Transportation Research Part C 183 (2026) 105451

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Data-driven optimization for maritime logistics: integrating transport network mining with ship fleet routing

Yimeng Zhang [a,b,c], Liang Huang [d,*], Jiaci Wang [d,e], He Lin [d], Shuyang Zhu [a], Mi Gan [a,c], Xiaobo Liu [a,c], Ruixue Ai [f]

[a] School of Transportation & Logistics, Southwest Jiaotong University, Chengdu, China
[b] Tangshan Institute, Southwest Jiaotong University, Tangshan, China
[c] National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Southwest Jiaotong University, Chengdu, China
[d] State Key Laboratory of Maritime Technology and Safety, Wuhan University of Technology, Wuhan, China
[e] Sanya Science and Education Innovation Park, Wuhan University of Technology, Sanya, China
[f] Akershus, University of Oslo, Oslo, Norway

ARTICLE INFO

Keywords:
Maritime logistics
Network construction
Data-driven optimization
AIS data
Ship fleet routing

ABSTRACT

We introduce a data-driven approach for optimizing maritime logistics by integrating the construction of maritime transport networks with the routing of the ship fleet. Utilizing data mining techniques, this approach identifies crucial nodes and routes from Automatic Identification System (AIS) data and builds a directed weighted transport network. Based on the obtained transport network, we then optimize ships' routes using Mixed Integer Programming and Adaptive Large Neighborhood Search. This data-driven method provides a complete solution that improves maritime logistics from data mining to route optimization and enhances the operational autonomy of both autonomous and traditional ships. The results using real-world AIS data illustrate how data mining can be leveraged to develop a detailed transport network that significantly enhances fleet routing optimization. We evaluate our approach against two benchmarks in the literature and demonstrate that it enhances identification accuracy by over 14 %. Furthermore, through numerical analyses under various scenarios, such as route disruptions and varying levels of port congestion, our routing approach proves capable of managing large-scale operations and adapting to transport time variations. Compared to disruptions on routes, severe port congestion notably increases operational costs as it extends loading and unloading times and causes higher delay penalties.

1. Introduction

Shipping plays a crucial role in the transport of cargo and products around the world and ships deliver over 80 % of world trade (Sirimanne et al., 2022). To remain competitive, shipping companies are actively improving fleet efficiency to minimize transport costs and time.

Maritime transport lacks a predefined physical network similar to the urban road network. While traditional routes rely on experience, international regulations, and general knowledge (Vettor and Soares, 2016), actual routes can differ from official routes

* Corresponding author.
E-mail addresses: yimengzhang@swjtu.edu.cn (Y. Zhang), leung.huang@whut.edu.cn (L. Huang), jiaci.wang1205@gmail.com (J. Wang), lh960930@whut.edu.cn (H. Lin), shuyangzhu150@gmail.com (S. Zhu), migan@swjtu.edu.cn (M. Gan), xiaobo.liu@swjtu.edu.cn (X. Liu), ruixue.ai@outlook.com (R. Ai).

https://doi.org/10.1016/j.trc.2025.105451
Received 28 August 2024; Received in revised form 7 September 2025; Accepted 7 November 2025

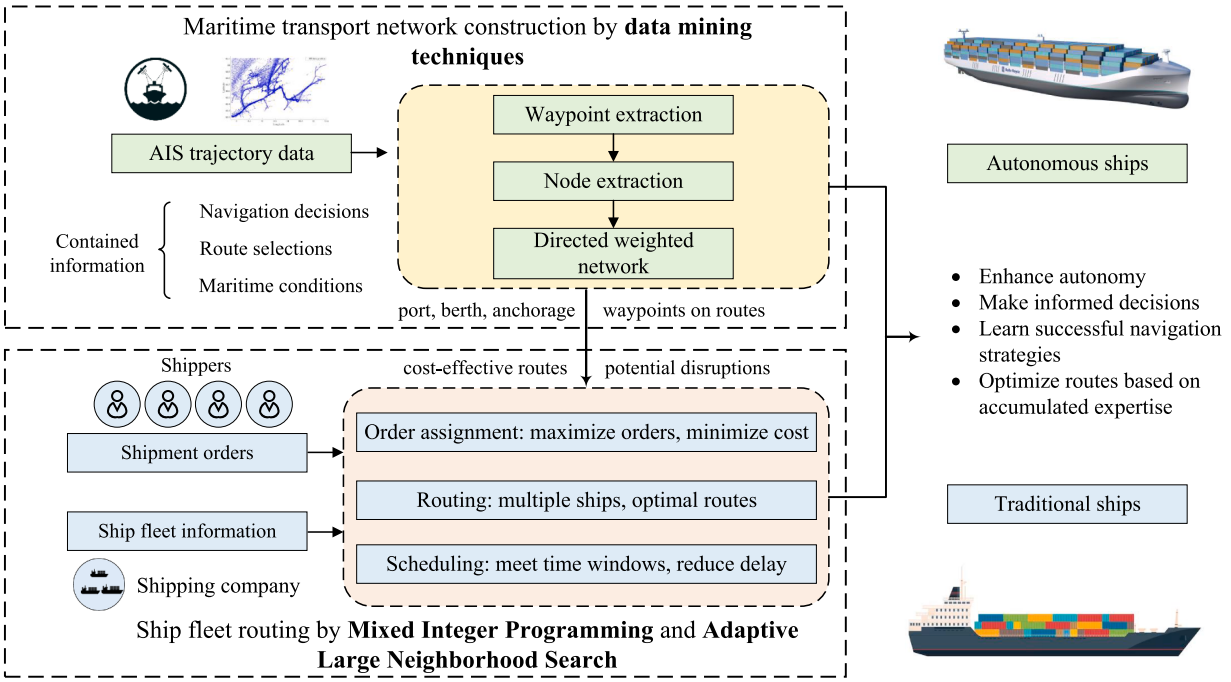


Fig. 1. The proposed approach for data-driven ship fleet routing.

due to complex and dynamic nature of maritime traffic, seasonal variations, and economic activities (Cullinane and Haralambides, 2021; Cong et al., 2024). In inland waterways and river deltas, route data is often incomplete or imprecise. Therefore, mining the maritime transport network and conducting routing based on accurate information of the actual routes is crucial.

In recent years, the integration of vast datasets such as Automatic Identification System (AIS) with advanced analytics has driven research on data-driven ship route optimization (Liu et al., 2023), which integrates both data mining and routing through large-scale data analysis (Gkerekos and Lazakis, 2020). This approach generates optimal routes by analyzing multiple factors, including weather, ocean currents, fuel consumption, and cargo volume, combining data mining and path planning techniques (Yang et al., 2019; Chu et al., 2023; Lai et al., 2022).

Data mining enables the extraction of actionable insights from massive data to construct a robust network (Liu et al., 2023). Ship trajectory data contains extensive information on habitual ship routes, including valuable insights from previous captains regarding the navigation characteristics of different waters (Cai et al., 2021). By integrating historical data on navigation decisions, route selections, and maritime condition handling, along with mining the experiential navigation route distributions, spatiotemporal activity patterns, and behavioral modes of ships in maritime spaces, the constructed network reflects the best practices of navigation in a specific region. Fleet operators can leverage this knowledge to replicate successful strategies, make informed decisions under similar circumstances, and optimize ship routes based on proven expertise. The data-driven optimization approach not only significantly enhances the safety, efficiency, and cost-effectiveness of operations but also highlights the crucial role of shipping companies in leveraging data-driven technologies to improve business performance.

In this paper, we propose a data-driven optimization approach for ship fleet routing. This approach constructs the transport network by a series of data mining techniques and optimizes shipping routes using Mixed Integer Programming (MIP) and Adaptive Large Neighborhood Search (ALNS), as shown in Fig. 1. This study addresses two key research questions: 1) How can a maritime transport network be derived from raw data? and 2) How can the routes of a fleet of ships be optimized using the maritime transport network? For the first question, the study tackles challenges such as data noise, uneven density distribution, and scalability by employing data mining techniques to develop a directed, weighted network. Utilizing the developed transport network, this study optimizes ship fleet routing and scheduling while tackling the complexities of maritime transport operations. It efficiently manages multiple shipment orders with varying time windows and mitigates disruptions to prevent delays, thereby addressing the second research question. The proposed approach enhances smart maritime logistics by automating the entire process from raw data mining to ship fleet routing, reducing the need for manual intervention in decision-making. By integrating these advanced data mining and optimization techniques, the approach optimizes routes based on accumulated expertise and transforms the way the ship fleet is managed, leading to improved efficiency and performance for both autonomous and traditional ships.

The rest of this paper is structured as follows: Section 2 provides a brief literature review. Section 3 describes the studied data-driven routing optimization problem for the ship fleet. In Section 4, we provide a detailed description of our approach, including the data mining techniques used to extract information from AIS data, and the MIP model and ALNS algorithm used to optimize ship routing. Section 5 provides the experiment results obtained from applying our approach to real-world shipping scenarios. Section 6

concludes this study, discusses the innovations of our approach and its limitations, and proposes possible future research directions in this area.

2. Literature review

We begin with a literature review on transport network mining in Section 2.1, followed by an examination of research on ship fleet routing in Section 2.2. The section concludes with a summary in Section 2.3, which identifies existing research gaps and contributions of this study.

2.1. Literature review on maritime transport network mining

In general, the maritime transport network and route extraction methods include the statistical method, grid-based method, and vector-based method (Xiao et al., 2019; Wang et al., 2020; Rong et al., 2022b; Liu et al., 2023).

The statistical method integrates digital image processing, clustering algorithms, and GIS for route feature extraction (Rong et al., 2022b). Yan et al. (2020) transform ship trajectory data into a ship trip semantic object (STSO), integrating STSO into the topological structure, leveraging graph theory for route extraction. Filipiak et al. (2020) combine a spatial index with a genetic algorithm to construct a directed graph for waypoint information extraction, mirroring real sea routes. Lee et al. (2020) fuse image processing technology with kernel density estimate (KDE) to accomplish route extraction, albeit requiring manual intervention. However, the statistical method merely offers basic analytics of traffic patterns, requiring collaboration with other technologies for advanced functions like anomaly detection (Xiao et al., 2019).

A grid-based method constructs the original traffic data within a set of index grids (Xiao et al., 2019). Both studies by Bomberger et al. (2007) and Dobrkovic et al. (2015) apply grid partitioning technology to analyze ship activity distribution and track information density within these grids. On the other hand, Wang et al. (2020) and Lu et al. (2020) adopt different approaches. Wang et al. (2020) utilize parallel grid merging with Quad-Trees and Delaunay triangulation for route boundary extraction, yet anomalous data hampers accuracy. To address this, Lu et al. (2020) adopt KDE to delineate route edges and mitigate the influence of outliers. However, while effective for small-area surveillance, grid-based method imposes high computational costs with area expansion (Vettor and Soares, 2015; Pallotta et al., 2013a,b) and necessitates preselecting optimal grid cell size (Pallotta et al., 2013a,b). To address these challenges, vector-based methods have emerged.

A vector-based approach is employed to construct maritime traffic networks, treating ports as nodes and voyages between them as edges (Mou et al., 2018; Wang et al., 2019). In the initial step, waypoints are identified and clustered using algorithms including DBSCAN (Kontopoulos et al., 2021; Lu et al., 2020), incremental DBSCAN (Pallotta et al., 2013b), K-means (Edelkamp and Schrödl, 2003; Filipiak et al., 2020), Ordering Points to Identify the Clustering Structure (OPTICS) (Yan et al., 2020), Genetic Algorithm (Dobrkovic et al., 2018; Filipiak et al., 2020), etc. After waypoint identification, trajectories connecting nodes are extracted to construct a network. Network structure, topology, and connectivity can be characterized by metrics including density, centrality indices, clustering coefficient, and average shortest path length (Kanrak et al., 2019). Filipiak et al. (2020) and Dobrkovic et al. (2018) generate networks by assigning trajectory points to nearest nodes or creating connections based on ship trajectory sequences, respectively, but may produce errors with missing data. Huang et al. (2023) cluster ship trajectories considering spatial distance and Course Over Ground differences, but effectiveness is limited by multiple threshold settings. Jin et al. (2017) and Yitao et al. (2020) use adaptive density incremental clustering and vessel dynamics for trajectory extraction and network construction, but scalability is a challenge with large datasets. Alternatively, Kaluza et al. (2010) construct a navigation network from ship journeys. Similarly, Wen et al. (2020) utilize the DBSCAN algorithm to identify key regions and connects them based on cluster similarity, thereby generating routes. Meanwhile, Murray and Perera (2022) harness machine learning to discern commonalities in trajectory segments.

While both Kontopoulos et al. (2021) and this study use the DBSCAN algorithm, our study differs in terms of objective, approach, and output. The objective of Kontopoulos et al. (2021)'s study is to detect anomalies, while this study aims to construct a maritime transport network and optimize ship fleet routing. They extract maritime traffic patterns by clustering sub-trajectories with a DBSCAN algorithm. In contrast, we first extract semantically meaningful feature points and then cluster them to detect spatially dense regions that reflect consistent navigational behavior, which are treated as functional nodes in the network. The outputs of their paper are convex hulls representing typical vessel movement patterns, while the output of this study is a weighted directional transport network and optimized routes of ships.

Arguedas et al. (2017) introduce a hierarchical maritime traffic network model comprising external and internal layers. While our study adopts a conceptual framework similar to Arguedas et al. (2017), we introduce substantial methodological enhancements. Specifically, our improved POI detection integrates Isolation Forest anomaly detection, leading to significantly higher accuracy in identifying stable staying points. Additionally, our refined turning-point detection algorithm outperforms the Douglas-Peucker (DP) method used in Arguedas et al. (2017), reducing false detections and providing more reliable breakpoints in vessel trajectories. In edge construction, we adopt precise trajectory segmentation methods and explicitly quantify maritime traffic intensity using weighted adjacency matrices, enabling direct support for downstream tasks like optimized fleet routing. These advancements offer a markedly more precise and practically relevant maritime transport network compared to previous studies.

Rong et al. (2022a) propose a framework for probabilistic maritime traffic prediction along a specific route. Compared with theirs, our work focuses on the construction of maritime transport network and differs significantly in node identification, edge construction, and network applications. Rong et al. (2022a) employ the OPTICS to cluster stationary points and identify departure/arrival zones, while also treating trajectory overlaps as route junctions. In contrast, we develop dedicated methods to extract three key types of

navigational points-turning points, staying points, and entry/exit points. To account for their differing spatial characteristics, we apply tailored DBSCAN clustering strategies and introduce the silhouette coefficient to evaluate clustering quality. Their network models route connections implicitly via trajectory aggregation and Gaussian process regression. In contrast, we explicitly construct edges from segmented trajectories and corresponding traffic statistics, generating a weighted edge matrix that directly reflects maritime traffic intensity. The application in Rong et al. (2022a) is primarily designed for behavior modeling tasks (e.g., destination prediction). Our structured network, by contrast, prioritizes spatial-topological accuracy and enables operational decision-making, particularly in fleet route planning and voyage optimization, through a clearly defined and interpretable edge weight matrix.

In summary, some achievements have been made in the construction of maritime transport networks, but these methods are easily interfered with by two factors during the construction process: (a) Non-feature trajectory points: For example, when extracting route turning points, these methods may easily extract turning points from non-turning behaviors such as ship collision-avoidance turns, staying turns, and zigzag maneuver; (b) Uneven density distribution of trajectory points, such as sparse trajectories in certain areas, may hinder the extraction of complete network information, thus affecting the accuracy of network construction. Aiming at the above problems, this paper introduces a novel approach to construct a maritime transport network.

2.2. Literature review on ship fleet routing

We classify ship route optimization approaches based on their data sources and the use of data mining techniques, as follows: (a) Approaches independent of AIS data and data mining techniques; (b) Approaches utilizing AIS data without employing data mining; and (c) Approaches leveraging AIS data and employing data mining techniques.

The following articles, rather than processing AIS data, utilize information from public databases, port records, or even hypothetical datasets. Kisialiou et al. (2018) and Kisialiou et al. (2019) integrate heuristic algorithms with simulation techniques using data from Statoil, the Norwegian Meteorological Institute, and a Norway operator Equinor. An ALNS algorithm for inland waterway transport is proposed by Zhang et al. (2020), leveraging data from Contargo's website and an online ship monitoring system. More recently, Utomo et al. (2023) optimize shipping routes from Indonesia to the Asia-Pacific region using data from 12 regional ports. In addition, a multi-objective programming framework for ship scheduling is introduced by Wen et al. (2024), utilizing data from six container terminals at the Port of Shanghai, emphasizing environmental concerns.

Some studies use and process AIS data to optimize fleet routes without utilizing data mining techniques. Guo et al. (2020) leverage Deep Deterministic Policy Gradient (DDPG) algorithm and Deep Reinforcement Learning (DRL) method to optimize route planning through interaction with environments constructed using AIS data. Jeong et al. (2019) propose a risk contour map and multi-criteria model for route planning, validated through numerical simulations and real AIS data analysis. Chian Tan et al. (2021) and Naus (2020) contribute to the field by utilizing data processing and analysis techniques. Chian Tan et al. (2021) select routes for autonomous ships based on risk estimation and nearest neighbor search, while Naus (2020) propose a grid reference system for efficient route planning, considering vessel density, speed, and course in AIS data.

Given that route planning must continually adapt to complex and changing conditions, analyzing historical AIS data to identify optimal routes becomes crucial (Chian Tan et al., 2021; Han and Yang, 2020; Li and Yang, 2023). Han and Yang (2020) construct a ship historical route network using DBSCAN and central line extraction, then optimize paths by combining Dijkstra's algorithm for finding the shortest path within a maritime network. Li and Yang (2023) introduce a machine learning approach that utilizes AIS data for unsupervised route planning of autonomous surface ships (MASS), improving maritime safety through feature measurement, pattern extraction, and route planning algorithms.

In summary, existing methodologies often rely on presupposed or predefined maritime networks, while real-world routes are dynamic and require data-driven derivation from actual data. Moreover, large-scale optimization and adaptability to disruptions remain challenges, highlighting the need for efficient solutions that handle scale and change effectively.

2.3. Summary

Table 1 summarizes studies in both data mining and ship routing optimization in maritime transport. Among the literature in the data mining section, a few studies (Filipiak et al., 2020; Dobrkovic et al., 2015; Liu et al., 2023; Pallotta et al., 2013b; Rong et al., 2022b) employ the same waypoint extraction technique as this study. The majority of the literature in this section focus solely on data mining, failing to leverage the extracted routes to generate transportation networks, only a few articles (Filipiak et al., 2020; Liu et al., 2023; Rong et al., 2022b; Han and Yang, 2020) succeed in constructing maritime transportation networks. Although (Tierney et al., 2019; Kisialiou et al., 2019) acknowledge the presence of disruptions, their analysis remains relatively superficial. Similarly, weighted networks are rarely considered in the existing literature, yet both aspects are vital components of our research.

Despite the high attractiveness of AIS data, few studies have applied it to ship route planning (Li and Yang, 2023). The data-driven methods offer a framework for detecting behavioral patterns from massive ship trajectories and generating ship routes. This not only aids in understanding ship trajectory data but also supports operators in making informed decisions (Li et al., 2023). The integration of historical AIS data and advanced algorithms for route optimization presents a significant opportunity to automate and improve navigation planning for ships (Naus, 2020). Therefore, a data-driven ship routing approach is needed to effectively utilize large volumes of maritime data for ship fleet routing.

This study leverages AIS data and focuses on integrating the construction of maritime transport networks with the optimization of ship fleet routing. Both Li and Yang (2023) and our work employ machine learning methodologies, yet our approach uniquely utilizes MIP and ALNS algorithms to refine routing optimization within the established transportation network. While Li and Yang

Table 1
Literature review on data mining and ship routing optimization.

Literature	Method	Data type	Mining object	Network	Routing	Weighted-network	Disruption
Data mining							
Filipiak et al. (2020)	CUSUM, GA	AIS	waypoint	✓			
Lee et al. (2020)	KDE DBSCAN,	AIS	boundary, centerline				
Dobrkovic et al. (2015)	GA, ACO	AIS	waypoint				
Wang et al. (2020)	PGM, FA	AIS	trajectory				
Pallotta et al. (2013b)	TREAD	AIS	waypoint				
Liu et al. (2023)	DNSCAN	AIS	waypoint	✓			
Wen et al. (2020)	DBSCAN	AIS	pattern				
Murray and Perera (2022)	GMM, LDA, GA	AIS	trajectory				
Huang et al. (2023)	MD-DBSCAN	AIS	route				
Rong et al. (2022b)	AWEM, KDE-T, ATNGM	AIS	waypoint	✓			
Ship routing optimization							
Tierney et al. (2019)	MIP, CC, Sim	ship service data supply base data,			✓		
Kisialiou et al. (2019)	CC, Sim	ship data			✓		
Wen et al. (2024)	MOP	ship arrival data			✓		
Jeong et al. (2019)	MCM	AIS			✓		
Han and Yang (2020)	DBSCAN	AIS		✓	✓		
Chian Tan et al. (2021)	1NN	AIS			✓		
Naus (2020)	GRS	AIS			✓		
Guo et al. (2020)	DDPG, DRL	AIS			✓		
Li and Yang (2023)	AADTW, SCAF, DO	AIS	pattern		✓		
This article	MIP, ALNS	AIS	waypoint	✓	✓	✓	✓

CUSUM: Cumulative Sum; GA: Genetic Algorithm; MIP: Mixed Integer Programming; CC: Chance Constraints; PGM: Parallel Grid Merging; FA: Filtering Algorithm; ALNS: Adaptive Large Neighborhood Search; DDPG: Deep Deterministic Policy Gradient; DRL: Deep Reinforcement Learning; DBSCAN: Density-Based Spatial Clustering of Applications with Noise; Sim: Simulation; MOP: Multi-objective Programming; MCM: Multi-Criteria Model; 1NN: Nearest Neighbor Search; AWEM: Adaptive Waypoint Extraction Model; GRS: Grid Reference System; KDE: Kernel Density Estimation; ATNGM: Automatic Traffic Network Generation Model; AADTW: Automatic and Adaptive Dynamic Time Warping; SCAF: Spectral Clustering with Affinity Feature; DO: Dynamic Optimization; TREAD: Traffic Route Extraction and Anomaly Detection; ACO: Ant-Colony Optimization; GMM: Gaussian Mixture Model; LDA: Linear Discriminant Analysis.

(2023) primarily emphasize the extraction of ship movement patterns from historical trajectory data, our study shifts the focus to the mining of waypoints, a more granular analysis. Furthermore, we quantify the enhancements in fleet routing efficiency and resilience to disruptions, providing managerial insights that are not directly addressed in Li and Yang (2023). The contributions of this study include: (a) we present a framework that transforms raw AIS data into optimized ship fleet routing decisions by integrating data mining techniques and advanced routing algorithms, thereby enhancing the efficiency of maritime logistics; (b) we achieve significant improvements in maritime transport network construction compared with benchmarks in the literature; (c) we formulate the mathematical model for ship fleet routing and propose heuristic algorithm which can handle large-scale instances; (d) we deliver valuable insights for managing disruptions in maritime transport.

3. Problem description

Data-driven ship fleet routing involves leveraging data mining techniques to extract valuable information from a transport network, which is then utilized by an optimization-based routing model. The mined transport network provides a detailed map of available routes, including nodes, waypoints, and directions. Additionally, it assigns weights to these routes, reflecting the number of trips made on each route, which serves as an indicator of potential congestion. The optimization model uses this information to identify the most efficient and disruption-free routes for the ship fleet, avoiding congested or high-traffic areas. This integration aims to enhance the overall efficiency of the fleet's operations by combining the strengths of data mining in uncovering hidden patterns within the network and the power of optimization techniques in determining the best possible routing decisions based on these patterns.

The used notation is presented in Table 2. The maritime transport network is generally expressed as a graph $G = (N, A)$, which includes three basic elements: nodes, edges, and weights (Xiao et al., 2019). Nodes $(N, i \in N)$ represent key waypoints on maritime routes such as ports, docks, anchorages, and turning points. Edges $(A, (i, j) \in A, i \neq j)$ are connections between nodes. Weights associated with these edges reflect the frequency of maritime routes connecting the nodes. Data mining techniques are needed to identify basic elements in the transport network. However, accurately extracting these elements is difficult due to the intricate and often high-noise nature of maritime data. First, it faces the challenge of data quality issues and potential errors in network edge connections. Additionally, the extraction of turning points may mistakenly include non-feature behaviors such as collision-avoidance maneuvers. Furthermore, the uneven density distribution of trajectory points hinders the extraction of complete network information.

After the maritime transport network is constructed, ship fleet's routes can be optimized. Ship routing is a complex optimization problem that involves various constraints and objectives. Some of the key factors that affect ship routing include:

Table 2
Notation.

Sets:	
R	Set of orders, $r \in R$. Each order r contains pickup and delivery ports, time windows for pickup and delivery, and number of containers.
N	Set of ports, $i, j \in N$. $P, D \subseteq N$, sets of pickup and delivery ports.
O, O'	Sets of origins and destinations of ships, $o \in O, o' \in O'$.
K	Set of ships indexed by k and l.
A	Set of edges. For $i, j \in N$, the route from i to j is denoted by $(i, j) \in A$. $A_p/A_d \subseteq A$ represents the set of pickup/delivery edges. For $(i, j) \in A_p$, $i \in P$.
Parameters:	
u_k	Ship k's capacity.
q_r	Number of containers (TEU) of order r.
τ_{ij}^k	Ship k's transport time (h) on route (i, j).
$p(r)/d(r)$	Pickup/delivery port of order r.
$[a_{p(r)}, b_{p(r)}]$	Order r's pickup time window.
$[a_{d(r)}, b_{d(r)}]$	Order r's delivery time window.
t''^k_i	Port i's service time (h) for ship k.
v_k	Ship k's speed (km/h).
d_{ij}	Distance (km) between ports i and j.
c_k^n	c_k^1 is the unit transport cost related to time, (CNY/TEU/h); $c_k^{1'}$ is the unit transport cost related to distance (CNY/TEU/km); c_k^2 is the unit loading/unloading cost (CNY/TEU); c_k^3 is the unit storage cost (CNY/TEU/h); c_k^4 is the unit waiting cost (CNY/TEU/h).
c_r^{delay}	Unit delay penalty (CNY/TEU/h) for order r.
M	A large positive number.
Variables:	
x_{ij}^k	Binary variable; equals 1 if ship k traverses the route (i, j), and 0 if not.
y_{ij}^{kr}	Binary variable; equals 1 if order r transported by ship k uses route (i, j), and 0 if not.
z_{ij}^k	Binary variable; equals 1 if port i precedes port j in the route of the ship k, and 0 if not.
$t_i^{kr}/t'^{kr}_i/\bar{t}_i^{kr}$	The time of arrival/service initiation/service completion for order r handled by ship k at port i.
$t_i^k/t'^k_i/\bar{t}_i^k$	The time of arrival/final service initiation/departure for ship k at port i.
t_{ki}^{wait}	The waiting time of ship k at port i.
t_r^{delay}	The delivery delay of order r at the destination port.

1. Ship capacities: Each ship has a limited capacity for cargo, which needs to be considered when scheduling the loading and unloading of cargo at different ports.
2. Cargo demands: The cargo demand at each port may vary, and the ship must be scheduled in a way that meets these demands while satisfies time windows of orders.
3. Port constraints: Ports may have constraints on the number and size of ships that can be handled at any given time.
4. Time constraints: The arrival and departure times of ships at different ports may be subject to operation time

A solution for ship fleet routing consists of a collection of $|K|$ routes that fulfill orders. The number of containers that ship k carries at any time cannot surpass its capacity u_k. Each order r must be handled by the same ship k, implying that $d(r)$ must be served after $p(r)$. Order r must be picked up within the time window $[a_{p(r)}, b_{p(r)}]$ and delivered within $[a_{d(r)}, b_{d(r)}]$, though deliveries past $b_{d(r)}$ are allowed with a delay penalty. Ship k is permitted to wait for containers at port i, and if the ship has not yet arrived, containers in order r may be stored at port i.

4. Data-driven route planning for the ship fleet

Section 4.1 presents the methodology for constructing transport networks using data mining technologies. It then introduces an optimization model for ship fleet routing in Section 4.2.

4.1. Transport network construction

Most vessels follow similar navigation patterns within constrained navigable waters. Analyzing these collective movement behaviors allows for the derivation of a structured maritime transportation network. Fig. 2 illustrates the overall framework of the study, which consists of four main steps: AIS data preprocessing, waypoint detection, node extraction, and network construction.

To ensure high-quality input, raw AIS data are first cleaned by removing noisy and drifting points, as well as anomalies in vessel speed and heading (Huang et al., 2020). Trajectories are then grouped by MMSI to reconstruct individual vessel paths. Trajectory segmentation is applied when the time interval between two consecutive points exceeds 30 minutes. This threshold is widely adopted in AIS trajectory analysis, effectively distinguishing separate voyages without excessive fragmentation. Finally, interpolation and smoothing techniques are applied to mitigate irregular sampling and improve trajectory continuity. Representative waypoints are then extracted from the processed trajectories to capture characteristic maneuvering behaviors. These waypoints are spatially clustered, and the centroids of the clusters are defined as nodes within the maritime network. Edges between nodes are established based on

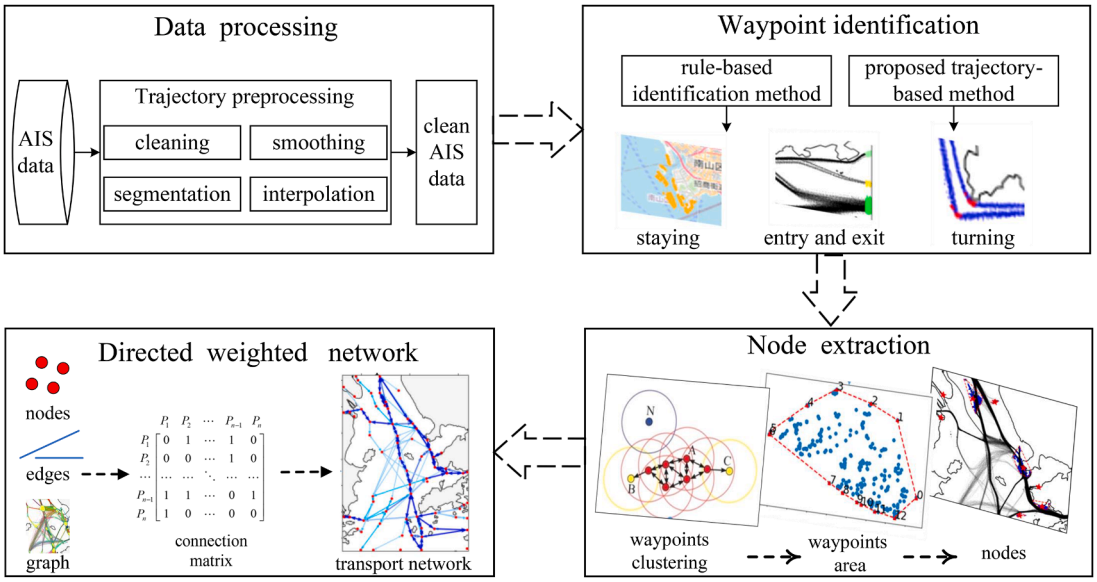


Fig. 2. Network construction flowchart.

actual voyage flows and directional movement patterns, with weights assigned to reflect traffic intensity. The result is a directed, weighted network that represents major shipping routes and captures their usage characteristics.

This study introduces several methodological enhancements over prior approaches. An enhanced algorithm based on cumulative turning features is developed for waypoint identification, enabling more accurate detection of vessel turning behaviors. A refined geometric extraction method is applied in edge generation to improve the representation of network edge geometry. Furthermore, both DBSCAN and OPTICS are employed for clustering, and their performance is systematically compared through experiments.

4.1.1. Waypoint extraction based on data mining

In this study, waypoints are classified into three categories according to ship behavioral characteristics: staying points refer to locations where ships are berthed or anchored; boundary-crossing points mark the positions where ships enter or exit the main channel within the study area; and route-turning points indicate locations where ships undergo significant directional changes. The following subsections describe the identification methods for each type.

Recognition method of staying points

Staying points are identified from ship trajectories by detecting stationary segments under distance, time, and point-count constraints. To eliminate outliers and extract highly aggregated stationary points, the Isolation Forest algorithm (Liu et al., 2008) is applied. Their spatial distribution and degree of aggregation are then quantified using three indicators: trajectory point repetition rate, average distance between adjacent points, and maximum inter-point distance. Finally, the K-nearest neighbor algorithm (Larose and Larose, 2014) is employed to automatically classify staying points into anchoring and berthing categories.

Recognition method of boundary-crossing points

Boundary-crossing points are identified by examining trajectory points near the study area boundaries that exhibit crossing behavior. The area is delineated by a set of boundary lines $B = \{b_i, b_{i+1}, \ldots, b_{i+n}\}$. For each boundary line b_i, trajectory points are filtered according to two criteria: (i) $dis(p, b_i) \leq 1$ km, which selects points in proximity to the boundary. The 1 km buffer, based on spatial analysis principles (Lee et al., 2022), extends the 500 m safety zone defined by UNCLOS to account for AIS noise and vessel maneuvers. (ii) $SOG_p > 1$ nm/h, a widely used threshold to identify underway vessels (Yoo et al., 2024; Duan et al., 2024), thereby excluding drifting, anchoring, or other low-speed points.

Points satisfying both criteria are marked as candidate boundary-crossing points. A geometric check is then applied: if the corresponding trajectory segment intersects the boundary line b_i, it is labeled as crossing; otherwise, it is excluded.

Recognition method of turning points

Traditional approaches for identifying turning points on maritime routes typically rely on course-change rules or trajectory shape features (Wen et al., 2020). While computationally efficient, these methods often fragment a complete turning behavior into isolated points and may misidentify non-turning behaviors as turning events. To address these limitations, we propose a filtering strategy based on cumulative turning features. The overall process is illustrated in Fig. 3, and the detailed procedures are described below.

1. Smoothing and interpolation. To mitigate oscillatory deviations caused by AIS positioning errors, trajectories are smoothed using the Savitzky-Golay filter (Phu et al., 2018). Cubic spline interpolation (McKinley and Levine, 1998) is then applied between staying points to obtain evenly sampled trajectories.

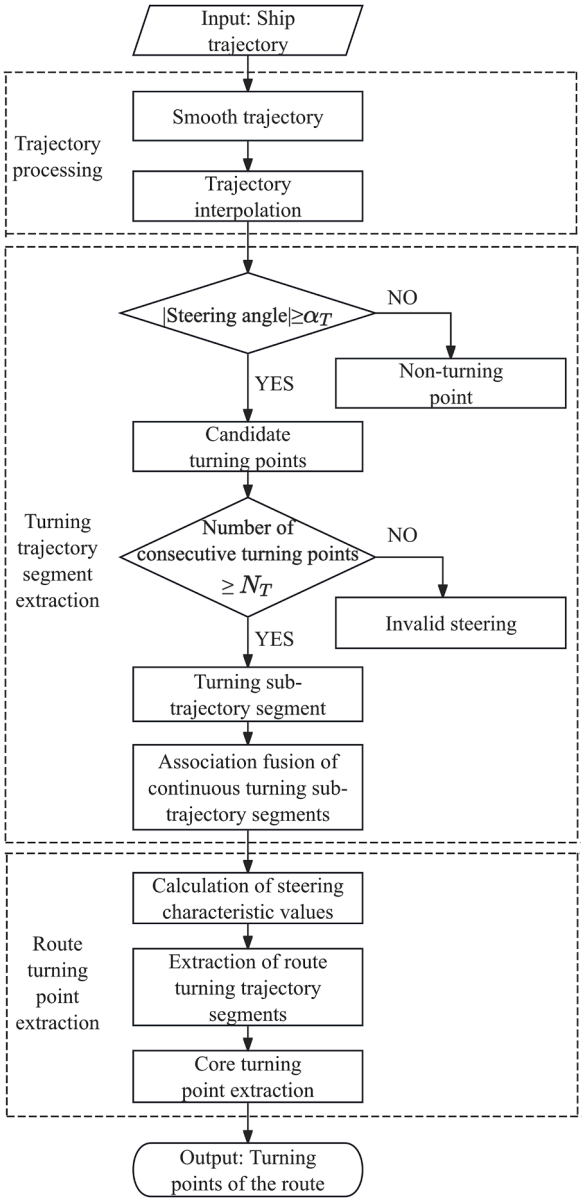


Fig. 3. Route turning point identification.

2. Extract turning sub-trajectory segments. For the interpolated trajectory, the turning angle $\alpha_{i-1,i}$ between consecutive points is calculated as the difference in geographic bearing (TCOG) between p_{i-1} and p_i, where positive and negative values denote right and left turns, respectively. If $|\alpha_{i-1,i}| > \alpha_T$ (a predefined threshold), p_i is marked as a turning point. Consecutive turning points are aggregated to form candidate turning sub-trajectories S; otherwise, points are discarded.
3. Merging of adjacent sub-trajectories. Candidate sub-trajectories are first detected (see Fig. 4(a)). For adjacent sub-trajectories S_i and S_{i+1}, the inter-segment distance $dis(S_i, S_{i+1})$ is computed (see Fig. 4(b)). If $dis(S_i, S_{i+1}) \leq D_s$, with D_s set to 1000 m in accordance with maneuvering guidelines and prior studies (Huang et al., 2021), the sub-trajectories are merged into a complete turning segment (Fig. 4(c)), thereby restoring continuous turning behaviors fragmented by data sparsity or noise.
4. Extraction of route turning points. This process consists of three steps: (i) For each complete turning segment, feature values are derived, including the total turning angle $\alpha_{tol} = \alpha_{(i,n)}$ (i starts from 1), average speed over ground (SOG), average turning rate (TR) during the turn, and turning consistency R_T (Eq. (1)).

$$R_T = max(\alpha_{rt}, \alpha_{lt}/(\alpha_{sum} \times 100\,\%) \tag{1}$$

where α_{rt} and α_{lt} denote the cumulative right- and left-turn angles (°), and α_{sum} is the total turning angle (°).

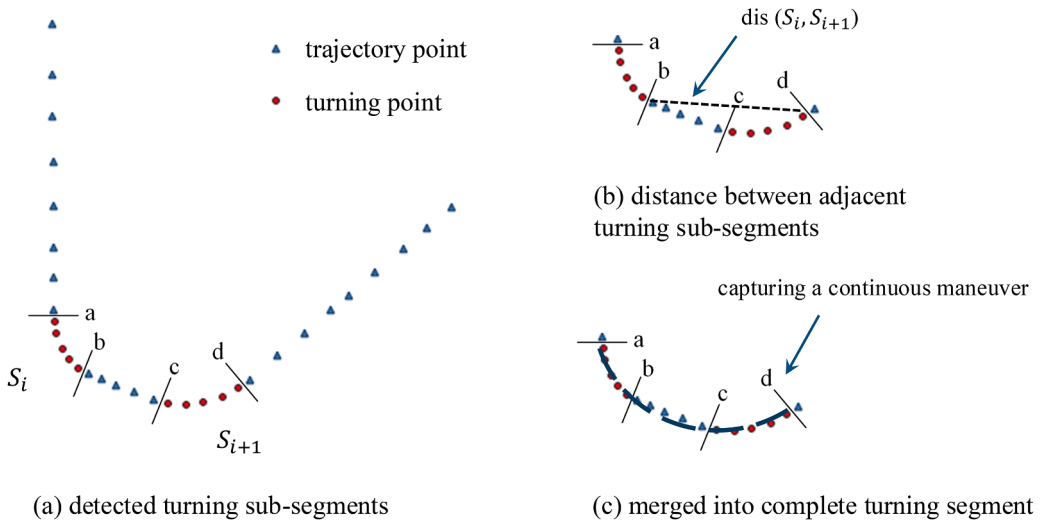


Fig. 4. Turning segment extraction and merging.

(ii) Multi-feature thresholds are applied to remove spurious points that do not satisfy turning constraints. Subsequently, the core turning points, which represent the most prominent turning features along a trajectory segment, are identified by defining a baseline l between the start p_1 and end p_n of each segment. For every intermediate point p_m, turning characteristics are then computed using Eq. (2).

$$\begin{cases} TR = \dfrac{\alpha_{m-1,m}}{\left(t_{p_m} - t_{p_{m-1}}\right)} \times 60 \\ TCR = \dfrac{TR_{p_m} - TR_{p_{m-1}}}{t_{p_m} - t_{p_{m-1}}} \times 60 \\ TD = \mathrm{dis}\left(p_m, l\right) \end{cases} \tag{2}$$

where TD denotes the perpendicular deviation from p_m to the baseline l (m), TR the turning rate (°/min), and TCR the rate of change of TR (°/min^2), reflecting angular acceleration.

(iii) To comprehensively quantify vessel turning behavior, a Turning Weight index (TW) is defined as $TW = 0.4 \times TD + 0.3 \times TR + 0.3 \times TCR$, with a higher weight assigned to TD to emphasize spatial deviation, while TR and TCR jointly capture maneuvering dynamics. For each turning segment, the three points with the highest TW values are selected as core turning points, ensuring representativeness and minimizing redundancy.

4.1.2. Maritime transport network construction

The maritime transport network is constructed by clustering trajectory waypoints into nodes and aggregating trajectory segments between nodes into sub-routes, from which a directed weighted network is established.

Recognition method of nodes in the network

DBSCAN (Kontopoulos et al., 2021; Lu et al., 2020) is applied to cluster the identified waypoints. For staying points, independent parameter settings distinguish berthing from anchoring clusters; for turning and boundary-crossing points, grid search experiments are used to determine optimal parameters for each dataset. Cluster boundaries are delineated using the convex hull algorithm (Barber et al., 1996), and cluster centroids are defined as representative nodes in the network.

Staying areas are derived from berthing and adjacent anchoring regions. As shown in Fig. 5, the two regions are delineated using the above method and merged into a unified polygon based on topological relationships. The centroid of this polygon is defined as the representative staying node in the network.

Recognition method of sub-routes in the network

Navigable sub-routes are identified by delineating polygonal regions with convex hulls around clustered trajectory points. These regions serve as spatial segmentation units for continuous trajectories. Trajectory points are assigned to regions using the ray casting algorithm (see Fig. 6), which determines spatial inclusion based on polygon intersections. An odd number of intersections (e.g., point a) indicates that the point is inside the region, while an even number (e.g., point b) signifies that it is outside. Continuous ship trajectories are thereby converted into ordered sequences of region identifiers (as shown in Fig. 7), representing ship flow paths through adjacent regions and capturing the topological structure of maritime movement.

Generation method of edges in the network

Once the navigable connectivity between node regions is identified (Fig. 8(a)), sub-routes are abstracted into topological edges. An edge-generation method is proposed that combines Alpha-Shapes with constrained Delaunay triangulation to capture the spatial characteristics of ship movement. Alpha-Shapes delineate route boundaries from outlier-filtered trajectory clusters (Fig. 8(b) and (c)).

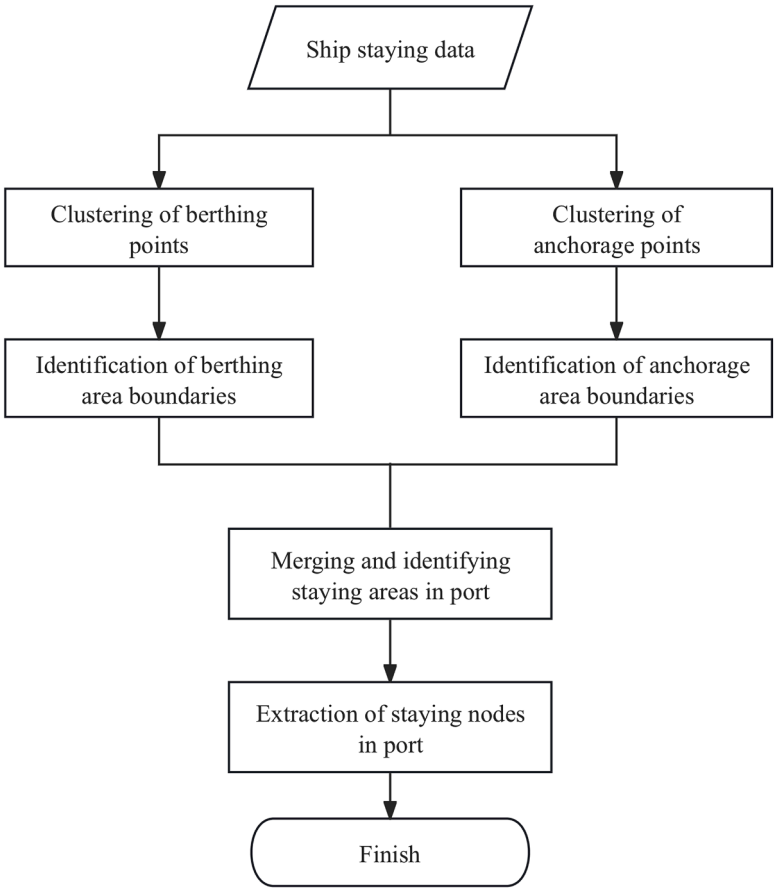


Fig. 5. Extraction process of staying nodes.

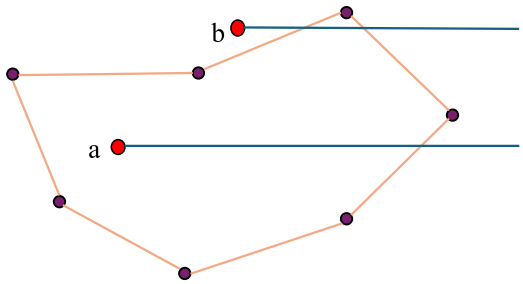


Fig. 6. The ray casting method.

Constrained Delaunay triangulation is applied within these boundaries to generate a triangular mesh (Fig. 8(d)). From this mesh, centerlines are extracted and taken as topological edges for the corresponding route segments (Fig. 8(e) and (f)). Repeating the procedure across all sub-routes yields a complete geometric-topological representation of the maritime route network.

An adjacency matrix is constructed to represent the maritime network as a weighted directed graph, where each entry indicates the existence and intensity of navigational connections between nodes. In the corresponding weighted adjacency matrix $(a_{ij})_{(N\times N)}$, each element a_{ij} denotes the number of observed trips from node i to node j, as defined in Eq. (3).

$$a_{ij} = \begin{cases} w_{ij} & \text{if } w_{ij} \geq n_f, \\ 0 & \text{if } w_{ij} < n_f. \end{cases} \tag{3}$$

where weight w_{ij} represents the number of navigational trips from node i to node j.

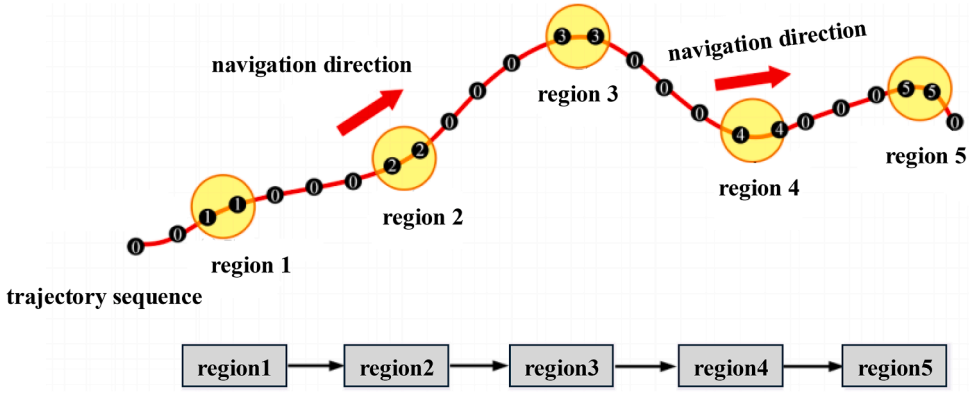
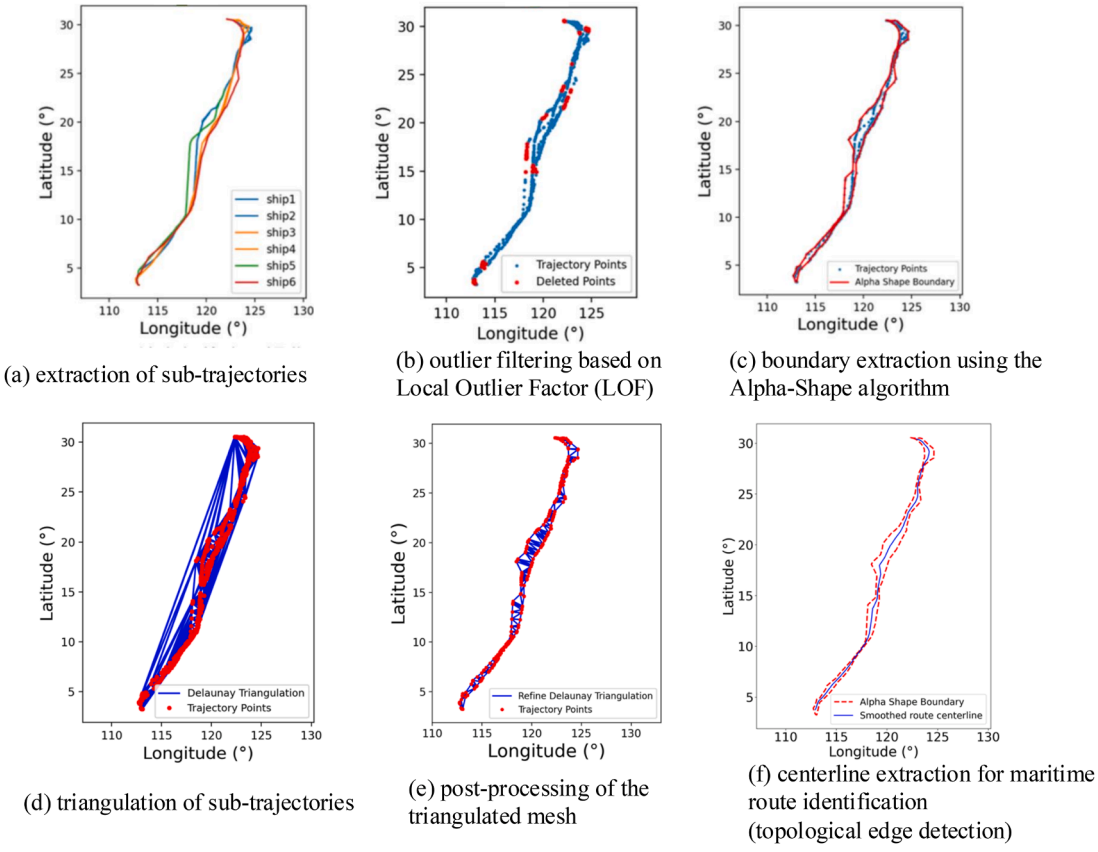


Fig. 7. Construction of region sequences.



(a) extraction of sub-trajectories

(b) outlier filtering based on Local Outlier Factor (LOF)

(c) boundary extraction using the Alpha-Shape algorithm

(d) triangulation of sub-trajectories

(e) post-processing of the triangulated mesh

(f) centerline extraction for maritime route identification (topological edge detection)

Fig. 8. Edge construction workflow.

4.2. Ship fleet routing

The maritime transport network constructed in Section 4.1 provides crucial data for optimizing ship fleet routing through its detailed mapping of nodes, edges, waypoints, and route weights. Nodes (N), representing key locations such as ports, allow for strategic planning of stops for loading and unloading. Edges (A), which connect these nodes, offer various route options with specific characteristics, including distance (d_{ij}) and typical travel time (τ_{ij}^{k}). Waypoints further refine these routes by indicating navigational points that ships must pass through, ensuring safe and efficient navigation. Weights (a_{ij}), representing frequency of trips, enable

the assessment and comparison of different routes. By leveraging this comprehensive network information, ship fleet operators can optimize routing to minimize costs, reduce travel time, enhance safety, and improve overall operational efficiency. To optimize routes of ships, this section begins with a detailed presentation of the MIP model in Section 4.2.1, followed by an introduction to the tailored ALNS algorithm in Section 4.2.2.

4.2.1. Mixed integer programming model

In this section, a mathematical model is introduced (the notation is in Table 2). The objective is minimizing the sum of transit cost, loading/unloading cost, storage cost, waiting cost, and delay penalty, as shown in Eq. (4).

$$\begin{aligned}\min\ F = &\sum_{k\in K}\sum_{(i,j)\in A}\sum_{r\in R}(c_k^1\tau_{ij}^k + c_k^{1'}d_{ij})q_r y_{ij}^{kr} + \\ &\sum_{k\in K}\sum_{(i,j)\in A_p}\sum_{r\in R}c_k^2 q_r y_{ij}^{kr} + \sum_{k\in K}\sum_{(i,j)\in A_d}\sum_{r\in R}c_k^2 q_r y_{ij}^{kr} \\ &\sum_{k\in K}\sum_{(i,j)\in A_p}\sum_{r\in R}c_k^3 q_r y_{ij}^{kr}(t'^{kr}_i - a_{p(r)}) + \\ &\sum_{k\in K}\sum_{i\in N}c_k^4 t_{ki}^{\text{wait}} + \sum_{r\in R}c_r^{\text{delay}} q_r t_r^{\text{delay}}\end{aligned} \tag{4}$$

Constraints (5) ensure that the number of served requests meets a predefined threshold, which is adjustable via the parameter ε. An order r is served when it departs from the origin $p(r)$ with vehicle k (i.e., $y_{p(r)j}^{kr} = 1$). Constraints (6) ensure that if each order is picked up, it must be delivered at their delivery port.

Constraints (7) and (8) ensure that a ship starts and finishes shipping at the designated departure and arrival depots, respectively.

$$\sum_{r\in R}\sum_{k\in K}\sum_{j\in N} y_{p(r)j}^{kr} \geqslant \varepsilon|R| \tag{5}$$

$$\sum_{i\in N} y_{id(r)}^{kr} \geqslant \sum_{j\in N} y_{p(r)j}^{kr} \quad \forall r\in R, \forall k\in K \tag{6}$$

$$\sum_{j\in N} x_{o(k)j}^{k} \leqslant 1 \quad \forall k\in K \tag{7}$$

$$\sum_{j\in N} x_{o(k)j}^{k} = \sum_{j\in N} x_{jo'(k)}^{k} \quad \forall k\in K \tag{8}$$

Constraints (9) and (10) ensure flow conservation for ships and orders, respectively. Constraints (11) guarantee that the ship needs to be traversing the associated route for an order to be transported by it.

$$\sum_{j\in N} x_{ij}^{k} - \sum_{j\in N} x_{ji}^{k} = 0 \quad \forall k\in K,\ \forall i\in N\setminus o(k), o'(k) \tag{9}$$

$$\sum_{j\in N} y_{ij}^{kr} - \sum_{j\in N} y_{ji}^{kr} = 0 \quad \forall k\in K,\ \forall r\in R,\ \forall i\in N, p(r), d(r) \tag{10}$$

$$y_{ij}^{kr} \leqslant x_{ij}^{k} \quad \forall (i,j)\in A,\ \forall k\in K,\ \forall r\in R \tag{11}$$

Constraints (12)–(14) serve to eliminate subtours. Constraints (15) define the capacity limitations.

$$x_{ij}^{k} \leqslant z_{ij}^{k} \quad \forall i,j\in N,\ \forall k\in K \tag{12}$$

$$z_{ij}^{k} + z_{ji}^{k} = 1 \quad \forall i,j\in N,\ \forall k\in K \tag{13}$$

$$z_{ij}^{k} + z_{jp}^{k} + z_{pi}^{k} \leqslant 2 \quad \forall i,j,p\in N,\ \forall k\in K \tag{14}$$

$$\sum_{r\in R} q_r y_{ij}^{kr} \leqslant u_k x_{ij}^{k} \quad \forall (i,j)\in A,\ \forall k\in K \tag{15}$$

Constraints (16) guarantee that the order's pickup time falls within the specified time window. Constraints (17) ensure that the service start time is after the containers' arrival time. Constraints (18) set that the service finish time equals the sum of the service start time and the service duration. Constraints (19) ensure that ships depart only after all services are concluded. Constraints (20) guarantee that an order's arrival time cannot precede the ship's arrival time. Constraints (21) set the ship's last service start time.

$$t'^{kr}_{p(r)} \geqslant a_{p(r)} y_{ij}^{kr},\ \bar{t}^{kr}_{p(r)} \leqslant b_{p(r)}(y_{ij}^{kr} + M(1 - y_{ij}^{kr})) \quad \forall (i,j)\in A, \forall r\in R,\ \forall k\in K \tag{16}$$

$$t_i^{kr} \leqslant t'^{kr}_i \quad \forall i\in N,\ \forall k\in K,\ \forall r\in R \tag{17}$$

$$t'^{kr}_i + t''^{kr}_i \sum_{j\in N} y_{ij}^{kr} \leqslant \bar{t}_i^{kr} \quad \forall i\in N,\ \forall k\in K,\ \forall r\in R \tag{18}$$

$$\bar{t}_i^{k} \geqslant \bar{t}_i^{kr} \quad \forall i\in N,\ \forall k\in K,\ \forall r\in R \tag{19}$$

$$t_i^{k} \leqslant t_i^{kr} \quad \forall i\in N,\ \forall k\in K,\ \forall r\in R \tag{20}$$

$$t'^{k}_i \geqslant t'^{kr}_i \quad \forall i\in N,\ \forall k\in K,\ \forall r\in R \tag{21}$$

Constraints (22) and (23) ensure that the travel time aligns with the distance and speed. Constraints (24) and (25) determine waiting time and delay time, respectively.

$$\bar{t}_i^k + \tau_{ij}^k - t_j^k \leqslant M(1 - x_{ij}^k) \quad \forall (i,j) \in A,\ \forall k \in K \tag{22}$$

$$\bar{t}_i^k + \tau_{ij}^k - t_j^k \geqslant -M(1 - x_{ij}^k) \quad \forall (i,j) \in A,\ \forall k \in K \tag{23}$$

$$t_{ki}^{\text{wait}} \geqslant t'^k_i - t_i^k \quad \forall i \in N,\ \forall k \in K \tag{24}$$

$$t_r^{\text{delay}} \geqslant (\bar{t}_{d(r)}^{kr} - b_{d(r)}) \sum_{i \in N} y_{id(r)}^{kr} \quad \forall r \in R,\ \forall k \in K \tag{25}$$

Solving the proposed MIP model using an exact approach is time-consuming due to the NP-hard nature of the studied problem (Zhang et al., 2020). Therefore, in next subsection, we propose the ALNS algorithm to reduce computation time.

4.2.2. Adaptive large neighborhood search

To solve the optimization problem, we develop algorithms that exploit the structure of the problem and utilize an advanced computational technique, i,e., ALNS (Zhang et al., 2022b). ALNS is a metaheuristic optimization technique that is particularly well-suited to solving complex routing problems with multiple constraints. By iteratively exploring the solution space and adapting the search strategy based on the quality of the solutions found, ALNS can quickly converge to high-quality solutions that meet a variety of criteria. The pseudocode for the ALNS developed for the ship fleet routing problem is provided in Algorithm 1. The adaptive mechanism of the ALNS has been documented in the literature (Ropke and Pisinger, 2006; Zhang et al., 2022a,b) and will not be repeated here. The proposed ALNS algorithm is specifically tailored to address ship fleet routing problems, incorporating several original methodological enhancements: (a) We customize and add operators tailored to maritime logistics, including Capacity Utilizing Insertion, Regret Insertion, Most Constrained First Insertion, Related Removal, History Removal, and Port Removal. For example, the Capacity Utilizing Insertion operator and the Route Removal operator are specifically designed to optimize the utilization of ship capacity. Additionally, the Port Removal operator strategically eliminates ports negatively impacting overall route efficiency, enhancing the practical effectiveness and economic viability of routing decisions. Prior studies have demonstrated that such operator customization alone has been recognized as a significant contribution (Schiffer and Walther, 2018; Gschwind and Drexl, 2019; François et al., 2019; Santos and de Carvalho, 2021; Chen et al., 2021; Voigt, 2025). (b) We benchmark ALNS against four alternative heuristics (2-opt, Genetic Algorithm, Simulated Annealing, and Greedy Algorithm), and show that ALNS consistently yields the lowest costs and highest service levels with acceptable runtime, demonstrating that our choice of ALNS is evidence-based. (c) we embed maritime-specific constraints and objectives, including ship capacities, dynamic port handling times, and comprehensive operational costs (fuel, port service fees, penalties). By using actual maritime nodes (N), edges (A), and realistic maritime route distances (d_{ij}), the optimization process closely mirrors real operational scenarios, leading to practically applicable routing decisions rather than abstract solutions. (d) Unlike studies using hypothetical or public datasets, we integrate ALNS with the constructed maritime transport network and extend it to handle disruption-induced re-planning, thereby bridging data-driven network generation and ship fleet routing.

Algorithm 1: The ALNS algorithm for ship fleet routing.

Input: K, R, N, A; **Output:** X_{best} ; // X_{best} represents the best solution.
Construct the initial solution X_{initial} using greedy insertion operator; set $T_{\text{Temp}} > 0$ based on the objective of the initial solution;
$X_{\text{last}} \leftarrow X_{\text{initial}}$; $X_{\text{best}} \leftarrow X_{\text{last}}$; // X_{last} means the last solution.
repeat
  at the start of each segment, select operators according to the updated weights;
  $X_{\text{current}} \leftarrow X_{\text{last}}$; iteratively using insertion and removal operators until R_{pool} is empty or the number of attempts to empty R_{pool} exceeds the preset limit; update X_{current} and R_{pool}; // R_{pool} is the pool of unserved orders.
  if $F(X_{current}) < F(X_{last})$ **then**
    $X_{\text{last}} \leftarrow X_{\text{current}}$;
  else
    $X_{\text{last}} \leftarrow X_{\text{current}}$ with $p = e^{\frac{-(F(X_{\text{current}}) - F(X_{\text{last}}))}{T_{\text{Temp}}}}$; // Update X_{last} by the simulated annealing (Ropke and Pisinger, 2006) and p is the probability of accepting X_{current} as X_{last}.
  end
  if $F(X_{last}) < F(X_{best})$ **then**
    $X_{\text{best}} \leftarrow X_{\text{last}}$;
  end
  $T_{\text{Temp}} \leftarrow T_{\text{Temp}} \cdot c$; // c is the cooling rate.
until *the preset number of iterations has been achieved*;

Table 3
AIS trajectory data.

MMSI	Receive time	Long. (°)	Lat. (°)	*SOG* (knot)	*COG* (°)	Heading (°)
205791000	1,592,490,228	114.03434	22.35127	14.2	254.5	255
205791000	1,594,267,182	114.02576	22.34898	14.2	255.7	260
205791000	1,592,490,732	113.99862	22.34642	15	268.4	268
…	…	…	…	…	…	…
412475740	1,593,848,407	113.68124	22.62808	9.1	159.4	159
412475740	1,593,850,209	113.68131	22.62807	6.1	287.4	287
412475740	1,593,852,011	113.71897	22.57139	6.1	278.7	278

All insertion operators share a fundamental operation: inserting an order into a ship k's route. The insertion operators used are described below:

Greedy Insertion operator evaluates all possible solutions and inserts the order into the optimal route (Ghilas et al., 2016).

Capacity Utilizing Insertion operator prioritizes inserting orders to ships with higher load factors, which aims to utilize ship capacity.

Regret Insertion operator places order r by evaluating various regret values. Initially, it assesses all possible placements of order r into every route, then computes the regret value for each option. The insertion with the highest regret value is chosen (Ropke and Pisinger, 2006).

Most Constrained First Insertion operator sorts the orders using a weighted function that considers the distance between pickup and delivery terminals, load, and time windows (Qu and Bard, 2012).

Random Insertion operator selects ships and positions at random, inserting the order as soon as a feasible solution is identified (Danloup et al., 2018).

All removal operators involve removing the order from routes. The primary difference between these removal operators lies in the selection of orders. The following are the removal operators used:

Worst Removal operator eliminates the orders with the highest cost within each route (Wolfinger, 2021).

Related Removal operator starts by randomly removing an order r, then proceeds to remove similar orders r' based on factors such as distance, time, load, and the ships capable of serving both r and r' (Ropke and Pisinger, 2006).

History Removal operator relies on historical cost data to identify and remove orders that may be suboptimal, having higher current costs than their historical counterparts.

Port Removal operator removes all orders associated with a port to manage port-specific constraints or bottlenecks effectively.

Route Removal operator is used when insertion operators fail to find feasible solutions with minimal removals. This involves clearing the entire route, returning all its orders to the order pool. The aim is to reduce the number of ships used and optimize capacity utilization.

Random Removal operator randomly selects ships and removes one order from each (Danloup et al., 2018).

By iteratively applying the insertion and removal operators until a predetermined number of iterations is reached, the most optimal solution among all generated solutions will be selected and reported as the final transport plan for the ship fleets.

5. Numerical experiments

Section 5.1 describes the experimental area and dataset. Section 5.2 presents the results of maritime transport network construction and compares the proposed approach with two benchmark methods. Section 5.3 analyzes ship fleet routing under various scenarios, highlighting performance and strategic implications. Collectively, these sections demonstrate the potential of data-driven technologies to enhance the efficiency and effectiveness of maritime logistics.

5.1. Experimental area and data description

To validate the proposed data-driven optimization model, the Pearl River Estuary is selected as the study area. This region extends from 113.52°E to 114.04°E in longitude and from 22.05°N to 22.72°N in latitude, as shown in Fig. 9. The analysis is based on AIS data provided by the Maritime Administration, covering the period from June 1, to December 31, 2020. This dataset contains over 1.215 million records for container ships and includes attributes such as MMSI, reception time (s), longitude (Long.), latitude (Lat.), speed over ground (SOG), course over ground (COG), and heading (detailed in Table 3).

The ships' information, such as capacity and speed, is also obtained from AIS data. The shipment orders are generated based on the approaches outlined in the literature on freight transport optimization (Zhang et al., 2022c, 2023). The container volume of orders is uniformly distributed in $[10, 30]$ TEUs. The earliest pickup time, $a_p(r)$, is uniformly distributed in $[1, 330]$. The latest delivery time, $b_d(r)$, is given by $b_d(r) = a_p(r) + LD_r$, where LD_r represents the lead time. The lead time LD_r is independently and identically distributed among $\{12, 24, 36\}$ hours, with corresponding probabilities $\{0.6, 0.25, 0.15\}$. We refer to Zhang et al. (2022b) for details on the unit cost parameters, such as waiting cost, loading cost, storage cost, and delay penalty.

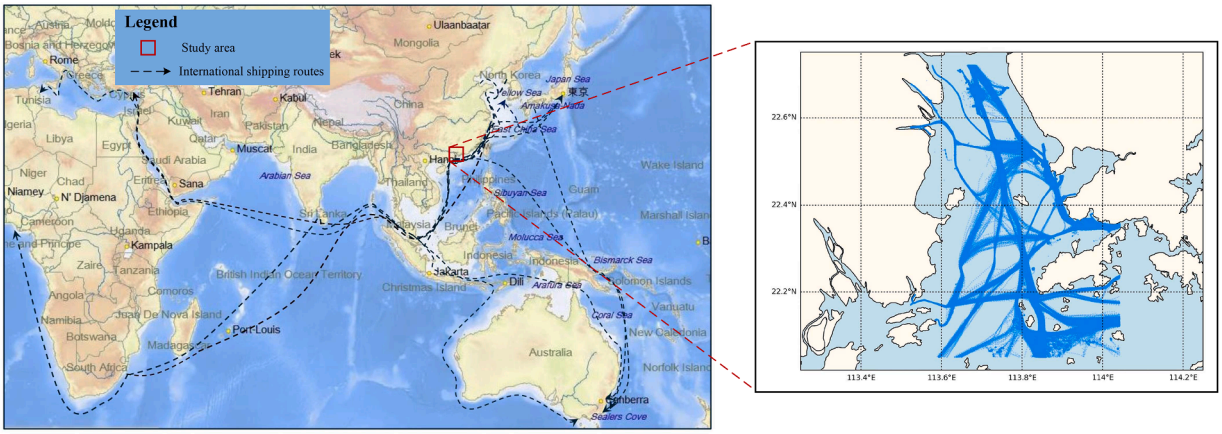


Fig. 9. Experimental area.

Table 4
Range of voyages for each class of routes.

Route level	Voyage Frequency Range
Main route	(180, ∞)
Secondary main route	(90,180]
Branch route	(25,90]
Secondary branch route	[10,25]

5.2. Maritime transport network construction results and comparison

Following the methodology outlined in Section 4.1, Section 5.2.1 displays the results of each construction step. Subsequently, Section 5.2.2 compares these results against two benchmarks in the literature (Lu et al., 2020; Varlamis et al., 2021), providing a comprehensive evaluation of the proposed approach.

5.2.1. Maritime transport network construction results

The results of trajectory feature-point identification are shown in Fig. 10. As shown in Fig. 10(a) and (b), the trajectory cleaning process effectively removes drift points and filters out segments crossing land, producing a trajectory that is more accurate and consistent. Berthing points (blue) and anchorage points (yellow) are identified in Fig. 10(c), while boundary-crossing points are shown in Fig. 10(d).

Turning points are identified using the cumulative turning feature method with $\alpha_T = 2°$ and $N_T = 5$, selected based on empirical evaluation (see Appendix B for the experimental setup and results). These parameters correspond to a cumulative turning angle of 10°, which provides the highest identification accuracy. Guided by navigation experience and experiments, turning feature constraints are set as: $SOG > 3$ knots, $TR > 5°/\text{min}$ and $R_T \geq 80\,\%$ (Abebe et al., 2020; Li et al., 2022). This enables identification of turning node areas, shown in Fig. 10(e).

DBSCAN clustering is applied to the three feature types. To account for spatial heterogeneity and ensure robust results, optimal parameters are determined through sensitivity analysis, as detailed in Appendix C. By clustering the identified trajectory feature points, 14 boundary-crossing clusters, 9 staying clusters, and 48 turning clusters are obtained, as shown in Fig. 11(a). For each cluster, a minimum convex hull is generated to delineate its boundary, and the geometric centroid is taken as the representative node in the network topology (Fig. 11(b)).

The voyage threshold for effective node connections is set to $n_f = 10$, and routes are classified into four levels according to the statistical distribution of voyage frequencies over six months of historical data (Table 4). Based on these settings, a weighted topological network is constructed, as shown in Fig. 11(c).

5.2.2. Comparison with benchmarks

To validate the effectiveness of the proposed method, we compare its network construction performance with two representative baseline approaches widely used in maritime route extraction from AIS data. The first is a heading threshold-based segmentation method (Varlamis et al., 2021; Huang et al., 2024; Corvino et al., 2025), which detects waypoint candidates from significant changes in ship heading and speed. The second benchmark is the Douglas-Peucker (DP) algorithm (Lu et al., 2020), a classical trajectory simplification technique that preserves key geometric features, particularly turning points, and remains a standard tool in AIS data

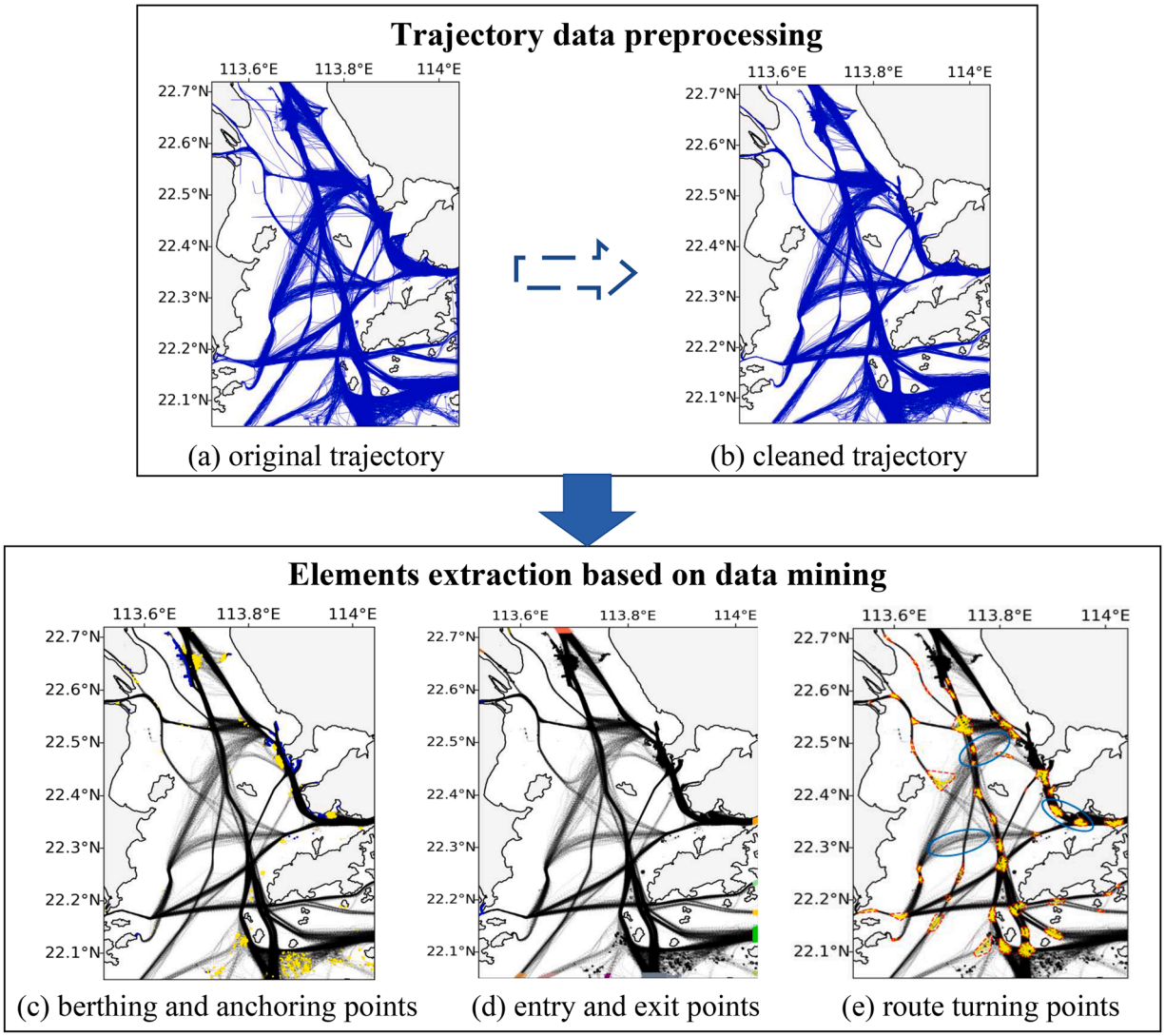


Fig. 10. Trajectory feature point identification results.

preprocessing and route reconstruction (Tang et al., 2021; Lee and Cho, 2022; Zhang et al., 2024; Lu et al., 2020). Together, these rule-based and geometry-based approaches provide complementary baselines for evaluating the accuracy of the proposed method in detecting navigational nodes at critical turning regions.

Fig. 12(a) and (b) show the node extraction results of the two baseline methods, whereas Fig. 12(c) presents the results of the proposed method. The experiment demonstrates that the proposed approach constructs a navigational network with 200 directional edges and achieves 95 % effective connectivity. In contrast, the baseline methods exhibit deviations from actual routes in some areas, with some nodes erroneously located on land.

The following metrics are computed to assess performance: correctly identified nodes (N_{ide}), misidentified nodes (N_{mis}), unidentified nodes (N_{un}), average node position error (d_{np}), identification accuracy (R_{acc}), and false identification rate (R_{mis}). The last three metrics are defined in (26)-(28). Ground truth is established from official electronic charts and ship navigation practices in the study area, as shown in Fig. 12(d).

$$d_{\mathrm{np}} = \frac{\sum_{i=1}^{n} \mathrm{dis}\left(p_i, p_i'\right)}{n} \tag{26}$$

$$R_{\mathrm{acc}} = \frac{N_{\mathrm{ide}} - N_{\mathrm{mis}}}{N_{\mathrm{mark}}} \times 100\,\% \tag{27}$$

$$R_{\mathrm{mis}} = \frac{N_{\mathrm{mis}}}{N_{\mathrm{mark}}} \times 100\,\% \tag{28}$$

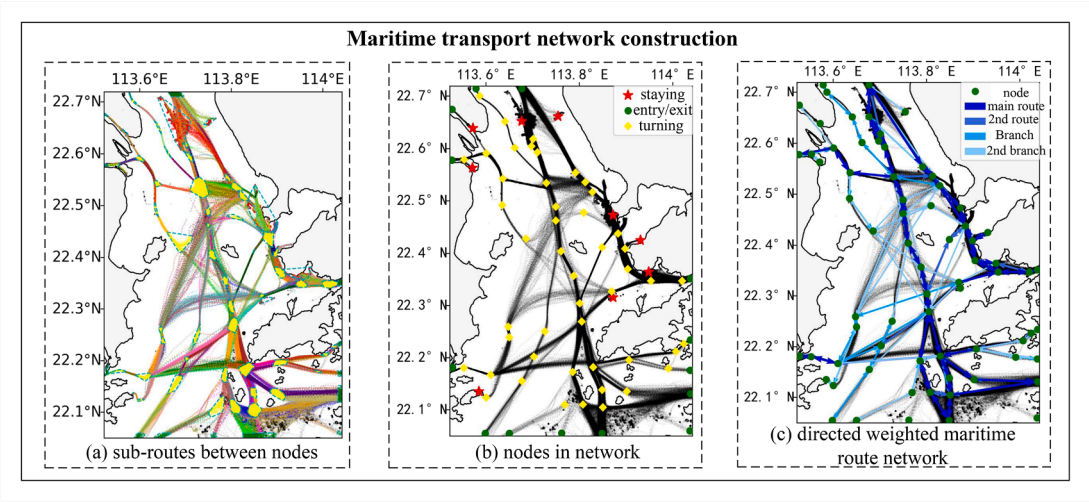


Fig. 11. Maritime transport network construction results.

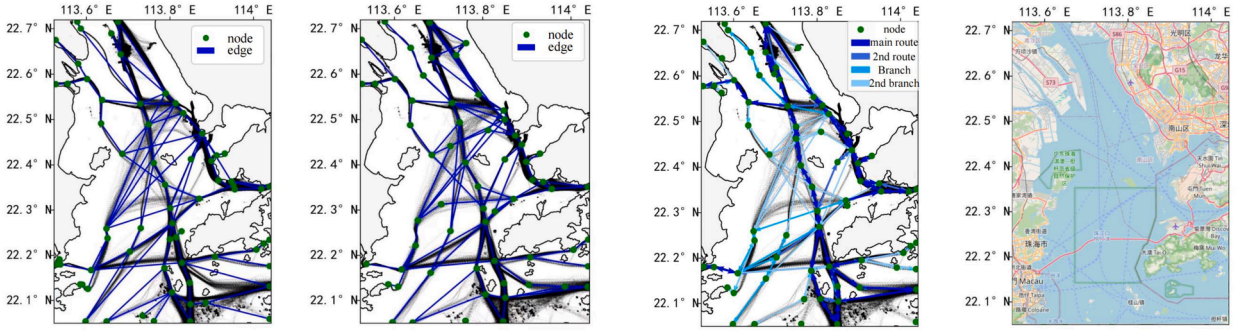


Fig. 12. Comparative experiment on the extraction of maritime transport network.

where p_i and p_i' are the nodes extracted by the identification method and the reference nodes for baseline match, respectively. n is the number of matched node pairs, and N_{mark} is the total number of baseline nodes.

The results of these indicators are summarized in Table 5. The proposed method achieves an overall node identification accuracy of 86.42 % with a false identification rate of only 1.23 %. For turning nodes, the identification accuracy reaches 82.46 %, significantly higher than the 66.67 % and 68.42 % obtained by the two baseline methods (Lu et al., 2020; Varlamis et al., 2021).

In summary, the proposed method outperforms two commonly used baselines by achieving higher accuracy in identifying critical navigational nodes and routes. Moreover, the resulting weighted network provides a richer representation of maritime traffic patterns, offering a practical foundation for ship fleet routing analysis.

5.3. Results and analysis of ship fleet routing

Building upon the constructed maritime transport network, this section examines ship fleet routing to assess both efficiency and robustness. First, the proposed heuristic algorithm is compared against an exact optimization method in Section 5.3.1. Routing performance under network disruptions, including route failures, node failures, and port congestion, is also analyzed, with particular emphasis on adaptive strategies for maintaining operational efficiency.

5.3.1. Comparison with an exact approach

Table 6 compares the performance of an exact approach (Gurobi) (Gurobi, 2025) and a heuristic algorithm across different scenarios, characterized by varying numbers of ships (K) and orders (R). Both methods yield identical results in terms of total cost (F) and the number of orders completed (R'), demonstrating their effectiveness in finding optimal or near-optimal solutions. However, a

Table 5
Comparative results of node extraction using different methods.

Article	Node type	N_{ide}	N_{mis}	N_{un}	d_{np}	R_{acc} (%)	R_{mis} (%)
Varlamis et al. (2021)	staying	10	4	4	1582.92	60	40
	entry/exit	14	0	0	157.84	100	0
	turning	42	4	19	568.38	66.67	7.02
	all nodes	66	8	23	635.02	71.60	9.87
Lu et al. (2020)	staying	6	0	4	793.68	60	0
	entry/exit	13	0	1	288.98	92.86	0
	turning	43	4	18	939.53	68.42	7.02
	all nodes	62	4	23	789.01	71.60	4.94
This study	staying	9	0	1	379.92	90	0
	entry/exit	14	0	0	136.79	100	0
	turning	48	1	10	282.30	82.46	1.75
	all nodes	71	1	11	266.65	86.42	1.23

Table 6
Comparison results with the exact approach.

		Exact approach			Heuristic algorithm		
K	R	F	R'	CPU (s)	F	R'	CPU (s)
3	5	10,495	5	65	10,495	5	4.2
3	10	28,251	8	433	28,251	8	13.8
3	20	50,542	13	4326	50,542	13	20.3
5	5	7345	5	343	7345	5	6.2
5	10	10,565	10	1656	10,565	10	14.5
5	20	15,349	15	7648	15,349	15	26.5
10	5	5646	5	1976	5646	5	8.3
10	10	8956	10	8565	8956	10	45.4
10	20	32,905	18	10800∗	14,984	18	52.6

∗: The time limitation (3 hours) is reached.
K: Number of ships; R: Number of orders; F: Total cost (CNY); R': Number of orders completed; CPU: Computation time of finding the optimal/best solution.

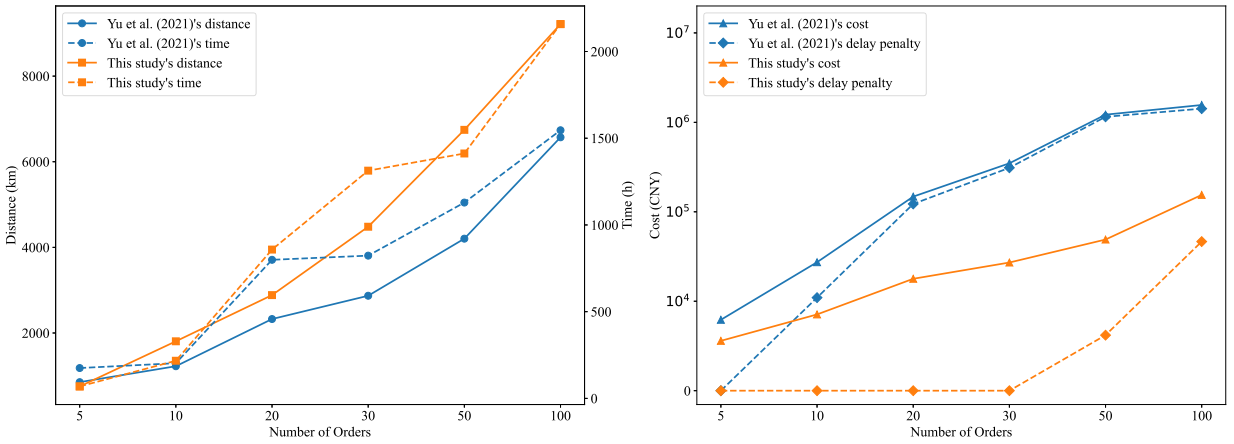
significant difference is observed in the computation time (CPU). The heuristic algorithm consistently outperforms the exact approach in terms of speed, with the exact approach requiring substantially more time, especially as the problem size increases. Notably, the exact approach reached a time limitation in one scenario (K = 10, R = 20), highlighting the heuristic's advantage in computational efficiency for larger, more complex problems.

5.3.2. *Comparison with state-of-the-art methods*

To evaluate the effectiveness of our method in comparison with state-of-the-art approaches, we conduct comparative experiments using the scenario presented in Yu et al. (2021). Results for normal and congested scenarios are shown in Figs. 13 and 14.

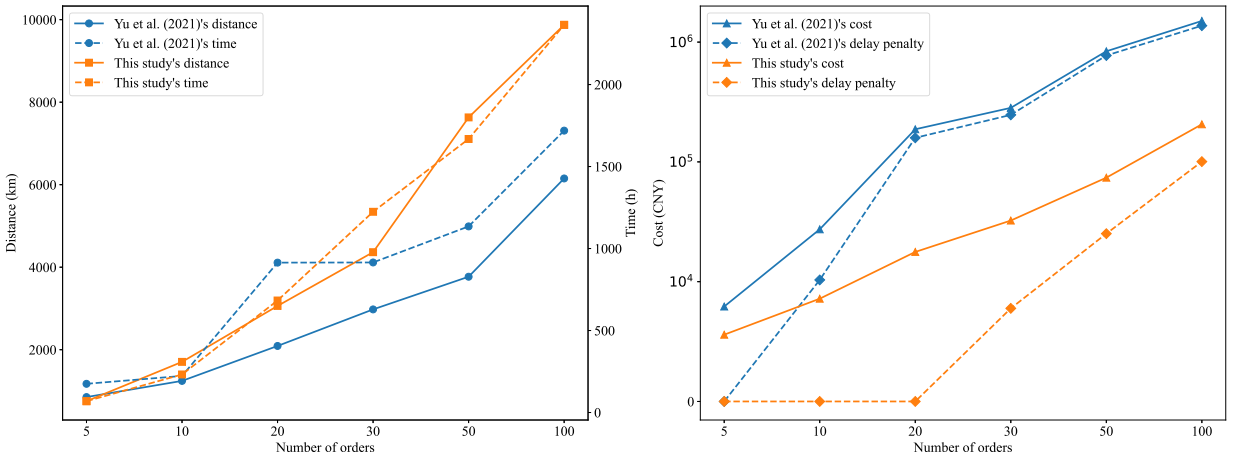
The comparison reveals that the approach proposed by Yu et al. (2021) generally produces routes with shorter sailing distances and travel times. However, it results in significantly higher total costs and delay penalties, particularly as the number of orders increases. This effect is especially pronounced under congested conditions, where their routing strategy does not consider longer waiting times in ports, causing increased cost and delay penalty. In contrast, our method adopts a comprehensive cost framework that integrates transit costs, loading/unloading costs, storage costs, waiting costs, and delay penalties. Although the resulting routes may be slightly longer in terms of distance and travel time, they consistently yield lower total costs and delay penalties across various operational scenarios. These results highlight our method's superior ability to account for congestion effects and cost-sensitive constraints, ultimately enhancing overall cost-efficiency and service reliability.

To evaluate the performance of different algorithms in route planning, we conduct comparative experiments using the 2-opt Algorithm (Englert et al., 2014), Genetic Algorithm (Utomo et al., 2023), Simulated Annealing Algorithm (Zhu et al., 2025), Greedy Algorithm (Zhao et al., 2025), and the proposed ALNS algorithm. The experiments are performed on seven instances ranging from 3 to 40 orders, with an analysis on three performance indicators: the number of served orders, total transportation cost, and computation time. The detailed results are presented in Table 7. As indicated, ALNS matches or outperforms all other algorithms in terms of served orders across all instances. It also consistently yields the lowest total cost (Fig. 15), demonstrating its strength in navigating large and complex solution spaces to find high-quality solutions. Although ALNS requires slightly more computation time than simpler heuristics like 2-opt and Greedy, its runtime remains under 1 second, which is acceptable for practical use. The trade-off between marginally longer runtime and significantly improved solution quality is favorable for real-world applications.



(a) Transport distance and time (b) Transport cost and delay penalty

Fig. 13. Comparison between Yu et al. (2021) and this study (normal condition).



(a) Transport distance and time (b) Transport cost and delay penalty

Fig. 14. Comparison between Yu et al. (2021) and this study (congested condition).

Table 7
Comparative analysis of five different heuristic algorithms.

		Total Cost (CNY)					Order Completed					Computation Time (s)				
$\|O\|$	$\|K\|$	2-opt	ALNS	Greedy	SA	Genetic	2-opt	ALNS	Greedy	SA	Genetic	2-opt	ALNS	Greedy	SA	Genetic
3	2	31863	29365	31863	31863	30134	3	3	3	3	3	0.000	0.009	0.026	0.001	0.014
5	2	37945	32704	37185	38329	34636	4	4	4	4	4	0.001	0.019	0.043	0.002	0.032
5	5	50147	36824	36824	38330	36824	5	5	5	5	5	0.001	0.027	0.116	0.003	0.043
10	3	57023	41063	44031	53150	50904	6	6	6	6	6	0.001	0.034	0.126	0.004	0.070
10	5	93717	74796	93717	89074	86842	10	10	10	10	10	0.001	0.058	0.193	0.005	0.092
15	4	80424	61581	68868	78182	71943	10	10	10	10	9	0.001	0.076	0.248	0.006	0.091
15	7	140,973	106554	116397	121166	108299	15	15	15	15	14	0.001	0.116	0.288	0.009	0.164
20	7	140341	122704	140341	136767	122010	17	17	17	17	16	0.001	0.201	0.440	0.011	0.216
20	12	173867	147705	173867	166572	158150	20	20	20	20	20	0.001	0.221	0.637	0.015	0.267
30	10	208380	191948	208380	208064	195571	26	26	26	26	23	0.002	0.444	0.932	0.020	0.392
30	15	254999	216747	245767	252198	243981	30	30	30	30	30	0.002	0.506	1.608	0.025	0.527
40	12	271281	243207	271281	246810	233439	30	30	30	29	29	0.002	0.592	1.801	0.032	0.619
40	40	344059	275492	302720	334284	319123	40	40	40	40	40	0.003	0.923	3.657	0.060	0.893

$|O|$: number of orders; $|K|$: number of ships; 2-opt: Two-Optimal Exchange Algorithm; Greedy: Greedy Algorithm; SA: Simulated Annealing Algorithm; Genetic: Genetic Algorithm; ALNS: Adaptive Large Neighborhood Search.

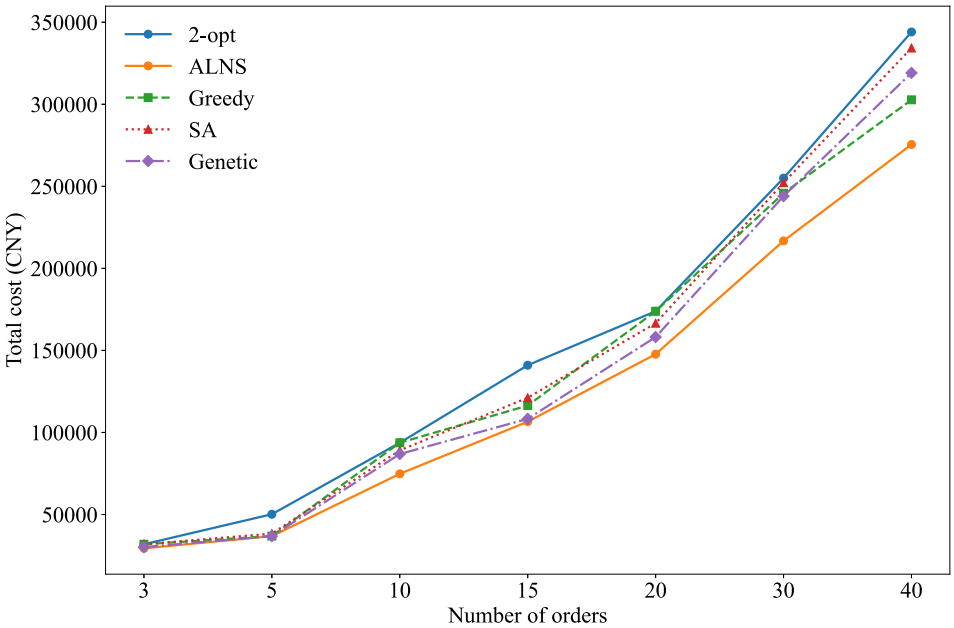


Fig. 15. Comparison of total cost among state-of-the-art algorithms.

Table 8
Ship fleet and shipment order information.

Type	Item	Origin	Destination
Ship	1	0	1
	2	0	1
	3	3	6
Shipment orders	1	0	2
	2	1	0
	3	2	5
	4	3	6
	5	0	8
	6	0	7
	7	5	6
	8	2	8
	9	1	2
	10	6	3

5.3.3. Results under disruptions on routes

Nine port nodes are identified in the extracted network. To assess the impact of route disruptions, comparative experiments are conducted using the route classes defined in Table 4, which categorizes transport routes into four levels based on historical voyage statistics. Ports 0 and 3, which record the highest traffic flows, are selected as origins for analysis. Ten shipment orders are generated for three cargo ships, as detailed in Table 8, with origins and destinations assigned according to the traffic density distribution. These experiments demonstrate how the proposed optimization model dynamically adjusts fleet routing in response to route disruptions.

Under normal conditions without disruptions, the proposed optimization model produces the transportation plan illustrated in Fig. 16). Ship 1 departs from port 0, loads order 9 at port 1, and unloads it at port 2; it then proceeds to port 6 to load order 10 and delivers it to port 3 (via a waypoint near port 4; see Fig. 16). Finally, the ship returns to port 1, completing its mission.

Ship 2 departs from port 0 with orders 1 and 5, continues to port 2 to load orders 3 and 8 while unloading order 1, and then sails to port 5 to unload order 3. It subsequently calls at port 8 to unload orders 5 and 8 before returning to port 1. Ship 3 departs from port 3 with order 4, proceeds to port 1 to load order 2, and then to port 5 to load order 7. Subsequently, it unloads order 2 at port 0 and orders 7 and 4 at port 6, where the mission concludes.

The weights (a_{ij}) in the transport network (see Eq. 3) indicate that the route between ports 0 and 1 is heavily trafficked, making it highly vulnerable to disruptions. To quantify the impact of such uncertainties, transport distances on these routes are ad-

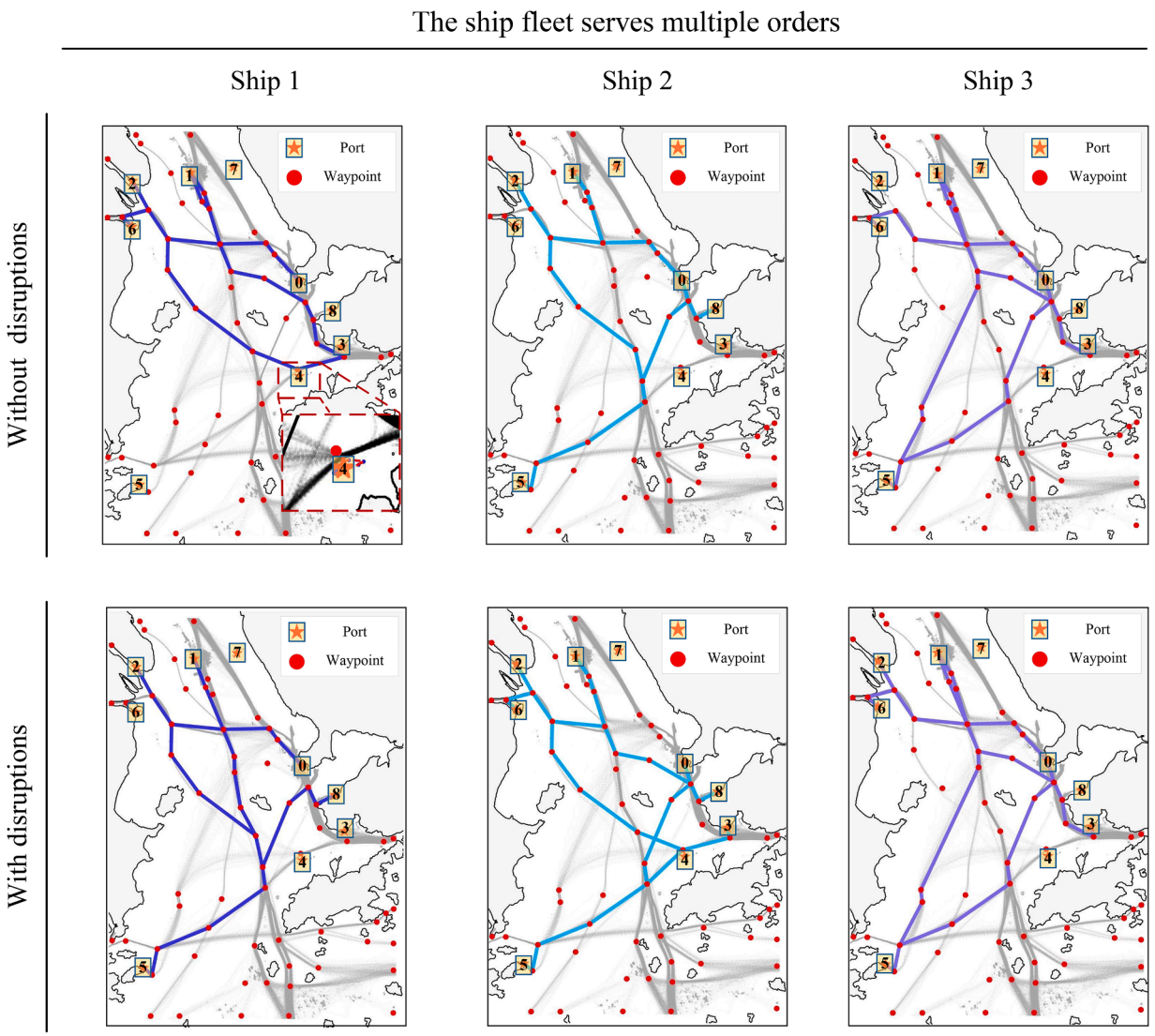


Fig. 16. Comparison of transport plans with and without disruptions.

justed proportionally to a_{ij}. As illustrated in the lower panels of Fig. 16, the model prioritizes routing efficiency over pure distance minimization to mitigate disruption effects on fleet operations. The most significant adjustments occur on the busy link between ports 0 and 1 for ships 1 and 2. However, under increased disruptions, the adjusted plan omits orders 1 and 2, underscoring that disruptions can severely degrade transport efficiency and even result in unfulfilled orders.

5.3.4. Results under different congestion levels in ports

This section examines the impacts of port congestion through two comparative experiments. Level 1 represents normal traffic with occasional congestion, while Level 2 simulates severe congestion. Congestion is modeled by increasing vessel service time at port, denoted as C (hours).

Under normal operational conditions (Level 1), the baseline service time of one hour is incrementally extended by 0.5-hour steps up to 2.5 hours. The resulting effects on transportation efficiency and economic cost are presented in Table 9. As expected, total cost F increases with longer service times and larger order volumes (R). However, a cost decrease is observed in some instances (e.g., when R increases from 60 to 70 and from 80 to 90), suggesting improved reduced operational costs due to economies of scale. With respect to service efficiency, the service rate (completion rate of shipment orders) remains stable despite increased service times and order volumes, demonstrating the model's capability to handle large-scale instances.

Algorithmic efficiency is evaluated using the Best/Total ratio, which measures the proportion of computation time required to identify the optimal solution. As shown in Fig. 17, shipment order size strongly influences this ratio. Moreover, as service times

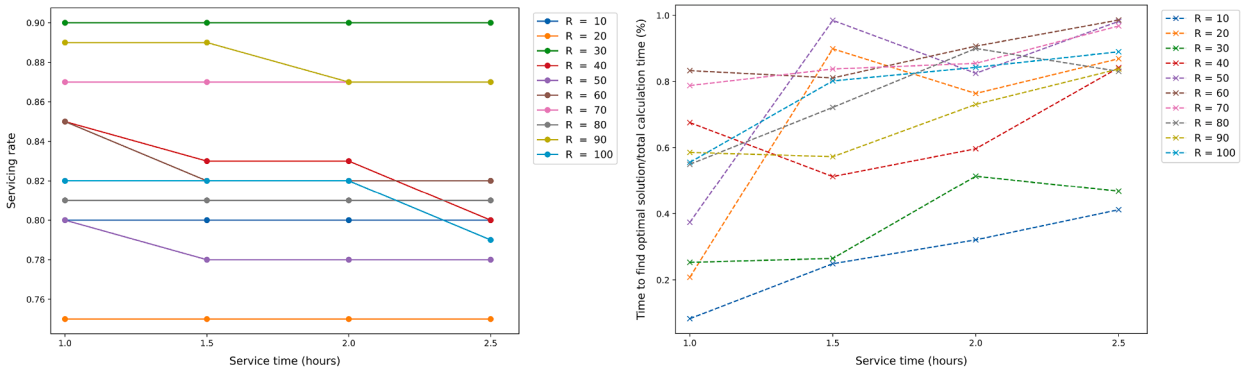


Fig. 17. Service rate and computation time vary with increasing service time at Level 1.

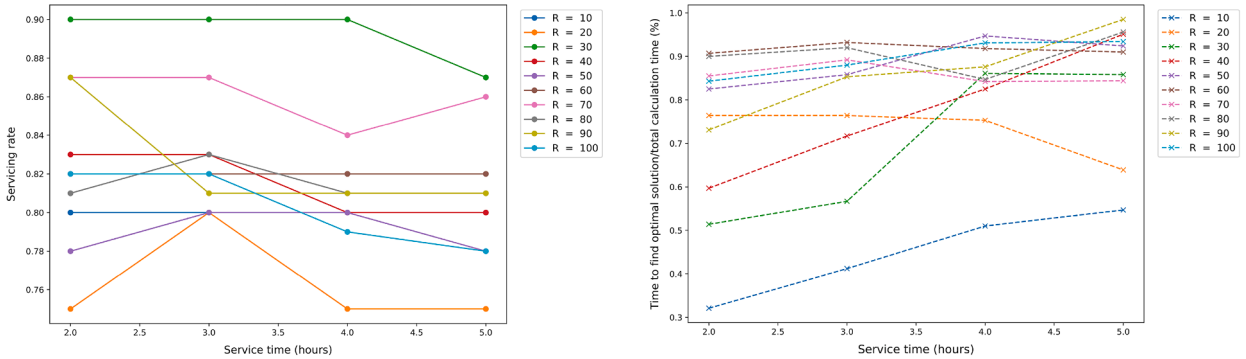


Fig. 18. Service rate and computation time vary with increasing service time at Level 2.

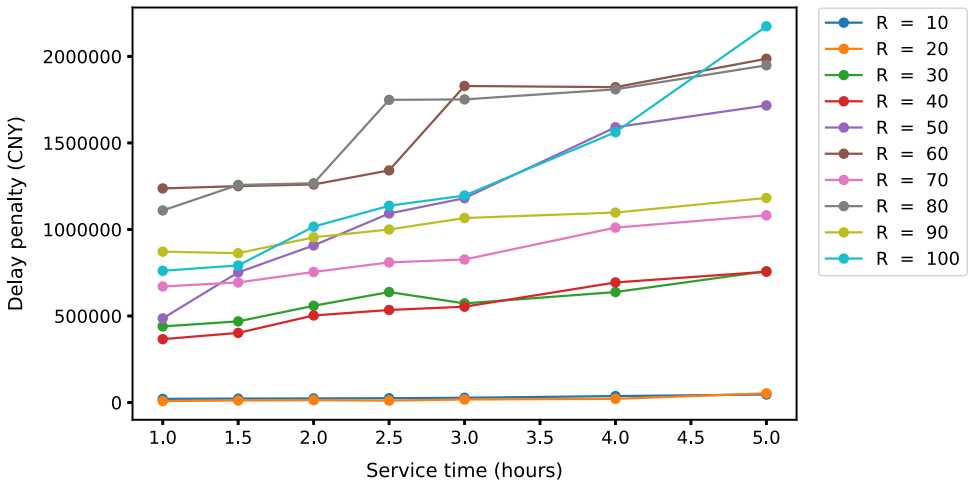


Fig. 19. Penalty cost changes under increasing service time.

increase, the growth of the Best/Total ratio is less pronounced for larger order sets, indicating that the optimization algorithm maintains stable performance even under complex planning conditions.

Heavily congested conditions (Level 2) are modeled by extending port service times, beginning at 2 hours and increasing in 1-hour increments. Four levels are defined to capture the effects of severe congestion, with the aggregated results reported in Table 10. Notably, unlike in Level 1, prolonged service times under Level 2 cause substantial fluctuations and significant increases in total cost. Moreover, as the number of shipment orders grows, the service rate declines sharply, particularly when order volumes exceed

Table 9
Experiment results at Level 1.

			Avg. Costs (CNY)				Service Efficiency		CPU (s)		
R	C	K	F	F1	F2	F3	Order completed	Service rate	Initial	Best	Best/Total (%)
10	1	3	28,251	340	822	21,040	8	0.8	1.17	13.81	0.083
10	1.5	3	29,389	398	1493	22,817	8	0.8	1.52	29.62	0.249
10	2	3	30,661	329	1524	23,540	8	0.8	1.57	35.73	0.321
10	2.5	3	32,029	326	1584	24,878	8	0.8	1.94	42.57	0.412
20	1	5	15,349	309	885	8520	15	0.75	1.73	26.51	0.208
20	1.5	5	23,375	432	1187	11,671	15	0.75	2.31	242.65	0.899
20	2	5	26,129	513	2292	13,571	15	0.75	2.68	230.16	0.764
20	2.5	5	24,117	681	1639	10,993	15	0.75	3.57	227.34	0.869
30	1	5	465,214	884	4728	439,777	27	0.9	3.89	119.85	0.253
30	1.5	5	498,267	961	6233	468,768	27	0.9	5.17	185.29	0.265
30	2	5	585,375	961	6468	558,962	27	0.9	6.05	321.88	0.514
30	2.5	5	616,659	776	7534	598,323	27	0.9	6.96	354.45	0.468
40	1	7	587,015	1229	7748	366,057	34	0.85	7.96	611.15	0.676
40	1.5	7	571,909	1415	10,039	402,327	33	0.83	8.78	940.14	0.513
40	2	7	401,445	1473	11,518	392,930	33	0.83	10.99	983.03	0.597
40	2.5	7	438,718	1562	12,563	424,650	32	0.8	13.13	1022.07	0.842
50	1	7	527,304	1131	12,657	486,243	40	0.8	18.38	657.73	0.374
50	1.5	7	794,668	1654	13,877	751,932	39	0.78	23.24	1974.6	0.985
50	2	7	947,077	1773	13,645	907,787	39	0.78	26.64	1970.64	0.825
50	2.5	7	1,133,907	1816	14,562	1,092,188	39	0.78	31.45	1939.24	0.982
60	1	7	1,288,711	1430	12,807	1,237,142	51	0.85	32.75	881.09	0.833
60	1.5	7	1,305,352	1443	17,386	1,250,307	49	0.82	39.78	2283.07	0.811
60	2	7	1,306,282	1581	19,468	1,259,770	49	0.82	40.25	2700.7	0.907
60	2.5	7	1,398,911	1704	20,846	1,341,295	49	0.82	60.43	2701.82	0.986
70	1	8	731,137	884	16,530	670,474	61	0.87	36.08	583.59	0.788
70	1.5	8	753,769	981	16,849	693,783	61	0.87	46.59	661.14	0.838
70	2	8	818,609	1064	17,623	754,275	61	0.87	45.4	854.51	0.855
70	2.5	8	870,338	1281	17,896	809,662	61	0.87	46.59	958.82	0.968
80	1	8	1,174,588	1490	14,214	1,109,725	65	0.81	12.11	1701.85	0.55
80	1.5	8	1,325,523	1587	20,253	1,257,298	65	0.81	21.13	2047.75	0.722
80	2	8	1,336,612	1664	20,204	1,266,832	65	0.81	29.17	2617.52	0.9
80	2.5	8	1,593,474	1696	27,463	1,548,961	65	0.81	39.24	2651.34	0.831
90	1	10	945,974	1548	18,057	871,869	80	0.89	63.03	615.38	0.586
90	1.5	10	938,942	1647	19,290	862,719	80	0.89	63.49	654.55	0.573
90	2	10	1,029,159	1681	21,279	954,603	78	0.87	67.84	733.31	0.731
90	2.5	10	1,075,792	1720	22,792	999,843	78	0.87	68.44	993.74	0.838
100	1	11	855,262	1319	27,940	761,230	82	0.82	53.35	823.9	0.556
100	1.5	11	860,331	1463	34,925	792,177	82	0.82	59.74	1097.57	0.802
100	2	11	1,106,209	1542	27,762	996,231	82	0.82	62.23	1106.57	0.843
100	2.5	11	1,222,826	1683	30,092	1,136,897	79	0.79	71.89	1199.06	0.89

R: number of orders; C: port service time (hours); K: number of ships in the fleet; F: total cost (CNY); F1: waiting cost (CNY); F2: storage cost (CNY); F3: delay penalty (CNY); Initial: computation time of the initial solution (s); Best: computation time of the best solution (s); Total: Total computation time.

90. These findings underscore the detrimental impact of severe port congestion and highlight the importance of robust fleet route planning to mitigate efficiency losses.

Level 2 exhibits lower service rates than Level 1 for the same order size, as shown in Fig. 18. Even with a small number of orders (e.g., $R = 10$), service rates decline noticeably, in contrast to the stability observed in Level 1. As order volumes increase, delay penalties rise progressively with longer service times (Fig. 19). In particular, when $R = 100$, delay penalties grow sharply, highlighting the challenges of handling large-scale instances under congestion. Severe port congestion extends vessel loading and unloading times, thereby requiring additional resources and leading to higher delay costs. Compared with inefficiencies caused by congested sailing routes, port congestion exerts a more significant impact on overall transport costs.

Table 10
Experiment results at Level 2.

			Avg. Costs (CNY)				Service Efficiency		CPU (s)		
R	C′	K	F	F1	F2	F3	Order completed	Service rate	Initial	Best	Best/Total (%)
10	2	3	30,661	329	1524	23,540	8	0.8	1.57	35.7	0.32
10	3	3	32,031	326	1596	27,405	8	0.8	1.73	42.6	0.41
10	4	3	43,918	377	1691	37,217	8	0.8	2.52	50.7	0.51
10	5	3	53,013	451	1776	46,190	8	0.8	2.83	71.3	0.55
20	2	5	26,129	513	2292	13,571	15	0.75	2.68	230.2	0.76
20	3	5	29,305	644	2009	17,518	16	0.8	3.57	230.2	0.76
20	4	5	32,839	785	2524	21,171	15	0.75	4.73	232.5	0.75
20	5	5	65,402	829	2776	53,133	15	0.75	7.78	252	0.64
30	2	5	585,375	961	6468	558,962	27	0.9	6.05	321.9	0.51
30	3	5	667,874	957	7861	572,781	27	0.9	5.95	364.6	0.57
30	4	5	670,558	1096	9332	638,323	27	0.9	6.96	402	0.86
30	5	5	787,886	1264	9608	758,692	26	0.87	7.7	452.2	0.86
40	2	7	401,445	1473	11,518	302,930	33	0.83	10.99	983	0.6
40	3	7	534,270	1698	13,877	453,686	33	0.83	17.21	1038.2	0.72
40	4	7	722,573	2278	14,532	693,757	32	0.8	20.55	1970.6	0.83
40	5	7	790,379	2357	15,962	755,771	32	0.8	26.74	2119.2	0.95
50	2	7	947,077	1790	13,645	907,787	39	0.78	26.64	1970.6	0.83
50	3	7	1,225,421	1654	15,734	1,180,925	40	0.8	34.23	2296	0.86
50	4	7	1,631,487	1854	15,962	1,590,641	40	0.8	39.48	2752.4	0.95
50	5	7	1,755,094	2153	1776	1,716,808	39	0.78	42.51	2939.2	0.92
60	2	7	1,306,282	1581	19,468	1,259,770	49	0.82	40.25	2700.7	0.91
60	3	7	1,876,838	1700	17,386	1,829,142	49	0.82	49.51	2862.9	0.93
60	4	7	1,875,488	1707	19,468	1,822,315	49	0.82	53.13	3105.7	0.92
60	5	7	2,040,656	1869	20,846	1,986,490	49	0.82	76.27	3384.8	0.91
70	2	8	818,609	1064	17,623	754,275	61	0.87	45.4	854.5	0.86
70	3	8	888,484	1275	19,515	826,559	61	0.87	52.03	990.5	0.89
70	4	8	1,070,275	1435	20,382	1,010,679	59	0.84	56.3	1022.1	0.84
70	5	8	1,148,524	1881	22,819	1,081,704	60	0.86	58.47	1138.2	0.84
80	4	8	1,336,612	1664	20,204	1,266,832	65	0.81	29.17	2617.5	0.9
80	3	8	1,820,347	1587	29,276	1,751,556	66	0.83	41.13	2688.4	0.92
80	4	8	1,824,096	1691	32,084	1,809,725	65	0.81	57.37	2701.9	0.85
80	5	8	2,015,164	1948	32,840	1,948,500	65	0.81	70.05	3107.7	0.96
90	2	10	1,029,159	1712	21,279	954,603	78	0.87	67.84	733.3	0.73
90	3	10	1,082,357	1808	23,403	966,124	73	0.81	69.29	1026.8	0.85
90	4	10	1,137,611	2014	25,792	1,097,648	73	0.81	70.98	1226.8	0.88
90	5	10	1,254,225	2129	26,429	1,182,259	73	0.81	75.31	1332.4	0.99
100	2	11	1,106,209	1542	27,762	996,231	82	0.82	62.23	1106.6	0.84
100	3	11	1,282,110	1736	31,275	1,196,085	82	0.82	75.47	1302.4	0.88
100	4	11	1,652,340	1749	33,767	1,562,130	79	0.79	81.89	1357.5	0.93
100	5	11	2,261,009	2149	34,925	2,174,214	78	0.78	103.89	1368.9	0.93

R: number of orders; C': port service time (hours); K: number of ships in the fleet; F: total cost (CNY); F1: waiting cost (CNY); F2: storage cost (CNY); F3: delay penalty (CNY); Initial: computation time of the initial solution (s); Best: computation time of the best solution (s); Total: Total computation time.

6. Conclusions

This study integrates maritime transport network construction with fleet routing to develop a data-driven optimization framework for maritime logistics. The proposed approach automates the pipeline from raw trajectory data to routing decision-making. Using data mining techniques, waypoints are identified from ship trajectories, and key behavioral regions (e.g., berthing and turning areas) are extracted. These features enable the construction of a directed, weighted transport network that captures realistic route patterns. Validation with data from the Pearl River Estuary identified 9 staying nodes, 14 boundary-crossing nodes, 48 turning nodes, and 200 sub-routes of varying scales and directions.

The constructed network serves as the foundation for fleet routing decisions, which integrate analysis of traffic patterns, waypoints, and congestion risks. This data-driven framework leverages historical insights to optimize routing, demonstrating a significant enhancement in fleet operational efficiency. A mathematical model combined with an Adaptive Large Neighborhood Search heuristic is applied to optimize routing performance. Experimental results under port and route congestion scenarios confirm the effectiveness and adaptability of the proposed framework. Overall, the synergy between network construction and fleet routing provides a holistic and proactive strategy for maritime logistics. The findings contribute to the literature on data-driven optimization while offering practical guidance for managing costs and improving service reliability in dynamic maritime environments.

The proposed data-driven routing optimization approach can enhance operational efficiency and safety for both manned and autonomous ships by optimizing routes based on maritime conditions revealed from massive maritime data. Utilizing the proposed approach offers several significant benefits:

1. Optimized route planning: By analyzing historical data on experienced routes with waypoints, shipping companies can identify the most efficient paths commonly used under varying conditions. This helps in crafting routes that maximize fuel efficiency and minimize transport time.
2. Congestion management: Understanding the potential congestion levels of each port and route through the weighted network allows operators to predict busy periods and adjust schedules accordingly. This proactive approach reduces waiting times and delays, improving operational efficiency and reducing costs associated with idle ships.
3. Enhanced safety: Data-driven insights into the usual traffic flow and common navigational patterns at specific waypoints help in predicting and avoiding potential conflict points. This improves navigational safety by reducing the likelihood of collisions and other maritime incidents.
4. Dynamic rerouting: With access to data on alternative routes, the proposed approach can re-plan the routes and schedules of ships in response to disruptions such as severe congestion, port delays, or maritime accidents. This flexibility ensures that ships can maintain schedules more reliably, enhancing service quality.
5. Autonomous fleet routing: Our model provides a comprehensive, end-to-end solution, integrating raw data processing, maritime transport network mining, and optimized fleet route planning into a cohesive pipeline. For autonomous vessels, the core challenge lies precisely in translating raw, unstructured AIS data into reliable, actionable routing and scheduling decisions. By addressing this gap, our method directly enables autonomous ships to perform optimal route selection, including detailed path planning and cargo allocation decisions, without manual intervention. Importantly, unlike previous research that predominantly focuses on route optimization for individual autonomous ships, our approach is specifically designed to handle fleet-level routing decisions, which is critical for practical implementation in maritime logistics operations.

In addition to the advantages highlighted in the paper, this study has the following potential limitations:

1. Dependence on sufficient historical data: The effectiveness of the proposed method relies on the availability of sufficient historical AIS data. If the amount of data collected is limited, the quality and reliability of the resulting route planning may be significantly compromised.
2. Limited ship type: The experiments in this study are conducted exclusively on container ship data. For other ship types (e.g., bulk carriers, small ships), parameter adjustments and further validation would be required.
3. Lack of microscopic operational considerations: Our method focuses on macro-level route planning for fleets and does not address micro-level operational factors such as collision avoidance or individual ship maneuvering.
4. Lack of real-time adaptation: The method relies solely on historical AIS data and does not incorporate real-time AIS updates, limiting its ability to adapt dynamically to sudden route changes.

Future research directions in the field of data-driven routing optimization for the ship fleet could explore several promising areas to further enhance the efficiency and sustainability of maritime logistics:

1. Data sharing among different shipping companies: Creating a centralized, secure platform where companies can share real-time and historical routing data, traffic conditions, port statuses, and other relevant maritime information is critical for improve the efficiency of maritime transport. The shared data could enhance the collective understanding of maritime operations, allowing for more accurate predictions of port congestion, optimal route adjustments, and better management of fleet capacities. Furthermore, this research could address the challenges of data privacy, economic competition, and the integration of heterogeneous data sources.
2. Collaborative routing: Collaborative routing approaches allow multiple ships or fleets to coordinate their routes in real time. Ships can not only receive optimized routes based on their individual parameters but also adjust their courses in harmony with nearby ships to maximize overall network efficiency.
3. Integration of predictive analytics: Develop predictive models that can forecast potential disruptions in shipping routes, such as severe congestion that may affect maritime corridors. By integrating these forecasts into the routing algorithms, shipping companies can proactively reroute ships to avoid delays and reduce risk.
4. Microscopic operational modeling: Future work can integrate microscopic operational considerations such as ship maneuvering and collision avoidance protocols to ensure feasible and safe execution in congested or complex maritime environments.
5. Real-time transport network updates: Integrate real-time AIS streams with incremental density-based clustering to update waypoint clusters online and trigger localized network updates without full re-clustering. By continuously incorporating real-time traffic shifts caused by weather, incidents, or port congestion, the transportation network model can remain current, facilitating dynamic rerouting and enhancing operational resilience and efficiency.

CRediT authorship contribution statement

Yimeng Zhang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization; **Liang Huang:** Writing – review & editing, Supervision, Funding acquisition, Data curation, Conceptualization; **Jiaci Wang:** Writing – review & editing, Software; **He Lin:** Writing – review & editing; **Shuyang Zhu:** Writing – review & editing; **Mi Gan:** Supervision, Data curation; **Xiaobo Liu:** Supervision, Data curation; **Ruixue Ai:** Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by National Natural Science Foundation of China (no 42471497, 52402521, 52232011, 52372306), Sichuan Natural Science Foundation (no 25QNJJ3599), Hebei Natural Science Foundation (no G2024105007), Sichuan Science and Technology Program (no 2025HJPJ0011), Fundamental Research Funds for the Central Universities (no 2682025CX056), and Tianfu Emei Young Talents Program of Sichuan Province (no 2948).

Appendix A. Performance comparison between DBSCAN and OPTICS

This appendix details the application of OPTICS to cluster the trajectory feature-point dataset (comprising turning, staying, and boundary crossing points). The detailed parameter tuning process of OPTICS can be found in the online documentation (https://github.com/hellocode1205/Tuning-process-of-OPTICS). The final parameter selections and corresponding clustering results are summarized in Table A.1 and visualized in Fig. A.1, facilitating a direct comparison with the DBSCAN outcomes.

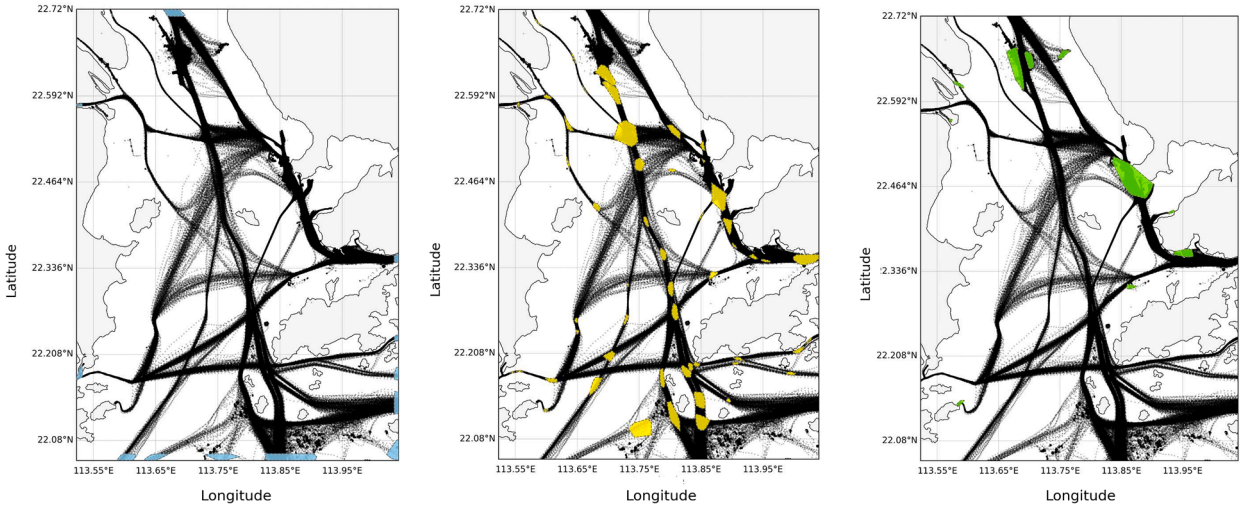
The difference in cluster count for turning points (DBSCAN: 48, OPTICS: 55) stems from OPTICS' superior ability to handle heterogeneous data with varying densities, evident in the uneven distribution of points in Fig. A.1(b). In contrast, DBSCAN relies on a fixed global *eps* value, which limits its capacity to detect clusters of highly divergent sizes and densities and causes it to merge some finer clusters identified by OPTICS. Nevertheless, both methods achieve robust clustering performance under their respective optimal parameters, as quantified in Table A.1.

Notably, different clustering algorithms yield varying numbers of clusters, resulting in transport networks with distinct topologies. To evaluate the downstream impact of these network topology differences on route planning, we use networks generated by DBSCAN and OPTICS as input to our route planning algorithm. Performance is assessed across multiple scenarios using metrics including total cost, delay cost, and service rate, with the results detailed in Table A.2. The results indicate that the differences between the

Table A.1
Clustering results of trajectory feature points.

Method	Feature	Parameters	Clusters	Noise rate	Silh.	DBI
OPTICS	boundary crossing	$\xi = 0.07$, $min\ samples = 43$	14	0.064	0.923	0.177
	turning	$\xi = 0.05$, $min\ samples = 50$	55	0.232	0.869	0.247
	staying	$\xi = 0.09$, $min\ samples = 30$	10	0.291	0.854	0.291
DBSCAN	boundary crossing	$eps = 800$, $min\ samples = 14$	14	0.0019	0.925	0.139
	turning	$eps = 1400$, $min\ samples = 42$	48	0.202	0.877	0.125
	staying	$eps = 1800$, $min\ samples = 90$	9	0.122	0.848	0.183

Clusters: the number of clusters; Silh: the Silhouette coefficient; DBI: the Davies-Bouldin Index.



(a) cluster of boundary crossing points (blue) (b) cluster of turning points (yellow) (c) cluster of staying points (green)

Fig. A.1. Clustering results of trajectory point features (boundary crossing, staying, turning) using OPTICS.

Table A.2
Comparison results under DBSCAN- and OPTICS-based transport networks.

	Total cost (CNY)			Delay penalty (CNY)			Service rate (%)	
$\|O\|$	DBSCAN	OPTICS	Gap (%)	DBSCAN	OPTICS	Gap (%)	DBSCAN	OPTICS
5	3608	3608	0.0	0	0	0.0	100	100
10	7169	7203	–0.5	0	0	0.0	100	100
20	17,682	16,994	4.1	0	0	0.0	100	100
30	29,143	29,020	0.4	1244	1668	–25.4	100	100
50	65,703	65,489	0.3	19441	18528	4.9	100	100
100	164563	179614	-8.4	61664	74010	-16.7	100	100

$|O|$: number of orders; Gap (%): (DBSCAN – OPTICS)/OPTICS × 100 %; Computation time: best solution time; Service rate: proportion of served orders.

two network structures lead to modest variations in route planning performance. This demonstrates the robustness of the routing algorithm to underlying topological changes and confirms that both clustering methods are functionally effective in this application.

Appendix B. Parameter determination

B.1 Turning-point algorithm tuning

Parameter tuning for the turning-point identification algorithm is conducted using two evaluation metrics, N_{all} and Accuracy.

We compare our approach with two widely used baselines: the threshold-based method (Varlamis et al., 2021) and the DP compression algorithm (Lu et al., 2020; Arguedas et al., 2017). Fig. B.1(a) illustrates the test (blue dashed polygon) and validation (red polygon) regions, with the comparison results summarized in Table B.1.

Yellow markers in the figures denote turning points identified under optimal parameter settings. The threshold-based method attains its highest accuracy of 36.70 % at a 2° threshold, though with the largest number of candidate points (Fig. B.1(b)). By contrast, the DP compression method achieves 58.45 % accuracy at 1 km, reflecting effective noise reduction (Fig. B.1(c)). The proposed method, based on cumulative turning features, achieves the highest accuracy of 59.45 % at a cumulative angle threshold of 10° (2°×5), as shown in Fig. B.1(d). Compared with the baselines, it produces fewer redundant turning points. This reduction is critical because the number and quality of candidate points directly influence cluster shapes and, ultimately, the extracted route network.

Based on these findings, the per-step angle threshold α_T is set to 2°, consistent with the optimal baseline setting, and the accumulation window is fixed at $N_T = 5$, corresponding to the 10° cumulative angle.

B.2 Heading-change threshold selection

This study evaluates the effect of different heading-change thresholds (3°/min, 5°/min, 7°/min, and 9°/min) on turning-point identification. As shown in Table B.2, the 3°/min threshold attains the highest accuracy (36.70 %) but produces an excessive number of candidate points (124,195), introducing noise. The 5°/min threshold achieves slightly lower accuracy (34.08 %) while substantially reducing candidates (59,353), offering a more reliable balance. Accordingly, 5°/min is adopted as the optimal heading-change threshold.

Table B.1
Comparison of different methods and parameters.

Method	Parameter	N_{all}	Accuracy (%)
DP compression method	0.5 km	51,091	53.26
	0.75 km	43,274	57.99
	1 km	**37274**	**58.45**
	1.25 km	29,617	56.77
	1.5 km	27,868	55.78
Threshold-based method	**2°**	**124195**	**36.70**
	4°	59,353	34.08
	6°	36,169	31.62
	8°	24,968	29.73
The proposed approach	**10°(2°×5)**	**16381**	**59.45**
	12°(2°×6)	10,583	51.86

N_{all}: the number of candidate turning points detected; Accuracy: the proportion of valid turning points within the validation region.

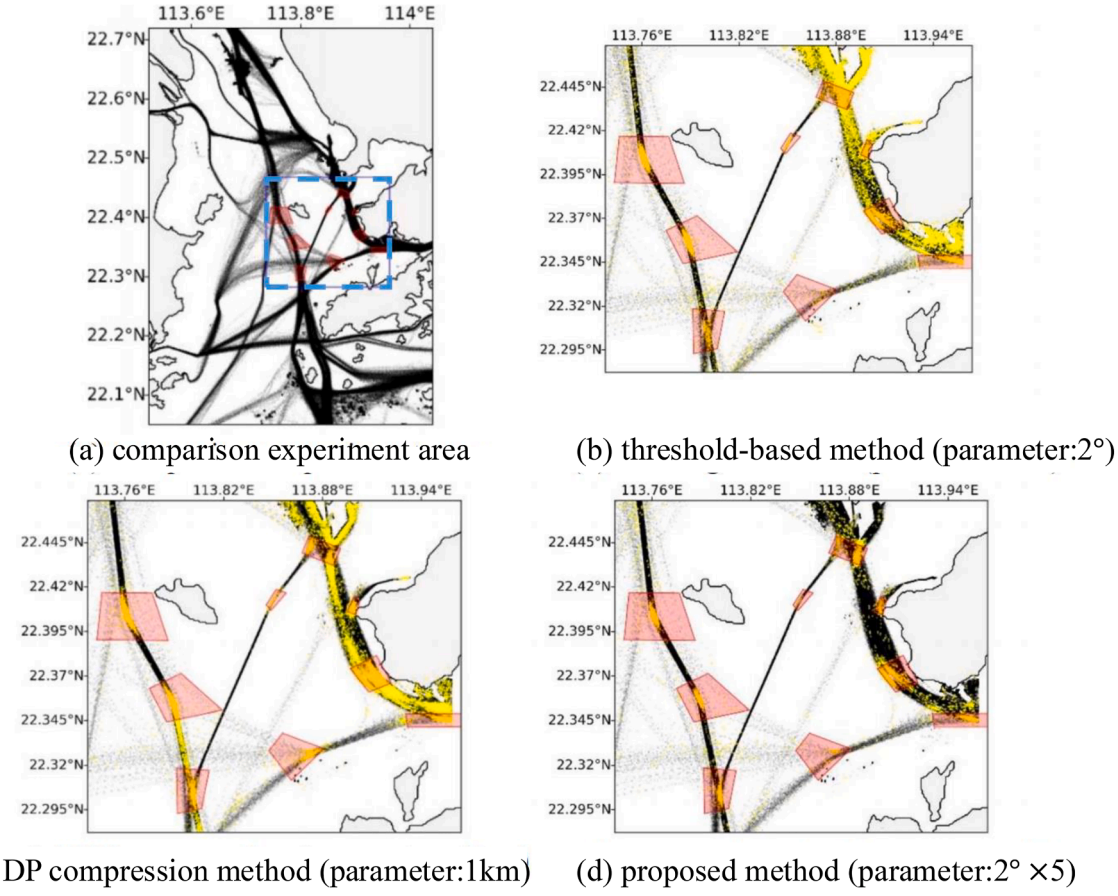


Fig. B.1. Comparison between baseline and proposed methods.

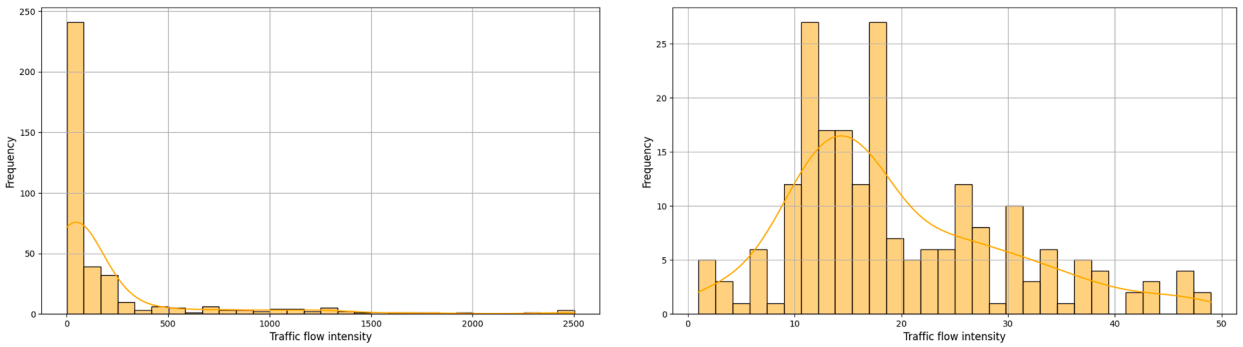
Table B.2
Impact of heading change threshold on turning point identification.

heading change threshold	turning points	accuracy (%)
3°/min	124,195	36.70
5°/min	**59,353**	**34.08**
7°/min	36,169	31.62
9°/min	24,968	29.73

B.3 Determination of traffic flow threshold

A threshold approximation method is applied to determine effective connections between nodes in the route network using traffic flow data. All non-zero flow values are first extracted from the weighted adjacency matrix $(a_{ij})_{N\times N}$, excluding node pairs without actual connections. Reasonable lower and upper bounds are then defined from the value distribution, within which candidate thresholds are evaluated against charted routes. This iterative process progressively narrows the threshold range, filtering out noise while preserving meaningful connections.

Fig. B.2(a) presents the distribution of non-zero traffic flow values, while Fig. B.2(b) shows that most values fall between 10 and 20. Examination of candidate thresholds (n_f = 10-20) indicates that $n_f = 10$ best filters low-frequency noise while maintaining the majority of meaningful routes. Accordingly, $n_f = 10$ is adopted for the study area. This threshold, however, should be calibrated to the statistical characteristics and operational realities of the local maritime transport network.



(a) distribution of non-zero traffic flow values extracted from the adjacency matrix

(b) variation in the number of effective connections under candidate thresholds

Fig. B.2. Distribution of traffic flow values extracted from the adjacency matrix.

Appendix C. Clustering method comparison and parameter selection

C.1 Comparative analysis of clustering methods

This appendix compares several density-based clustering methods for AIS trajectory data to evaluate whether advanced approaches offer tangible benefits in our application. Two representative datasets are used: dataset 1, characterized by high-density trajectory points, and dataset 2, characterized by substantial density variation and noise. Results are summarized in Table C.1.

Across both datasets, DBSCAN achieves competitive performance, with low noise ratios and validity scores comparable to those of more specialized methods. Alternative methods show minor improvements in certain metrics, but the overall cluster structures remain consistent. On dataset 2, DBSCAN's global density threshold (*eps*) yields slightly coarser segmentation and fewer clusters, yet the key topological features are preserved and validity indices remain within acceptable ranges. Therefore, DBSCAN is adopted for clustering feature-point datasets due to its effectiveness and reliability.

C.2 Parameter selection for DBSCAN

The silhouette coefficient and noise ratio are used as the primary evaluation criteria for DBSCAN, supplemented by the Davies-Bouldin Index (DBI). High silhouette values and low noise ratios and DBI scores indicate robust and well-separated clustering structures (Monshizadeh et al., 2022; Gaido, 2023; Wang et al., 2025). To reduce threshold subjectivity, empirical distributions of these metrics across all parameter configurations are analyzed. Thresholds are set at the top 10 % for silhouette and the lowest 25 % for noise ratio and DBI. Configurations meeting these criteria are then ranked using the composite score described below.

$$\text{Score} = \frac{1}{3}\left(\hat{S} + (1 - \hat{N}) + (1 - \hat{D})\right)$$

where $\hat{S}$, $\hat{N}$, and $\hat{D}$ are the min-max normalized silhouette, noise ratio, and DBI, with $\hat{N}$ and $\hat{D}$ inverted to align with the silhouette score.

For the boundary-crossing point set ($eps \in [100, 1000]$ m, $min\ samples \in [2, 20]$), the silhouette threshold is set at 0.924 (top 10 %), while the thresholds for the noise ratio and DBI are set at 0.003 and 0.15 (lowest 25 %), respectively (see Fig. C.1). Among configurations satisfying silhouette > 0.924, noise ratio < 0.003, and DBI < 0.15, the parameter set $eps = 800$ and $min\ samples = 14$ achieves a composite score of 0.656, ranking it among the top candidates.

Table C.1
Clustering results for different methods.

Exp. ID	Method	Parameters	Noise	K	DBI	Silh
dataset 1	DBSCAN	eps = 0.8, *min samples* = 18	0.0027	14	0.1340	0.9244
	HDBSCAN	min cluster size = 60	0.0038	12	0.1714	0.9364
	VDBSCAN	eps ∈ [0.5, 1, 1.5], *min samples* = 16	0.0026	15	0.1758	0.9245
	OPTICS	ξ = 0.07, *min samples* = 45	0.0639	14	0.1773	0.9205
dataset 2	DBSCAN	eps = 0.4, *min samples* = 12	0.1921	19	0.2947	0.8089
	HDBSCAN	min cluster size = 10	0.2115	21	0.5069	0.8328
	VDBSCAN	eps ∈ [0.3, 0.7, 1.3], *min samples* = 18	0.1595	20	0.3906	0.8218
	OPTICS	ξ = 0.05, *min samples* = 20	0.4132	21	0.3525	0.8526

Noise: Noise ratio; K: Number of clusters; DBI: Davies-Bouldin Index; Silh: Silhouette score; eps: distance threshold in DBSCAN and VDBSCAN (in km).

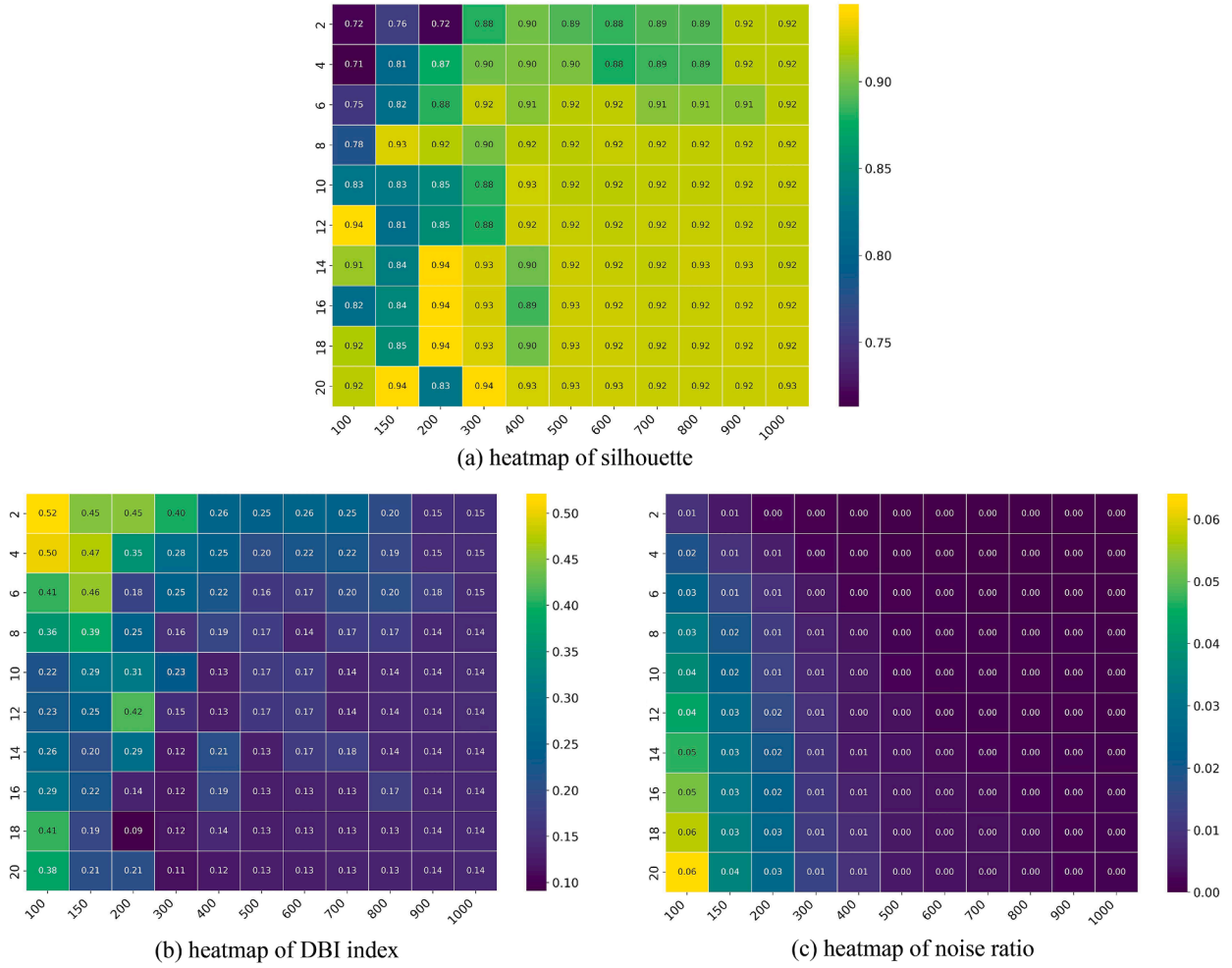


Fig. C.1. Performance metrics for boundary-crossing points (x-axis: *min samples*; y-axis: *eps* (m)).

For turning points ($eps \in [1000, 2000]$ m, $min\ samples \in [2, 92]$), thresholds are silhouette > 0.854, noise < 0.293, and DBI < 0.2334 (see Fig. C.2). The best configuration is $eps = 1400$ and $min\ samples = 42$, meets all criteria with a composite score of 0.754.

Lastly, for staying points ($eps \in [1000, 3000]$ m, $min\ samples \in [10, 100]$), the applied thresholds are silhouette > 0.8248, DBI < 0.2773, and noise ratio < 0.1319 (see Fig. C.3). The parameter set $eps = 1800$, $min\ samples = 90$ meets all criteria, achieving a top-ranked composite score of 0.760.

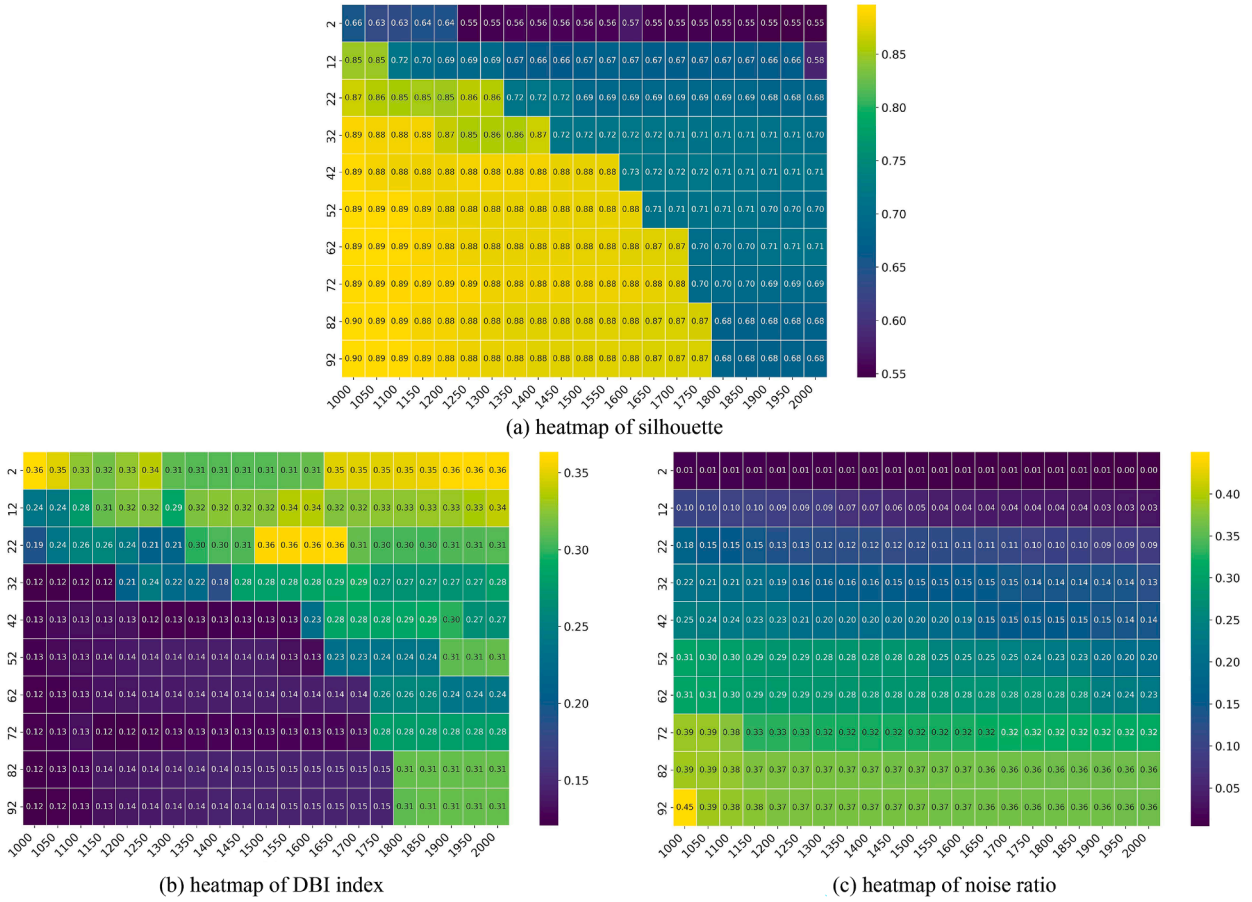


Fig. C.2. Performance metrics for turning points (x-axis: *min samples*; y-axis: *eps* (m)).

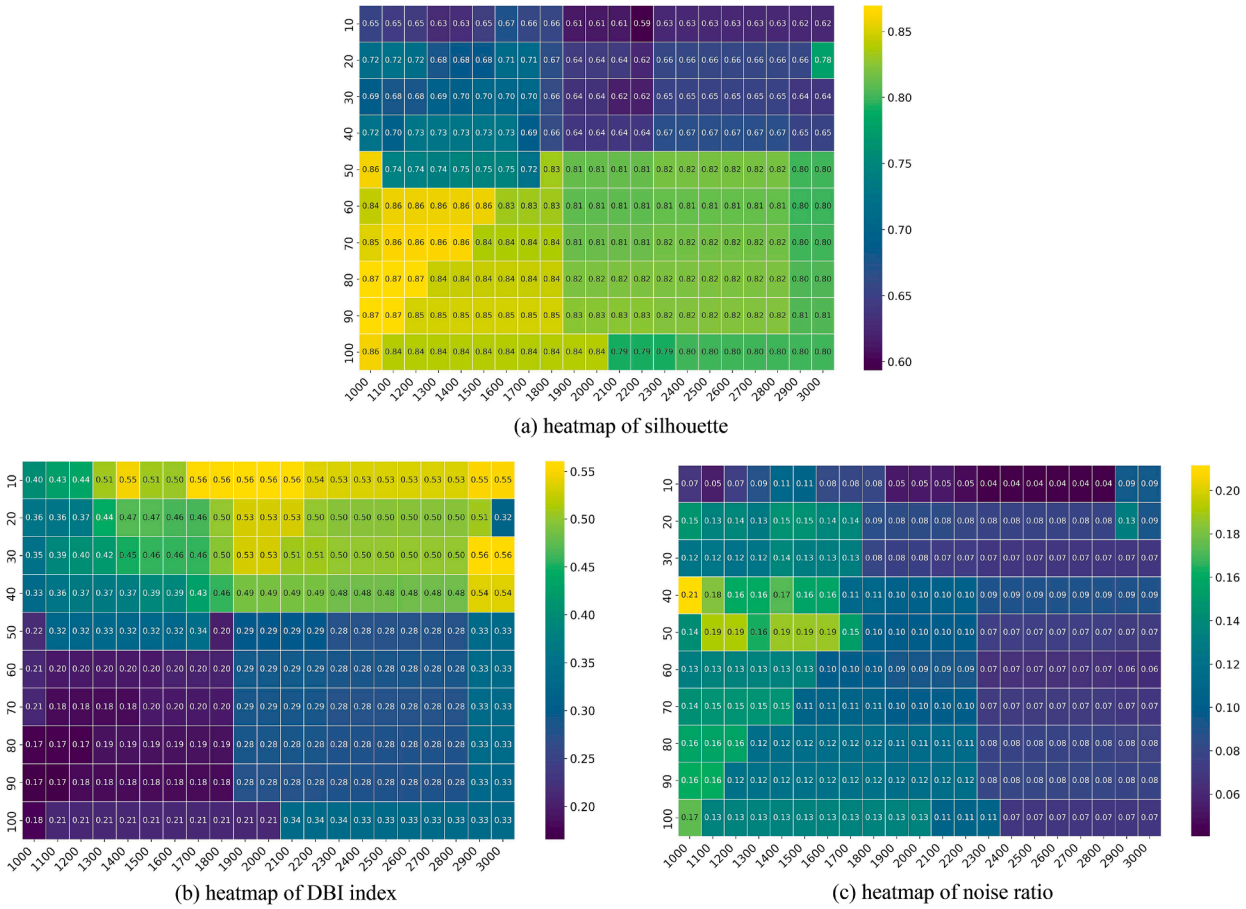


Fig. C.3. Performance metrics for staying points (x-axis: *min samples*; y-axis: *eps* (m)).

References

Abebe, M., Shin, Y., Noh, Y., Lee, S., Lee, I., 2020. Machine learning approaches for ship speed prediction towards energy efficient shipping. Appl. Sci. 10 (7), 2325.

Arguedas, V.F., Pallotta, G., Vespe, M., 2017. Maritime traffic networks: from historical positioning data to unsupervised maritime traffic monitoring. IEEE Trans. Intell. Transp. Syst. 19 (3), 722–732.

Barber, C.B., Dobkin, D.P., Huhdanpaa, H., 1996. The quickhull algorithm for convex hulls. ACM Transact. Math. Softw. (TOMS) 22 (4), 469–483.

Bomberger, N.A., Waxman, A.M., Rhodes, B.J., Sheldon, N.A., 2007. A new approach to higher-level information fusion using associative learning in semantic networks of spiking neurons. Inform. Fusion 8 (3), 227–251.

Cai, J., Chen, G., Lützen, M., Rytter, N. G.M., 2021. A practical AIS-based route library for voyage planning at the pre-fixture stage. Ocean Eng. 236, 109478.

Chen, C., Demir, E., Huang, Y., 2021. An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots. Eur. J. Oper. Res. 294 (3), 1164–1180.

Chian Tan, W., Hian Chua, K., Wu, Y., 2021. The automated risk estimation for the navigation of autonomous ships by learning with navigation feature. Int. J. Comput. Methods 18 (03), 2041003.

Chu, H., Zhang, W., Bai, P., Chen, Y., 2023. Data-driven optimization for last-mile delivery. Compl. Intell. Syst. 9 (3), 2271–2284.

Cong, L., Zhang, H., Wang, P., Chu, C., Wang, J., 2024. Impact of the russia–ukraine conflict on global marine network based on massive vessel trajectories. Remote Sens. (Basel) 16 (8), 1329.

Corvino, M., Daffinà, F., Francalanci, C., Giacomazzi, P., Magliani, M., Ravanelli, P., Stahl, T., 2025. A methodology to extract geo-referenced standard routes from AIS data. arXiv preprint arXiv:2503.22734.

Cullinane, K., Haralambides, H., 2021. Global trends in maritime and port economics: the COVID-19 pandemic and beyond. Maritime Econ. Logist. 23, 369–380.

Danloup, N., Allaoui, H., Goncalves, G., 2018. A comparison of two meta-heuristics for the pickup and delivery problem with transshipment. Comput. Oper. Res. 100, 155–171.

Dobrkovic, A., Iacob, M.-E., van Hillegersberg, J., 2018. Maritime pattern extraction and route reconstruction from incomplete AIS data. Int. J. Data Sci. Analyt. 5, 111–136.

Dobrkovic, A., Iacob, M.-E., Van Hillegersberg, J., 2015. Using machine learning for unsupervised maritime waypoint discovery from streaming AIS data. In: Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business, pp. 1–8.

Duan, K., Li, Q., Liu, S., Liu, Y., Wang, S., Li, S., Wang, X., Ma, N., Ma, Y., 2024. Ais-based operational phase identification using progressive ablation feature selection with machine learning for improving ship emission estimates. J. Air Waste Manage. Assoc. 74 (2), 100–115.

Edelkamp, S., Schrödl, S., 2003. Route planning and map inference with global positioning traces. In: Computer Science in Perspective: Essays Dedicated to Thomas Ottmann. Springer, pp. 128–151.

Englert, M., Röglin, H., Vöcking, B., 2014. Worst case and probabilistic analysis of the 2-opt algorithm for the TSP. Algorithmica 68 (1), 190–264.

Filipiak, D., Wecel, K., Milena, S., Michalak, M., Abramowicz, W., 2020. Extracting maritime traffic networks from AIS data using evolutionary algorithm. Bus. Inform. Syst. Eng. 62 (5), 435–450.
François, V., Arda, Y., Crama, Y., 2019. Adaptive large neighborhood search for multitrip vehicle routing with time windows. Transp. Sci. 53 (6), 1706–1730.
Gaido, M., 2023. Distributed silhouette algorithm: Evaluating clustering on big data. arXiv preprint arXiv:2303.14102.
Ghilas, V., Demir, E., Van Woensel, T., 2016. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. Comput. Oper. Res. 72, 12–30.
Gkerekos, C., Lazakis, I., 2020. A novel, data-driven heuristic framework for vessel weather routing. Ocean Eng. 197, 106887.
Gschwind, T., Drexl, M., 2019. Adaptive large neighborhood search with a constant-time feasibility test for the dial-a-ride problem. Transp. Sci. 53 (2), 480–491.
Guo, S., Zhang, X., Zheng, Y., Du, Y., 2020. An autonomous path planning model for unmanned ships based on deep reinforcement learning. Sensors 20 (2), 426.
Gurobi, 2025. Gurobi optimization. https://www.gurobi.com.
Han, P., Yang, X., 2020. Big data-driven automatic generation of ship route planning in complex maritime environments. Acta Oceanolog. Sin. 39 (8), 113–120.
Huang, C., Qi, X., Zheng, J., Zhu, R., Shen, J., 2023. A maritime traffic route extraction method based on density-based spatial clustering of applications with noise for multi-dimensional data. Ocean Eng. 268, 113036.
Huang, L., Chen, X., Zhang, W., 2021. Identification and classification of ship stop behaviors based on trajectory features. J. Traffic Transp. Eng. (Chin. Edit.) 21 (5), 109–139. (in Chinese).
Huang, L., Wan, C., Wen, Y., Song, R., van Gelder, P., 2024. Generation and application of maritime route networks: overview and future research directions. IEEE Trans. Intell. Transp. Syst. 26 (1), 620–637.
Huang, L., Wen, Y., Guo, W., Zhu, X., Zhou, C., Zhang, F., Zhu, M., 2020. Mobility pattern analysis of ship trajectories based on semantic transformation and topic model. Ocean Eng. 201, 107092.
Jeong, M.-G., Lee, E.-B., Lee, M., Jung, J.-Y., 2019. Multi-criteria route planning with risk contour map for smart navigation. Ocean Eng. 172, 72–85.
Jin, X., Yang, Y., Qiu, X., 2017. Framework of frequently trajectory extraction from AIS data. In: Proceedings of the 2017 the 7th International Conference on Computer Engineering and Networks, pp. 22–23.
Kaluza, P., Kölzsch, A., Gastner, M.T., Blasius, B., 2010. The complex network of global cargo ship movements. J. Roy. Soc. Interface 7 (48), 1093–1103.
Kanrak, M., Nguyen, H.O., Du, Y., 2019. Maritime transport network analysis: a critical review of analytical methods and applications. J. Int. Logist. Trade 17 (4), 113–122.
Kisialiou, Y., Gribkovskaia, I., Laporte, G., 2018. Robust supply vessel routing and scheduling. Transp. Res. Part C: Emerg. Technolog. 90, 366–378.
Kisialiou, Y., Gribkovskaia, I., Laporte, G., 2019. Supply vessel routing and scheduling under uncertain demand. Transp. Res. Part C: Emerg. Technolog. 104, 305–316.
Kontopoulos, I., Varlamis, I., Tserpes, K., 2021. A distributed framework for extracting maritime traffic patterns. Int. J. Geograph. Inform. Sci. 35 (4), 767–792.
Lai, Y., Yang, F., Meng, G., Lu, W., 2022. Data-driven flexible vehicle scheduling and route optimization. IEEE Trans. Intell. Transp. Syst. 23 (12), 23099–23113.
Larose, D.T., Larose, C.D., 2014. K-nearest neighbor algorithm. In: Discovering Knowledge in Data: An Introduction to Data Mining. Wiley Data and Cybersecurity, pp. 149–164.
Lee, J.-S., Lee, H.-T., Cho, I.-S., 2022. Maritime traffic route detection framework based on statistical density analysis from AIS data using a clustering algorithm. IEEE Access 10, 23355–23366.
Lee, J.-S., Son, W.-J., Lee, H.-T., Cho, I.-S., 2020. Verification of novel maritime route extraction using kernel density estimation analysis with automatic identification system data. J. Mar. Sci. Eng. 8 (5), 375.
Lee, W., Cho, S.-W., 2022. Ais trajectories simplification algorithm considering topographic information. Sensors 22 (18), 7036.
Li, G., Liu, M., Zhang, X., Wang, C., Lai, K.-h., Qian, W., 2022. Semantic recognition of ship motion patterns entering and leaving port based on topic model. J. Mar. Sci. Eng. 10 (12), 2012.
Li, H., Jiao, H., Yang, Z., 2023. Ais data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods. Transp. Res. Part E: Logist. Transp. Rev. 175, 103152.
Li, H., Yang, Z., 2023. Incorporation of AIS data-based machine learning into unsupervised route planning for maritime autonomous surface ships. Transp. Res. Part E: Logist. Transp. Rev. 176, 103171.
Liu, F.T., Ting, K.M., Zhou, Z.-H., 2008. Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining. IEEE, pp. 413–422.
Liu, Z., Gao, H., Zhang, M., Yan, R., Liu, J., 2023. A data mining method to extract traffic network for maritime transport management. Ocean Coast. Manage. 239, 106622.
Lu, N., Liang, M., Zheng, R., Liu, R.W., 2020. Historical AIS data-driven unsupervised automatic extraction of directional maritime traffic networks. In: 2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA). IEEE, pp. 7–12.
McKinley, S., Levine, M., 1998. Cubic spline interpolation. College Redwood. 45 (1), 1049–1060.
Monshizadeh, M., Khatri, V., Kantola, R., Yan, Z., 2022. A deep density based and self-determining clustering approach to label unknown traffic. J. Netw. Comput. Applic. 207, 103513.
Mou, N., Liu, C., Zhang, L., Fu, X., Xie, Y., Li, Y., Peng, P., 2018. Spatial pattern and regional relevance analysis of the maritime silk road shipping network. Sustainability 10 (4), 977.
Murray, B., Perera, L.P., 2022. Ship behavior prediction via trajectory extraction-based clustering for maritime situation awareness. J. Ocean Eng. Sci. 7 (1), 1–13.
Naus, K., 2020. Drafting route plan templates for ships on the basis of AIS historical data. J. Navigat. 73 (3), 726–745.
Pallotta, G., Vespe, M., Bryan, K., 2013a. Traffic knowledge discovery from AIS data. In: Proceedings of the 16th International Conference on Information Fusion. IEEE, pp. 1996–2003.
Pallotta, G., Vespe, M., Bryan, K., 2013b. Vessel pattern knowledge discovery from AIS data: a framework for anomaly detection and route prediction. Entropy 15 (6), 2218–2245.
Phu, Q. N.P., Nguyen, V., Do, T., Ngo, T.D., 2018. Measuring crowd collectiveness with trajectory smoothing. In: 2018 1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR). IEEE, pp. 1–6.
Qu, Y., Bard, J.F., 2012. A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. Comput. Oper. Res. 39 (10), 2439–2456.
Rong, H., Teixeira, A.P., Soares, C.G., 2022a. Maritime traffic probabilistic prediction based on ship motion pattern extraction. Reliab. Eng. Syst. Safe. 217, 108061.
Rong, Y., Zhuang, Z., He, Z., Wang, X., 2022b. A maritime traffic network mining method based on massive trajectory data. Electron. (Basel) 11 (7), 987.
Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.
Santos, V. G.M., de Carvalho, M. A.M., 2021. Tailored heuristics in adaptive large neighborhood search applied to the cutwidth minimization problem. Eur. J. Oper. Res. 289 (3), 1056–1066.
Schiffer, M., Walther, G., 2018. An adaptive large neighborhood search for the location-routing problem with intra-route facilities. Transp. Sci. 52 (2), 331–352.
Sirimanne, S.N., Hoffman, J., et al., 2022. Review of maritime transport 2022. In: United Nations Conference on Trade and Development, Geneva, Switzerland.
Tang, C., Wang, H., Zhao, J., Tang, Y., Yan, H., Xiao, Y., 2021. A method for compressing AIS trajectory data based on the adaptive-threshold douglas-peucker algorithm. Ocean Eng. 232, 109041.
Tierney, K., Ehmke, J.F., Campbell, A.M., Müller, D., 2019. Liner shipping single service design problem with arrival time service levels. Flex. Serv. Manufact. J. 31, 620–652.
Utomo, A., Pambudi, G., Hamada, K., et al., 2023. Optimization of shipping routes for container ships from indonesia to the asia-pacific using heuristic algorithms. J. Mar. Sci. Eng. 11 (7), 1360.
Varlamis, I., Kontopoulos, I., Tserpes, K., Etemad, M., Soares, A., Matwin, S., 2021. Building navigation networks from multi-vessel trajectory data. Geoinformatica 25, 69–97.
Vettor, R., Soares, C.G., 2015. Detection and analysis of the main routes of voluntary observing ships in the north atlantic. J. Navigat. 68 (2), 397–410.
Vettor, R., Soares, C.G., 2016. Development of a ship weather routing system. Ocean Eng. 123, 1–14.

Voigt, S., 2025. A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems. Eur. J. Oper. Res. 322 (2), 357–375.
Wang, G., Meng, J., Li, Z., Hesenius, M., Ding, W., Han, Y., Gruhn, V., 2020. Adaptive extraction and refinement of marine lanes from crowdsourced trajectory data. Mob. Netw. Applic. 25, 1392–1404.
Wang, S.-M., Yang, W.-R., Zhuang, Q.-Y., Lin, W.-H., Tian, M.-Y., Su, T.-J., Cheng, J.-C., 2025. Application of three-dimensional hierarchical density-based spatial clustering of applications with noise in ship automatic identification system trajectory-cluster analysis. Appl. Sci. 15 (5), 2621.
Wang, Z., Claramunt, C., Wang, Y., 2019. Extracting global shipping networks from massive historical automatic identification system sensor data: a bottom-up approach. Sensors 19 (15), 3363.
Wen, X., Chen, Q., Yin, Y.-Q., Lau, Y.-y., Dulebenets, M.A., 2024. Multi-objective optimization for ship scheduling with port congestion and environmental considerations. J. Mar. Sci. Eng. 12 (1), 114.
Wen, Y., Sui, Z., Zhou, C., Xiao, C., Chen, Q., Han, D., Zhang, Y., 2020. Automatic ship route design between two ports: a data-driven method. Appl. Ocean Res. 96, 102049.
Wolfinger, D., 2021. A large neighborhood search for the pickup and delivery problem with time windows, split loads and transshipments. Comput. Oper. Res. 126, 105110.
Xiao, Z., Fu, X., Zhang, L., Goh, R. S.M., 2019. Traffic pattern mining and forecasting technologies in maritime traffic service networks: a comprehensive survey. IEEE Trans. Intell. Transp. Syst. 21 (5), 1796–1825.
Yan, Z., Xiao, Y., Cheng, L., He, R., Ruan, X., Zhou, X., Li, M., Bin, R., 2020. Exploring AIS data for intelligent maritime routes extraction. Appl. Ocean Res. 101, 102271.
Yang, D., Wu, L., Wang, S., Jia, H., Li, K.X., 2019. How big data enriches maritime research–a critical review of automatic identification system (AIS) data applications. Transp. Rev. 39 (6), 755–773.
Yitao, W., Lei, Y., Xin, S., 2020. Route mining from satellite-AIS data using density-based clustering algorithm. J. Phys. Conf. Ser. 1616 (1), 012017.
Yoo, S.-L., Onyango, S.O., Kim, J.-S., Kim, K.-I., 2024. Anchor dragging risk estimation strategy from supervised cost-sensitive learning. J. Mar. Sci. Eng. 12 (10), 1817.
Yu, H., Murray, A.T., Fang, Z., Liu, J., Peng, G., Solgi, M., Zhang, W., 2021. Ship path optimization that accounts for geographical traffic characteristics to increase maritime port safety. IEEE Trans. Intell. Transp. Syst. 23 (6), 5765–5776.
Zhang, T., Wang, Z., Wang, P., 2024. A method for compressing AIS trajectory based on the adaptive core threshold difference douglas–peucker algorithm. Sci. Rep. 14 (1), 21408.
Zhang, Y., Atasoy, B., Negenborn, R.R., 2022a. Preference-based multi-objective optimization for synchromodal transport using adaptive large neighborhood search. Transp. Res. Rec. 2676 (3), 71–87.
Zhang, Y., Atasoy, B., Souravlias, D., Negenborn, R.R., 2020. Pickup and delivery problem with transshipment for inland waterway transport. In: 11th International Conference of Computational Logistics, ICCL 2020, Enschede, the Netherlands, September 28–30, 2020, Proceedings 11. Springer, pp. 18–35.
Zhang, Y., Guo, W., Negenborn, R.R., Atasoy, B., 2022b. Synchromodal transport planning with flexible services: mathematical model and heuristic algorithm. Transp. Res. Part C: Emerg. Technolog. 140, 103711.
Zhang, Y., Heinold, A., Meisel, F., Negenborn, R.R., Atasoy, B., 2022c. Collaborative planning for intermodal transport with eco-label preferences. Transp. Res. Part D: Transp. Environ. 112, 103470.
Zhang, Y., Negenborn, R.R., Atasoy, B., 2023. Synchromodal freight transport re-planning under service time uncertainty: an online model-assisted reinforcement learning. Transp. Res. Part C: Emerg. Technolog. 156, 104355.
Zhao, Y., Peng, P., Zhou, J., Wang, Y., 2025. Heuristic algorithm for integrated ship scheduling, routing and stowage problem in multi-vessel roll-on/roll-off shipping. J. Heurist. 31 (1), 15.
Zhu, Z., Zhang, S., Cao, Z., 2025. Exact and heuristic methods for the integrated optimisation of vessel deployment and liner shipping routing schedule. Int. J. Syst. Sci.: Oper. Logist. 12 (1), 2505920.