IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 November 2022, accepted 29 December 2022, date of publication 5 January 2023, date of current version 23 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3234121

APPLIED RESEARCH

Compressing AIS Trajectory Data Based on the Multi-Objective Peak Douglas–Peucker Algorithm

ZHENG ZHOU, YINGJIAN ZHANG, XIAOYU YUAN, AND HONGBO WANG
State Key Laboratory on Integrated Optoelectronics, College of Electronic Science and Engineering, Jilin University, Changchun 130012, China

Corresponding author: Hongbo Wang (wang_hongbo@jlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant u1964202.

ABSTRACT The automatic identification system (AIS) provides a massive database for ocean science. The original AIS data are redundant. Direct use will cause a waste of data storage space and computation costs; hence, data compression must be performed. The Douglas–Peucker algorithm (DP) is an effective trajectory compression algorithm that can well preserve the spatial characteristics of a trajectory but has the following shortcomings: first, it has poor track recovery when compressing multi-turn routes; second, it does not consider the ship speed and heading; and third, it may have the wrong result of the compressed trajectory crossing the obstacle. To address these situations, this study proposes a multi-objective peak DP algorithm (MPDP) that adopts a peak sampling strategy, considers three optimization objectives (spatial characteristics, heading and speed) of trajectory and adds an obstacle detection mechanism to realize a compression algorithm more suitable for curved trajectories. The classical DP algorithm is compared with the MPDP algorithm by simulating trajectory and real trajectory experiments. The results show that the MPDP algorithm optimizes the length loss rate, simultaneous Euclidean distance, and average deviations of the speed and the heading while maintaining a high compression rate similar to that of the DP algorithm. Moreover, it can also successfully avoid obstacles. The optimization effect is most obvious for the multi-turn or hovering trajectory. The optimization rate of length loss, synchronous Euclidean distance, and average deviation of the heading can reach 40%.

INDEX TERMS AIS data, Douglas–Peucker algorithm (DP), multi-objective peak Douglas–Peucker algorithm (MPDP), trajectory compression.

I. INTRODUCTION

Since 2002, the International Convention on the Safety of Life at Sea has required all ships over 300 gross tonnage to be equipped with automatic identification system (AIS) equipment [1]. The AIS is an automatic tracking system that provides identification and location services for ships by exchanging data with neighboring ships, AIS shore stations, and satellites, among other devices. Although AIS systems were originally designed for radar and vessel traffic services, researchers can collect and store the AIS data generated over a certain period of time for a given water area. These data then become a large data source for studying the navigational behavior of ships in that water area.

Over the last decade, researchers have increasingly applied AIS data to the construction work of smart oceans. For example, for ship collision avoidance decision research [2], [3], [4], Shi et al. [5] identified the collision risk based on the ship domain and input the data of successful ship encounters extracted from the AIS dataset into a double-gated recurrent unit neural network to generate a collision avoidance decision for unmanned ships through neural network learning. For ship trajectory prediction research [6], [7], [8], Volkova et al. [9] input the ship positioning information from the AIS data into a neural network for ship trajectory prediction to solve the weak positioning problem of the satellite signals obscured by obstacles in inland waters. For route planning [10], [11], [12], Zhang et al. [13] proposed a new automatic maritime route generation algorithm. In this algorithm, given a set of ship track AIS data, the data are first compressed and clustered,

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Wang.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

and then an ant colony algorithm is used to perform a route search on the clustered clusters and recommend a better route. For maritime logistics and transportation [14], Sungil Kim et al. [15] applied the AIS data to logistics and transportation economics and used real-time AIS ship tracking data in combination with historical ship data to propose an effective method for ship delay detection. For ship emissions, Huang et al. [16] used the AIS data extracted from ships to establish a quantitative model for estimating ship engine emissions under different operating conditions. They further applied spatiotemporal analysis to quantify exhaust ship emissions. In addition to the abovementioned applications, the AIS data are also widely used in fields like ocean current detection [17], [18], [19], waterway traffic [20], [21], [22], environmental noise [23], marine fisheries [24], [25], [26], and marine trade [27], [28].

However, the AIS was originally designed to aid communication in ship traffic services; thus, their usage brings many challenges to scientific research. The most problematic of which is how to store and analyze them [29]. AIS systems transmit information every 2–12 s when the ship is in motion and every 6 min when it is semi-static and static. In other words, the AIS data are sometimes updated much faster than the changes in the ship speed, position, and heading during the voyage. Therefore, the original AIS data contain much redundant information, which is not conducive to subsequent research and calculation.

AIS data compression must be performed to reduce the storage and computational costs of data processing. The commonly used compression algorithms are the Douglas–Peucker (DP) algorithm [30], Bellman algorithm [31], STTrace algorithm [32], sliding window [33], top–down time ratio algorithm (TD-TR), and opening window algorithm (OPW) and its improved algorithms (i.e., opening window time ratio (OPW-TR), and opening window spatiotemporal algorithm (OPW-SP)). The DP algorithm is considered one of the most accurate and effective data compression methods [34]. This view is also shared by Muckell et al. [35], who used the results of GPS data compression experiments to perform a comprehensive comparison of the compression effects of the abovementioned algorithms. The results showed that the DP algorithm outperforms the TD-TR, STTrace, and Bellman algorithms in terms of the algorithmic computing time. The authors also compared the spatial error of the compressed trajectories using simultaneous Euclidean distances and found that the DP algorithm outperforms the TD-TR, OPW-TD, OPW-SP, STTrace, and Bellman algorithms. As regards the DP algorithm-based track compression, Gerben et al. [36] improved the DP algorithm in the direction of better retention of ship stopping and movement information. They also proposed a machine learning framework to analyze the moving object trajectories from ships. The ship trajectory data were clustered, classified, and detected by anomalies. Zhang et al. [37] used the DP algorithm to extract the feature points to simplify the AIS trajectory data and improve the subsequent processing efficiency. Their study mainly discussed the compression threshold of the unique input parameter of the DP algorithm and presented an artificial intelligence-based minimum ship domain evaluation method used as the criteria for determining the compression threshold. The final determination threshold was set to 0.8 times the ship length, at which the DP algorithm compression rate can effectively be balanced with other performance indicators. Li et al. [38] conducted a similar study to implement channel density visualization based on the channel density estimation on a simplified trajectory dataset to ensure a good balance between AIS trajectory simplification and visualization performance. Accordingly, a large number of experiments were conducted to optimally select the appropriate threshold value for the DP algorithm. Zhao et al. [39] proposed a compression method that considered an improved DP algorithm for determining the ship trajectory shape using the trajectory information of the track points. The ship trajectory was divided into straight routes and turning routes. Different retention strategies were used to retain fewer track points in the straight line section and more track points in the turning section. The experiments were conducted based on the AIS data in Zhoushan Islands, China. Compared with the traditional DP algorithm, the method significantly reduces the compression time and shows a better performance under high compression intensity. The compression times of the DP and improved DP algorithms were verified as better than those of the OPW, TD-TR, and OPW-TR algorithms. In 2019, Tang et al. [40] analyzed the ship motion behavior at each track point in terms of the motion velocity, stop point, and motion direction and proposed the trajectory partition method based on combined motion features (TPMF) for trajectory partition. In the TPMF, the change points, where nodes move with significant velocity changes, are first extracted. Next, the stop points are extracted by detecting the node velocity changes. Finally, the DP algorithm is applied to partition the sub-trajectories according to the extracted feature points (i.e., change and stop points). Simulations were performed on the Geolife trajectory dataset. The simulation results showed that the TPMF achieves a good balance between the simplification rate and the trajectory partitioning error while shortening the running time. Liu et al. [41] proposed the adaptive DP algorithm (ADP) with automatic thresholding for AIS-based vessel trajectory compression, segmentation framework, and average distance improvements and additions to the original DP algorithm. Tang et al. [42] also addressed the adaptive DP algorithm and proposed another adaptive-threshold DP algorithm. The investigation of the changes in the critical threshold during the compression process showed that most of the critical threshold changes had obvious inflection points. Hence, the critical threshold change rate is determined by the mathematical statistics of the track, which no longer relies on the ship static information. The advantages of matrix operations and the point reduction method were used to improve the algorithm's computational efficiency. Zhong et al. [43]

proposed a data compression algorithm based on the spatiotemporal characteristics of the AIS data for velocity change or stopping points. This algorithm compresses the trajectory data with orientation and velocity differences and time intervals as its parameters.

Most of the research and discussion on the DP algorithm have focused on reducing the time of algorithm operation and setting or adapting the appropriate threshold value. However, the compression effect of the DP algorithm is not satisfactory in the application of some complex trajectories, such as multi-turn and circling routes. There is still room for improvement in preserving the spatial feature information of the trajectory data. Although some scholars have started paying attention to other ship maneuvering information like heading and speed in addition to the spatial feature information of the trajectory, only a few have discussed the DP algorithm to improve the heading and speed recovery while maintaining a high compression rate. As the first work that must be performed in AIS data-based research, data compression must consider map information; otherwise, the simplified trajectory may intersect with obstacles, consequently affecting the results' correctness in subsequent studies. Therefore, the MPDP is proposed here based on the DP algorithm. The main contributions of the proposed algorithm are as follows:

1) The algorithm adopts a peak retention strategy to improve the spatial feature retention effect of multi-turn trajectories, such as circling routes, while reducing the number of compression layers and improving the compression efficiency.
2) The algorithm uses a multi-objective fitness function to comprehensively consider the spatial characteristics of routes, heading, and speed. It improves the recovery effect of track heading and speed while maintaining a high compression rate.
3) The algorithm considers the map information and performs secondary preservation of waypoints when the compressed track crosses obstacle situations, thereby successfully avoiding obstacles and ensuring the correctness of the compressed track.

The rest of this paper is organized as follows: Section II introduces the trajectory compression method that includes the principle and implementation of the classical DP algorithm; Section III introduces the proposed improved DP algorithm, namely the multi-objective peak DP algorithm; Section IV presents the compression simulation experiments using real trajectories with simulated trajectories, in which the compression results of the MPDP and classical DP algorithms are compared and analyzed in terms of six performance metrics (i.e., compression time, compression rate, length loss rate, synchronous Euclidean distance, average velocity deviation, and heading average deviation); and Section V discusses the research conclusions and future research directions.

II. TRAJECTORY COMPRESSION METHOD

A. PREPROCESSING

Using the DP algorithm requires the calculation of the distance between a point and a straight line. The original AIS data of the ship trajectory coordinate information are established based on the geographic coordinate system. Calculating the distance between two points on a sphere is complicated. Directly calculating the distance between the trajectory point and the straight line is difficult, as well. Therefore, we convert the original geographic coordinates (λ, φ) of the trajectory points into coordinates (x, y) of the Mercator projection to facilitate the data calculation in the DP algorithm:

$$\begin{aligned} x &= r_0 \times \lambda, \\ y &= r_0 \times q, \\ r_0 &= \frac{l \times \cos(\varphi_0)}{\sqrt{1 - (e^2 \times \sin^2(\varphi_0))}}, \\ q &= \ln\left(\tan(\frac{\pi}{4} + \frac{\varphi}{2}) \times \left(\frac{1 - e \times \sin\varphi}{1 + e \times \sin\varphi}\right)^{e/2}\right), \end{aligned} \tag{1}$$

where λ and φ denote the longitude and the latitude of the track point, respectively; x and y denote the horizontal and vertical coordinates of the track point, respectively; φ_0 denotes the standard latitude in the Mercator projection; l denotes the long radius of the Earth's ellipsoidal sphere; e denotes the first eccentricity of the Earth's ellipsoidal sphere; r_0 denotes the radius of the parallel circle of standard latitude; and q denotes the equatorial latitude.

After the Mercator projection transformation, the distance *Distance* from the point to the line is calculated by the vector method [39], [44] as

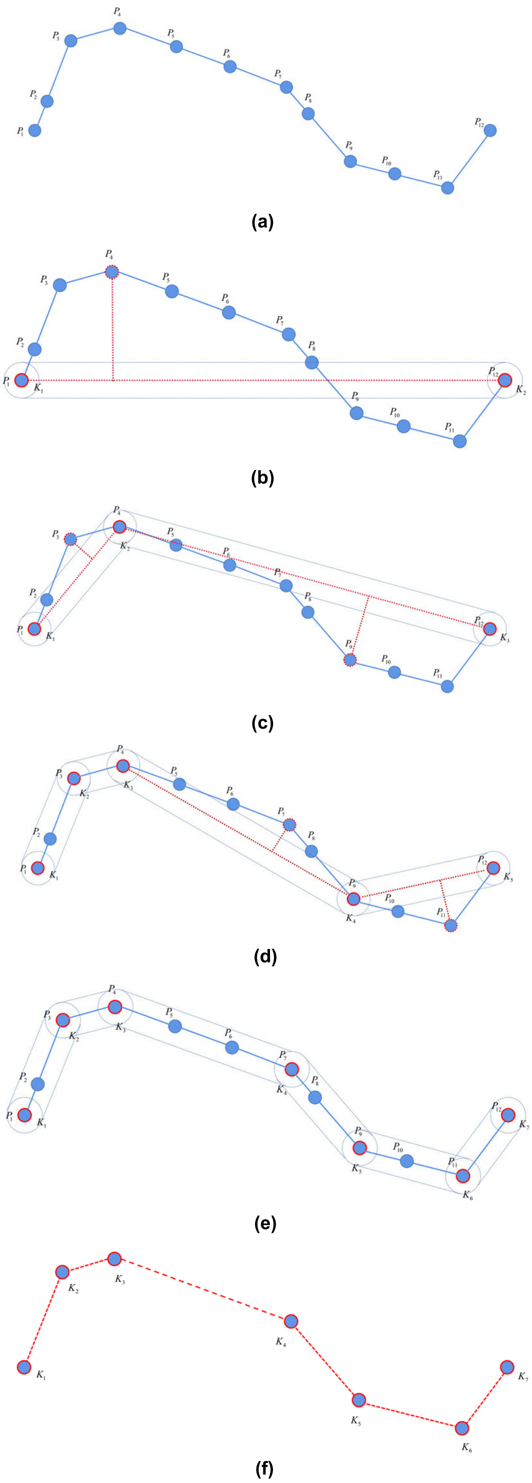
$$Distance = \frac{|a \times b|}{|a|}, \tag{2}$$

where a denotes the vector from the start of the line segment to the end of the line segment, and b denotes the vector from the start of the line segment to the target point.

B. CLASSICAL DP ALGORITHM

The DP algorithm is the most widely used trajectory compression algorithm, with the main task of extracting a key set of waypoints $\boldsymbol{KS} = \left\{K_1, \ldots K_j, \ldots K_M\right\} j \in (1, M)$ that can reflect the main morphological features of the original trajectory from the set of waypoints $\boldsymbol{OS} = \{P_1, \ldots P_i, \ldots P_N\}\, i \in (1, N)$ of the original trajectory. Fig. 1 shows the compression process of the DP algorithm. The solid line indicates the original trajectory, while the long dashed line depicts the compressed trajectory. The compression steps of the DP algorithm are as follows:

- Step 1: Set the compression threshold ε $(\varepsilon > 0)$;
- Step 2: Add the starting and ending points of set $\boldsymbol{OS}$ to set $\boldsymbol{KS}$;
- Step 3: Update set $\boldsymbol{KS}$:
- Step 3-1: Divide set $\boldsymbol{OS}$ into $M - 1$ subsets by taking the points in set $\boldsymbol{KS}$ as the splitting points. Let the



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 1. Schematic of the Douglas–Peucker algorithm. (a) Origin trajectory. (b) Finding the farthest point. (c) Judging and splitting. (d) Judging and retaining. (e) All points meet the threshold. (f) Simplified trajectory.

subsets corresponding to two adjacent points K_j to K_{j+1} be $\boldsymbol{OS\_Sub}_j$. Take the line between K_j and K_{j+1} as the baseline using Eq. (1) to calculate the distance of each point in subset $\boldsymbol{OS\_Sub}_j$ to the baseline set $\boldsymbol{DS}$.

- Step 3-2: Take the maximum value $d_{\max}$ in the distance set $\boldsymbol{DS}$ corresponding to the index $Index$. If $d_{\max} \geq \varepsilon$, then point $\boldsymbol{OS\_Sub}_j\,[Index]$ of the maximum distance taken will be added to the set of key waypoints $\boldsymbol{KS}$.
- Step 4: Assume that the maximum value of all subsets $\boldsymbol{OS\_Sub}$ is $D_{\max}$. If $D_{\max} \geq \varepsilon$, then repeat Step 3; otherwise, end the loop.

C. COMPRESSION PERFORMANCE

This section defines the six performance metrics for measuring compression effectiveness.

Compression time T_c: run time of compressing the trajectory using the DP algorithm.

Compression ratio R_c: ratio of the number of trajectory points discarded in the compression process to the number of original trajectory points expressed as

$$R_c = \frac{n}{m}, \tag{3}$$

where n is the number of eliminated trajectory points, and m is the total number of original trajectory points.

Length loss rate R_l: ratio of the difference between the original trajectory and recovered lengths of the compressed waypoint to the original trajectory length expressed as

$$R_l = \frac{\sum_{i=1}^{N-1} \overline{P_iP_{i+1}} - \sum_{j=1}^{M-1} \overline{K_jK_{j+1}}}{\sum_{i=1}^{N-1} \overline{P_iP_{i+1}}}, \tag{4}$$

where N denotes the number of original trajectory points of each ship; $\overline{P_iP_{i+1}}$ denotes the distance between two adjacent points of the original trajectory; M denotes the number of retained trajectory points of each ship after compression; and $\overline{K_jK_{j+1}}$ denotes the distance between two adjacent points of the compressed trajectory.

Simultaneous Euclidean distance D_{SE}: let $P_i = (x_i, y_i, t_i)$ be a point in the original trajectory $\boldsymbol{OS}$, $K_j = (x_j, y_j, t_j)$ and $K_{j+1} = (x_{j+1}, y_{j+1}, t_{j+1})$ be the points in the compressed trajectory that lie before and after P_i $(t_j \leq t_i \leq t_{j+1})$, respectively, in time order, and $P'_i = (x'_i, y'_i, t_i)$ be the waypoint after linear recovery using the adjacent points in the compressed trajectory set $\boldsymbol{KS}$:

$$\begin{aligned} x'_i &= x_j + \left(\frac{t_i - t_j}{t'_{j+1} - t'_j}\right)(x_{j+1} - x_j), \\ y'_i &= y_j + \left(\frac{t_i - t_j}{t_{j+1} - t_j}\right)(y_{j+1} - y_j). \end{aligned} \tag{5}$$

Let the distance D_{SE}^i be the distance between the recovered point P'_i and the original point P_i. The simultaneous Euclidean distance D_{SE} is the average distance between the original and recovered trajectories often used to respond to the compression effect and expressed as:

$$\begin{aligned} D_{SE}^i &= \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2}, \\ D_{SE} &= \frac{1}{N}\sum_{i=1}^{N} D_{SE}^i, \end{aligned} \tag{6}$$

where N denotes the number of original trajectory points for each ship.

The average deviation of speed R_v used to reflect the average difference in speed between the recovered P_i' and original P_i track points is expressed as

$$\begin{aligned} v_i' &= v_j, \\ R_v^i &= \left| v_i' - v_i \right|, \\ R_v &= \frac{1}{N} \sum\nolimits_{i=1}^{N} R_v^i, \end{aligned} \tag{7}$$

where N denotes the number of original trajectory points for each ship; v_i denotes the speed of the ship traveling to the trajectory point P_i; v_i' denotes the speed of the ship traveling to the recovery trajectory point P_i', v_j denotes the speed of the ship traveling to the critical trajectory point K_j.

Average deviation of heading R_θ: the average value of the heading difference between each recovery track point P_i' and the original track point P_i is expressed as

$$\begin{aligned} \theta_i' &= \arctan\left(\frac{y_{j+1} - y_j}{x_{j+1} - x_j}\right), \\ R_\theta^i &= \left| \theta_i' - \theta_i \right|, \\ R_\theta &= \frac{1}{N} \sum\nolimits_{i=1}^{N} R_\theta^i, \end{aligned} \tag{8}$$

where N denotes the number of original trajectory points for each ship; θ_i denotes the heading angle of the ship traveling to the trajectory point P_i; θ_i' denotes the heading angle of the ship traveling to the recovery trajectory point P_i'; and (x_j, y_j) and (x_{j+1}, y_{j+1}) denote the coordinates at the key trajectory points K_j and K_{j+1}, respectively.

D. COMPRESSION THRESHOLD

The DP algorithm steps clearly show that the compression threshold ε is the only parameter that must be set by the user in the DP algorithm. Increasing the compression threshold will reduce the number of retained nodes. This reduces the data storage cost and improves the compression rate, and at the same time, reduces the accuracy of the simplified data. The choice of the threshold value is uncertain and varies under different application domains. For different scenarios, appropriate thresholds need to be set, which allows the data to be compressed and still contain the important information needed for subsequent calculations. The link between ship size and compression threshold settings needs to be considered when compressing AIS data. For example, large ships usually sail smoothly in open water. Track points slightly away from the main course do not affect the overall shape of the ship trajectory. However, small ships usually navigate in narrow waters, making the above deviated track points important for recovering their trajectory. The ship turning radius is also related to its own size; therefore, if a large compression threshold suitable for large ships is set for a small ship, many important track points will be discarded in the trajectory compression result. On the contrary, if a small compression threshold suitable for a small ship is used, then several unnecessary trajectory points will be retained in the trajectory of a large ship, resulting in a lower compression rate.

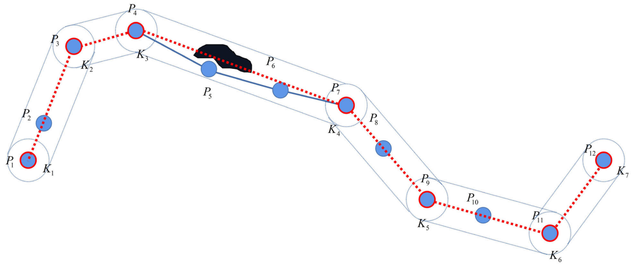


FIGURE 2. Situation of the retained route crossing obstacles after the DP algorithm compression.

Zhang et al. [37] used the ship domain as a threshold for ship route planning and set it to 0.8 times the captain. Tang et al. [42] converted the threshold selection problem into a critical threshold change rate setting problem and used a statistical analysis of the data within a region to propose an adaptive critical threshold change rate applicable to the region to control the compression effect. In threshold selection, Zhao et al. [39] set the compression threshold to 0.5 times the captain. Considering the area of water, they stated that the ratio in open water can be set greater than that in small water, and in areas with watercourses, the maximum threshold value can be set according to the river width. The appropriate ratio for waters near islands is usually between 0.1 and 10 times.

In this study, we used 0.8 times captain as the reference interval for setting the compression threshold. For different users, the compression threshold can be adjusted according to the compression intensity demand in their own research. An unreasonable compression threshold setting in the application may lead to the situation depicted in Fig. 2, in which the compression threshold is set as too large, resulting in a constant directional line between the key waypoints crossing non-navigable areas (e.g., obstacles after compression). The compression result in this case has errors. To address this situation, we propose herein **Improved Strategy IV,** which introduces an obstacle detection mechanism. When the reserved waypoints are connected by the constant direction line to form the recovery navigation trajectory through obstacles, we believe that the compression threshold size set at this time is not applicable to the route and will use Algorithm 2 to re-compress and avoid the non-navigable area while generating the recommended compression threshold ε_{new}. By contrast, when the recovery route does not have the error result of crossing the non-navigable area, the compression threshold is applicable at this time and does not generate the recommended compression threshold.

III. MPDP ALGORITHM

This section presents the proposed MPDP algorithm.

First, the traditional DP algorithm in each compression layer only selects a distance farthest point as the split point

Algorithm 1 Trajectory Compression

Input: Original track points set OS, Threshold of the compression distance ε, Threshold of the number of peak points *Th_n*, Threshold of compression layers *ThJ*

Output: Key track point set KS

```
L is the number of compression layers, and the initial value is 0
KS <- MPDP( OS, L, c, Th.n, ThJ j
function MPDP(PS, Zr} e,Thn, ThJ )
    n is the size of point set PS
    L=L+1
    if PS[O] is same as PS [n—1] then
        for i=l to u-2 do
            Calculate the point-to-point distance d from PS[i] to PS[0] through Equation (11)
            DS[i| = d
        end for
    else
        for i=l to n-2 do
            Calculate the point-to-line distance d from PS[i] to PS[0] PS[n-l] through Equation (2)
            DS[i| = d
        end for
    end if
    D is the set of distance set DS normalized by Equation (9)
    A, V are the change rate of heading angle and speed of each track point after normalization by the Equation (9)
    FS=oc * D + $\beta$ * A+ $\omega$ * V
    if L<= ThJ then
        Index—findpeaks(FS)
        m is the size of index set Index of points
        while (m>Th_n) do
            Index—lindpeaksf FS[Index])
        end while
    else
        [fmax. Index] =max(FS)
    end if
    Index ={0. Index, n-1}
    k is the size of index set. Index of points
    for i=l to k-2 do
        if DS[Index[i]] > e then
            KS r- MPDP( PS[Index[i-l]: Index'i] - 1], L, c., Th_n, ThJ 1
            KS r- MPDPf PS[Index[i]: Index[i+1]], c, Th.n, ThJ. )
        else
            Add PS[Index[i]] into KS
            E = [E,DC[i]}
        end if
    end for
    return KS
end function
```

Algorithm 2 Compressed Trajectory Obstacle Detection and Obstacle Avoidance

Input: Original track points set OS, Key track point set KS

Output: Key track point set KS, Recommended compression threshold $£_{neiv}$

```
KS, $\varepsilon_{new}$ ←-OBAVOID(OS,KS)
function OBAVOID(OS, KS)
    i=l
    while i do
        if the obstacles between KS[i] and KS[i+l] then
            PS is the subset of all trackpoints from KS[i] to KS[i+l] in OS
            n is the size of point set PS
            dmax and Index are respectively the maximum distance between each
point in PS and the line KS[i] KS[i+l] and the index for obtaining the maximum distance
            if KS(i] is same as KS [i+1] then
                for j=0 to n-1 do
                    Calculate the point-to-point distance d from PS[j] to KS[i] through Equation (11)
                    if d> dmax then
                        dmax = d
                        Index=i
                    end if
                end for
            else
                for j=0 to n-1 do
                    Calculate the point-to-line distance d from PS[j] to KS[i] PS[i+l] through Equation (2)
                    if d> dmax then
                        dmax = d
                        Index=i
                    end if
                end for
            end if
            Add PS[fndej] into KS
            E = {E, dmtiz}
        else
            if KS[i+l] is same as the last trajectory point in OS then
                break
            else
                i=i+l
            end if
        end if
    end while
    $\varepsilon_{new}$ =min(E)
    return KS, $\varepsilon_{new}$
end function
```

and continues the compression by the chain decomposition of the route. Hence, when the compressed route trajectory is multi-turn, some track points at the turn cannot be collected closer to the route point with the largest degree of turn. We illustrate this situation by assuming an ideal turning case 1 (Fig. 4 (a)). This time, we can clearly see that the ship turning case is exactly the same at the (ABCD) position. However, after compression by the DP algorithm, the track points retained in the four positions show different compression effects, which is not expected. We hope that in such cases of track compression, the DP compression algorithm can retain the same track points for the same turn cases and recover the track characteristics better.

Therefore, we propose **Improved Strategy I**, which is a multi-peak retention strategy. In Step 3-2 of the DP compression algorithm, we find the peaks in the distance set ***DS*** to form point set ***ES***. The distances corresponding to the waypoints in ***ES*** are compared with the threshold ε one by one. The points are stored in set ***KS*** if they are greater than the threshold ε. This method reduces the number of compression layers of the chain decomposition of the DP algorithm, consequently reducing the traversal-style point-to-linear distance calculation that must be performed when the subset $\boldsymbol{OS\_Sub_j}$ is compressed and speeding up the DP algorithm compression.

There are many small angle changes in the ship's heading during navigation; hence, many small peaks are obtained when finding the peaks, resulting in the retention of excessive track points. This is different from the ideal situation, where the track points are not at the critical locations in the trajectory and need not be retained. Accordingly, we set the threshold of the number of peaks *Th_n*. When the number of peaks in the set of extreme points ***ES*** is greater than *Th_n*, the peaks are obtained again for the set of extreme points ***ES*** until the number threshold is satisfied. At this point, the retained extreme points are the peak values with the greatest change in distance from the set. In general, the *Th_n* value must be an integer from 3 to 6. The whole compression process shows a multi-peak situation for the distance set ***DS*** in the early stage. In the later stage of the compression process, the distance set mostly remains as a single-peak set at this time because of the decomposition into small track segments when the great distance value in ***DS*** is the same as the maximum value. Therefore, we set the peak compression layer threshold *Th_l* and use the multi-peak retention strategy when the number of compression layers is less than the threshold *Th_l* to retain multiple track points. We take the maximum value of ***DS*** to retain a track point when the number of compression layers is greater than the threshold. In general, the value must be an integer from 1 to 3.

Second, another shortcoming of the DP algorithm is that speed changes are made at certain moments during the ship's navigation, which affects the recovery of the ship's navigation behavior. However, the DP algorithm does not consider the influence of the speed change points in the compression process. At the same time, the current use of the DP algorithm in the compression process selects the farthest point as the splitting point. For the AIS data, the waypoint is dense, and the farthest point is not the largest point of the turning arc of the route. Taking the simulation experiment Case 2 as an example, Fig. 11 shows a certain distance between the farthest distance point and the point with the largest course bend rate. Only using the distance as the track point retention criterion cannot reflect the influence of the speed change on the ship trajectory.

Therefore, we introduce **Improved Strategy II** using the fitness function instead of the point-to-line distance. The fitness function contains three parts: point-to-linear distance, heading change rate, and speed change rate. The units and magnitudes of the three greatly differ and are not suitable for direct comprehensive comparative analysis. Therefore, each cost must be normalized separately to eliminate the influence of units and magnitudes. The min–max normalization method is adopted here. It is the linear transformation of the original data such that the resultant values are mapped to between [0,1]. The transformation equation is presented as follows:

$$F = \frac{f - \min}{\max - \min}, \tag{9}$$

where max and min are the maximum and minimum values of the group data, respectively; f is the original objective function, including the point-to-line distance, heading change rate and speed change rate of each waypoint; and F is the normalized objective function.

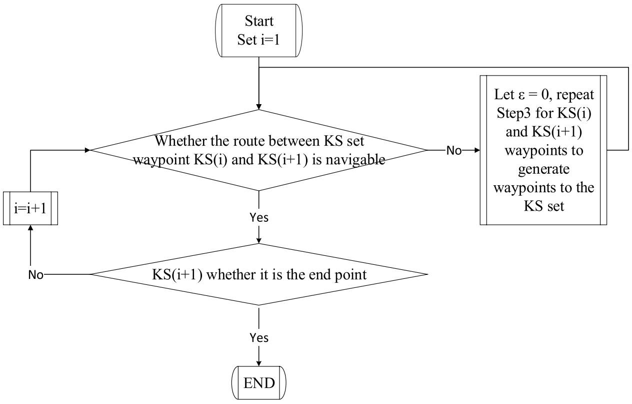

Start
Set i=1
Whether the route between KS set waypoint KS(i) and KS(i+1) is navigable
No
Let ε = 0, repeat Step3 for KS(i) and KS(i+1) waypoints to generate waypoints to the KS set
i=i+1
Yes
No
KS(i+1) whether it is the end point
Yes
END

FIGURE 3. Testing the connectivity steps of the retained route after the DP algorithm compression.

After conversion, the adaptation function *fitness* is calculated as

$$\begin{aligned} &\mathit{fitness} = \alpha \times d + \beta \times \eta + \omega \times \upsilon, \\ &\alpha + \beta + \omega = 1, \end{aligned} \tag{10}$$

where d is the normalized point-to-line distance, η is the normalized heading change rate; υ is the normalized speed change rate; and α, β, ω is the weight coefficient that can be adjusted according to the actual user requirements. We use set ***FS*** to store the fitness function value of each point. The extreme or maximum value is calculated using ***FS*** instead of ***DS***. The index set ***Index*** of the corresponding points is obtained. The distances of the points corresponding to index ***Index*** in ***FS*** are then read to determine whether or not the compression threshold requirement is satisfied. This ensures that all the waypoints added to the key waypoint set ***KS*** satisfy the compression threshold

When the DP algorithm compression threshold is not set reasonably, or when the original trajectory is very close to land or other obstacles, the recovery route obtained using the key waypoint set ***KS*** will be under the situation of crossing non-navigable areas (e.g., obstacles) (Fig. 2). Therefore, we introduce **Improved Strategy III**, which is an obstacle detection mechanism, and add Step 5 to the original DP algorithm.

- Step 5: Check the conductivity of the compressed segment. Fig. 3 presents the specific steps.

As shown in Algorithm 2, when a problematic route (e.g., obstacle crossing) is detected, the original trajectory will be compressed and re-calculated with ***KS*** (i) and ***KS*** $(i+1)$ as the start and end points, respectively. The new critical waypoint will also be retained until the recovered trajectory is smoothly navigated. At this time, we define the critical distance set as ***KS*** to store the point-to-line distance in the corresponding point when the trajectory point is added to the critical path set. The minimum value in the updated critical distance set $\boldsymbol{E}$ is selected as the recommended compression threshold ε_{new} when the obstacle detection is completed.

Finally, the coordinates of the start and end points of the segment overlap when the ship is in in-situ hovering or a round-trip motion. That is, the geographic coordinates of two points K_j and K_{j+1} overlapping during the compression process will lead to the following error when the DP algorithm executes Step 3-1: when calculating the distance set $\boldsymbol{DS}$ from each point in subset $\boldsymbol{OS\_Sub}_j$ to the baseline using the straight line between K_j and K_{j+1} as the baseline, the point-to-straight line distance Eq. (2) cannot be calculated because the coordinates of the K_j and K_{j+1} points are the same. In this case, we introduce **Improved Strategy IV**, the position overlap judgment mechanism. When the coordinates of two points are same, the point-to-line distance *Distance* is changed to use Eq. (11) to calculate the point-to-point distance *Distance'*.

$$Distance' = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}, \tag{11}$$

where x_i, x_j and y_i, y_j are the horizontal and vertical coordinates of the two points, respectively. At this point, the algorithm's correctness is guaranteed without changing its principle.

Combining the abovementioned improved strategies, we propose herein the improved MPDP algorithm (Algorithm 1):

1) To reduce the number of DP algorithm compression layers and speed up the compression process, the peak sampling method is used instead of the maximum sampling method.
2) The fitness function is used instead of the point-to-straight line distance as the sampling criterion, which can improve the algorithm's preservation effect in both heading and speed.
3) The obstacle detection mechanism is introduced.
4) The inner position overlap judgment mechanism is used to replace the point-to-line distance formula with the two-point distance equation when there is an end-point overlap.

IV. RESULTS

A. TEST BACKGROUND

We verified the effectiveness and feasibility of our algorithm by comparing it with the conventional DP algorithm. It was comprehensively compared in terms of the following six performance metrics: compression time T_c, compression rate R_c, length loss rate R_l, simultaneous Euclidean distance D_{SE}, average deviation of speed R_v, and average deviation of heading R_θ (Section II, Part C).

We designed four simulated and three real trajectory experiments. The four simulation experiments mainly corresponded to the four proposed improvement strategies and verified the effectiveness of each improvement strategy. Two real trajectory experiments were conducted to verify the effectiveness of the MPDP algorithm in real scenarios. The real trajectory was tested using the AIS data from the Shiyezhou section of the Yangtze River waters on January 18, 2022. The algorithm was implemented and run on a computer with the following specifications: 64-bit Windows with eight cores (Intel (R) Core (TM) i7-10700 CPU at 2.9 to 4.8 GHz) and 16 GB RAM) using MATLAB R2020a in the configuration presented below.

For illustration purposes, the algorithm names were defined according to Table 1.

TABLE 1. Names of the algorithms used in this paper and their definitions.

Algorithm	Explanation
DP	Classical Douglas–Peucker Algorithm
DP-i	Classical Douglas–Peucker Algorithm with **Improved Strategy I**
DP-ii	Classical Douglas–Peucker Algorithm with **Improved Strategies I** and **II**
DP-iii	Classical Douglas–Peucker Algorithm with **Improved Strategies I**, **II**, and **III**
DP-iv	Classical Douglas–Peucker Algorithm with **Improved Strategy IV**
MPDP	Classical Douglas–Peucker Algorithm with **Improved Strategies I**, **II**, **III** and **IV**

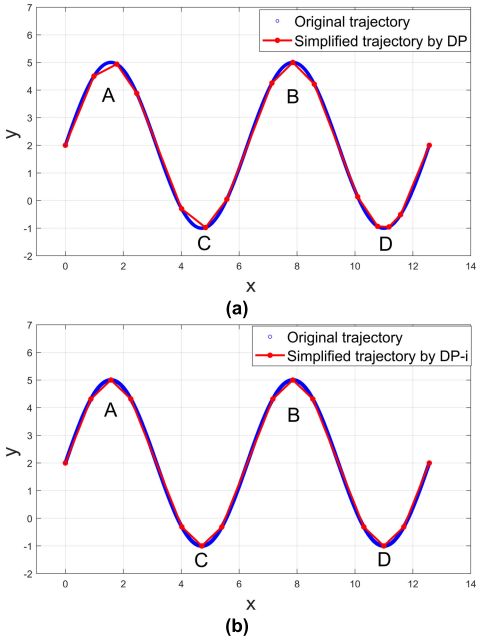
B. SIMULATION EXPERIMENTS

Case 1: Let the ship navigate according to the trajectory that can be synthesized $y = 3 \times \sin(x)$. There are 12567 simulated track points. The DP and DP-i algorithms are used for the track point compression experiments. Table 2 presents the algorithm parameters.

TABLE 2. Settings of the experimental parameters for Case 1.

Algorithm	Compression threshold ε (m)	Threshold for the number of peaks Th_n	Threshold for the number of compression layers Th_l
DP	0.1	-	-
DP-i	0.1	4	2

Fig. 4 clearly shows that the ship turns are exactly the same at the (ABCD) position. However, after the compression of the DP algorithm, the track points retained in the four positions showed different recovery effects. After the DP-i algorithm was compressed by adding **Improved Strategy I**, the ship track points at (ABCD) were identical and fitted the original trajectory better. This was because when the DP algorithm was used (Fig. 5), it kept one track point in the first layer to split the track into two segments and slit downward in turn. As shown in Fig. 6, using the DP-i algorithm retained four waypoints in the first compression layer. At this time, the trajectory was divided into five segments. Fig. 7 depicts the critical distance change, where dots of the same color indicate the waypoints retained under the same compression layer L.



(a)

(b)

FIGURE 4. Compression results of Case 1 trajectory using the (a) DP and (b) DP-i algorithms.

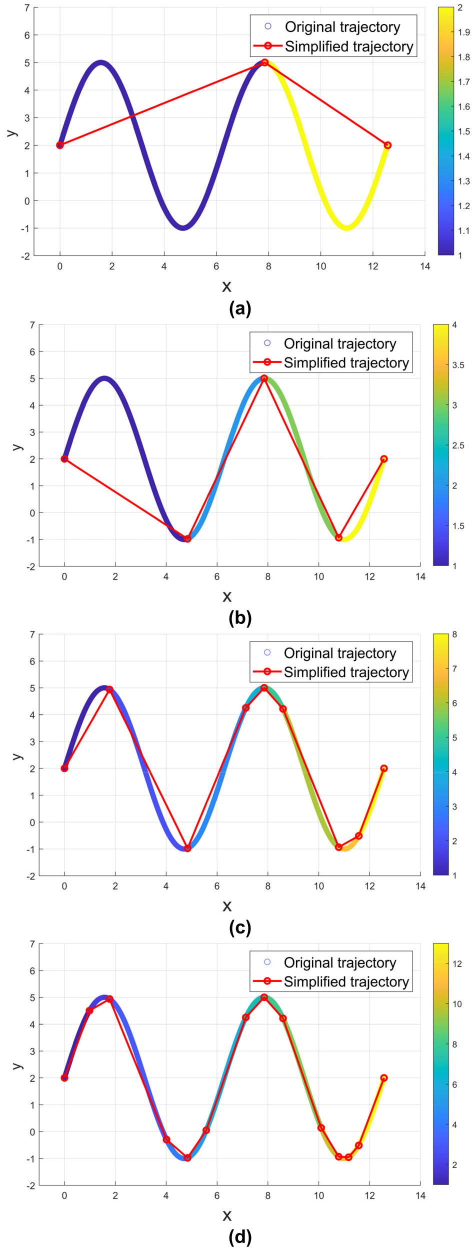
TABLE 3. Comparison of compression performance indicators for Case1.

Algo-rithm	T_c (ms)	R_c	R_l	D_{SE} (m)	R_θ (rad)
DP	109.5	99.88%	1.46%	0.0963	0.1380
DP-i	54.9	99.89%	1.28%	0.0941	0.1348

The original DP algorithm retained all critical waypoints after five compression layers, whereas the improved DP algorithm retained all critical waypoints after two compression layers.

Table 3 presents the experimental results. The compression time T_c of the DP-i algorithm was reduced by 54.6ms compared with that of the DP algorithm, which was approximately 49.86% optimized. The compression rate R_c was increased by 0.01%, indicating that the compression situation was almost the same. Fig. 12 also illustrates that the DP algorithm finally retains 14 key waypoints, while the DP-i algorithm retains 12 key waypoints. The length loss rate R_l was reduced by 0.18%, which was approximately 12.33% optimized. The simultaneous Euclidean distance D_{SE} was reduced by 0.0022m, approximately 2.28% optimized. Fig. 9 shows the overall heading recovery. The average heading deviation R_θ was reduced by 0.0032rad, approximately 2.32% optimized.

The DP-i algorithm preserved fewer key waypoints with less computing time in this experimental scenario. Meanwhile, the performance metrics of the simplified recovered routes outperformed the routes preserved by the original DP algorithm in all aspects. **Improved Strategy I** can find all the key nodes to be preserved in fewer compression layers, thereby speeding up the compression. It also showed better preservation of the route characteristic for such curved routes.



(a)

(b)

(c)

(d)

FIGURE 5. Case 1 trajectory compression process using the DP algorithm: compressions after (a) one, (b) two, (c) three, and (d) four layers. Note: the same color track points in the figure are in the same subset.

Case 2: Let the ship navigate according to the trajectory that can be synthesized $y = 3 \times \sin(x) + x$. There are 12567 simulated track points. The DP and DP-ii algorithms were used for the track point compression experiments. Table 4 shows the algorithm parameters.

In, Fig. 8 the simplified route obtained after the compression by the DP-ii algorithm was closer to the original route trajectory compared to the DP algorithm. Fig. 9 shows

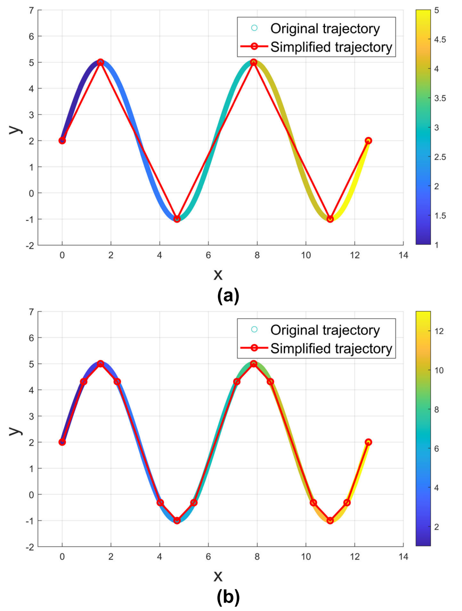


FIGURE 6. Case 1 trajectory compression process using the DP-i algorithm: compression after (a) one and (b) two layers. Note: the same color track points in the figure are in the same subset.

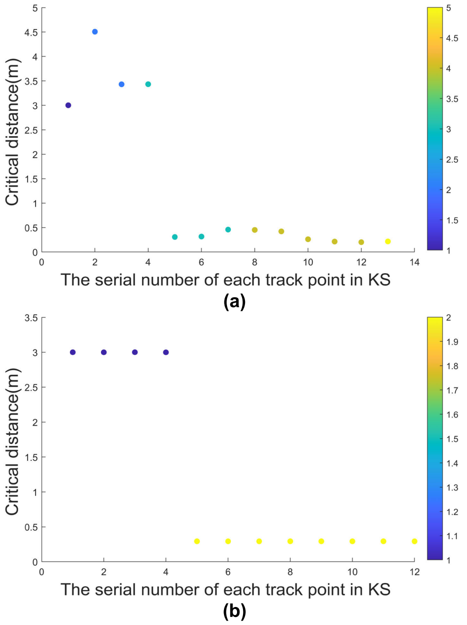


FIGURE 7. Variation of the critical distance for Case 1. Note: points of the same color indicate the waypoints retained under the same number of compression layers.

the values of each objective function in the first-layer of compression. The normalized distance d, normalized heading rate of change η, and normalized speed rate of change υ of this trajectory were optimal at different locations; hence, the multi-objective fitness function calculation using Eq. (10) can balance the relationship between the optimization objectives to retain better route points. The DP-ii algorithm used a peak retention strategy for the set of fitness functions $\boldsymbol{FS}$ while adding the waypoints at the peak to the set of critical

TABLE 4. Settings of experimental parameters for Case2.

Algorithm	ε (m)	Th_n	Th_l	α	β	ω
DP	0.2	-	-	-	-	-
DP-ii	0.2	4	2	0.5	0.3	0.2

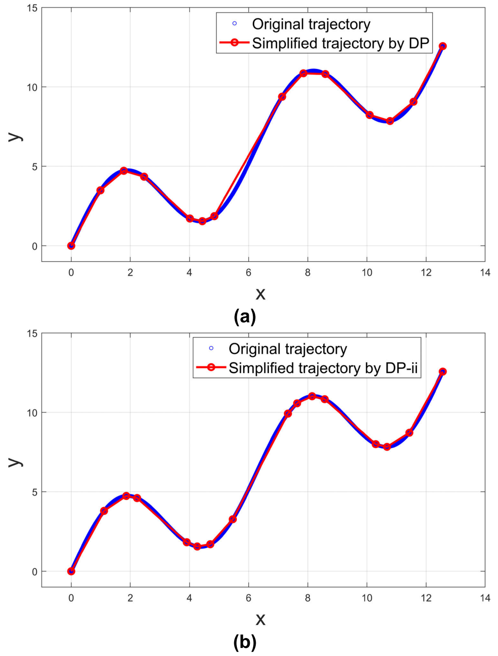


FIGURE 8. Compression results of the Case 2 trajectory using the (a) DP and (b) DP-ii algorithms.

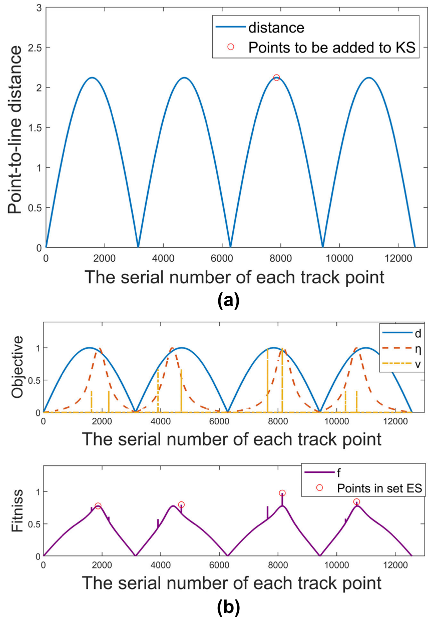
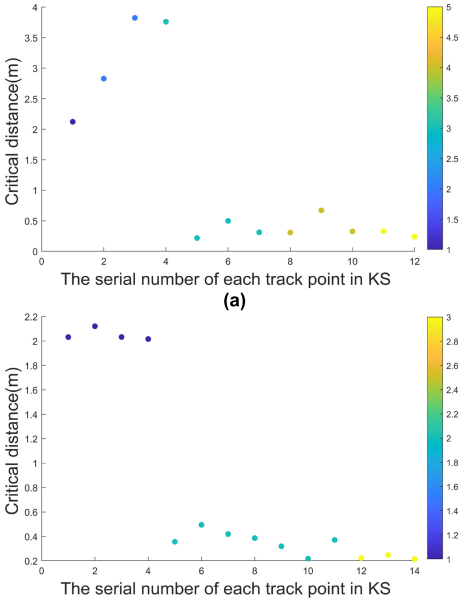


FIGURE 9. Distance set DS value, fitness function value, and key point selection of the Case 2 trajectory in the first-layer compression using the (a) DP and (b) DP-ii algorithms.



(a)

(b)

FIGURE 10. Critical distance variation for Case 2. Note: points of the same color indicate the waypoints retained under the same number of compression layers.

TABLE 5. Comparison of the compression performance indicators for Case 2.

Algo-rithm	T_c (ms)	R_c	R_l	D_{SE} (m)	R_θ (rad)	R_v (kn)
DP	117.3	99.89%	1.3%	0.1427	0.1371	0.2572
DP-ii	73.5	99.87%	0.81%	0.1010	0.1118	0.0177

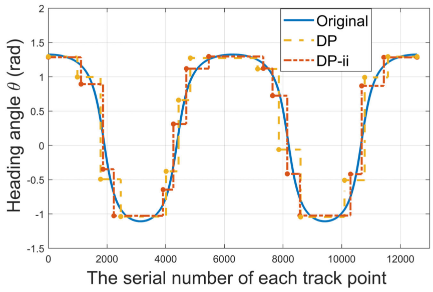


FIGURE 11. Comparison of the heading angles in Case 2.

nodes ***KS***. Fig. 10 depicts the critical distance variation. The original DP algorithm retained all critical waypoints after five compression layers, while the improved DP algorithm retained all critical waypoints after three compression layers.

Table 5 presents the experimental results. The compression time was reduced by 43.8ms and optimized by 37.3% using the DP-ii algorithm, and the compression rate were approximately equal. Fig. 10 depicts that the DP algorithm finally retained 12 key waypoints, while the DP-i algorithm retained 14 key waypoints with almost the same compression intensity. The length loss rate was reduced by 0.49% and optimized by 37.69%. The simultaneous Euclidean distance was reduced by 0.0417m and optimized by 29.22%. Fig. 11 displays the overall heading recovery. The average heading deviation was reduced by 0.0253rad and optimized by 18.45%. Fig. 12 depicts the overall speed recovery with a

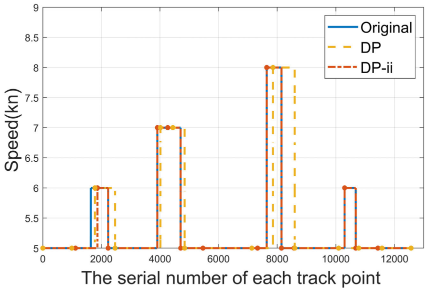
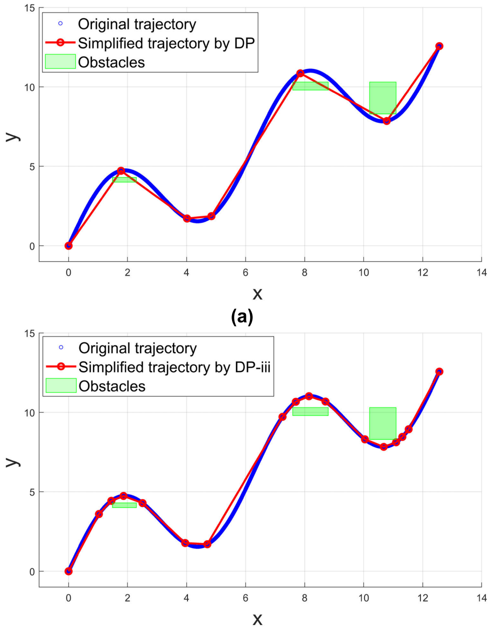


FIGURE 12. Comparison of the speeds in Case 2.



(a)

(b)

FIGURE 13. Compression results of the Case 3 trajectory using the (a) DP and (b) DP-iii algorithms.

TABLE 6. Settings of experimental parameters for Case3.

Algo-rithm	ε (m)	Th_n	Th_l	α	β	ω
DP	0.5	-	-	-	-	-
DP-iii	0.5	4	2	0.5	0.3	0.2

TABLE 7. Comparison of compression performance indicators for Case3.

Algo-rithm	T_c (ms)	R_c	R_l	D_{SE} (m)	R_θ (rad)	R_v (kn)	ε_{new}
DP	98.1	99.94%	3.49%	0.40	0.24	0.83	-
DP-iii	85.7	99.86%	1.10%	0.14	0.12	0.09	0.038

TABLE 8. Settings of the experimental parameters for Case 3 using the recommended thresholds.

Algorithm	ε (m)	Th_n	Th_l	α	β	ω
DP	0.038	-	-	-	-	-
DP-iii	0.038	4	2	0.5	0.3	0.2

TABLE 9. Settings of the experimental parameters for Case 3 using the recommended thresholds.

Algo-rithm	T_c (ms)	R_c	R_l	D_{SE} (m)	R_θ (rad)	R_v (kn)
DP	142.8	99.73%	0.18%	0.0293	0.0533	0.1565
DP-iii	125.5	99.72%	0.17%	0.0227	0.0526	0

TABLE 10. Settings of experimental parameters for Case4.

Algorithm	ε (m)	Th_n	Th_l	α	β	ω
DP	0.5	-	-	-	-	-
DP-iv	0.5	-	-	-	-	-
MPDP	0.5	5	2	0.5	0.5	0

0.2395kn reduction in the average speed deviation optimized by 93.12%.

After the addition of **Improved Strategies I** and **II**, the DP-ii algorithm used less computing time in this experimental scenario. With nearly the same compression rate, the performance indices of the retained routes were better than those retained by the original DP algorithm, with the most significant optimization effect in speed recovery.

Case 3: Let the ship navigate according to the trajectory that can be fitted as $y = 3 \times \sin(x) + x$. There are 12567 simulated track points. Three obstacles were set up in the green area in Fig. 13. The DP and DP-iii algorithms were used for the track point compression experiments. Table 6 lists the algorithm parameters.

In Fig. 13, the DP preserved the course through the green obstacle, while the DP-iii algorithm performed another split and successfully avoided the obstacle by including the obstacle detection mechanism from **Improved Strategy III**. Fig. 14 shows the critical distance variation. The original DP algorithm underwent four compression layers and retained all critical waypoints, whereas the improved DP algorithm retained all critical waypoints after 13 compression layers, of which the first two layers were the waypoints retained by Algorithm 1, and the last 11 layers were the waypoints obtained by re-splitting during the obstacle detection by Algorithm 2.

Table 7 presents the experimental results. The compression time was reduced by 12.4ms using the DP-iii algorithm and optimized by approximately 13.21%. The compression rate was reduced by 0.08%. In Fig. 14, the DP algorithm finally retained five key waypoints, whereas the DP-iii algorithm retained 15 key waypoints. The length loss rate was reduced by 2.39%, showing approximately 68.48% optimization. The simultaneous Euclidean distance was reduced by 0.26m and optimized by approximately 63.45%. The average heading deviation was reduced by 0.12rad, which was approximately 48.26% optimized. The average speed deviation was reduced by 0.74kn and optimized by 88.85%.

The experimental results revealed that the DP-iii algorithm used less computing time. In addition, no error cases of

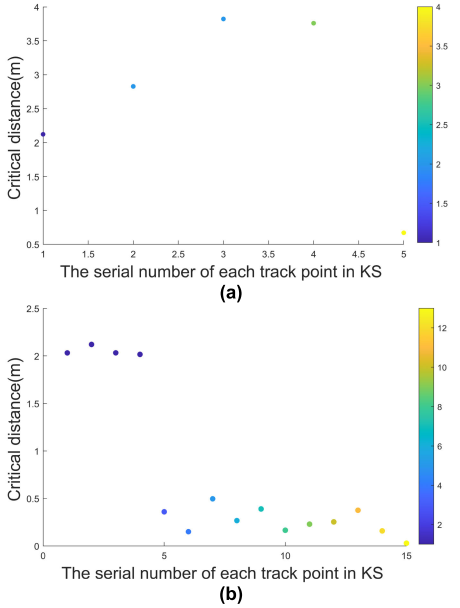


FIGURE 14. Critical distance variation for Case 3. Note: points of the same color indicate the waypoints retained under the same number of compression layers.

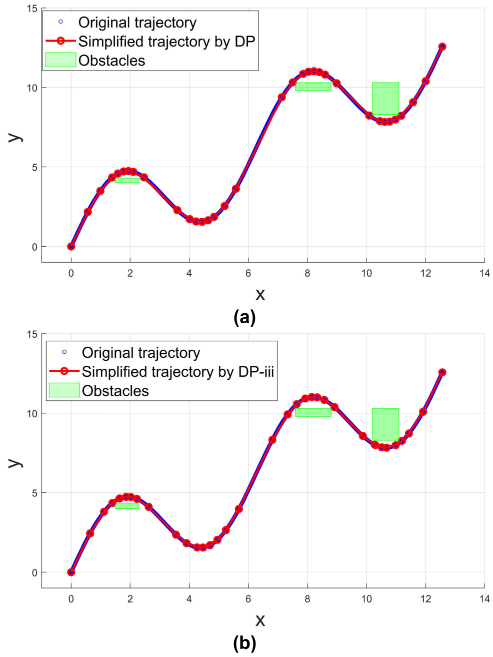


FIGURE 15. Compression results of case3 trajectory (Use recommended thresholds), where (a) using DP algorithm (b) using DP-iii algorithm.

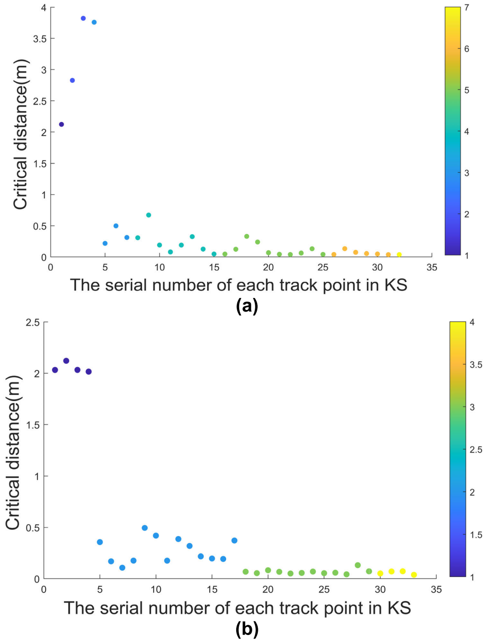


FIGURE 16. Variation of critical distance for case3 (Use recommended thresholds). Note: Points of the same color indicate waypoints retained under the same number of compression layers.

obstacle crossing occurred in the reserved routes. Meanwhile, all performance indices of the simplified routes of the DP-iii algorithm were better than those of the routes reserved by the original DP algorithm. This proved that the algorithm has a better effect of preserving route characteristics for such routes with excessively large threshold settings or routes that are too close to obstacles.

The route compression threshold was set 0.038m which is the recommended threshold of the DP-iii algorithm. Table 8 presents the remaining parameters to reverify the DP and DP-iii algorithms.

Fig. 15 shows that both the DP and DP-iii algorithms retained routes to avoid obstacles when using the compression threshold recommended by the DP-iii algorithm. In other words, the DP-iii algorithm recommends an effective compression threshold. Table 9 shows that the DP-iii algorithm computing time was reduced by 17.3ms, optimized by approximately 12.11%. In Fig. 16, the DP-iii algorithm reduced three compression layers. The compression rate was reduced by 0.01%, and the length loss rate was reduced by 0.01%. The synchronous Euclidean distance was reduced by 0.0066m and optimized by approximately 22.53%. The average heading deviation was reduced by 0.0007rad, while the average speed deviation was reduced by 0.1565kn and approximately 100% optimized.

The DP-iii algorithm showed a slight advantage in time, while the retained track points were optimized best in terms of both synchronous Euclidean distance and average airspeed deviation for approximately the same compression rate. These results indicated that the DP-iii algorithm compressed tracks better in terms of both spatial location and airspeed recovery.

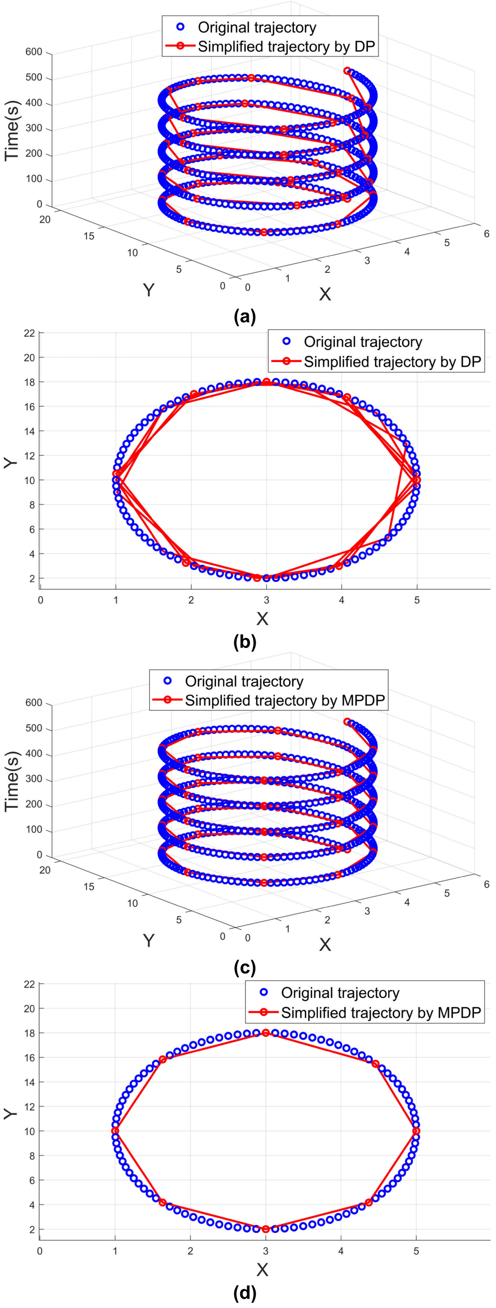
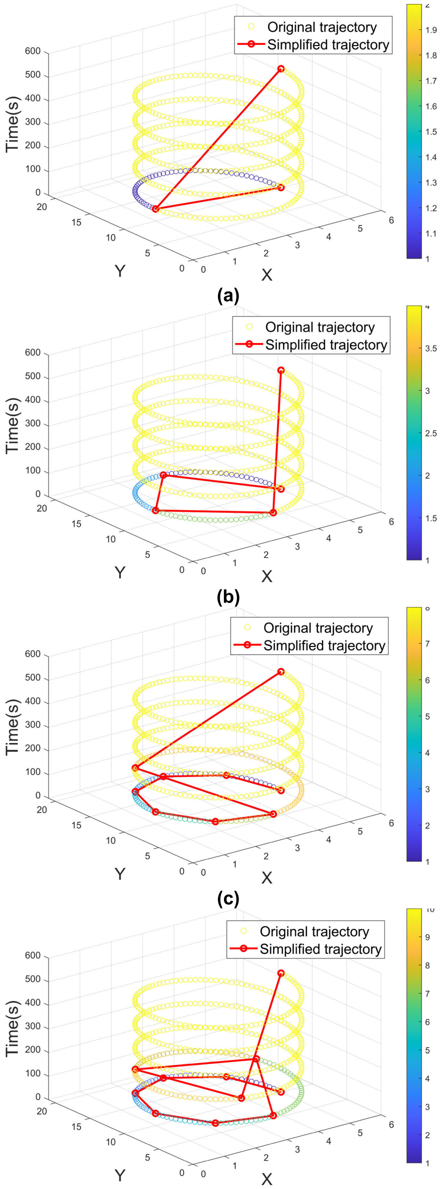


FIGURE 17. Compression results of the Case 3 trajectory: (a) and (b) plain and top views using the DP algorithm, respectively, and (c) and (d) plain and top views using the MPDP algorithm, respectively.

Case 4: Let the ship hover with a uniform speed in the in-situ elliptical trajectory, forming point 505 simulated waypoints. The start and end points were geographically identical. The DP, DP-iv, and MPDP algorithms were used to compress the trajectory points. Table 10 lists the algorithm parameters.

Fig. 17 illustrates that the MPDP algorithm retained the trajectory closer to the original trajectory and maintained



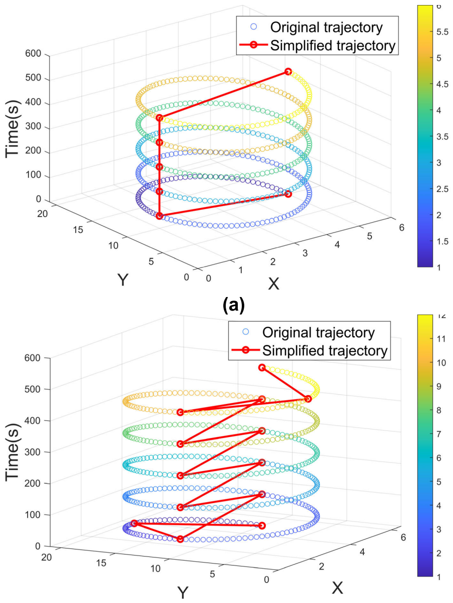
(a)

(b)

(c)

(d)

FIGURE 18. Case 4 trajectory compression process using the DP algorithm: compressions after (a) one, (b) two, (c) three, and (d) four layers. Note: the same color track points in the figure are in the same subset.



(a)

(b)

FIGURE 19. Case 4 trajectory compression process using the MPDP algorithm: compression after (a) one and (b) two layers. Note: the same color track points in the figure are in the same subset.

TABLE 11. Comparison of the compression performance indicators for Case 2.

Algo-rithm	T_c (ms)	R_c	R_l	D_{SE} (m)	R_θ (rad)
DP	-	-	-	-	-
DP-iv	8.8	91.88%	1.75%	0.3002	0.2412
MPDP	3.1	91.88%	1.45%	0.2751	0.1962

a consistent effect for each trajectory compression loop. Combining Fig. 18 with Fig. 19, for the in-situ circling trajectory, the MPDP algorithm retained five track points in the first compression layer when **Improved Strategy I** was introduced. This was equivalent to splitting what was originally a five-turn overlapping track into six non-overlapping segments in time and performing further splitting. The DP algorithm cannot spatially split the route faster because only one track point was retained in the first compression layer. Thus, it can only spatially and gradually split the route. This resulted in a track that was originally identical per lap but retained track points with completely different positions after the DP algorithm calculation.

Table 11 shows that the DP algorithm failed to produce results in this simulation experiment because the point-to-linear distance was required in the DP algorithm solution process. Correspondingly, Eq. (2) failed when the coordinates of the two breakpoints were same. We then compared the DP-iv algorithm, which introduced **Improved Strategy IV**, with the MPDP algorithm. The operation time of the MPDP algorithm was reduced by 5.7ms. The compression ratio was equal. The DP-iv algorithm retained all key nodes after 12 compression layers, while the MPDP algorithm retained all track points after only four compression layers (Fig. 20). The lost track length obtained by the MPDP algorithm was reduced by 0.30% and optimized by 17.14%. The simultaneous Euclidean distance was reduced by 0.0251m and optimized by 8.36%. The average heading deviation was reduced by 0.0450rad and optimized by 18.66%.

The MPDP algorithm maintains a high compression rate with less computing time in this experimental scenario. Meanwhile, the performance indices of the simplified recovery routes were better than those retained by the original

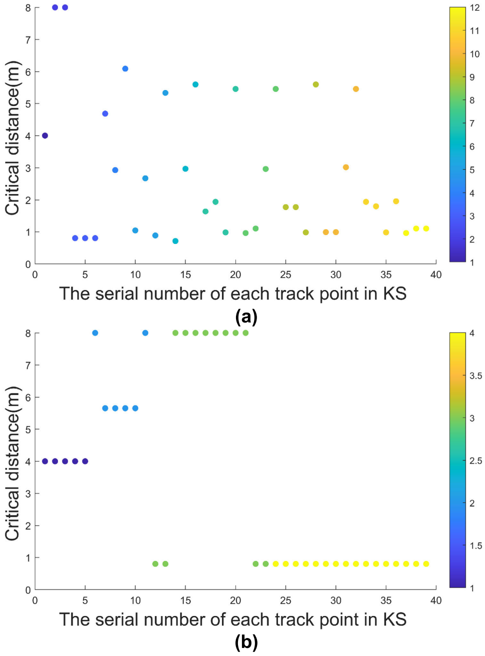


FIGURE 20. Critical distance variation for Case 4. Note: points of the same color indicate the waypoints retained under the same number of compression layers.

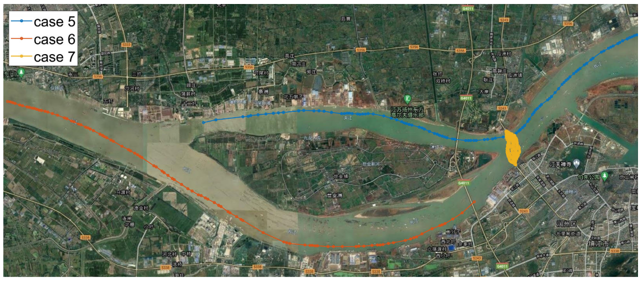


FIGURE 21. Case 5 to 7 original AIS data.

TABLE 12. Case 5 to 7 basic route information.

Number of experiment	MMSI	Length of the ship (m)	Number of track points
Case5	412751530	53	252
Case6	413779797	43	422
Case7	413788463	76	377

TABLE 13. Settings of experimental parameters for Case 5 to 7.

Number of experiment	Algorithm	ε (m)	Th_n	Th_l	α	β	ω
Case5	DP	42.40	-	-	-	-	-
	MPDP	42.40	4	2	0.5	0.3	0.2
Case6	DP	34.40	-	-	-	-	-
	MPDP	34.40	4	2	0.5	0.3	0.2
Case7	DP	60.80	-	-	-	-	-
	MPDP	60.80	4	2	0.5	0.3	0.2

TABLE 14. Comparison of compression performance indicators for Case5 to 7.

Number of experiment	Algorithm	T_c (ms)	R_c	R_l
Case5	DP	6.9	91.67%	0.17%
	MPDP	7.2	89.68%	0.16%
Case6	DP	8.0	95.73%	0.01%
	MPDP	7.5	94.08%	0.01%
Case7	DP	13.5	89.12%	4.42%
	MPDP	11.5	85.68%	2.64%
Number of experiment	**Algorithm**	**D_{SE} (m)**	**R_θ (rad)**	**R_v (kn)**
Case5	DP	124.7261	0.0603	0.0778
	MPDP	120.9807	0.0581	0.0548
Case6	DP	235.7260	0.0422	0.0571
	MPDP	224.2394	0.0407	0.0405
Case7	DP	74.5930	0.6246	2.2894
	MPDP	39.6593	0.6028	1.2987

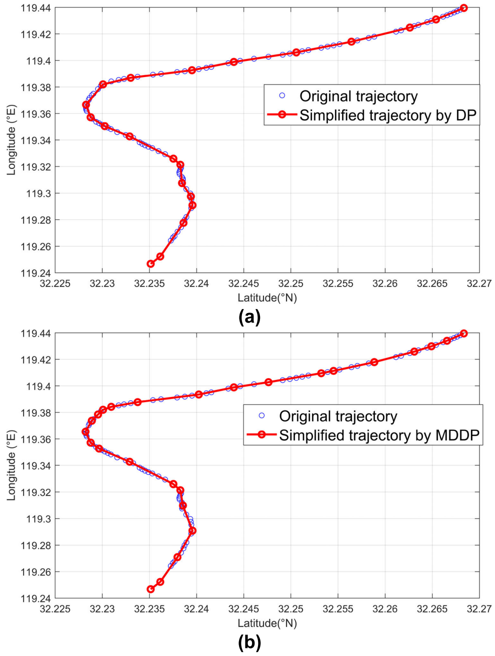


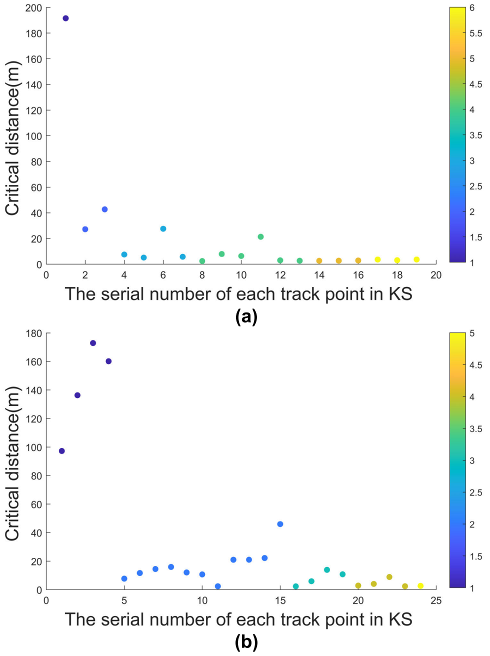
FIGURE 22. Compression results of the Case 5 trajectory using the (a) DP and (b) MPDP algorithms.

DP algorithm in all aspects, showing the most significant optimization effects in the route length and heading recovery.

C. REAL SCENARIO VERIFICATION

The real trajectory was tested using the AIS data from the Shiyezhou section of the Yangtze River waters on January 18, 2022. Fig. 21 depicts the three test routes taken. Table 12 shows their basic information, and Table 13 presents the experimental parameters of cases 5 to 7.

The compression results of Case 5 in Fig. 22 combined with the MPDP algorithm in Table 14 showed a computation time increased by 0.3ms. The compression rate was reduced by 1.99%. The MPDP algorithm retained five more waypoints (Fig. 23). The length loss rate was reduced by 0.01%. The simultaneous Euclidean distance was reduced by 3.7454m. Fig. 24 depicts the overall heading recovery.



(a)

(b)

FIGURE 23. Critical distance variation for Case 5. Note: points of the same color indicate the waypoints retained under the same number of compression layers.

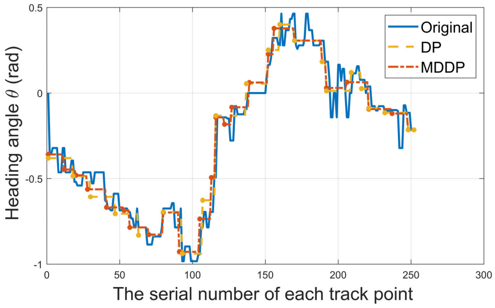


FIGURE 24. Comparison of the heading angles of Case 5.

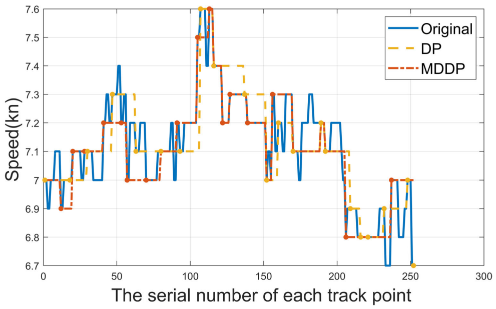
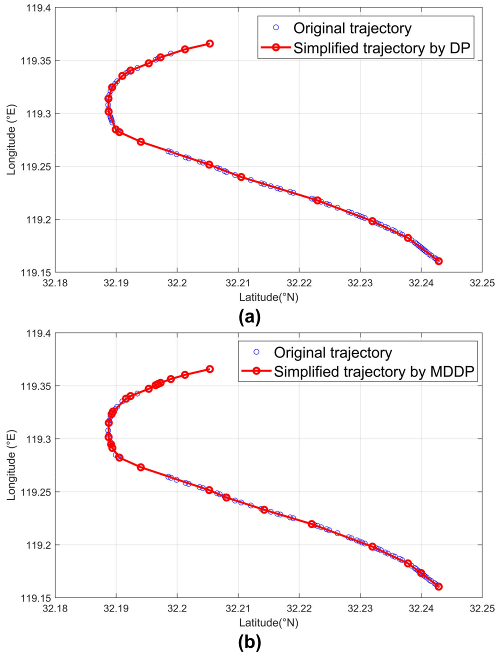


FIGURE 25. Comparison of the speeds of Case 5.

The average heading deviation was reduced by 0.0022rad. Fig. 25 illustrates the overall speed recovery. The average speed deviation was reduced by 0.023kn and optimized by approximately 29.56%.



(a)

(b)

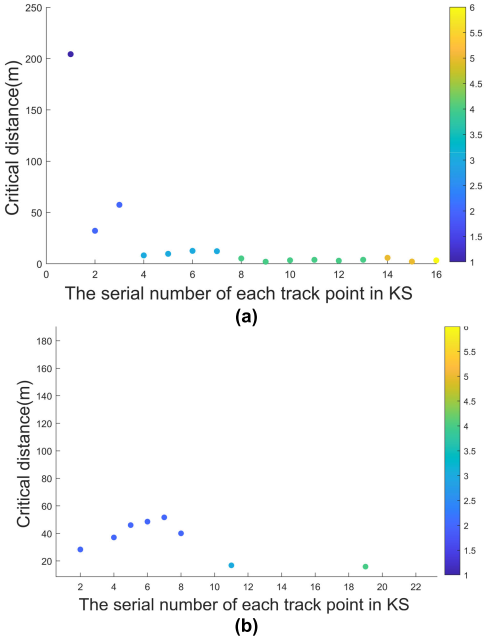
FIGURE 26. Compression results of the Case 6 trajectory using the (a) DP and (b) MPDP algorithms.

TABLE 15. Comparison of the average compression performance indicators for 2-hour AIS data from Shiyezhou section of the Yangtze River waters.

Algo-rithm	T_c (ms)	R_c	R_l	D_{SE} (m)	R_θ (rad)	R_v (kn)
DP	426.6	96.08%	0.65%	145.8896	0.1271	0.1946
MPDP	345.5	94.28%	0.59%	125.4893	0.1103	0.1525

The compression results of Case 6 in Fig. 26 combined with the MPDP algorithm in Table 14 showed a computation time reduced by 0.5ms. The compression ratio was reduced by 1.65%. Nine more waypoints were retained by the MPDP algorithm (Fig. 27). The length loss rate was approximately equal. The simultaneous Euclidean distance was reduced by 11.4866m. Fig. 28 displays the overall heading recovery. The average heading deviation was reduced by 0.0012rad. Fig. 29 presents the overall speed recovery. The average speed deviation was reduced by 0.0166kn and optimized by approximately 29.07%.

Fig. 30 and Table 14 show that the operation time of the MPDP algorithm was reduced by 2ms in Case 7. The compression was reduced by 3.44%. In Fig. 31, the MPDP algorithm retained 13 more track points. The length loss rate was reduced by 1.78% and optimized by 40.27%. The synchronous Euclidean distance was reduced by 34.9337m and optimized by 46.83%. Fig. 32 presents the overall course recovery. The average heading deviation decreased by 0.0218rad and was optimized by 3.49%. Fig. 33 displays the overall speed recovery. The average speed deviation was reduced by 0.9907kn and optimized by approximately 43.27%.



(a)

(b)

FIGURE 27. **Critical distance variation for Case 6. Note: points of the same color indicate the waypoints retained under the same number of compression layers.**

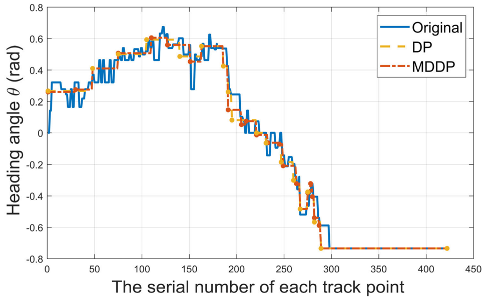


FIGURE 28. **Comparison of the heading angles of Case 6.**

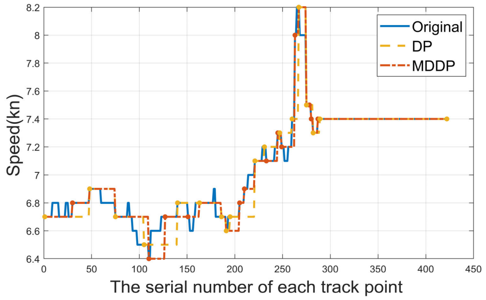
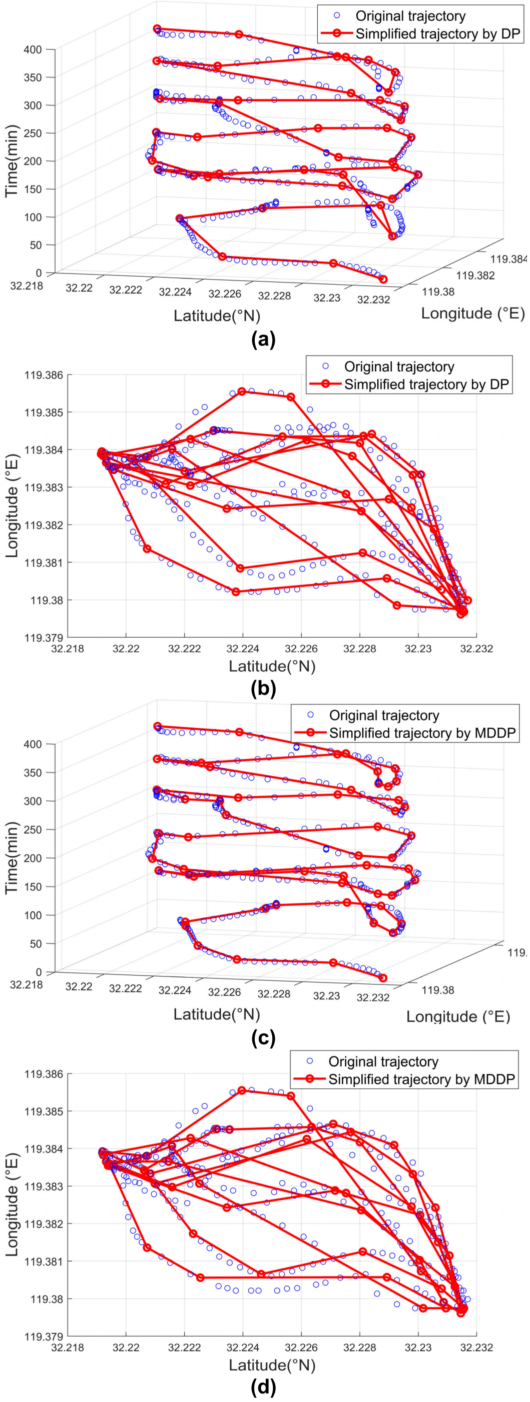


FIGURE 29. **Comparison of the speeds of Case 6.**



(a)

(b)

(c)

(d)

FIGURE 30. **Compression results of the Case 7 trajectory: (a) and (b) plain and top views using the DP algorithm, respectively, and (c) plain and (d) top views using the MPDP algorithm, respectively.**

Fig. 34 (a) shows the 2-hour AIS data for the Shiyezhou section of the Yangtze River waters, which contains 262 vessel trajectories with a total of 84,716 waypoints. Fig. 34 (b) and (c) shows the compression results, where 3118 waypoints were retained after using the DP algorithm and 4540 waypoints were retained after using the MPDP algorithm. Table 15 shows that the average operation time of the MPDP algorithm was reduced by 81.1ms. The average compression rate was reduced by 1.8%. The average length loss rate was reduced by 0.06%. The average synchronous Euclidean distance was reduced by 20.4003m and optimized by 13.98%. The average heading deviation decreased by 0.0168rad and was optimized by 13.22%. The average speed

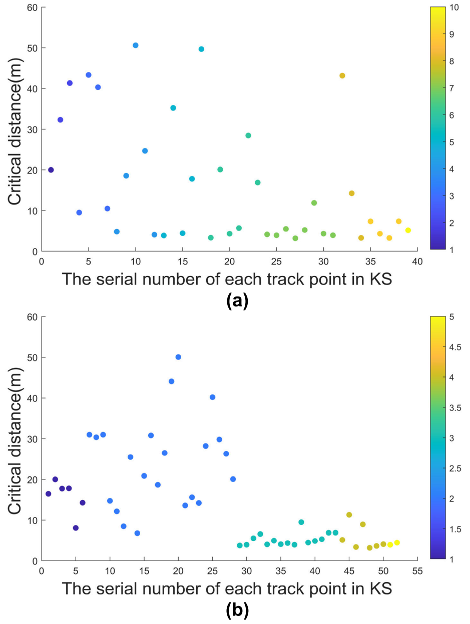


FIGURE 31. Critical distance variation for Case 7. Note: points of the same color indicate the waypoints retained under the same number of compression layers.

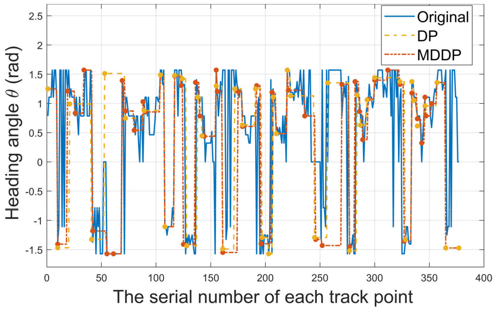


FIGURE 32. Comparison of heading angles of Case 7.

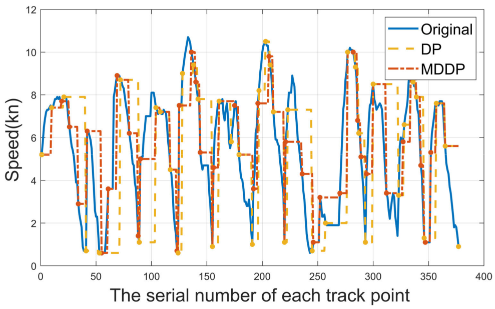


FIGURE 33. Comparison of speeds in Case 7.

deviation was reduced by 0.0421kn and optimized by approximately 21.63%.

V. CONCLUSION

The massive AIS trajectory data generated by ships every moment during navigation provide a big database for studying maritime traffic and related information. The deep development of the AIS data by researchers is greatly significant to the construction of maritime intelligence. However, unprocessed AIS data contain a large amount of duplicated and redundant information, leading to the wastage of storage space and calculation costs and largely affecting the data's application scope and effectiveness. In this work, we conducted a study related to AIS data compression and proposed the multi-objective peak DP algorithm incorporating four improvement strategies. These strategies compensated for the deficiencies of the DP algorithm in terms of poor compression of multi-bend tracks and its failure to consider two important ship maneuvering factors (i.e., heading speed and failure) to examine the correctness of compressed routes in combination with maps. **Improved Strategy I** (multi-peak retention strategy) can effectively reduce the number of compression layers and time. Using **Improved Strategy I** alone can save up to nearly 50% of the running time. However, even though introducing **Improved Strategy III** (obstacle detection mechanism) can effectively solves error cases (e.g., classical DP algorithm compression resulted crossing obstacles), it adds additional computation to the algorithm, making the MPDP algorithm less effective in optimizing the running time. With fewer waypoints, the running time may be slightly

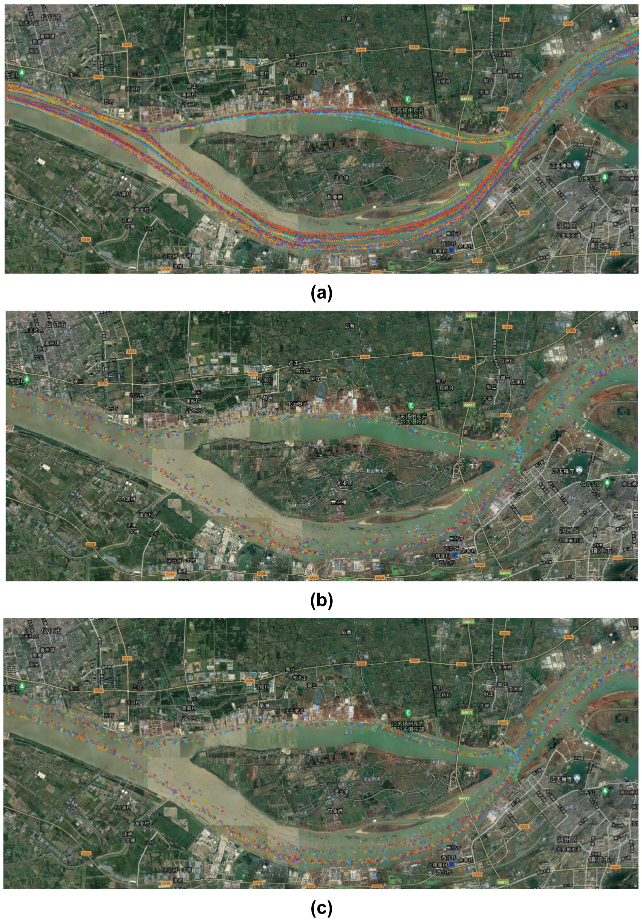


FIGURE 34. Compression results of 2-hour AIS data from Shiyezhou section of the Yangtze River waters, where (a) is original AIS data, (b) using the DP algorithm and (c) using the MPDP algorithm.

larger than that of the classical DP algorithm. By introducing **Improved Strategy II** (multi-objective optimization of the trajectory in spatial characteristics), the speed and the heading were optimized using the fitness function instead of the point-to-linear distance. Finally, **Improved Strategy IV** (position overlap judgment mechanism) was introduced to effectively solve the unsolvable situation caused by the spatial overlap of ship track points.

The results of the simulated and real trajectory experiments showed that the MPDP algorithm was approximately equal to or slightly smaller than the classical DP algorithm in terms of the compression rate. The performance indices of length loss rate, synchronous Euclidean distance, and average speed and heading deviations were optimized. In other words, the MPDP algorithm compressed the trajectory more closely to the original route while maintaining the high compression rate of the classical DP algorithm. The compressed trajectory showed a better recovery effect on the maneuvering performance of ship heading and speed. The MPDP algorithm was particularly effective in compressing circling and reciprocating trajectories, with an optimization rate of up to 40% for the length loss rate, simultaneous Euclidean distance, and average speed deviation. Multi-turn or circling trajectories often exist in inland or multi-island waters. Therefore, in the research of maritime traffic and related fields in these waters, the MPDP algorithm can be utilized as a preprocessing method for the AIS data, which can retain more critical information than what must be extracted from the route. In our future research work, we will develop a route-planning algorithm based on the AIS data compression for inland waters by using the MPDP algorithm suitable for the inland waterway characteristics and navigation rules. We will also conduct further research on the running time aspect of the MPDP algorithm to better reduce the overall computation time of a large volume of AIS data and improve the research efficiency.

REFERENCES

[1] *Maritime Navigation and Radiocommunication Equipment and Systems*, Standard BS PD IEC PAS 61174-1:2021, Apr. 16, 2021, doi: 10.3403/30431631.

[2] W. Zhang, F. Goerlandt, J. Montewka, and P. Kujala, "A method for detecting possible near miss ship collisions from AIS data," *Ocean Eng.*, vol. 107, pp. 60–69, Oct. 2015.

[3] H. Rong, A. P. Teixeira, and C. G. Soares, "Ship collision avoidance behaviour recognition and analysis based on AIS data," *Ocean Eng.*, vol. 245, Feb. 2022, Art. no. 110479.

[4] A. Hörteborn and J. W. Ringsberg, "A method for risk analysis of ship collisions with stationary infrastructure using AIS data and a ship manoeuvring simulator," *Ocean Eng.*, vol. 235, Sep. 2021, Art. no. 109396.

[5] J.-H. Shi and Z.-J. Liu, "Deep learning in unmanned surface vehicles collision-avoidance pattern based on AIS big data with double GRU-RNN," *J. Mar. Sci. Eng.*, vol. 8, no. 9, p. 682, Sep. 2020.

[6] G. Pallotta, M. Vespe, and K. Bryan, "Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction," *Entropy*, vol. 15, no. 6, pp. 2218–2245, Jun. 2013.

[7] T. Liu and J. Ma, "Ship navigation behavior prediction based on AIS data," *IEEE Access*, vol. 10, pp. 47997–48008, 2022.

[8] K. Bao, J. Bi, M. Gao, Y. Sun, X. Zhang, and W. Zhang, "An improved ship trajectory prediction based on AIS data using MHA-BiGRU," *J. Mar. Sci. Eng.*, vol. 10, no. 6, p. 804, Jun. 2022.

[9] T. A. Volkova, Y. E. Balykina, and A. Bespalov, "Predicting ship trajectory based on neural networks using AIS data," *J. Mar. Sci. Eng.*, vol. 9, no. 3, p. 254, Feb. 2021.

[10] Z. Yan, Y. Xiao, L. Cheng, R. He, X. Ruan, X. Zhou, M. Li, and R. Bin, "Exploring AIS data for intelligent maritime routes extraction," *Appl. Ocean Res.*, vol. 101, Aug. 2020, Art. no. 102271.

[11] K. Naus, "Drafting route plan templates for ships on the basis of AIS historical data," *J. Navigat.*, vol. 73, no. 3, pp. 726–745, May 2020.

[12] Y. K. He, D. Zhang, J. Zhang, M. Y. Zhang, and T. W. Li, "Ship route planning using historical trajectories derived from AIS data," *TransNav, Int. J. Mar. Navigat. Saf. Sea Transp.*, vol. 13, no. 1, pp. 69–76, 2019.

[13] S.-K. Zhang, G.-Y. Shi, Z.-J. Liu, Z.-W. Zhao, and Z.-L. Wu, "Data-driven based automatic maritime routing from massive AIS trajectories in the face of disparity," *Ocean Eng.*, vol. 155, pp. 240–250, May 2018.

[14] L. Wu, Y. Xu, Q. Wang, F. Wang, and Z. Xu, "Mapping global shipping density from AIS data," *J. Navigat.*, vol. 70, no. 1, pp. 67–81, Jan. 2017.

[15] S. Kim, H. Kim, and Y. Park, "Early detection of vessel delays using combined historical and real-time information," *J. Oper. Res. Soc.*, vol. 68, no. 2, pp. 182–191, Feb. 2017.

[16] L. Huang, Y. Wen, X. Geng, C. Zhou, C. Xiao, and F. Zhang, "Estimation and spatio-temporal analysis of ship exhaust emission in a port area," *Ocean Eng.*, vol. 140, pp. 401–411, Aug. 2017.

[17] C. Le Goff, B. Boussidi, A. Mironov, Y. Guichoux, Y. Zhen, P. Tandeo, S. Gueguen, and B. Chapron, "Monitoring the greater Agulhas current with AIS data information," *J. Geophys. Res., Oceans*, vol. 126, no. 5, May 2021, Art. no. e2021JC017228.

[18] R. Pelich, M. Chini, R. Hostache, P. Matgen, C. Lopez-Martinez, M. Nuevo, P. Ries, and G. Eiden, "Large-scale automatic vessel monitoring based on dual-polarization Sentinel-1 and AIS data," *Remote Sens.*, vol. 11, no. 9, p. 1078, May 2019.

[19] N. Molina-Padron, F. Cabrera-Almeida, V. Arana, M. Tichavska, and B.-P. Dorta-Naranjo, "Monitoring in near-real time for amateur UAVs using the AIS," *IEEE Access*, vol. 8, pp. 33380–33390, 2020.

[20] S. Zohoori, M. J. Kang, M. Hamidi, and B. Craig, "A vectorized algorithm for waterway traffic analysis using AIS data," *J. Ocean Technol.*, vol. 16, no. 4, pp. 68–95, 2021.

[21] F. Xiao, H. Ligteringen, C. van Gulijk, and B. Ale, "Comparison study on AIS data of ship traffic behavior," *Ocean Eng.*, vol. 95, pp. 84–93, Feb. 2015.

[22] Z. Yan, L. Cheng, R. He, and H. Yang, "Extracting ship stopping information from AIS data," *Ocean Eng.*, vol. 250, Apr. 2022, Art. no. 111004.

[23] S. Roul, C. Kumar, and A. Das, "Ambient noise estimation in territorial waters using AIS data," *Appl. Acoust.*, vol. 148, pp. 375–380, May 2019.

[24] J. L. Shepperson, N. T. Hintzen, C. L. Szostek, E. Bell, L. G. Murray, and M. J. Kaiser, "A comparison of VMS and AIS data: The effect of data coverage and vessel position recording frequency on estimates of fishing footprints," *ICES J. Mar. Sci.*, vol. 75, no. 3, pp. 988–998, Jun. 2018.

[25] C. Zhang, Y. Chen, B. Xu, Y. Xue, and Y. Ren, "The dynamics of the fishing fleet in China seas: A glimpse through AIS monitoring," *Sci. Total Environ.*, vol. 819, May 2022, Art. no. 153150.

[26] D. S. Pedroche, D. Amigo, J. García, and J. M. Molina, "Architecture for trajectory-based fishing ship classification with AIS data," *Sensors*, vol. 20, no. 13, p. 3782, Jul. 2020.

[27] Z. Yan, Y. Xiao, L. Cheng, S. Chen, X. Zhou, X. Ruan, M. Li, R. He, and B. Ran, "Analysis of global marine oil trade based on automatic identification system (AIS) data," *J. Transp. Geogr.*, vol. 83, Feb. 2020, Art. no. 102637.

[28] H. Jia, V. Prakash, and T. Smith, "Estimating vessel payloads in bulk shipping using AIS data," *Int. J. Shipping Transp. Logistics*, vol. 11, no. 1, pp. 25–40, 2019.

[29] M. Robards, G. Silber, J. Adams, J. Arroyo, D. Lorenzini, K. Schwehr, and J. Amos, "Conservation science and policy applications of the marine vessel automatic identification system (AIS)—A review," *Bull. Mar. Sci.*, vol. 92, no. 1, pp. 75–103, Jan. 2016.

[30] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica, Int. J. Geograph. Inf. Geovis.*, vol. 10, no. 2, pp. 112–122, Dec. 1973.

[31] R. Bellman and B. Kotkin, "On the approximation of curves by line segments using dynamic programming," *Commun. ACM*, vol. 4, no. 6, p. 284, Feb. 1962.

[32] M. Potamias, K. Patroumpas, and T. Sellis, "Sampling trajectory streams with spatiotemporal criteria," Presented at the 18th Int. Conf. Sci. Stat. Database Manag. (SSDBM), Vienna, Austria, 2006.

[33] E. J. Keogh, S. Chu, D. Hart, and M. J. Pazzani, "An online algorithm for segmenting time series," Presented at the IEEE Int. Conf. Data Mining, Nov./Dec. 2001.

[34] N. Meratnia and R. A. de By, "Spatiotemporal compression techniques for moving point objects," in *Proc. Int. Conf. Extending Database Technol.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecure Notes in Bioinformatics, vol. 2992, no. 1, 2004, pp. 765–782.

[35] J. Muckell, J. H. Hwang, C. T. Lawson, and S. S. Ravi, "Algorithms for compressing GPS trajectory data: An empirical evaluation," Presented at the 18th SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst. (GIS), San Jose, CA, USA, 2010.

[36] G. K. D. de Vries and M. van Someren, "Machine learning for vessel trajectories using compression, alignments and domain knowledge," *Expert Syst. Appl.*, vol. 39, no. 18, pp. 13426–13439, Dec. 2012.

[37] S.-K. Zhang, Z.-J. Liu, Y. Cai, Z.-L. Wu, and G.-Y. Shi, "AIS trajectories simplification and threshold determination," *J. Navigat.*, vol. 69, no. 4, pp. 729–744, 2016.

[38] Y. Li, R. W. Liu, J. Liu, Y. Huang, B. Hu, and K. Wang, "Trajectory compression-guided visualization of spatio-temporal AIS vessel density," Presented at the 8th Int. Conf. Wireless Commun. Signal Process. (WCSP), 2016.

[39] L. Zhao and G. Shi, "A method for simplifying ship trajectory based on improved Douglas–Peucker algorithm," *Ocean Eng.*, vol. 166, pp. 37–46, Oct. 2018.

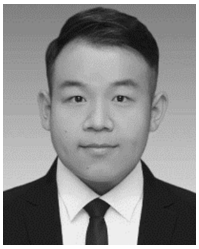
[40] J. Tang, L. Liu, and J. Wu, "A trajectory partition method based on combined movement features," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–13, Jul. 2019.

[41] J. Liu, H. Li, Z. Yang, K. Wu, Y. Liu, and R. W. Liu, "Adaptive Douglas–Peucker algorithm with automatic thresholding for AIS-based vessel trajectory compression," *IEEE Access*, vol. 7, pp. 150677–150692, 2019.

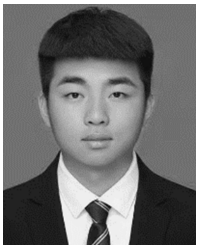
[42] C. Tang, H. Wang, J. Zhao, Y. Tang, H. Yan, and Y. Xiao, "A method for compressing AIS trajectory data based on the adaptive-threshold Douglas–Peucker algorithm," *Ocean Eng.*, vol. 232, Jul. 2021, Art. no. 109041.

[43] Y. Zhong, J. Kong, J. Zhang, Y. Jiang, X. Fan, and Z. Wang, "A trajectory data compression algorithm based on spatio-temporal characteristics," *PeerJ Comput. Sci.*, vol. 8, Oct. 2022, Art. no. e1112.

[44] M. Gao and G. Y. Shi, "Ship spatiotemporal key feature point online extraction based on AIS multi-sensor data using an improved sliding window algorithm," *Sensors*, vol. 19, no. 12, p. 2706, Jun. 2019.



ZHENG ZHOU was born in Changchun, China, in 1998. He received the bachelor's degree in electronic information science and technology, in 2020. He is currently pursuing the M.D. degree in circuits and systems with Jilin University, Jilin. He studied at Jilin University. His research interest includes route planning based on AIS.



YINGJIAN ZHANG was born in Chifeng, China, in 1998. He received the bachelor's degree in electronic information science and technology, in 2020. He is currently pursuing the M.D. degree in circuits and systems with Jilin University, Jilin. He studied at Jilin University. His research interest includes ship route planning.

XIAOYU YUAN was born in Taiyuan, China, in 2000. He received the bachelor's degree in electronic information engineering, in 2022. He is currently pursuing the M.D. degree in circuits and systems with Jilin University. He studied at Qinghai Normal University, Qinghai. His research interest includes ship route planning.

HONGBO WANG was born in Changchun, China, in 1969. She received the Ph.D. degree from Saint Petersburg State University, Russia. She is currently working with the College of Electronic Science and Engineering, Jilin University. Her research interests include ship motion control, weather routing, and ship collision avoidance.