. [\[CrossRef\]](#)
9. Hadi, B.; Khosravi, A.; Sarhadi, P. Deep reinforcement learning for adaptive path planning and control of an autonomous underwater vehicle. *Appl. Ocean Res.* **2022**, *129*, 103326. [\[CrossRef\]](#)
10. Li, L.; Wu, D.; Huang, Y.; Yuan, Z.-M. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Appl. Ocean Res.* **2021**, *113*, 102759. [\[CrossRef\]](#)
11. Xu, X.; Cai, P.; Ahmed, Z.; Yellapu, V.S.; Zhang, W. Path planning and dynamic collision avoidance algorithm under COLREGs via deep reinforcement learning. *Neurocomputing* **2022**, *468*, 181–197. [\[CrossRef\]](#)
12. Lee, W.; Choi, G.-H.; Kim, T. Visibility graph-based path-planning algorithm with quadtree representation. *Appl. Ocean Res.* **2021**, *117*, 102887. [\[CrossRef\]](#)
13. Zhou, X.; Wang, X.; Xie, Z.; Li, F.; Gu, X. Online obstacle avoidance path planning and application for arc welding robot. *Robot. Comput.-Integr. Manuf.* **2022**, *78*, 102413. [\[CrossRef\]](#)
14. Yao, J.; Gang, L. Ship Collaborative Path Planning Method Based on CS-STHA. *J. Mar. Sci. Eng.* **2022**, *10*, 1504. [\[CrossRef\]](#)
15. Matos, D.M.; Costa, P.; Sobreira, H.; Valente, A.; Lima, J. Efficient multi-robot path planning in real environments: A centralized coordination system. *Int. J. Intell. Robot. Appl.* **2024**, *9*, 217–244. [\[CrossRef\]](#)
16. Zaccone, R.; Martelli, M. A collision avoidance algorithm for ship guidance applications. *J. Mar. Eng. Technol.* **2019**, *19*, 62–75. [\[CrossRef\]](#)
17. Cao, S.; Fan, P.; Yan, T.; Xie, C.; Deng, J.; Xu, F.; Shu, Y. Inland Waterway Ship Path Planning Based on Improved RRT Algorithm. *J. Mar. Sci. Eng.* **2022**, *10*, 1460. [\[CrossRef\]](#)
18. Yan, X.; Wang, S.; Ma, F.; Liu, Y.; Wang, J. A novel path planning approach for smart cargo ships based on anisotropic fast marching. *Expert Syst. Appl.* **2020**, *159*, 113558. [\[CrossRef\]](#)
19. Chen, P.; Huang, Y.; Papadimitriou, E.; Mou, J.; van Gelder, P. Global path planning for autonomous ship: A hybrid approach of Fast Marching Square and velocity obstacles methods. *Ocean Eng.* **2020**, *214*, 107793. [\[CrossRef\]](#)
20. Gan, L.; Yan, Z.; Zhang, L.; Liu, K.; Zheng, Y.; Zhou, C.; Shu, Y. Ship path planning based on safety potential field in inland rivers. *Ocean Eng.* **2022**, *260*, 111928. [\[CrossRef\]](#)
21. Gao, P.; Zhou, L.; Zhao, X.; Shao, B. Research on ship collision avoidance path planning based on modified potential field ant colony algorithm. *Ocean Coast. Manag.* **2023**, *235*, 106482. [\[CrossRef\]](#)
22. Wang, M.; Ma, T.; Li, G.; Zhai, X.; Qiao, S. Ant Colony Optimization with an Improved Pheromone Model for Solving MTSP with Capacity and Time Window Constraint. *IEEE Access* **2020**, *8*, 106872–106879. [\[CrossRef\]](#)
23. Wu, L.; Huang, X.; Cui, J.; Liu, C.; Xiao, W. Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot. *Expert Syst. Appl.* **2023**, *215*, 119410. [\[CrossRef\]](#)
24. Wang, X.; Yao, X.; Zhang, L. Path Planning under Constraints and Path Following Control of Autonomous Underwater Vehicle with Dynamical Uncertainties and Wave Disturbances. *J. Intell. Robot. Syst.* **2020**, *99*, 891–908. [\[CrossRef\]](#)
25. Wang, Z.; Li, G.; Ren, J. Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm. *Comput. Commun.* **2021**, *166*, 49–56. [\[CrossRef\]](#)

26. Zhao, W.; Wang, Y.; Zhang, Z.; Wang, H. Multicriteria Ship Route Planning Method Based on Improved Particle Swarm Optimization–Genetic Algorithm. *J. Mar. Sci. Eng.* **2021**, *9*, 357. [\[CrossRef\]](#)
27. Shen, L.; Wang, Y.; Liu, K.; Yang, Z.; Shi, X.; Yang, X.; Jing, K. Synergistic path planning of multi-UAVs for air pollution detection of ships in ports. *Transp. Res. Part E* **2020**, *144*, 102128. [\[CrossRef\]](#)
28. Tjandra, S.S.; Setiawan, F.; Salsabila, H. Application of Genetic Algorithms to Solve MTSP Problems with Priority (Case Study at the Jakarta Street Lighting Service). *J. Optim. Syst. Ind.* **2022**, *21*, 75–86. [\[CrossRef\]](#)
29. Muthugala, M.A.V.J.; Samarakoon, S.M.B.P.; Elara, M.R. Toward energy-efficient online Complete Coverage Path Planning of a ship hull maintenance robot based on Glasius Bio-inspired Neural Network. *Expert Syst. Appl.* **2022**, *187*, 115940. [\[CrossRef\]](#)
30. Shu, Y.; Zhu, Y.; Xu, F.; Gan, L.; Lee, P.T.-W.; Yin, J.; Chen, J. Path planning for ships assisted by the icebreaker in ice-covered waters in the Northern Sea Route based on optimal control. *Ocean Eng.* **2023**, *267*, 113182. [\[CrossRef\]](#)
31. Xiaofei, Y.; Yilun, S.; Wei, L.; Hui, Y.; Weibo, Z.; Zhengrong, X. Global path planning algorithm based on double DQN for multi-tasks amphibious unmanned surface vehicle. *Ocean Eng.* **2022**, *266*, 112809. [\[CrossRef\]](#)
32. Chen, X.; Liu, S.; Zhao, J.; Wu, H.; Xian, J.; Montewka, J. Autonomous port management based AGV path planning and optimization via an ensemble reinforcement learning framework. *Ocean Coast. Manag.* **2024**, *251*, 107087. [\[CrossRef\]](#)
33. Hu, Y.; Yao, Y.; Lee, W.S. A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs. *Knowl.-Based Syst.* **2020**, *204*, 106244. [\[CrossRef\]](#)
34. Akshya, J.; Priyadarsini, P.L.K. Graph-based path planning for intelligent UAVs in area coverage applications. *J. Intell. Fuzzy Syst.* **2020**, *39*, 8191–8203. [\[CrossRef\]](#)
35. Kim, K.; Park, J.; Kim, J. Optimized Area Partitioning for Cooperative Underwater Search Using Multiple Autonomous Underwater Vehicles. *Int. J. Control Autom. Syst.* **2025**, *23*, 392–404. [\[CrossRef\]](#)
36. Cai, C.; Chen, J.; Ayub, M.S.; Liu, F. A Task Allocation Method for Multi-AUV Search and Rescue with Possible Target Area. *J. Mar. Sci. Eng.* **2023**, *11*, 804. [\[CrossRef\]](#)
37. Ji, H.; Hu, H.; Peng, X. Multi-Underwater Gliders Coverage Path Planning Based on Ant Colony Optimization. *Electronics* **2022**, *11*, 3021. [\[CrossRef\]](#)
38. Mu, X.; Gao, W. Coverage path planning for multi-AUV considering ocean currents and sonar performance. *Front. Mar. Sci.* **2025**, *11*, 1483122. [\[CrossRef\]](#)
39. Zhang, Y.; Wang, Q.; Shen, Y.; Wang, T.; Dai, N.; He, B. Multi-AUV cooperative search method based on dynamic optimal coverage. *Ocean Eng.* **2023**, *288*, 116168. [\[CrossRef\]](#)
40. Zhang, M.; Cai, W.; Pang, L. Predator-Prey Reward Based Q-Learning Coverage Path Planning for Mobile Robot. *IEEE Access* **2023**, *11*, 29673–29683. [\[CrossRef\]](#)
41. Jensen-Nau, K.R.; Hermans, T.; Leang, K.K. Near-Optimal Area-Coverage Path Planning of Energy-Constrained Aerial Robots with Application in Autonomous Environmental Monitoring. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 1453–1468. [\[CrossRef\]](#)
42. Kapoutsis, A.C.; Chatzichristofis, S.A.; Kosmatopoulos, E.B. DARP: Divide Areas Algorithm for Optimal Multi-Robot Coverage Path Planning. *J. Intell. Robot. Syst.* **2017**, *86*, 663–680. [\[CrossRef\]](#)
43. Zhang, B.; Si, X.; Cai, Y.; Gu, X.; Cao, Y.; Xi, X. Research on Path Planning Method for Weeding Robots in Paddy Fields Based on DARP Algorithm. *Trans. Chin. Soc. Agric. Mach.* **2025**, *56*, 335–343.
44. Apostolidis, S.D.; Kapoutsis, P.C.; Kapoutsis, A.C.; Kosmatopoulos, E.B. Cooperative multi-UAV coverage mission planning platform for remote sensing applications. *Auton. Robot.* **2022**, *46*, 373–400. [\[CrossRef\]](#)
45. Cai, C.; Chen, J.; Yan, Q.; Liu, F. A Multi-Robot Coverage Path Planning Method for Maritime Search and Rescue Using Multiple AUVs. *Remote Sens.* **2023**, *15*, 93. [\[CrossRef\]](#)
46. Ma, Y.; Liu, Y.; Bai, X.; Guo, Y.; Yang, Z.; Wang, L.; Zhang, L. DivideMerge: A multi-vessel optimization approach for cooperative operation and maintenance scheduling in offshore wind farm. *Renew. Energy* **2024**, *229*, 120758. [\[CrossRef\]](#)
47. Tang, Z.; Wang, L.; Wang, Y.; He, H.; Li, B. Distributed-integrated model predictive control for cooperative operation with multi-vessel systems. *J. Mar. Sci. Technol.* **2022**, *27*, 1281–1301. [\[CrossRef\]](#)
48. Natural Earth. Free Vector and Raster Map Data at 1:10m, 1:50m, and 1:110m Scales. Available online: <https://www.naturalearthdata.com/> (accessed on 7 November 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.